



EDUCACIÓN
SECRETARÍA DE EDUCACIÓN PÚBLICA



TECNOLÓGICO
NACIONAL DE MÉXICO®



Instituto Tecnológico de San Luis Potosí

Reporte Final del Año Sabático bajo Modalidad B1.2 Elaboración de apuntes de asignatura.

“Fundamentos de Base de Datos, clave AEF-1031”



Programa de estudios: Ingeniería en Sistemas Computacionales

Autor: Dubelza Beatriz Oliva Garza

Docente del Departamento de Sistemas y Computación

San Luis Potosí, S.L.P. 25 de agosto de 2025



Contenido

Contribución Académica	4
Temario oficial:	6
Introducción a la asignatura.....	8
Ubicación del programa de estudios en la malla reticular.....	9
Tema 1 Introducción a las Bases de Datos	10
1.1 Conceptos básicos	10
1.2 Objetivos de las bases de datos	12
1.3 Áreas de Aplicación de los Sistemas de Bases de datos	13
1.4 Modelos de bases de datos.....	16
1.5 Clasificación de Bases de Datos	19
1.6 Arquitectura de base de datos	22
1.7 Arquitectura del SGBD	24
Tema 2 Diseño de Bases de Datos con el modelo Entidad - Relación.	33
2.1 El Proceso de Diseño.....	33
2.2 Modelo Entidad-Relación	35
2.3 Diseño con diagramas E-R	42
2.4 Modelo E-R extendido	43
2.5 La Notación E-R con UML.....	44
Tema 3: Modelo relacional	50
3.1 Introducción al modelo relacional	50
3.2 Conversión de Modelo E-R a Modelo relacional.....	51
3.3 Esquema de la base de datos.....	53
3.4 Restricciones	54
3.5 Integridad de dominio	55
Tema 4: Normalización de Bases de Datos.....	60
4.1 Conceptos básicos	60
4.2 Primera forma normal.	61
4.3 Dependencias funcionales y transitivas.	62



4.4 Segunda Forma Normal (2FN):	63
4.5 Tercera Forma Normal (3FN):	64
4.6 Forma Normal de Boyce-Codd (FNBC o BCNF):	64
4.7 Otras formas normales:	65
Tema 5 Álgebra relacional	89
5.1 Operaciones fundamentales del álgebra relacional:	89
5.2 Álgebra relacional extendida	94
TEMA 6 Introducción al lenguaje sql:	119
6.1 Características	119
6.2 Lenguaje de Definición de Datos (LDD)	120
6.3 Lenguaje de Manipulación de Datos (DML):	122
Referencias	141
Instrumentación Didáctica	142
Carta de reconocimiento del autor de los derechos a favor del TecNM	162
Reporte de Turnitin (Software antiplagio)	163



Contribución Académica

La asignatura de Fundamentos de Bases de Datos contribuye significativamente a los objetivos formativos del programa educativo de Ingeniería en Sistemas Computacionales al que pertenece. Al analizar su aporte, se destaca que esta asignatura capacita a los estudiantes en aspectos fundamentales para el desarrollo eficaz y eficiente del software de gestión, así como para diseñar y gestionar el almacenamiento de información en empresas y otros contextos. Además, se enfoca en habilidades clave como el análisis, diseño y gestión de sistemas de bases de datos, garantizando la integridad, disponibilidad y confidencialidad de la información.

La asignatura de Fundamentos de Bases de Datos también prepara a los estudiantes para desarrollar e implementar sistemas de información que apoyen la toma de decisiones, utilizando metodologías basadas en estándares internacionales. Esto es esencial para que los futuros profesionales en Ingeniería en Sistemas Computacionales puedan crear aplicaciones robustas que manejen la información de manera segura y eficiente.

La asignatura de Fundamentos de Bases de Datos aporta directamente a los objetivos formativos del programa educativo de Ingeniería en Sistemas Computacionales al proporcionar a los estudiantes las competencias necesarias para trabajar en el diseño, gestión y desarrollo de sistemas de bases de datos, cumpliendo con los estándares de calidad y eficiencia requeridos en el campo de la informática.

Además, es importante mencionar los siguientes beneficios adicionales:

1. Mejora el recuerdo y la comprensión de la información:
2. Fomenta un rol activo en el aprendizaje



3. Facilita el repaso y el estudio: Contar con un conjunto de apuntes bien estructurados permite a los estudiantes repasar y estudiar el material de manera eficiente antes de exámenes y trabajos.
4. Desarrolla habilidades valiosas

La elaboración de apuntes de la materia Fundamentos de Base de datos se justifica académicamente por las siguientes razones: Refuerza el aprendizaje, al mejorar la atención, facilitar el procesamiento cognitivo, organizar la información y estimular el pensamiento crítico. Además, proporciona un recurso útil para la revisión y consolidación del conocimiento adquirido, personaliza el contenido, desarrolla habilidades, es una guía para los docentes que imparten por primera vez esta materia.

Competencia a desarrollar de la asignatura:

- Analiza requerimientos y diseña bases de datos para generar soluciones al tratamiento de información basándose en modelos y estándares.

Competencias previas:

- Esta asignatura requiere como competencia previa que el estudiante comprenda y aplique los conceptos y propiedades de álgebra de conjuntos, relaciones y álgebra booleana adquiridas en matemáticas discretas.
- Comprende y aplica los conceptos básicos de lógica matemática, conjuntos y relaciones para aplicarlos en modelos que resuelvan problemas computacionales.



Temario oficial:



TECNOLÓGICO NACIONAL DE MÉXICO
Secretaría Académica, de Investigación e Innovación
Dirección de Docencia e Innovación Educativa

1. Datos Generales de la asignatura

Nombre de la asignatura:	Fundamentos de Base de Datos
Clave de la asignatura:	AEF-1031
SATCA¹:	3-2-5
Carrera:	Ingeniería Informática, Ingeniería en Sistemas Computacionales e Ingeniería en Tecnologías de la Información y Comunicaciones

No.	Temas	Subtemas
1.	Introducción a las bases de datos	1.1 Conceptos básicos 1.2 Objetivos de las Bases de Datos 1.3 Áreas de Aplicación de los Sistemas de Bases de datos 1.4 Modelos de bases de datos 1.5 Clasificación de Bases de Datos 1.6 Arquitectura de base de datos 1.7 Arquitectura del SGBD
2.	Diseño de Bases de Datos con el modelo E-R.	2.1 El Proceso de Diseño 2.2 Modelo Entidad-Relación 2.3 Diseño con diagramas E-R 2.4 Modelo E-R extendido 2.5 La Notación E-R con UML.
3.	Modelo relacional	3.1 Introducción al modelo relacional 3.2 Conversión de Modelo E-R a Modelo relacional 3.3 Esquema de la base de datos 3.4 Restricciones 3.3.1 Integridad de entidad 3.3.2 Integridad referencial 3.5 Integridad de dominio
4.	Normalización de bases de datos.	4.1 Conceptos básicos 4.2 Primera forma normal. 4.3 Dependencias funcionales y transitivas. 4.4 Segunda forma normal. 4.5 Tercera forma normal. 4.6 Forma normal Boyce-Codd.



		4.7 Otras formas normales.
5.	Álgebra relacional.	5.1 Operaciones fundamentales del álgebra relacional 5.2 Álgebra relacional extendida.
6.	Introducción al lenguaje SQL.	6.1 Características 6.2 Lenguaje de Definición de Datos (LDD) 6.3 Lenguaje de Manipulación de Datos (LMD)



Introducción a la asignatura

El programa de la asignatura de Fundamentos de Base de datos se encuentra ubicada en el 4º semestre del plan reticular, no teniendo pre-requisito, pero siendo muy recomendable la asignatura de Matemáticas Discretas de 1er semestre.

Este programa está diseñado para fortalecer las competencias específicas de un Ingeniero en Sistemas Computacionales, la capacidad para analizar, diseñar y gestionar sistemas de bases de datos conforme a los requerimientos del entorno para garantizar la integridad, disponibilidad y confidencialidad de la información, así como para desarrollar e implementar sistemas de información para la gestión de procesos y apoyo en la toma de decisiones, utilizando metodologías basadas en estándares internacionales.

Es importante porque el estudiante adquiere las competencias en el análisis y el diseño de base de datos, que le permitirán desarrollar aplicaciones para sistemas de información robustos que ofrezcan garantía en el manejo de la información.

Es conveniente mencionar que hoy en día la información forma parte del capital intangible de las organizaciones y cada vez se demandan sistemas de información que garanticen la integridad y seguridad de esta.

La asignatura propicia el dominio de modelos de diseño de base de datos basados en reglas de normalización, de integridad y de seguridad

Dentro de las competencias genéricas desarrolladas en este programa de estudios de manera transversal se encuentran capacidad de investigación, capacidad de abstracción, análisis y síntesis, habilidades para buscar, procesar y analizar información procedente de diversas fuentes, habilidad en el uso de Tecnologías de la Información y de la Comunicación (TIC), habilidad de comunicación de expresión oral y escrita, trabajo en equipo, trabajo en grupos colaborativos, trabajo bajo presión, todas ellas de suma importancia para su inserción en el mundo laboral.



El programa de la asignatura se organiza en seis temas, en los cuales se incluyen aspectos teóricos y de aplicación relacionados con modelado lógico de base de datos y lenguajes de acceso a base de datos.

Ubicación del programa de estudios en la malla reticular.

INSTITUTO TECNOLÓGICO DE SAN LUIS POTOSÍ
RETÍCULA DE LA CARRERA DE INGENIERÍA EN SISTEMAS COMPUTACIONALES
CLAVE: ISIC-2010-224 **MODALIDAD: PRESENCIAL**

<table><tr><td>Fundamentos de Base de Datos DBB4</td></tr><tr><td>AEF-1031</td></tr><tr><td>22 3 2 5</td></tr></table>	Fundamentos de Base de Datos DBB4	AEF-1031	22 3 2 5	<table><tr><td>22</td><td>Taller de Base de Datos CBB5</td></tr><tr><td></td><td>SCA-1025</td></tr><tr><td>28</td><td>0 4 4</td></tr></table>	22	Taller de Base de Datos CBB5		SCA-1025	28	0 4 4	<table><tr><td>28</td><td>Administración de Base de Datos CBB6</td></tr><tr><td></td><td>SCB-1001</td></tr><tr><td>34</td><td>1 4 5</td></tr></table>	28	Administración de Base de Datos CBB6		SCB-1001	34	1 4 5
Fundamentos de Base de Datos DBB4																	
AEF-1031																	
22 3 2 5																	
22	Taller de Base de Datos CBB5																
	SCA-1025																
28	0 4 4																
28	Administración de Base de Datos CBB6																
	SCB-1001																
34	1 4 5																



Tema 1 Introducción a las Bases de Datos

1.1 Conceptos básicos

1.2 Objetivos de las Bases de Datos

1.3 Áreas de Aplicación de los Sistemas de Bases de datos

1.4 Modelos de bases de datos

1.5 Clasificación de Bases de Datos

1.6 Arquitectura de base de datos

1.7 Arquitectura del SGBD

Competencias Específica(s): Conoce y comprende los conceptos básicos de base de datos para proponer soluciones en el tratamiento de información.

Competencias Genéricas:

- Habilidades para buscar, procesar y analizar información procedente de diversas fuentes.
- Capacidad de abstracción, análisis y síntesis.
- Capacidad de comunicación oral y escrita.

1.1 Conceptos básicos

Conceptos fundamentales de las bases de datos

- **Base de datos:** Conjunto organizado de datos interrelacionados que se almacenan electrónicamente con el objetivo de ser gestionados, consultados y actualizados eficientemente. Permiten almacenar información sobre personas, productos, transacciones, entre otros elementos relevantes para una organización.
- **Dato:** Representación simbólica (numérica, alfabética, etc.) de un atributo o característica de una entidad. Por sí solo, un dato no tiene valor semántico, pero procesado en contexto genera información útil.



- **Información:** Conjunto de datos procesados que adquiere significado y valor para la toma de decisiones o la gestión de procesos.
- **Campo:** Área de almacenamiento que contiene datos del mismo tipo (por ejemplo, nombre, fecha de nacimiento).
- **Registro:** Conjunto de campos relacionados que representan una entidad completa (por ejemplo, todos los datos de una persona).
- **Tabla:** Colección de registros estructurados en filas y columnas dentro de una base de datos relacional.
- **SGBD:** Es un software que permite crear, administrar, organizar y manipular una base de datos. Actúa como una interfaz entre los usuarios, las aplicaciones y la base de datos, facilitando el almacenamiento, recuperación y gestión eficiente de grandes cantidades de información.
 - Funciones principales de un SGBD:
 - Permitir la definición de la estructura de la base de datos (tablas, campos, tipos de datos, restricciones).
 - Facilitar la manipulación de datos, incluyendo inserción, actualización, eliminación y consultas.
 - Asegurar la integridad y seguridad de los datos, controlando accesos y manteniendo la consistencia.
 - Realizar copias de seguridad y recuperación ante fallos del sistema.
 - Gestionar y optimizar el almacenamiento físico de la información para garantizar un acceso rápido y eficiente.
 - Componentes típicos:
 - Motor de almacenamiento que interactúa con el sistema operativo.
 - Lenguajes para definición (DDL), manipulación (DML) y consulta (DQL) de datos.



- Procesador y optimizador de consultas.
 - Catálogo de metadatos para registrar la estructura y objetos de la base.
 - Herramientas para monitoreo, informes, copia de seguridad y recuperación.
-
- Ejemplos comunes de SGBD: MySQL, MariaDB, PostgreSQL, Oracle Database, Microsoft SQL Server, Microsoft Access, entre otros.

Un SGBD es fundamental para gestionar datos en cualquier sector, garantizando que la información sea accesible, segura y organizada para su uso efectivo.

1.2 Objetivos de las bases de datos

- **Almacenamiento eficiente:** Optimizar espacio y velocidad de acceso a grandes volúmenes de datos.
- **Acceso rápido y sencillo:** Permitir búsquedas, consultas y actualizaciones de datos eficazmente por múltiples usuarios.
- **Integridad y consistencia:** Aplicar reglas que eviten inconsistencias y errores en la información.
- **Seguridad:** Proteger la información sensible mediante permisos de acceso y controles de confidencialidad.
- **Reducción de redundancia:** Minimizar la duplicación de datos, garantizando que cada dato exista solo una vez.
- **Independencia de datos:** Facilitar que los cambios en la estructura de la base de datos no afecten a las aplicaciones que la usan.



- **Confiabilidad y respaldo:** Asegurar recuperación ante fallos y pérdida de información.

Usos de las bases de datos

- **Centralización y compartición de datos:** Permitir que múltiples usuarios y aplicaciones accedan y manipulen la misma base de datos simultáneamente.
- **Control de acceso:** Definir quién puede ver o modificar los datos.
- **Facilitar la toma de decisiones:** Proveer información precisa y actualizada que apoye la administración y operaciones de una organización.
- **Automatización de procesos:** Permitir consultas y reportes automáticos sobre los datos almacenados.

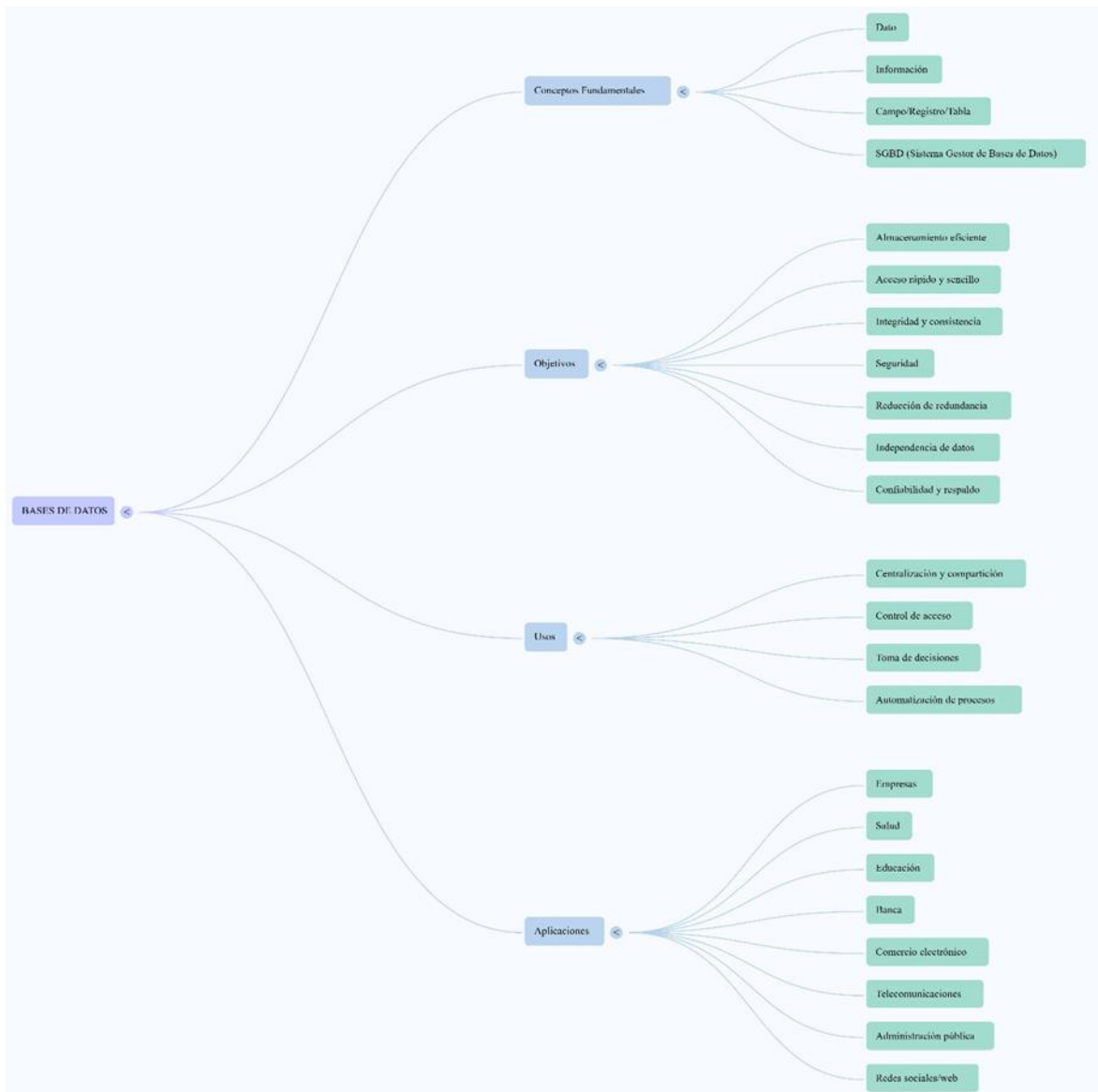
1.3 Áreas de Aplicación de los Sistemas de Bases de datos

Las bases de datos tienen aplicaciones en una gran variedad de sectores y sistemas, incluyendo:

- **Sector empresarial:** Gestión de inventarios, procesamiento de transacciones, seguimiento de clientes y ventas, análisis de mercado.
- **Salud:** Mantenimiento de historiales médicos, programación de citas, gestión de pacientes.
- **Educación:** Administración de registros estudiantiles, inscripciones, calificaciones y gestión académica.
- **Banca:** Manejo de información sobre clientes, cuentas, préstamos y transacciones financieras.
- **Comercio electrónico:** Gestión de catálogos de productos, pedidos y recomendaciones en línea.



- **Telecomunicaciones:** Almacenamiento de registros de llamadas, mantenimiento de saldos, facturación.
- **Administración pública:** Registros de ciudadanos, impuestos, servicios y trámites gubernamentales.
- **Redes sociales y aplicaciones web:** Almacenamiento de perfiles, publicaciones y datos de usuarios en plataformas digitales.



Mapa conceptual: Bases de datos

Elaboración propia

Este esquema ofrece una visión global sobre los conceptos, objetivos, usos y aplicaciones de las bases de datos en el entorno actual.



1.4 Modelos de bases de datos

Existen varios modelos de bases de datos que definen la manera en que los datos son organizados, almacenados, gestionados y relacionados. Cada modelo se adapta a diferentes necesidades y tipos de información.

1. Modelo jerárquico

- La información se organiza en una estructura de árbol invertido.
- Cada registro tiene un único "padre" y puede tener varios "hijos" (relaciones uno a muchos).
- Utilizado especialmente en aplicaciones donde las relaciones son estrictamente jerárquicas, como directorios organizacionales o archivos de sistemas antiguos.

2. Modelo de red

- Los registros están conectados entre sí mediante múltiples relaciones, permitiendo que cada registro tenga varios padres y varios hijos (relaciones muchos a muchos).
- Facilita representar estructuras complejas, como rutas de transporte, genealogías o mapas de relaciones en empresas.

3. Modelo relacional

- El más difundido actualmente.
- La información está estructurada en tablas que pueden relacionarse entre sí gracias a claves primarias y foráneas.
- Permite consultas flexibles y es ampliamente utilizado en sistemas empresariales, financieros y de gestión de datos.



4. Modelo orientado a objetos

- Los datos se almacenan en forma de objetos, similares a los que usan los lenguajes de programación orientados a objetos.
- Ideal para aplicaciones complejas que requieren manipular datos jerárquicos y relaciones sofisticadas, como software de ingeniería o simulaciones.

5. Modelo documental (NoSQL)

- Almacena información en documentos estructurados (generalmente en formato JSON, XML o BSON).
- Popular en aplicaciones web modernas, sistemas de gestión de contenidos y plataformas que requieren alta escalabilidad y flexibilidad en el almacenamiento de datos heterogéneos.

6. Modelo multidimensional

- Pensado para análisis y consulta de grandes volúmenes de datos, normalmente en contextos de inteligencia empresarial y procesamiento analítico (OLAP).
- Representa los datos en estructuras de cubos que permiten consultas complejas y agrupaciones según varios criterios (dimensiones).

7. Modelo clave-valor (NoSQL)

- Cada dato se almacena como un par “clave-valor”.
- Es extremadamente simple y rápido, ideal para gestionar grandes volúmenes de datos no estructurados o para sistemas como cachés y sesiones.

8. Modelo de grafos

- La información se representa como nodos (entidades) y aristas (relaciones), facilitando el modelado de redes sociales, rutas y estructuras de conectividad.



- Excelente para analizar patrones complejos de conexiones e interrelaciones entre conjuntos de datos.

Tabla 1. Características principales de los modelos de datos. Elaboración propia.

Modelo	Característica principal	Usos comunes
Jerárquico	Estructura de árbol invertido, relaciones 1:N	Archivos antiguos, directorios, sistemas legacy
Red	Múltiples padres e hijos, relaciones N:M	Genealogías, rutas, aplicaciones complejas
Relacional	Datos en tablas, claves primarias/foráneas	Negocios, finanzas, sector industrial
Orientado a objetos	Almacena objetos completos (atributos y métodos)	Ingeniería, ciencia, simulaciones
Documental (NoSQL)	Documentos como JSON/XML, datos flexibles	Web, CMS, Big Data
Multidimensional	Estructura de cubos para análisis	BI, OLAP, analítica de datos
Clave-valor (NoSQL)	Pares clave-valor, acceso rápido	Cachés, aplicaciones de alta demanda
Grafo	Nodos y aristas, relaciones complejas	Redes sociales, mapas, análisis de relaciones



Estos modelos pueden combinarse o especializarse para adaptarse a contextos muy diversos según las necesidades de la organización o la aplicación concreta. La elección del modelo adecuado es clave para garantizar eficiencia, escalabilidad y correcto manejo de la información.

1.5 Clasificación de Bases de Datos

Las bases de datos pueden clasificarse desde distintas perspectivas según su estructura, uso, forma de organización, variabilidad y tecnología. A continuación, se presentan las clasificaciones más aceptadas y frecuentes por algunos autores:

1. Según su Estructura o Modelo de Datos

- **Bases de datos jerárquicas:** Organizan la información en forma de árbol, donde cada nodo tiene uno o varios nodos hijos, pero solo un nodo padre.
- **Bases de datos de red:** Permiten relaciones más complejas, donde un nodo puede tener múltiples padres y múltiples hijos (relaciones muchos a muchos).
- **Bases de datos relacionales:** Basadas en tablas que se relacionan mediante claves primarias y foráneas. Son las más usadas actualmente y permiten consultas flexibles.
- **Bases de datos orientadas a objetos:** Almacenan información en forma de objetos, integrando conceptos de la programación orientada a objetos.
- **Bases de datos multidimensionales:** Estructuran la información en "cubos", permitiendo el análisis multidireccional, típicas en entornos de inteligencia de negocios.
- **Bases de datos de grafos:** Representan los datos como nodos y aristas, ideales para modelar redes sociales y relaciones complejas.



- **Bases de datos documentales (NoSQL):** Almacenan los datos en documentos estructurados como JSON o XML, permitiendo gran flexibilidad.
- **Bases de datos de clave-valor (NoSQL):** Los datos se guardan como pares clave/valor, son muy rápidas y escalables.

2. Según su Variabilidad

- **Bases de datos estáticas:** Solamente permiten consultar los datos; normalmente se utilizan para análisis históricos donde los datos no cambian.
- **Bases de datos dinámicas:** Permiten la actualización, modificación y borrado de registros de manera continua.

3. Según su Distribución

- **Bases de datos centralizadas:** Todos los datos se almacenan en una única ubicación física.
- **Bases de datos distribuidas:** La información se encuentra almacenada en diferentes ubicaciones, conectadas por una red, pero para el usuario funciona como una sola base de datos.

4. Según el Contenido o Finalidad

- **Bases de datos bibliográficas:** Contienen referencias a documentos, libros, artículos, etc.
- **Bases de datos de texto completo:** Almacenan el texto íntegro de los documentos.
- **Bases de datos multimedia:** Gestionan imágenes, audios y videos.
- **Bases de datos científicas, biológicas, legales, comerciales, etc.:** Diseñadas específicamente para disciplinas o sectores particulares.



5. Según la Tecnología o el Acceso

- **Bases de datos en la nube:** Almacenadas y gestionadas en plataformas cloud.
- **Bases de datos en memoria:** Los datos residen en la RAM para mayor velocidad de consulta.
- **Bases de datos transaccionales:** Centran su funcionamiento en la realización de operaciones transaccionales con garantías de integridad.

Tabla 2. Clasificación de las bases de datos. Elaboración propia.

Criterio	Clasificaciones principales
Por estructura	Jerárquica, Red, Relacional, Objetos, Multidimensional, Grafo, NoSQL
Por variabilidad	Estática, Dinámica
Por distribución	Centralizada, Distribuida
Por contenido	Bibliográfica, Texto completo, Multimedia, Específicas
Por tecnología	En la nube, En memoria, Transaccional

Esta clasificación permite elegir el tipo y modelo más adecuado de base de datos según las necesidades de la organización, el volumen de datos, la estructura de la información y su finalidad específica.



1.6 Arquitectura de base de datos

La **arquitectura de base de datos** se refiere a la estructura general y organización de los sistemas que almacenan, gestionan y permiten el acceso eficiente a los datos. Su diseño está compuesto por varios niveles y componentes que trabajan de manera coordinada para ofrecer seguridad, disponibilidad y rendimiento óptimo.

Niveles en la arquitectura de una base de datos

Se reconocen principalmente tres niveles o capas, siguiendo la arquitectura estándar ANSI-SPARC (American National Standard Institute - Standards Planning and Requirements Comité), también denominados **Niveles de abstracción de una base de datos**:

- **Nivel interno (físico):**

Describe cómo y dónde se almacenan realmente los datos en dispositivos físicos (discos duros, SSD, memoria). Incluye detalles sobre la organización interna, la gestión del espacio y el rendimiento.

- **Nivel conceptual (lógico):**

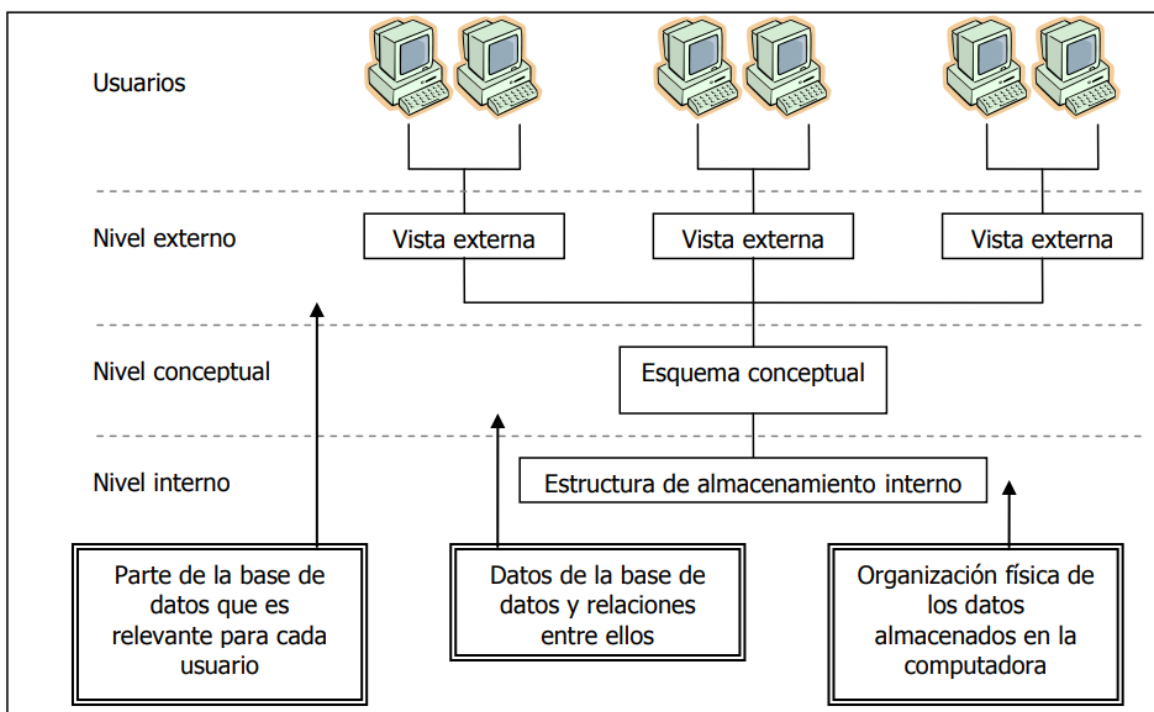
Se centra en la representación estructurada de los datos y sus relaciones sin preocuparse por cómo están almacenados físicamente. Aquí se definen entidades, atributos, relaciones y reglas de integridad.

- **Nivel externo (de usuario o de vistas):**

Proporciona diferentes vistas o perspectivas de los datos adaptadas a las necesidades de cada usuario o aplicación. Cada vista puede mostrar sólo una parte de la base de datos, facilitando seguridad y personalización.

**Tabla 3.** Resumen de niveles y componentes. Elaboración propia.

Nivel/Capa	Enfoque principal	Componentes clave
Interno/Físico	Almacenamiento real	Motor de almacenamiento, organización física, gestión del espacio
Conceptual/Lógico	Estructura y relaciones	Diccionario de datos, definición de entidades/relaciones, restricciones
Externo/De vistas	Vista del usuario	Vistas personalizadas, interfaces de usuario/cliente, control de acceso

**Figura 1** Niveles de abstracción de datos. Elaboración propia.



Arquitecturas comunes

- **Arquitectura cliente-servidor:**

Divide el sistema en dos partes:

- **Servidor de bases de datos:** Gestiona el almacenamiento, acceso, seguridad y procesamiento de datos.
- **Cliente:** Interfaz o aplicación que accede al servidor mediante consultas y recibe los resultados.

Esta arquitectura permite la centralización de los datos y la escalabilidad al poder añadir más clientes fácilmente.

- **Arquitectura distribuida:**

Los datos se almacenan en diferentes ubicaciones físicas, pero se presentan como una única base de datos lógica al usuario. Este modelo mejora la redundancia, disponibilidad y acceso global a la información.

La adecuada arquitectura de una base de datos garantiza eficiencia, seguridad, facilidad en el acceso y manipulación de la información, y una base sólida para el crecimiento futuro del sistema.

1.7 Arquitectura del SGBD

La arquitectura del SGBD describe la estructura interna y los componentes principales que permiten almacenar, gestionar, proteger y recuperar datos de manera eficiente. Su diseño modular permite la independencia de cada función y la coordinación para garantizar la integridad, seguridad y rendimiento de la base de datos.

Componentes principales del SGBD

1. **Gestor de almacenamiento**

Encargado de la gestión física de los datos en disco o memoria. Controla la



escritura, lectura y borrado, garantizando el correcto almacenamiento de la información.

- **Control de acceso (autorizaciones e integridad)**

Administra los permisos de los usuarios y asegura que solo personas autorizadas puedan consultar o modificar la información, garantizando la confidencialidad y seguridad de los datos.

- **Gestor de transacciones**

Maneja la ejecución de operaciones agrupadas (transacciones), asegurando la atomicidad, consistencia, aislamiento y durabilidad. Permite revertir cambios ante errores y proteger la integridad de la base.

- **Planificador y control de concurrencia**

Coordina el acceso simultáneo de múltiples usuarios, evitando conflictos y garantizando la integridad cuando varios procesos interactúan con los datos al mismo tiempo.

- **Gestor de archivos**

Administra la transferencia de datos entre la memoria principal y los dispositivos de almacenamiento, optimizando el acceso a los datos.

- **Gestor de recuperación**

Permite restaurar la base de datos a un estado consistente tras fallos, mediante backups y registros de transacciones.

2. **Gestor de consultas (procesador de consultas)**

Analiza y ejecuta las consultas hechas por los usuarios, transformando solicitudes en operaciones internas optimizadas para acceder a los datos. Incluye un optimizador que selecciona el plan de ejecución más eficiente.

- **Lenguaje de definición de datos (LDD):** crea y modifica la estructura de la base.

- **Lenguaje de manipulación de datos (LMD):** para insertar, modificar, eliminar y consultar datos.



- **Motor de evaluación de consultas.** Ejecuta las instrucciones de bajo nivel generadas por el compilador del LMD.
- **Compilador del LMD:** Traduce las instrucciones del LMD en un lenguaje de consultas a un plan de evaluación que consiste en instrucciones de bajo nivel que entiende el motor de evaluación de consultas.

3. Almacenamiento en disco

- **Diccionario de datos**

Subsistema que almacena los metadatos, es decir, la descripción de las estructuras y restricciones de la base de datos (tablas, índices, relaciones, etc.), usados por otros módulos para operar sobre los datos.

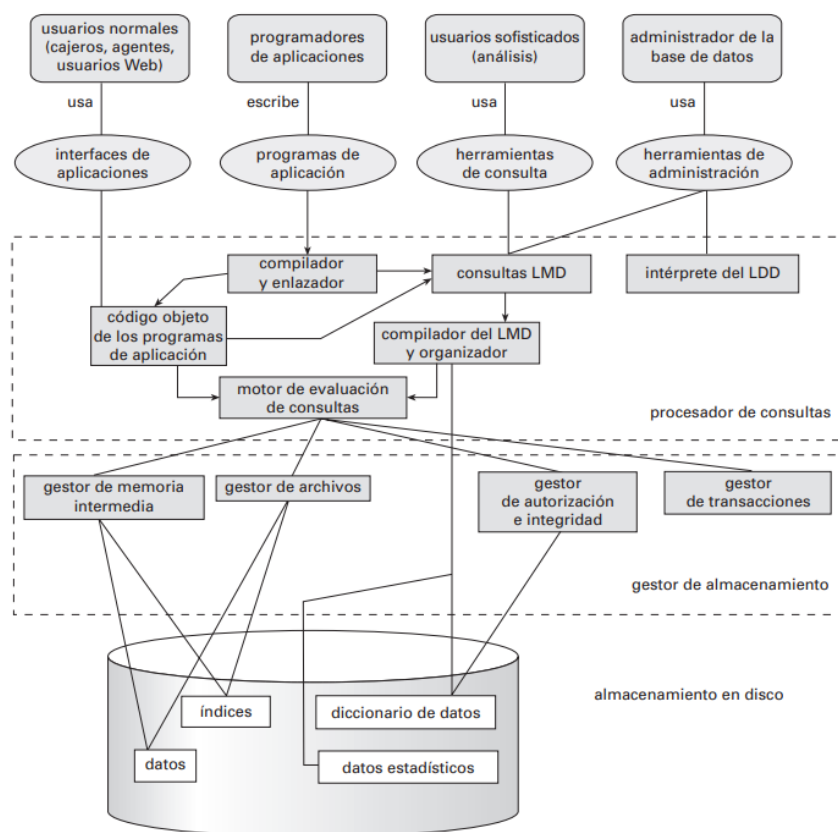


Figura 2 Arquitectura de un Sistema gestor de base de datos

Fundamentos de Base de datos Korth y Silverchatz



Usuarios y administradores de una base de datos

Podemos definir a los usuarios como aquella persona que tenga todo tipo de contacto con el sistema de base de datos desde que este se diseña, elabora, termina y se usa. Los usuarios que accedan a una base de datos pueden clasificarse como:

- **Administrador de bases de datos:** Denominado por sus siglas como DBA (DataBase Administrator). Es la persona encargada y que tiene el control total sobre el sistema de base de datos. Este usuario y sus funciones se definirán a mayor detalle más adelante en esta misma sección.
- **Programadores de aplicaciones:** Se encargan de desarrollar las aplicaciones que manejan datos de la base de datos. Estas aplicaciones contendrán solicitudes de datos al DBMS que luego serán procesados por los programas de la aplicación que tendrán como finalidad resolver problemas específicos de la empresa.
- **Usuarios finales:** Son personas que no tienen por qué tener conocimientos informáticos y que pueden manipular los datos (examinarlos y actualizarlos) con la ayuda de las aplicaciones. Son quienes interactúan con el sistema invocando a uno de los programas de aplicación permanentes que se han escrito anteriormente en el sistema de base de datos.



Ejemplos de modelo de datos

1. Ejemplo de modelo de datos conceptual

Las entidades que se pueden modelar en un negocio de ventas podrían ser el siguiente:

- ✚ Sucursales: Toda la información de las ventas.
- ✚ Productos: Información sobre los productos y de su nivel de inventario.
- ✚ Ventas: Información sobre cada una de las ventas.
- ✚ Clientes: información sobre todas las personas que han realizado una compra.
- ✚ Vendedores: Información de los trabajadores.

2. Ejemplo de modelo de datos lógico

La información para considerar en el modelo lógico pudiera ser la siguiente:

- ✚ Sucursales: contaría con datos como el nombre, la dirección, el correo electrónico y el número de teléfono de cada punto de venta.
- ✚ Productos: número del producto, nombre y marca.
- ✚ Clientes: nombre, dirección, el correo electrónico y su teléfono.
- ✚ Ventas: información de los productos, del cliente, del vendedor, la fecha de compra y el precio.



Examen

Tema 1 Introducción a las bases de datos con Respuestas

Instrucciones

Responda las siguientes preguntas seleccionando la opción correcta o respondiendo según se indique.

Preguntas de Selección Múltiple

1. ¿Qué es una base de datos?

- a) Un programa para editar textos
- b) Un conjunto organizado de datos relacionados
- c) Un sistema operativo
- d) Una hoja de cálculo

Respuesta: b) Un conjunto organizado de datos relacionados

2. ¿Cuál es la función del sistema gestor de bases de datos que permite realizar consultas para obtener datos específicos?

- a) Función de definición
- b) Función de integración
- c) Función de control
- d) Función de manipulación

Respuesta: d) Función de manipulación

3. El modelo relacional organiza los datos en:

- a) Árboles jerárquicos
- b) Redes complejas
- c) Tablas relacionadas por claves
- d) Documentos JSON

Respuesta: c) Tablas relacionadas por claves

4. ¿Cuál es uno de los principales objetivos de una base de datos?

- a) Aumentar la redundancia de datos
- b) Almacenamiento eficiente y acceso rápido



- c) Borrar información automáticamente
- d) Mantener datos en archivos planos

Respuesta: b) Almacenamiento eficiente y acceso rápido

5. ¿Qué modelo de base de datos utiliza una estructura de árbol invertido?

- a) Jerárquico
- b) Red
- c) Relacional
- d) Orientado a objetos

Respuesta: a) Jerárquico

6. En la arquitectura ANSI/SPARC, ¿qué nivel describe la vista del usuario?

- a) Nivel interno
- b) Nivel conceptual
- c) Nivel externo
- d) Nivel de almacenamiento

Respuesta: c) Nivel externo

7. ¿Qué es un registro en una base de datos?

- a) Un solo dato aislado
- b) Un conjunto de campos relacionados que representan una entidad completa
- c) Un índice para acelerar consultas
- d) Un usuario autorizado para el sistema

Respuesta: b) Un conjunto de campos relacionados que representan una entidad completa

8. La reducción de redundancia en bases de datos se logra para:

- a) Incrementar la duplicidad de datos
- b) Mejorar la integridad y eficiencia
- c) Facilitar la creación de datos duplicados
- d) Guardar copias de seguridad

Respuesta: b) Mejorar la integridad y eficiencia

9. ¿Qué es una transacción en una base de datos?

- a) Una consulta compleja
- b) Una secuencia de operaciones que se ejecuta como una unidad indivisible
- c) Un respaldo de la base de datos
- d) Un registro de usuario

Respuesta: b) Una secuencia de operaciones que se ejecuta como una unidad indivisible



10. ¿Qué arquitectura divide el sistema en cliente y servidor para gestionar bases de datos?

- a) Arquitectura monolítica
- b) Arquitectura cliente-servidor
- c) Arquitectura distribuida
- d) Arquitectura en red

Respuesta: b) Arquitectura cliente-servidor

Preguntas Verdadero o Falso

11. El modelo jerárquico es un modelo de red mejorado.

- a) Verdadero
- b) Falso

Respuesta: b) Falso

12. Las bases de datos orientadas a objetos almacenan la información en forma de objetos con atributos y métodos.

- a) Verdadero
- b) Falso

Respuesta: a) Verdadero

13. El control de acceso en un SGBD se encarga de definir los permisos de los usuarios.

- a) Verdadero
- b) Falso

Respuesta: a) Verdadero

14. La independencia de datos permite que los cambios en la estructura física afecten a las aplicaciones.

- a) Verdadero
- b) Falso

Respuesta: b) Falso

15. Las bases de datos distribuidas almacenan información en una única ubicación física.

- a) Verdadero
- b) Falso

Respuesta: b) Falso

Preguntas Abiertas



16. Define brevemente qué es un Sistema Gestor de Bases de Datos (SGBD) y menciona dos funciones principales.

Respuesta ejemplo:

Un SGBD es un software que permite almacenar, administrar y recuperar datos de una base de datos. Dos funciones principales son el procesamiento de consultas y el control de acceso a los datos.

17. ¿Cuál es la importancia de la integridad y consistencia en una base de datos?

Respuesta ejemplo:

La integridad y consistencia garantizan que los datos almacenados sean correctos y coherentes, evitando errores y duplicación, lo que es esencial para tomar decisiones confiables.

18. Menciona tres aplicaciones comunes de bases de datos en el mundo real.

Respuesta ejemplo:

- Gestión de clientes en empresas
- Mantenimiento de historiales médicos en salud
- Almacenamiento de perfiles y publicaciones en redes sociales



Tema 2 Diseño de Bases de Datos con el modelo Entidad - Relación.

2.1 El Proceso de Diseño

2.2 Modelo Entidad-Relación

2.3 Diseño con diagramas E-R

2.4 Modelo E-R extendido

2.5 La Notación E-R con UML.

2.1 El Proceso de Diseño

El **proceso de diseño de una base de datos** es un conjunto estructurado de fases destinadas a crear una base de datos eficiente, organizada y alineada con las necesidades de los usuarios y de la organización. Este proceso cuenta con varios pasos que garantizan que los datos estén completos, integrados y fácilmente accesibles. A continuación, se resumen las fases principales, según varios autores más relevantes y actualizados:

- **Recopilación y análisis de requisitos:** Consiste en identificar los datos que deben almacenarse, sus fuentes, los usuarios que accederán a la base de datos y las necesidades específicas de gestión e informes. Esta etapa se apoya en entrevistas, reuniones y análisis de procesos existentes para definir los requisitos de información y restricciones.
- **Diseño conceptual:** Se crea un *modelo de alto nivel* independiente del sistema de gestión de bases de datos a utilizar. Aquí se representan las entidades principales, sus atributos y relaciones, normalmente mediante diagramas entidad-relación (ER), lo que permite visualizar la estructura básica de la base de datos antes de preocuparse por detalles tecnológicos.
- **Elección del SGBD:** Se elige el sistema de gestión de base de datos (SGBD) más adecuado al proyecto, tomando en cuenta factores como volumen de datos,



necesidades de seguridad, presupuesto y experiencia técnica (por ejemplo, Oracle, MySQL, SQL Server, PostgreSQL).

- **Diseño lógico:** En esta fase se transforma el modelo conceptual en un *modelo lógico* adecuado al tipo de base de datos escogido. Se definen las tablas, filas, columnas y sus relaciones, y se identifican claves principales y foráneas. El modelo lógico depende del SGBD seleccionado.
- **Diseño físico:** Se decide cómo se almacenarán físicamente los datos. Aquí se eligen estructuras de almacenamiento específicas, índices, particiones y otras optimizaciones, y se redacta el código SQL necesario para implementar la base de datos en el motor seleccionado.
- **Implementación:** Es la materialización del diseño. Se crean las tablas, relaciones, restricciones y demás objetos en el SGBD, y se cargan los datos iniciales según lo planificado.
- **Perfeccionamiento y normalización:** Se revisa el funcionamiento de la base de datos con datos de prueba. Se aplican reglas de normalización para evitar la redundancia, mejorar la integridad y optimizar el almacenamiento. Se ajustan diseños según problemas detectados en esta etapa previa al uso en producción.

Tabla 4 Fases del proceso de diseño. Elaboración propia.

Fase	Objetivo principal
Recopilación/análisis	Identificar datos y necesidades del usuario
Diseño conceptual	Modelar entidades, atributos y relaciones a gran escala
Elección de SGBD	Seleccionar la tecnología de base de datos adecuada



Fase	Objetivo principal
Diseño lógico	Transformar el diseño conceptual en tablas y relaciones específicas
Diseño físico	Definir el almacenamiento real y la estructura interna
Implementación	Crear la base de datos en el SGBD seleccionado
Perfeccionamiento/Normalización	Revisar, optimizar y aplicar reglas para perfeccionar el modelo

En proyectos pequeños, algunas fases pueden solaparse o simplificarse, pero en bases de datos corporativas seguir cada una garantiza un modelo robusto y eficiente.

2.2 Modelo Entidad-Relación

El **Modelo Entidad-Relación (ER)** es una herramienta fundamental para el diseño de bases de datos que permite representar de manera gráfica y simplificada cómo personas, objetos o conceptos (llamados **entidades**) se relacionan entre sí dentro de un sistema de información.

Origen y concepto

Este modelo fue propuesto por **Peter Chen** en 1976 con el fin de proporcionar una visión unificada y clara de la estructura de los datos, facilitando el diseño y la comprensión de bases de datos mediante un esquema visual.

Elementos básicos del modelo entidad-relación



El modelo ER se compone principalmente de tres elementos: Entidades, Atributos y Relaciones.

2.2.1 Entidades y conjuntos de entidades.

- **Entidades:** Son las cosas, personas, objetos o conceptos sobre los cuales se desea almacenar información. Por ejemplo, en una base de datos de ventas, entidades pueden ser *Cliente*, *Producto*, *Factura* o *Dirección*. Gráficamente, las entidades se representan con **rectángulos**.
- **Atributos:** Son las propiedades o características que describen a cada entidad. Para la entidad *Cliente*, algunos atributos pueden ser *Nombre*, *Primer apellido* y *Fecha de nacimiento*. Estos se muestran como **óvalos** unidos a la entidad.
- **Conjuntos de Entidades:** es un conjunto de entidades del mismo tipo, es decir entidades que tienen los mismos atributos, por ejemplo: un conjunto de entidades *Alumno*.

Las entidades compartiendo las mismas propiedades se agrupan en conjuntos, los atributos que describen un conjunto de entidades definen su tipo.

Los conjuntos de entidades no son necesariamente disjuntos, Por ejemplo, es posible definir el conjunto de entidades de todos los maestros del tecnológico (maestros) y el conjunto de entidades de todos los alumnos (alumnos), y una entidad persona puede ser una entidad maestro, una entidad alumno, ambas cosas, o ninguna.



Figura 3 Conjunto de entidades débiles. Elaboración propia.

2.2.2 Relaciones y conjuntos de relaciones.

- **Relaciones:** Representan los vínculos o asociaciones entre entidades. Por ejemplo, la relación entre *Cliente* y *Producto* puede ser *compra* o *solicita*. Se representan con **rombos o diamantes**, y suelen expresarse con un verbo que define la naturaleza de la relación.

Conjunto de Relaciones: Un conjunto de relaciones es un grupo de relaciones del mismo tipo, por ejemplo, el conjunto de relaciones Cursó entre Alumnos y Materias:

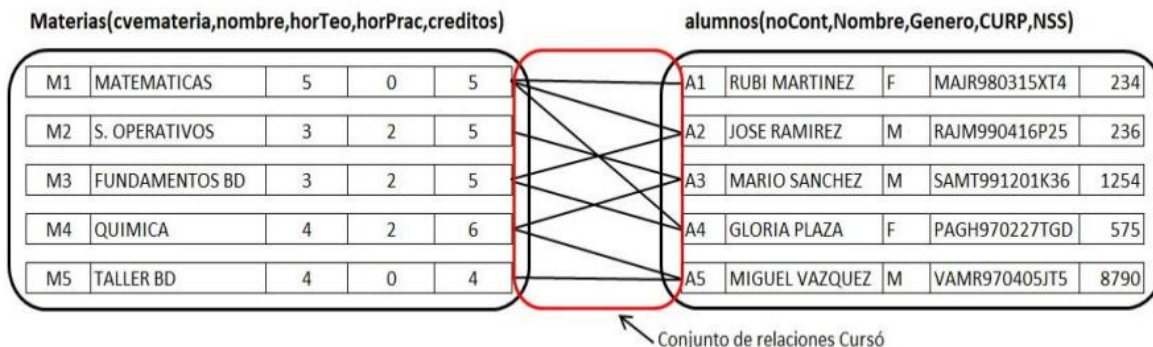


Figura 4 Conjunto de relaciones. Elaboración propia.

Una relación también puede tener atributos descriptivos. Los atributos descriptivos se utilizan para registrar información sobre la relación, en lugar de sobre cualquiera de las entidades participantes; por ejemplo, podríamos indicar la calificación que cada alumno obtuvo al cursar la materia.

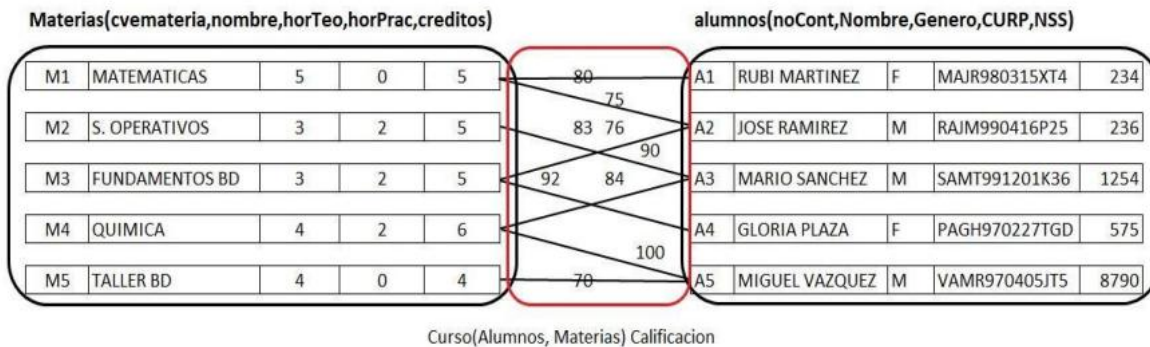


Figura 5. Atributos descriptivos. Elaboración propia.

Los conjuntos de entidades que participan en un conjunto de relaciones no son necesariamente distintos; a veces una relación puede involucrar dos entidades en el mismo conjunto de entidades. Por ejemplo, en el conjunto de relaciones prerequisite, que se muestra en la figura 6 entre entidades del conjunto de entidades materias:



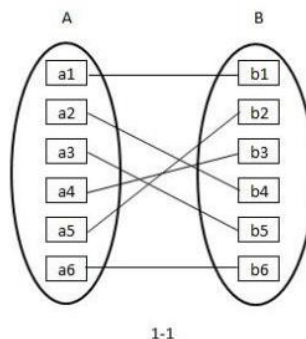
Figura 6 Conjunto de relaciones prerequisite.

Grado de una Relación: Es el número de entidades que participan en cada relación, la relación. Curso entre alumnos y materias es binaria o de grado 2, ya que en cada relación participa únicamente una entidad Alumno y una entidad Materia, el grado de la relación prerequisite también es 2 (binaria), ya que en cada relación participan 2 entidades materia, es posible tener relaciones que asocien a más de dos entidades, aunque es raro en la práctica.

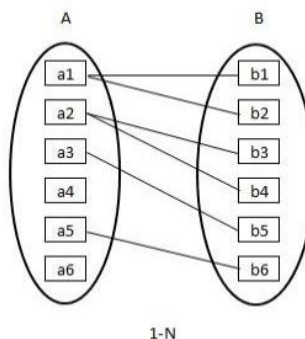


Restricciones: Nos indican la forma como están asociadas las diferentes entidades en una relación, estas son:

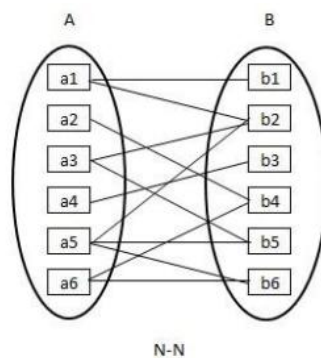
- **Cardinalidad.** - Expresa el número de entidades a las que una entidad puede estar asociada en un conjunto de relaciones, aunque existen conjuntos de relaciones que asocian entre sí a tres o más conjuntos de entidades, la gran mayoría de ellas son binarias por lo que en este momento nos vamos a enfocar en este tipo de relaciones. Para un conjunto de relaciones binarias R entre los conjuntos de entidades A y B, cardinalidad debe ser:
 - a) **Una a una.** Una entidad en A está asociada a lo sumo con una entidad en B, y una de B a lo sumo con una de A.



- b) **Una a varios.** Una entidad en A está asociada con un número cualquiera de entidades de B, pero una de B sólo puede estar asociada a una de A.



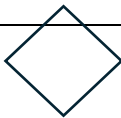
- c) **Varios a varios.** Una entidad en A está asociada con un número cualquiera en B, y una en B está asociada con un número cualquiera en A.



La cardinalidad adecuada para un conjunto de relaciones determinado es dependiente del mundo real que el conjunto de relaciones está modelando.

Representación gráfica

Tabla 5. Representación gráfica del modelo entidad-relación.

Elemento	Símbolo gráfico	Descripción
Entidad	Rectángulo 	Personas, objetos o conceptos
Atributo	Óvalo 	Propiedades de las entidades
Relación	Rombo (diamante) 	



Elemento	Símbolo gráfico	Descripción
		Conexión entre entidades (verbo)
Conexiones	 Líneas	Asociación entre entidades y atributos; pueden variar para indicar cardinalidades

El diagrama entidad-relación utiliza símbolos específicos para facilitar la interpretación visual:

Las líneas que conectan los elementos gráficamente pueden tener diferentes estilos para indicar tipos y cardinalidades de relación (uno a uno, uno a muchos, muchos a muchos).

Uso y aplicación

El modelo ER es principalmente usado en la fase de **diseño conceptual** de bases de datos, proporcionando un esquema claro y comprensible del dominio de información antes de la implementación física. Además, ayuda a identificar reglas, restricciones y dependencias entre datos.

Resumen de utilidad

- Facilita la visualización de la estructura de datos.
- Permite identificar claramente las entidades y sus relaciones.
- Proporciona un esquema para construir el modelo lógico y físico de la base de datos.



- Sirve para documentar y comunicar el diseño de datos entre desarrolladores y usuarios.

2.3 Diseño con diagramas E-R

El **Modelo Entidad-Relación** es una representación gráfica esencial para estructurar y entender cómo se organiza la información en una base de datos, siendo la base para un diseño organizado y eficiente.

El modelado de datos es un proceso descendente que comienza con la creación de un modelo conceptual para proporcionar una visión general, seguido por un modelo lógico y, finalmente, un diseño detallado del modelo físico. El modelo conceptual transforma una idea en una representación gráfica, similar a un diagrama de flujo utilizado por programadores. Las herramientas modernas de modelado de datos simplifican la creación y definición de bases de datos, así como de modelos lógicos y físicos. A continuación, se describen algunas técnicas y pasos comunes en el modelado de datos:

- **Identificar entidades y elaborar un diagrama de entidad-relación (ERD):** Las entidades representan "elementos clave de datos relevantes para el negocio". Por ejemplo, "cliente" y "ventas" son entidades. Un ERD muestra cómo estas entidades están interconectadas dentro de la organización y las relaciones que existen entre ellas.
- **Definir hechos, indicadores y dimensiones:** Los hechos son eventos o transacciones específicas en los datos, como la venta de un producto. Los indicadores son métricas cuantitativas, como ingresos, costos o cantidades. Las dimensiones son atributos cualitativos, como ubicaciones, fechas o descripciones.
- **Crear vistas de datos utilizando herramientas gráficas o consultas SQL:** Para quienes no están familiarizados con SQL, las herramientas gráficas son más intuitivas, ya que permiten arrastrar y soltar elementos para construir el modelo



visualmente. Estas vistas permiten combinar tablas o incluso otras vistas en una sola salida. Al arrastrar una fuente sobre otra ya vinculada a la salida, se puede establecer una unión entre las tablas.

2.4 Modelo E-R extendido

El Modelo Entidad-Relación Extendido (Modelo ER extendido o EER, por sus siglas en inglés) es una versión avanzada del modelo Entidad-Relación básica para el diseño de bases de datos. Se utiliza para representar de manera más precisa y compleja los requisitos y estructuras de bases de datos extensas o con características especiales.

Características principales del Modelo ER Extendido:

- Mayor nivel de abstracción: Permite describir con más detalle la distribución de los datos en el sistema.
- Incluye todos los conceptos básicos del modelo ER tradicional: entidades, relaciones, atributos.
- Incorpora extensiones o conceptos adicionales importantes:
 - Subtipos y supertipos (subclases y superclases)
 - Especialización (crear subgrupos dentro de una entidad)
 - Generalización (sintetizar varios conjuntos de entidades en uno de nivel superior basado en características comunes)
 - Jerarquías de entidades
 - Herencia de atributos (los subtipos heredan atributos del supertipo)
 - Agregación (tratar varias relaciones como una única entidad de nivel más alto)



- Categorías o tipos de unión (para modelar herencia múltiple en algunos casos)
- Soporta restricciones de integridad específicas para definir qué entidades pueden pertenecer a subtipos y con qué condiciones.

Este modelo es muy útil para representar bases de datos que requieren modelar con detalle jerarquías y relaciones complejas, como en programación orientada a objetos y sistemas con estructuras complejas de datos.

El Modelo ER Extendido es una herramienta de modelado más potente y expresiva que el modelo ER tradicional, diseñada para abordar las complejidades de bases de datos avanzadas y facilitar un diseño más claro y organizado.

2.5 La Notación E-R con UML.

La notación Entidad-Relación (ER) con UML integra dos enfoques fundamentales para el diseño y modelado de sistemas: el modelo ER, que se enfoca en la representación abstracta de datos y sus relaciones, y UML (Lenguaje Unificado de Modelado), usado para especificar varios componentes de un sistema de software.

Resumen de la Notación ER con UML:

- Representación de entidades y atributos: En UML, las entidades (llamadas clases) se representan mediante cuadros que incluyen sus atributos dentro de ellos, a diferencia del modelo ER tradicional que usa elipses para los atributos separados. Las clases en UML modelan objetos que, además de atributos, pueden tener métodos (funciones).
- Relaciones: Las relaciones binarias entre entidades se muestran con líneas que conectan las clases en UML. El nombre de la relación se escribe junto a la línea.



Roles y atributos de las relaciones pueden expresarse mediante cuadros conectados con líneas discontinuas a la relación, similar a la agregación en ER.

- Cardinalidad: Las restricciones de cardinalidad (mínimos y máximos participantes en una relación) se indican en UML con anotaciones como "0..1", "1..*", que representan el rango de participación. La posición de estas restricciones en UML es inversa al modelo ER clásico.
- Equivalencias y diferencias:
 - UML modela objetos (clases) con atributos y métodos, mientras que ER modela entidades con atributos.
 - La notación UML está más orientada a objetos y adecuada para un diseño detallado que abarca no solo datos sino también comportamiento y estructura del sistema.
 - Relaciones no binarias en ER deben convertirse en relaciones binarias para representar en UML.
- Ventajas: La combinación de ER con UML permite un diseño conceptual y lógico más claro y detallado, integrando tanto la perspectiva de datos como la de comportamiento, facilitando la comunicación entre desarrolladores y diseñadores

La notación ER con UML utiliza diagramas de clases UML para representar entidades y relaciones de manera más orientada a objetos, aportando claridad visual y detallando tanto estructura como comportamiento de sistemas. Esto hace que sea una herramienta poderosa para modelar sistemas complejos que requieren un diseño integrado de datos y funcionalidad.



Ejemplo de diagrama de clases UML

Un diagrama de clases representa las clases en un sistema, sus atributos y métodos, así como las relaciones entre ellas. Por ejemplo, para un sistema de alquiler de coches online, podríamos tener:

- Clase Usuario
 - Atributos: id_usuario, nombre_usuario, contraseña, contacto
 - Métodos: iniciar_sesión(), cerrar_sesión(), solicitud_salida()
- Director de orquesta
 - Atributos: conductor_id, nombre, contacto, estado (disponible o no)
 - Métodos: aceptar_viaje()
- Clase Viaje
 - Atributos: id_viaje, ubicación_recogida, ubicación_entrega, tarifa
 - Métodos: calcular_tarifa()
- Clase Vehículo
 - Atributos: vehículo_id, conductor_id, modelo, estado (disponible o reservado)

Las relaciones entre estas clases se mostrarían, por ejemplo, que un Conductor maneja uno o varios Vehículos, y un Usuario puede solicitar varios Viajes.

Un diagrama UML de clases incluye los siguientes elementos básicos:

1. Clase: Representada como un rectángulo dividido en tres secciones:
 - Sección superior: Contiene el nombre de la clase (obligatorio).
 - Sección central: Contiene los atributos o propiedades de la clase (opcional).



- Sección inferior: Contiene los métodos o funciones (operaciones) de la clase (opcional).
2. Atributos: Características o propiedades que describen la clase. Cada atributo se anota con su visibilidad, tipo de dato y nombre.
 3. Métodos (Operaciones): Las funciones o comportamientos que puede realizar la clase. También llevan visibilidad y parámetros si es necesario.
 4. Visibilidad: Se indica con símbolos antes del nombre del atributo o método:
 - Público: +
 - Privado: -
 - Protegido: #
 - Paquete (predeterminado): ~
 5. Relaciones entre clases: Representadas con líneas y flechas, incluyen:
 - Asociación: relaciones simples entre clases.
 - Herencia (Generalización): flecha abierta que muestra la relación "es un".
 - Agregación/Composición: relaciones "tiene un" con diferentes grados de dependencia.
 - Dependencia: línea discontinua que muestra que una clase usa a otra.
 6. Paquetes: Agrupan clases relacionadas dentro de un rectángulo con pestañas para organizar el diagrama.
 7. Interfaces: Son conjuntos de operaciones que una clase puede implementar, se representan generalmente con un estereotipo «interfaz».

Otros elementos posibles incluyen objetos (instancias de clases), enumeraciones y señales para comunicaciones asincrónicas, pero lo esencial son las clases con sus atributos, métodos y relaciones.



El diagrama de clases UML sirve para mostrar la estructura estática de un sistema orientado a objetos a través de clases, atributos, métodos y sus relaciones.



Complemento educacional CE. 2.1 Proyecto integrador

Al finalizar el curso deberá generar el proyecto integrador de una base de datos para un sistema de información analizado y diseñado en iniciando con la programación del modelo de datos, la creación de la base de datos. el Si podrá ser hasta de 3 integrantes por equipo.

Como primera entrega deberá utilizar una técnica de ingeniería de requisitos para obtener la información que dé lugar a la propuesta de SI, presente sus resultados en un informe técnico de acuerdo con lo visto en esta unidad plenaria en el grupo de clases con los siguientes puntos:

- Definición del problema o área de oportunidad.
- Diagrama Entidad – Relación
- Identificando en el diagrama cada una de las claves primarias y los diferentes tipos de entidades y sus relaciones.



Rúbrica para la evaluación del Complemento educacional del proyecto integrador.

Criterio de Evaluación	Excelente	Bueno	Aceptable	Bajo
Comprensión de conceptos	Demuestra comprensión completa y precisa de los conceptos.	Comprende adecuadamente los conceptos, aunque con algunas imprecisiones.	Muestra comprensión básica, con dificultades para aplicar los conceptos.	Comprensión limitada o incorrecta de los conceptos.
Aplicación práctica	Aplica los conceptos a situaciones reales de manera excepcional.	Aplica los conceptos correctamente, aunque con algunas dificultades.	Aplica los conceptos, pero con errores significativos.	No logra aplicar los conceptos a situaciones reales.
Uso de herramientas tecnológicas	Utiliza herramientas tecnológicas de manera efectiva y creativa.	Utiliza las herramientas adecuadamente, con algunos errores menores.	Utiliza herramientas, pero con dificultades notables.	No utiliza las herramientas recomendadas o lo hace incorrectamente.
Calidad del producto final	El producto final es de alta calidad y cumple todos los requisitos.	El producto final es adecuado y cumple la mayoría de los requisitos.	El producto final es básico y cumple algunos requisitos.	El producto final no cumple con los requisitos establecidos.
Colaboración y trabajo en equipo	Trabaja excepcionalmente bien en equipo, contribuyendo significativamente.	Colabora adecuadamente en el equipo, aunque con algunas limitaciones.	Participa en el equipo, pero su contribución es mínima.	No colabora efectivamente en el trabajo en equipo.



Tema 3: Modelo relacional

3.1 Introducción al modelo relacional

El modelo relacional es un enfoque para organizar y representar datos en bases de datos mediante tablas, conocidas también como relaciones. Este modelo fue propuesto por Edgar F. Codd en 1970 y se fundamenta en la teoría de conjuntos y la lógica de predicados. Las bases de datos relacionales almacenan la información en tablas donde:

- Cada tabla representa una relación.
- Cada fila (tupla) corresponde a un registro único.
- Cada columna (atributo) representa una característica o propiedad del dato.

El modelo relacional permite establecer relaciones entre tablas a través de claves primarias y extranjeras, lo que facilita la integridad y consistencia de los datos. Su objetivo principal es abstraer al usuario de la estructura física de los datos, permitiendo una gestión eficiente y flexible.

Además, las bases de datos relacionales emplean el lenguaje SQL (Structured Query Language) para la manipulación y consulta de los datos, facilitando operaciones complejas como búsquedas avanzadas, actualizaciones y análisis.

Este modelo es ampliamente utilizado en diversos sectores y sistemas como banca, comercio electrónico y salud, con ejemplos de sistemas de gestión relacional como MySQL, Oracle Database, Microsoft SQL Server y PostgreSQL.

El modelo relacional:

- Organiza datos en tablas con filas y columnas.
- Garantiza la integridad y evita duplicidades.
- Facilita consultas complejas con SQL.
- Permite relaciones robustas entre datos mediante claves.



- Es el modelo predominante en gestión de bases de datos actualmente.

Así, el modelo relacional es fundamental para almacenar, gestionar y acceder a grandes volúmenes de datos de manera estructurada, coherente y segura.

3.2 Conversión de Modelo E-R a Modelo relacional

1. **Entidades:** Cada entidad del modelo E-R se convierte en una tabla. Los atributos de la entidad se transforman en columnas de esa tabla, y los atributos clave primaria se designan como claves primarias en la tabla.
2. **Entidades débiles:** Se convierten en tablas que incluyen sus atributos y la clave primaria de la entidad fuerte de la que dependen, formando parte de su clave primaria.
3. Relaciones:
 - Relaciones 1:N: La clave primaria de la entidad con cardinalidad 1 se propaga como clave foránea en la tabla de la entidad con cardinalidad N. Si la relación tiene atributos propios, éstos se añaden a la tabla que recibe la clave foránea.
 - Relaciones N:M: Se crea una tabla nueva que incluye las claves primarias de ambas entidades como clave primaria compuesta, junto con los atributos de la relación si los hay.
 - Relaciones 1:1: Se añade la clave primaria de una de las entidades como clave foránea en la otra tabla, manteniendo la restricción de unicidad.
4. **Atributos multivaluados:** Se manejan creando tablas separadas que incluyan la clave primaria de la entidad original más el atributo multivaluado.



5. **Generalización/Especialización:** Se puede modelar creando tablas separadas para la superclase y subclases, propagando claves primarias o utilizando un atributo discriminador, dependiendo del caso.

Este proceso convierte el modelo conceptual abstracto del E-R en un esquema relacional que puede ser implementado y gestionado mediante bases de datos relacionales, manteniendo la integridad y las relaciones de los datos.

Las reglas básicas para transformar entidades, relaciones y atributos, así como consideraciones para relaciones de diferentes cardinalidades y tipos de atributos especiales. Es una guía clave para diseñar bases de datos relacionales a partir de diagramas E-R.

Las ventajas de convertir un diagrama Entidad-Relación (E-R) en un esquema relacional para bases de datos son las siguientes:

- **Facilita la implementación práctica:** El esquema relacional se puede implementar directamente en un sistema de gestión de bases de datos relacional (DBMS), mientras que el modelo E-R es más conceptual y abstracto.
- **Representación estructurada:** El modelo relacional organiza los datos en tablas con filas y columnas, lo que permite un almacenamiento y acceso eficientes.
- **Integridad y consistencia:** Al convertir las entidades y relaciones en tablas con claves primarias y foráneas, se aseguran las restricciones y la integridad referencial de los datos en la base.
- **Flexibilidad para consultas:** El esquema relacional es compatible con SQL, lo que facilita la recuperación y manipulación avanzada de datos mediante consultas estructuradas.
- **Base para normalización:** La conversión permite aplicar técnicas de normalización para eliminar redundancias y mejorar la eficiencia del almacenamiento.



- Mayor comprensión y comunicación: El diagrama E-R facilita la representación visual y el entendimiento del diseño, y su conversión al modelo relacional traduce esta representación a una forma lógica lista para implementarse.
- Compatibilidad con herramientas y tecnologías actuales: La mayoría de los DBMS utilizan el modelo relacional, por lo que la conversión es clave para aprovechar sus funcionalidades.
- Simplificación en el mantenimiento y escalabilidad: El esquema relacional facilita modificaciones y ampliaciones posteriores al diseño inicial.

Convertir un diagrama E-R a un esquema relacional permite transformar un modelo conceptual intuitivo en un diseño lógico y práctico que puede implementarse, manejarse y consultarse eficientemente en bases de datos modernas.

3.3 Esquema de la base de datos

Un esquema de base de datos es la estructura o el diseño que define cómo se organizan, relacionan y almacenan los datos en una base de datos. En particular, en una base de datos relacional, el esquema describe:

- Las tablas que componen la base de datos.
- Los campos o columnas presentes en cada tabla.
- Las relaciones entre las tablas mediante claves primarias y foráneas.
- Los tipos de datos de cada campo.
- Restricciones e integridad de los datos (por ejemplo, reglas de unicidad y dependencias).

El esquema actúa como un "plano" o marco que sirve para gestionar cómo se guardan y acceden los datos, pero no contiene los datos en sí. Se representa generalmente con diagramas y es una parte fundamental del diseño de bases de datos, ayudando a los



administradores y programadores a entender la estructura lógica y física del sistema de datos.

Se puede hablar de distintos niveles de esquemas, como el esquema conceptual (representación general), el esquema lógico (detalles de tablas y relaciones) y el esquema físico (cómo se almacenan físicamente los datos).

El esquema es el diseño técnico que organiza y define la estructura de una base de datos para asegurar un manejo eficiente, coherente y seguro de la información.

3.4 Restricciones

Las restricciones en el modelo relacional de bases de datos son reglas que garantizan la coherencia, integridad y consistencia de los datos almacenados. Se divide en dos grandes tipos: restricciones inherentes al modelo y restricciones semánticas definidas por el usuario.

1. Restricciones inherentes al modelo relacional:

- Cada tabla representa una relación que es un conjunto de tuplas (filas), por lo que no pueden existir tuplas duplicadas.
- El orden de las tuplas y de los atributos no importa.
- Cada atributo en una tupla solo puede tener un valor atómico del dominio definido para ese atributo (un solo valor, no múltiple).
- La clave primaria es obligatoria y no puede tener valores nulos, garantizando unicidad y obligatoriedad en la identificación de cada tupla.

2. Restricciones semánticas (definidas por el usuario):

- Clave primaria (PRIMARY KEY): Un atributo o conjunto de atributos que identifican unívocamente a una tupla. No se permiten valores repetidos ni nulos.



- Clave alternativa o candidata: Atributos que podrían servir como clave primaria y también deben ser únicos y no nulos.
- Integridad referencial (FOREIGN KEY): Defina que un atributo en una tabla está relacionado con la clave primaria de otra tabla, asegurando que los valores existan previamente en la tabla referenciada o sean nulos.
- Unicidad (UNIQUE): Impide valores duplicados en un conjunto de atributos.
- Obligatoriedad de valores (NOT NULL): Atributos que no pueden tener valores nulos.
- Validación (CHECK): Reglas que verifican que los datos cumplen ciertas condiciones específicas.
- Desencadenantes: Reglas o eventos que se activan ante ciertas operaciones para mantener la integridad.

Estas restricciones permiten que los datos reflejen fielmente la realidad modelada y eviten inconsistencias en la base de datos.

Las restricciones en el modelo relacional aseguran la integridad de la entidad, la integridad referencial, la unicidad, y controlan los valores permitidos para los atributos según reglas definidas por el modelo y el usuario. Estos son fundamentales para el correcto diseño, almacenamiento y manipulación de los datos en bases relacionales.

3.5 Integridad de dominio

La integridad de dominio en una base de datos relacionales consiste en garantizar que los valores almacenados en cada columna (atributo) de una tabla sean válidos y pertenezcan al conjunto permitido o dominio definido para esa columna. Esto incluye aspectos como:

- El tipo de dato permitido (por ejemplo, entero, cadena de texto, fecha).
- La longitud o tamaño máximo del valor.



- El rango de valores numéricos permitidos.
- La posibilidad o no de que el valor sea nulo.
- Conjuntos específicos de valores válidos (por ejemplo, listas definidas por el usuario).
- Reglas de formato y validaciones específicas a través de restricciones CHECK.

Este tipo de integridad previene que se ingresen datos incorrectos o inconsistentes que no cumplen con las reglas del dominio definidas para ese atributo, asegurando que solo se almacenen datos coherentes con el modelo.

Por ejemplo, si un atributo "edad" está definido con un dominio que permite solo números enteros entre 16 y 65, no se podrá insertar un valor fuera de ese rango ni texto. Si el atributo "DNI" es un entero, no se podrá ingresar una cadena como "Luis".

La integridad de dominio es clave para mantener la calidad, precisión y consistencia de los datos en la base de datos, y es una funcionalidad que el sistema gestor de bases de datos relacional (RDBMS) aplica automáticamente mediante la definición de tipos de datos, restricciones NOT NULL, CHECK y otras reglas.

La integridad de dominio consiste en que cada valor almacenado en la base de datos debe pertenecer al dominio declarado para esa columna, respetando su tipo, formato, rango y reglas asociadas. Esto protege la base de datos contra datos erróneos e inconsistentes.



Ejemplos prácticos que ilustran la importancia de la integridad de dominio en aplicaciones reales:

- En una base de datos de empleados con atributos como DNI, nombre, apellido y edad, el atributo DNI puede estar definido para aceptar solo números enteros. La integridad de dominio impide que se introduzca un DNI con valor textual como "Luis". Esto evita errores graves de tipo de dato y asegura que los datos tengan sentido para la aplicación.



- En la misma base de datos de empleados, el atributo edad puede tener un dominio definido para permitir solo enteros entre 16 y 65 años. La integridad de dominio impide que se registre un empleado con 90 años, garantizando que la información sea coherente con las reglas de negocio y evite datos erróneos.
- La integridad de dominio también evita consultas lógicas inválidas. Por ejemplo, si se consulta a los empleados cuyo DNI sea igual a un nombre como "Elena", el sistema rechazaría la consulta porque esos valores pertenecen a dominios incompatibles, evitando confusión y errores en los datos.
- En aplicaciones bancarias, se utiliza para asegurar que los campos como montos de transacciones o números de cuenta cumplan con formatos y rangos válidos, previniendo el ingreso de datos inválidos y protegiendo la precisión financiera y la confianza.
- En el ámbito sanitario, la integridad de dominio garantiza que campos como números de historia clínica, fechas de consulta o valores de laboratorio estén dentro de parámetros válidos, lo que es crítico para la seguridad y atención adecuada del paciente.

Estos ejemplos muestran cómo la integridad de dominio actúa como una primera barrera para mantener la calidad, validez y coherencia en los datos almacenados y manipulados en las bases de datos, facilitando que las aplicaciones funcionen correctamente y que las decisiones basadas en datos sean confiables. La integridad de dominio también ayuda a detectar errores desde el ingreso, evitando consecuencias posteriores más costosas.

Así, la integridad de dominio es fundamental para asegurar que los datos respeten el tipo, formato, rango y reglas definidas por el negocio o el sistema manejador de base de datos, haciendo que las aplicaciones reales sean robustas y confiables.



Examen sobre el Modelo Relacional de Base de Datos

Preguntas

1. ¿Qué es un modelo relacional de base de datos?
2. ¿Qué es una relación en el modelo relacional?
3. ¿Qué entiendes por tupla y atributo en el modelo relacional?
4. ¿Qué es una clave primaria y cuál es su función?
5. ¿Cuáles son las principales restricciones del modelo relacional?
6. ¿Qué es la integridad de dominio? Da un ejemplo.
7. ¿Qué significa la integridad referencial?
8. Explique la diferencia entre una clave primaria y una clave foránea.
9. ¿Por qué es importante la unicidad en las claves?
10. ¿Qué sucede si se rompe la integridad referencial en una base de datos relacional?

Respuestas

1. **Modelo relacional de base de datos** : Es un modelo para organizar y estructurar datos mediante el uso de tablas llamadas relaciones, donde cada tabla está formada por filas (tuplas) y columnas (atributos).
2. **Relación** : Es una tabla que representa un conjunto de entidades u objetos con atributos específicos y contiene un conjunto de tuplas (filas) que representan los datos.



3. **Tupla y atributo:** Una tupla es una fila dentro de la tabla que representa un registro completo. Un atributo es una columna de la tabla que define una característica o propiedad de la relación.
4. **Clave primaria:** Es un atributo o conjunto de atributos que identifica unívocamente cada tupla en una relación, garantizando que no haya duplicados ni valores nulos en esa columna o conjunto de columnas.
5. **Principales restricciones:** Integridad de entidad (clave primaria única y no nula), integridad referencial (claves foráneas referencian claves primarias existentes), integridad de dominio (valores de atributos deben cumplir el tipo y rango definido), unicidad y no nulidad.
6. **Integridad de dominio:** Asegura que los valores almacenados en un atributo pertenecen a un dominio válido, como tipo de dato, rango o formato. Ejemplo: El atributo “edad” solo acepta valores enteros entre 0 y 120.
7. **Integridad referencial:** Garantizar que un valor de clave foránea en una tabla debe existir previamente en la tabla referenciada como clave primaria o ser nulo, manteniendo la consistencia entre tablas relacionadas.
8. **Diferencia entre clave primaria y clave foránea:** La clave primaria identifica unívocamente una fila dentro de su propia tabla, mientras que la clave foránea es un atributo que hace referencia a la clave primaria de otra tabla para mantener relaciones.
9. **Importancia de la unicidad en las claves:** La unicidad evita duplicados y asegura que cada registro puede ser identificado de forma inequívoca, facilitando consultas, actualizaciones y eliminación precisas.
10. **Si se rompe la integridad referencial:** Se generarán datos inconsistentes o huérfanos, lo que puede provocar errores en las consultas, pérdida de datos relacionados o fallos en la lógica del negocio.



Tema 4: Normalización de Bases de Datos

4.1 Conceptos básicos

La **normalización** es un proceso fundamental en el diseño de bases de datos relacionales. Su objetivo principal es **mejorar el esquema lógico de la base de datos** para satisfacer ciertas restricciones que **minimizan la redundancia de datos y facilitan su gestión posterior**. Esto ayuda a proteger la integridad de los datos y disminuir los problemas al actualizarlos.

El proceso implica someter un esquema de relación a una serie de pruebas para "certificar" si pertenece o no a una cierta forma normal. Si una regla no se cumple, la relación debe descomponerse en varias relaciones que sí la cumplan. Cada regla que se cumple aumenta el grado de normalización, lo que hace que las relaciones sean menos vulnerables a las **anomalías de actualización** (inserción, borrado y modificación).

Problemas que resuelve la normalización: Un diseño de base de datos inadecuado puede presentar varios problemas:

- **Redundancia de datos:** Los mismos datos se repiten innecesariamente.
- **Anomalías de modificación (actualización):** Dificultad para cambiar datos sin generar inconsistencias, ya que la modificación en una tupla podría no replicarse en otras.
- **Anomalías de inserción:** Imposibilidad de agregar cierta información si no se cumplen condiciones de clave primaria o si se necesita insertar múltiples tuplas para un solo hecho.
- **Anomalías de borrado:** La eliminación de una tupla puede llevar a la pérdida de información relacionada que no debería desaparecer.
- **Incapacidad para almacenar ciertos hechos.**
- **Ambigüedades.**
- **Pérdida de información (tuplas espúreas):** Puede aparecer información incorrecta al unir tablas.



- **Pérdida de dependencias funcionales.**
- **Existencia de valores nulos (inaplicables).**

4.2 Primera forma normal.

Formas Normales (FN)

Las formas normales son un conjunto de reglas progresivas que una tabla debe cumplir para asegurar un buen diseño de la base de datos. Las primeras tres fueron propuestas por E.F. Codd.

1. Primera Forma Normal (1FN):

- **Definición:** Una relación está en 1FN si **todos los atributos son atómicos** (contienen un único valor indivisible) y **no hay grupos repetitivos**.
- **Características:**
 - Cada tabla debe tener su nombre único.
 - No puede haber dos filas iguales (no se permiten duplicados).
 - Todos los datos en una columna deben ser del mismo tipo.
 - Debe existir una clave primaria.
 - No debe haber variación en el número de columnas.
 - Independencia del orden de filas y columnas.
- **Problema que aborda:** Elimina la repetición de grupos de datos, ya sea a través de múltiples columnas o como una lista de valores dentro de una sola columna.
- **Solución:** Descomponer la tabla en relaciones donde cada atributo contenga un solo valor.
- **Es obligatoria** para el modelo relacional.



4.3 Dependencias funcionales y transitivas.

Conceptos Clave

La normalización se basa en reglas relacionadas con las **claves primarias** y las **dependencias funcionales**.

1. Dependencia Funcional (DF):

- Es una **restricción entre dos conjuntos de atributos** de la base de datos.
- Se representa como $X \rightarrow Y$, donde X es el "determinante" o "miembro izquierdo" e Y es el "determinado" o "miembro derecho".
- Significa que **cada valor de X tiene asociado en todo momento un único valor de Y**. Es decir, si dos tuplas coinciden en los valores de X, deben coincidir en los valores de Y.
- **Utilidad:** Describir un esquema de relación especificando restricciones que deben cumplirse.
- **Tipos de Dependencias Funcionales:**
 - **DF Trivial:** Una DF $X \rightarrow Y$ es trivial si Y es un subconjunto de X ($Y \subseteq X$). Por ejemplo, $\text{cod_libro} \rightarrow \text{cod_libro}$.
 - **DF Completa:** Si X es un descriptor compuesto (ej. $X = \{X1, X2\}$), Y tiene dependencia funcional completa respecto de X si depende funcionalmente de X, pero no de ningún subconjunto propio de X. Se representa como $X \Rightarrow Y$.
 - **DF Parcial:** Una DF $X \rightarrow Y$ es parcial si es posible eliminar un atributo A de X y la dependencia sigue siendo válida. Esto ocurre cuando un atributo no clave depende solo de una *parte* de una clave primaria compuesta.
 - **DF Transitoria:** Sean X, Y, Z tres atributos (o grupos de atributos). Si $X \rightarrow Y$ y $Y \rightarrow Z$, pero **Y no es una clave candidata** (o no $X \rightarrow Y$), entonces Z depende transitivamente de X. Es decir, un atributo no



clave (Z) depende de otro atributo no clave (Y), que a su vez depende de la clave primaria (X). La 3FN busca eliminar este tipo de problemas.

- **Reglas de Inferencia (Axiomas de Armstrong):** Son reglas fundamentales para deducir nuevas dependencias funcionales a partir de un conjunto dado.
 - **Reflexividad (RI1):** Si $Y \subseteq X$, entonces $X \rightarrow Y$.
 - **Aumento (RI2):** Si $X \rightarrow Y$, entonces $XZ \rightarrow YZ$.
 - **Transitividad (RI3):** Si $X \rightarrow Y$ y $Y \rightarrow Z$, entonces $X \rightarrow Z$.
 - Otras reglas derivadas incluyen Descomposición ($X \rightarrow YZ$ implica $X \rightarrow Y$), Unión ($X \rightarrow Y$ y $X \rightarrow Z$ implica $X \rightarrow YZ$), y Pseudotransitividad ($X \rightarrow Y$ y $WY \rightarrow Z$ implica $WX \rightarrow Z$).

2. Claves:

- **Superclave:** Un subconjunto de atributos de una relación que puede identificar de manera única cada tupla.
- **Clave (Clave Candidata):** Una superclave que es **mínima**; si se le quita cualquiera de sus atributos, deja de ser una superclave. Una relación puede tener varias claves candidatas.
- **Clave Primaria:** Una de las claves candidatas elegida arbitrariamente para identificar las tuplas de una relación. Se aconseja que tenga el menor número de columnas.
- **Atributo Primo:** Cualquier atributo que forma parte de una clave candidata.
- **Atributo No Primo (No clave):** Un atributo que no es miembro de ninguna clave.

4.4 Segunda Forma Normal (2FN):

- **Definición:** Una relación está en 2FN si está en **1FN** y **ningún atributo no primo depende parcialmente de una clave de la relación**. Es decir, todos



los atributos que no son parte de ninguna clave candidata deben depender funcionalmente de *toda* la clave primaria.

- **Problema que aborda:** Las dependencias parciales, que causan redundancia y anomalías.
- **Solución:** Descomponer la tabla para que los atributos no primos se asocien únicamente con la parte de la clave principal de la que dependen completamente.

4.5 Tercera Forma Normal (3FN):

- **Definición:** Una relación está en 3FN si está en **2FN** y **ningún atributo no primo depende transitivamente de una clave de la relación**.
Formalmente, para toda $DF\ X \rightarrow A$, X es una superclave de la relación O A es un atributo primo.
- **Problema que aborda:** Las dependencias transitivas, donde un atributo no clave depende de otro atributo no clave que, a su vez, depende de la clave.
- **Solución:** Separar la tabla en nuevas tablas, donde cada atributo no clave dependa directamente de la clave primaria, no de otro atributo no clave.
- **Es el nivel de normalización recomendado** para la mayoría de las bases de datos para evitar anomalías de actualización.

4.6 Forma Normal de Boyce-Codd (FNBC o BCNF):

- **Definición:** Una relación está en BCNF si, para toda dependencia funcional no trivial $X \rightarrow A$, **X es una superclave de la relación**.
- **Relación con 3FN:** Es una forma normal **más estricta que la 3FN**. Una relación en BCNF siempre está en 3FN, pero no viceversa.
- **Diferencia clave con 3FN:** La BCNF no permite la condición de la 3FN de que A pueda ser un atributo primo si X no es una superclave. La BCNF es diferente de 3NF cuando existen dos o más claves candidatas y más de un atributo en las claves candidatas.



- **Problema que aborda:** Anomalías que aún pueden existir en 3FN, especialmente cuando hay claves candidatas que se solapan.

2. Cuarta Forma Normal (4FN):

- **Definición:** Una tabla está en 4FN si está en 3FN y **todas las dependencias multivaluadas no funcionales ($X \twoheadrightarrow Y$) en ella tienen a X como superclave.**
- **Dependencia Multivaluada (DMV):** Una generalización de las dependencias funcionales, donde un conjunto de valores del determinado (Y) son determinados por un determinante (X), independientemente de otros atributos. Se denota como $X \twoheadrightarrow Y$.
- **Problema que aborda:** Ocurre cuando una relación contiene dos o más atributos multivaluados independientes.

4.7 Otras formas normales.

Quinta Forma Normal (5FN) o Forma Normal de Proyección-Reunión (FNDR):

- **Definición:** Una tabla está en 5FN si está en 4FN y su información **no puede ser descompuesta en varias relaciones menores sin pérdida**, a menos que la descomposición esté definida por claves candidatas (es decir, todas las dependencias de reunión no triviales están implicadas por claves candidatas).
- **Uso:** Es la menos usada y solo se aplica en casos muy concretos, generalmente en tablas con muchas columnas o donde las uniones puedan generar redundancia/incoherencia.
- **Problema que aborda:** Redundancias restantes en casos complejos donde los hechos multivaluados no son independientes, que no fueron cubiertos por formas normales anteriores.



Propiedades Deseables de una Descomposición

Además de cumplir con las formas normales, un buen diseño de base de datos busca que la descomposición de las relaciones posea ciertas propiedades:

1. Descomposición y Reunión sin Pérdidas (Lossless-Join Decomposition - JSP):

- Garantiza que **no se generarán tuplas espúreas (falsas)** cuando se aplique una operación de **join natural** a las relaciones resultantes de la descomposición.
- Significa que, al unir las relaciones descompuestas, se recupera exactamente la relación original sin agregar información incorrecta.

2. Preservación de Dependencias (Dependency Preservation):

- Asegura que **todas las dependencias funcionales originales se conserven** en la descomposición, ya sea directamente en una de las nuevas relaciones o que puedan inferirse de las dependencias que aparecen en ellas.
- Esto es crucial porque las dependencias funcionales representan restricciones de integridad del mundo real.

En el proceso de diseño, la **compatibilidad entre estas propiedades y las formas normales no siempre es perfecta**. Por ejemplo, un esquema en BCNF podría no siempre preservar todas las dependencias fácilmente. Los algoritmos de diseño buscan equilibrar estas propiedades para obtener un esquema robusto y eficiente.



Complemento educativo CE. 4.1 Explorando la Normalización de Bases de Datos

Objetivo General: El estudiante investigará a fondo el concepto de normalización en bases de datos relacionales, identificará las reglas y el propósito de cada forma normal, analizará las anomalías que cada una resuelve, y elaborará un mapa conceptual que sintetice su comprensión.



Público Objetivo: Estudiantes de bases de datos, profesionales que buscan reforzar conocimientos o cualquier persona interesada en el diseño de bases de datos relacionales.

Duración Estimada: 4-6 horas (incluyendo investigación y elaboración del mapa conceptual).

Materiales Necesarios:

- Acceso a las fuentes de información proporcionadas (los documentos "Dependencia Transitivas con Ejemplos | Base de Datos - Informático Sin Límites", "Modelo Relacional: Dependencias Funcionales y Normalización - Piazza", "Normalización - BASES DE DATOS", "Normalización - FING", "Primera forma normal - Wikipedia, la enciclopedia libre", "Quinta Forma Normal Para Principiantes + Ejemplos | Base de Datos" y "[AYUDA] Diferencia entre 3NF y BCNF : r/SQL - Reddit").
- Herramienta para crear mapas conceptuales (ej. CmapTools, Miro, Lucidchart, o papel y lápiz).

Instrucciones Detalladas de la Actividad:

Fase 1: Fundamentos de la Normalización (Tiempo estimado: 1 hora)

1. Investiga el concepto de Normalización:

- Lee las secciones introductorias de las fuentes.
- **Define la normalización** de bases de datos relacionales, destacando su objetivo principal: **minimizar la redundancia de datos y proteger la integridad.**
- Identifica **qué problemas o anomalías de diseño resuelve la normalización.** Enumera y explica cada una de las siguientes anomalías, proporcionando un breve ejemplo o descripción de cómo se manifiestan:



- **Redundancia de datos y posibilidades de inconsistencias:**
cuando la misma información se repite, lo que puede llevar a datos contradictorios si no se actualiza en todos los lugares.
- **Anomalías de modificación (actualización):** Dificultad para cambiar datos sin generar inconsistencias en otras tuplas.
- **Anomalías de inserción:** Imposibilidad de agregar cierta información si no se cumplen condiciones de clave primaria o si se necesita insertar múltiples tuplas para un solo hecho.
- **Anomalías de borrado:** La eliminación de una tupla puede llevar a la pérdida de información relacionada que no debería desaparecer.
- **Incapacidad para almacenar ciertos hechos y existencia de valores nulos (inaplicables):** Cuando el diseño de la tabla impide almacenar información relevante o requiere el uso excesivo de valores nulos.
- **Ambigüedades y pérdida de información (tuplas espúreas):**
Cuando la estructura del esquema puede llevar a interpretaciones incorrectas o a la generación de tuplas falsas al combinar información.

2. Comprende los Conceptos Clave para la Normalización:

- Investiga los siguientes términos, que son la base de las reglas de normalización:
 - **Dependencia Funcional (DF):** Explica que es una restricción entre dos conjuntos de atributos ($X \rightarrow Y$) y que significa que **cada valor de X tiene asociado en todo momento un único valor de Y**. Menciona su utilidad para describir un esquema de relación.
 - **DF Trivial:** Si Y es un subconjunto de X.
 - **DF Completa:** Y depende funcionalmente de X, pero no de ningún subconjunto propio de X.



- **DF Parcial:** Cuando un atributo no clave depende solo de una *parte* de una clave primaria compuesta.
- **DF Transitoria:** Define esta dependencia como $X \rightarrow Y$ y $Y \rightarrow Z$, **donde Y no es una clave candidata** (o X no depende de Y directamente, según la fuente), y Z depende transitivamente de X. Proporciona el ejemplo de Evento $\rightarrow$ Cantantes y Cantantes $\rightarrow$ Genero para ilustrarla.
- **Claves:**
 - **Superclave:** Un subconjunto de atributos que identifica de manera única cada tupla.
 - **Clave (Candidata):** Una superclave que es *mínima*.
 - **Clave Primaria:** La clave candidata elegida para identificar tuplas.
 - **Atributo Primo:** Cualquier atributo que forma parte de una clave candidata.
 - **Atributo No Primo (No clave):** Un atributo que no es miembro de ninguna clave.

Fase 2: Reglas de las Formas Normales y Anomalías Resueltas (Tiempo estimado: 2-3 horas)

Investiga cada una de las siguientes formas normales, describiendo su definición (regla) y las anomalías específicas que resuelve. Incluye ejemplos claros para cada una:

1. Primera Forma Normal (1FN):

- **Regla:** Una relación está en 1FN si **todos los atributos son atómicos** (contienen un único valor indivisible) y **no hay grupos repetitivos**.

Menciona también que cada tabla debe tener su nombre único, no puede haber dos filas iguales, todos los datos en una columna deben ser del mismo tipo, debe existir una clave primaria, no debe haber variación en el



número de columnas y debe haber independencia del orden de filas y columnas.

- **Anomalías/Problemas que resuelve:** Elimina los "grupos repetidos", ya sea a través de múltiples columnas (ej. Telefono 1, Telefono 2) o como una lista de valores dentro de una sola columna (ej. Telefono con múltiples números). Aborda la dificultad en las consultas y la imposibilidad de aplicar unicidad.
- **Ejemplo:** Utiliza el ejemplo de la tabla Cliente con el atributo Teléfono multivaluado o con múltiples columnas de Teléfono. Explica cómo se descompone en Cliente y Telefono del cliente para cumplir 1FN.

2. Segunda Forma Normal (2FN):

- **Regla:** Una relación está en 2FN si está en 1FN y **ningún atributo no primo depende parcialmente de una clave de la relación**. Es decir, todos los atributos que no son clave principal deben depender *completamente* de toda la clave primaria.
- **Anomalías/Problemas que resuelve:** Las dependencias parciales que causan redundancia y anomalías de actualización (modificación, inserción, borrado).
- **Ejemplo:** Utiliza el ejemplo de la relación PRESTAMO(num socio, nombre socio, cod libro, fec prest, editorial, pais) donde editorial y pais dependen solo de cod libro, que es parte de una clave compuesta (num socio, cod libro). Muestra la solución descomponiendo en PRESTAMO1 y LIBRO. También puedes usar el ejemplo EMP_PROY donde NombreE depende de Dni (parte de la clave {Dni, NumProyecto}).

3. Tercera Forma Normal (3FN):

- **Regla:** Una relación está en 3FN si está en 2FN y **ningún atributo no primo depende transitivamente de una clave de la relación**. Formalmente, para toda $DF X \rightarrow A$, X es una superclave de la relación O A es un atributo primo.



- **Anomalías/Problemas que resuelve:** Las dependencias transitivas, donde un atributo no clave depende de otro atributo no clave que, a su vez, depende de la clave. Esto evita redundancias y anomalías de actualización.
- **Ejemplo:** Usa el ejemplo de la tabla Películas donde Nacionalidad director depende de Director, y Director depende de Título (clave primaria). Muestra la descomposición en Películas y Autores. También el ejemplo EMP_DEPT donde DniDirector depende transitivamente de Dni a través de NúmeroDpto.

4. Forma Normal de Boyce-Codd (FNBC o BCNF):

- **Regla:** Una relación está en BCNF si, para toda dependencia funcional no trivial $X \rightarrow A$, **X es una superclave de la relación.**
- **Diferencia clave con 3FN:** Es una forma más estricta. La BCNF no permite la condición de la 3FN de que A pueda ser un atributo primo si X no es una superclave. Se diferencia de 3FN cuando existen dos o más claves candidatas y más de un atributo en las claves candidatas.
- **Anomalías/Problemas que resuelve:** Anomalías que aún pueden existir en 3FN, especialmente cuando hay claves candidatas que se solapan.
- **Ejemplo:** Utiliza el ejemplo IMPARTE(ESTUDIANTE,CURSO,PROFESOR) con dependencias $\{ESTUDIANTE,CURSO\} \rightarrow PROFESOR$ y $PROFESOR \rightarrow CURSO$. Explica cómo esta relación está en 3FN pero no en BCNF porque PROFESOR no es una superclave, a pesar de que CURSO es un atributo primo. Menciona la dificultad de la descomposición y la posible pérdida de dependencias.

5. Cuarta Forma Normal (4FN):

- **Regla:** Una tabla está en 4FN si está en 3FN y **todas las dependencias multivaluadas no funcionales ($X \twoheadrightarrow Y$) en ella tienen a X como superclave.**
- **Dependencia Multivaluada (DMV):** Explica que es una generalización de las dependencias funcionales, donde un conjunto de valores del



determinado (Y) son determinados por un determinante (X), independientemente de otros atributos.

- **Anomalías/Problemas que resuelve:** Ocurre cuando una relación contiene dos o más atributos multivaluados independientes, lo que puede causar redundancia.
- **Ejemplo:** La tabla NOMBREE, NOMBREP, NOMBRED donde un empleado (NOMBREE) tiene múltiples proyectos (NOMBREP) y múltiples dependientes (NOMBRED) de forma independiente. No está en 4FN porque NOMBREE no es una superclave, lo que lleva a redundancia. Muestra la descomposición en NOMBREE, NOMBREP y NOMBREE, NOMBRED.

6. Quinta Forma Normal (5FN) o Forma Normal de Proyección-Reunión (FNDR):

- **Regla:** Una tabla está en 5FN si está en **4FN** y su información **no puede ser descompuesta en varias relaciones menores sin pérdida**, a menos que la descomposición esté definida por claves candidatas.
- **Anomalías/Problemas que resuelve:** Redundancias restantes en casos complejos donde los hechos multivaluados no son independientes, que no fueron cubiertos por formas normales anteriores.
- **Uso:** Es la menos usada y solo se aplica en casos muy concretos, generalmente en tablas con muchas columnas o donde las uniones puedan generar redundancia/incoherencia.
- **Ejemplo:** El caso de VENDEDOR, COMPAÑIA, PRODUCTO donde existe una restricción de simetría (si un vendedor vende un producto X, y representa a una compañía Y que produce X, entonces el vendedor vende el producto X de la compañía Y). Muestra la descomposición en VENDEDOR PRODUCTO, VENDEDOR COMPAÑIA y COMPAÑIA PRODUCTO.

Fase 3: Propiedades Deseables de una Descomposición (Tiempo estimado: 0.5 horas)



1. Investiga las dos propiedades adicionales que un buen diseño de base de datos debe buscar en la descomposición:
 - **Descomposición y Reunión sin Pérdidas (Lossless-Join Decomposition - JSP):** Explica que garantiza que **no se generarán tuplas espúreas (falsas)** cuando se aplique un JOIN NATURAL a las relaciones resultantes de la descomposición.
 - **Preservación de Dependencias (Dependency Preservation):** Explica que asegura que **todas las dependencias funcionales originales se conserven** en la descomposición, ya sea directamente en una de las nuevas relaciones o que puedan inferirse de las dependencias que aparecen en ellas.

Fase 4: Elaboración del Mapa Conceptual (Tiempo estimado: 1-2 horas)

1. **Crea un mapa conceptual detallado** que integre todos los conceptos investigados. Asegúrate de que el mapa incluya:
 - El concepto central de **Normalización**.
 - Los **problemas/anomalías** que la normalización resuelve.
 - Los **conceptos clave** (Dependencias Funcionales, tipos de DF, Claves).
 - Cada **Forma Normal (1FN, 2FN, 3FN, BCNF, 4FN, 5FN)** con:
 - Su definición/regla.
 - Las anomalías específicas que resuelve.
 - Una conexión visual clara entre las formas normales (ej. 2FN implica 1FN, 3FN implica 2FN, etc.).
 - Las **Propiedades Deseables de la Descomposición** (JSP y Preservación de Dependencias).
 - Utiliza conectores y palabras de enlace para mostrar las relaciones entre los conceptos.

Evaluación:



La actividad se evaluará en función de:

- **Precisión y exhaustividad:** Que el mapa conceptual y la investigación reflejen una comprensión precisa y completa de todos los conceptos solicitados, apoyándose en las fuentes proporcionadas.
- **Claridad y organización:** La estructura lógica del mapa conceptual, la facilidad de lectura y la coherencia de las explicaciones.
- **Uso de ejemplos:** La inclusión de ejemplos claros y pertinentes para cada forma normal y tipo de anomalía.



Complemento educativo CE. 4.2 Normalización de Esquemas de Bases de Datos

Objetivo: Los estudiantes aplicarán los principios de normalización (1FN, 2FN, 3FN y FNBC) a esquemas de bases de datos, identificando dependencias funcionales y resolviendo anomalías para crear un diseño más eficiente e íntegro.

Introducción: El diseño de bases de datos relacionales busca organizar los datos de manera que se **minimice la redundancia** y se **maximice la integridad**. La **normalización** es un proceso fundamental para lograr esto, aplicando una serie de reglas conocidas como "formas normales" (FN). Estas reglas se basan principalmente en el concepto de **dependencias funcionales (DFs)**, que describen cómo ciertos atributos de una relación determinan de manera única los valores de otros atributos.

Un diseño bien normalizado ayuda a prevenir las **anomalías de actualización** (problemas de inserción, borrado y modificación de datos) y a asegurar la consistencia de los mismos. El proceso implica someter un esquema de relación a una serie de pruebas para "certificar" si pertenece o no a una cierta forma normal, y si es necesario, descomponerlo en esquemas más pequeños con propiedades deseables.



Instrucciones de la Actividad:

Parte 1: Análisis de Dependencias Funcionales para "ESTUDIANTE_SOLICITA_BECA"

1. **Escenario de la Base de Datos:** Analicen el siguiente esquema de relación para una tabla que almacena datos sobre estudiantes que solicitan becas, información sobre las becas y la fecha de solicitud: ESTUDIANTE_SOLICITA_BECA (Cod_Estud, Nombre_E, Apellido, DNI, Dirección, Cod_Beca, Nombre, Requisito, Fecha).

Consideren las siguientes reglas de negocio y semántica de los datos para inferir las dependencias funcionales:

- Cada Cod_Estud identifica unívocamente a un estudiante y determina su Nombre_E, Apellido, DNI y Dirección.
- Cada DNI es único para cada estudiante y determina la Dirección del estudiante.
- Cada Cod_Beca identifica unívocamente una beca y determina su Nombre y Requisito.
- La Fecha de solicitud depende de la combinación del Cod_Estud y Cod_Beca.
- El Nombre de una beca, si es único, podría determinar su Requisito (e.g., "METRICA" siempre tiene "Ing. Téc." como requisito).

2. Identificación de Clave Primaria y DFs:

- Identifiquen la **clave primaria** (o claves candidatas) para la tabla ESTUDIANTE_SOLICITA_BECA basándose en las reglas de negocio.
- Listado de **todas las dependencias funcionales (DFs)** que se pueden inferir del escenario y de la tabla. Utilicen la notación $X \rightarrow Y$.

Parte 2: Aplicación de Formas Normales

Para cada una de las siguientes formas normales, aplicarán los principios al esquema inicial de ESTUDIANTE_SOLICITA_BECA y a las relaciones resultantes de cada paso. En cada etapa:



- Expliquen la definición de la forma normal y por qué la relación actual la viola (si lo hace).
- Realicen las descomposiciones necesarias para que la relación(es) resultante(s) cumpla(n) con la forma normal, asegurando la **preservación de dependencias** y la **reunión sin pérdida de información**.
- Escriban el nuevo(s) esquema(s) de relación con su(s) clave(s) primaria(s) y sus DFs resultantes.

1. Primera Forma Normal (1FN):

- **Definición:** Una relación está en 1FN si y solo si todos sus atributos toman un **único valor atómico** de su dominio subyacente, lo que significa que **no hay grupos repetitivos** ni atributos multivaluados. También implica que no debe existir variación en el número de columnas.
- **Análisis para "ESTUDIANTE_SOLICITA_BECA":** El esquema proporcionado, tal como se muestra en la fuente, parece estar en 1FN, ya que los valores en cada columna son atómicos (simples e indivisibles). Sin embargo, si un atributo como Requisito pudiera contener múltiples valores (ej., "Ing. Téc., Excel Avanzado") o si hubiera columnas como Telefono1, Telefono2 para el mismo concepto, la tabla no estaría en 1FN.
- **Tarea:** Asumiendo que su tabla ESTUDIANTE_SOLICITA_BECA ya está en 1FN, justifiquen por qué. Si su análisis de las reglas de negocio o la estructura de los datos iniciales hubiera implicado una violación (como un campo multivaluado para Requisito o múltiples columnas de Teléfono), expliquen cómo la hubieran normalizado para alcanzar la 1FN, similar a los ejemplos de ALUMNO o Cliente en las fuentes.

2. Segunda Forma Normal (2FN):

- **Definición:** Una relación está en 2FN si está en 1FN y **ningún atributo no primo depende parcialmente de cualquier clave** de la relación. Un **atributo no primo** es aquel que no forma parte de ninguna clave candidata.



- **Análisis para "ESTUDIANTE_SOLICITA_BECA":** Consideren la clave primaria compuesta que identificaron. ¿Alguno de los atributos no clave (Nombre_E, Apellido, DNI, Dirección, Nombre de beca, Requisito, Fecha) depende solo de una parte de esa clave compuesta? Por ejemplo, en la tabla EMP_PROY de los recursos, NombreE dependía parcialmente de Dni (parte de la clave {Dni, NumProyecto}), violando 2FN.
- **Descomposición:** Si existen dependencias parciales, descompongan la tabla ESTUDIANTE_SOLICITA_BECA en nuevas relaciones para eliminar estas dependencias. Presenten los nuevos esquemas y sus DFs.

3. Tercera Forma Normal (3FN):

- **Definición:** Una relación está en 3FN si está en 2FN y si **no existe ninguna dependencia funcional transitiva** en los atributos que no son clave. Una dependencia transitiva $X \rightarrow Z$ ocurre a través de Y si $X \rightarrow Y$ y $Y \rightarrow Z$, donde Y no es una superclave. Es decir, un atributo no clave depende de otro atributo no clave que, a su vez, depende de la clave primaria.
- **Análisis para las tablas obtenidas en 2FN:** Revisen cada una de las tablas resultantes del paso anterior. ¿Existen atributos no clave que dependan de otros atributos no clave? Consideren el ejemplo del "Superbowl" (Cantantes $\rightarrow$ Genero donde Cantantes es atributo no clave pero depende de la clave Eventos) o el ejemplo de "Películas" (Director $\rightarrow$ Nacionalidad).
- **Descomposición:** Si encuentran dependencias transitivas, descompongan las tablas para eliminarlas. Presenten los nuevos esquemas y sus DFs.

4. Forma Normal de Boyce-Codd (FNBC):

- **Definición:** Una relación está en FNBC si **cada determinante (atributo o conjunto de atributos que determina a otro) es una clave candidata**. Se diferencia de 3FN en que no se permite que un determinante no clave determine a un atributo primo.
- **Análisis para las tablas obtenidas en 3FN:** Revisen si alguna de sus tablas resultantes de 3FN viola la FNBC. Un ejemplo clásico de una relación en 3FN



pero no en FNBC es la relación IMPARTE con atributos (ESTUDIANTE, CURSO, PROFESOR) y DFs {ESTUDIANTE,CURSO} → PROFESOR y PROFESOR → CURSO. Aquí, PROFESOR es un determinante, pero no es una superclave (ya que no determina a ESTUDIANTE), y CURSO es un atributo primo (parte de una clave candidata).

- **Descomposición (si aplica):** Si identifican alguna violación de FNBC, realicen la descomposición correspondiente. Presenten los esquemas finales y sus DFs.

Entregables:

1. Un documento que contenga:
 - La clave primaria identificada para la tabla inicial ESTUDIANTE_SOLICITA_BECA.
 - El listado completo de todas las dependencias funcionales identificadas para el escenario.
 2. Para cada forma normal (1FN, 2FN, 3FN, FNBC):
 - Una breve explicación de la regla.
 - Un análisis claro de si el esquema de relación actual (en ese paso de normalización) la cumple o la viola, justificando con las DFs identificadas y citando las fuentes si es relevante.
 - Si hay una violación, la(s) nueva(s) tabla(s) resultante(s) con su(s) clave(s) primaria(s) y DFs, explicando el proceso de descomposición.
 3. (Opcional, pero recomendado) Un diagrama simple que muestre la evolución de las tablas a través del proceso de normalización, desde el esquema original hasta las tablas final(es) normalizada(s).
-



Complemento educativo CE. 4.3 Normalización de Esquemas de Bases de Datos

Objetivo: Los estudiantes aplicarán los principios de normalización (1FN, 2FN, 3FN y FNBC) al esquema de base de datos ESTUDIANTE_SOLICITA_BECA, identificando dependencias funcionales y resolviendo anomalías para crear un diseño más eficiente e íntegro.

Introducción: El diseño de bases de datos relacionales busca organizar los datos para **minimizar la redundancia** y **maximizar la integridad**. La **normalización** es un proceso fundamental para lograr esto, aplicando una serie de reglas conocidas como "formas normales" (FN). Estas reglas se basan en el concepto de **dependencias funcionales (DFs)**, que describen cómo ciertos atributos de una relación determinan de manera única los valores de otros atributos. Un diseño bien normalizado ayuda a prevenir las **anomalías de actualización** (problemas de inserción, borrado y modificación de datos) y a asegurar la consistencia de los mismos. El proceso implica someter un esquema de relación a pruebas para "certificar" su forma normal, y si es necesario, descomponerlo en esquemas más pequeños con propiedades deseables.

Instrucciones de la Actividad:

Parte 1: Análisis de Dependencias Funcionales para "ESTUDIANTE_SOLICITA_BECA"

1. **Escenario de la Base de Datos:** Analicen el siguiente esquema de relación para una tabla que almacena datos sobre estudiantes que solicitan becas, información sobre las becas y la fecha de solicitud: ESTUDIANTE_SOLICITA_BECA (Cod_Estud, Nombre_E, Apellido, DNI, Dirección, Cod_Beca, Nombre_B, Requisito, Fecha) (Se ha renombrado "Nombre" a "Nombre_B" para evitar ambigüedad con el nombre del estudiante).

Consideren las siguientes reglas de negocio y semántica de los datos para inferir las dependencias funcionales:



- Cada Cod_Estud identifica unívocamente a un estudiante y determina su Nombre_E, Apellido, DNI y Dirección.
- Cada DNI es único para cada estudiante y, por lo tanto, determina el Cod_Estud y la Dirección del estudiante, así como Nombre_E y Apellido.
- Cada Cod_Beca identifica unívocamente una beca y determina su Nombre_B y Requisito.
- La Fecha de solicitud depende de la combinación del Cod_Estud y Cod_Beca.
- El Nombre_B de una beca, si es único para un tipo de beca (independientemente de su Cod_Beca), podría determinar su Requisito.

2. Identificación de Clave Primaria y DFs:

- Identifiquen la **clave primaria** (o claves candidatas) para la tabla ESTUDIANTE_SOLICITA_BECA basándose en las reglas de negocio.
- Listado de **todas las dependencias funcionales (DFs)** que se pueden inferir del escenario y de la tabla. Utilicen la notación $X \rightarrow Y$.

Parte 2: Aplicación de Formas Normales

Para cada una de las siguientes formas normales, aplicarán los principios al esquema inicial de ESTUDIANTE_SOLICITA_BECA y a las relaciones resultantes de cada paso. En cada etapa:

- Expliquen la definición de la forma normal y por qué la relación actual la viola (si lo hace).
- Describan las **anomalías de actualización** (inserción, borrado, modificación) que la violación de la forma normal causa.
- Realicen las descomposiciones necesarias para que la relación(es) resultante(s) cumpla(n) con la forma normal, asegurando la **preservación de dependencias** y la **reunión sin pérdida de información**.
- Escriban el nuevo(s) esquema(s) de relación con su(s) clave(s) primaria(s) y sus DFs resultantes.



Documento Paso a Paso: Normalización del Esquema "ESTUDIANTE_SOLICITA_BECA"

Esquema Inicial: ESTUDIANTE_SOLICITA_BECA (Cod_Estud, Nombre_E, Apellido, DNI, Dirección, Cod_Beca, Nombre_B, Requisito, Fecha)

Parte 1: Análisis de Dependencias Funcionales

- **Clave Primaria Identificada:** Basándonos en la semántica de los datos y las reglas de negocio, la **clave primaria** de la relación ESTUDIANTE_SOLICITA_BECA es la combinación de {Cod_Estud, Cod_Beca}. Esta combinación identifica unívocamente cada solicitud de beca en una fecha específica.
- **Dependencias Funcionales (DFs) Inferidas:**
 1. **DF1:** {Cod_Estud, Cod_Beca} → Fecha (La combinación de un estudiante y una beca determina una única fecha de solicitud).
 2. **DF2:** Cod_Estud → Nombre_E, Apellido, DNI, Dirección (El código de estudiante determina unívocamente todos sus datos personales).
 3. **DF3:** DNI → Cod_Estud, Nombre_E, Apellido, Dirección (Dado que el DNI es único para cada estudiante, determina todos sus datos personales, incluyendo su Cod_Estud y Dirección). Esto implica que DNI es una clave candidata para la información del estudiante.
 4. **DF4:** Cod_Beca → Nombre_B, Requisito (El código de la beca determina su nombre y los requisitos asociados a ella).
 5. **DF5:** Nombre_B → Requisito (Asumiendo que el nombre de una beca, si es único, determina un requisito específico, independientemente del Cod_Beca).

Parte 2: Aplicación de Formas Normales

1. Primera Forma Normal (1FN)

- **Definición:** Una relación está en 1FN si y solo si todos sus atributos toman un **único valor atómico** de su dominio subyacente, lo que significa que **no hay grupos repetitivos** ni atributos multivaluados.



- **Análisis para ESTUDIANTE_SOLICITA_BECA:** El esquema proporcionado, tal como se ilustra en la tabla, **ya se encuentra en 1FN**. Los valores en cada columna son atómicos; por ejemplo, Dirección es una única dirección, y Requisito es un único requisito ("Ing. Téc."), no una lista de múltiples requisitos separados por comas. Tampoco hay columnas con grupos repetitivos (Telefono1, Telefono2).
- **Anomalías encontradas (persisten en 1FN):** A pesar de cumplir con 1FN, la relación ESTUDIANTE_SOLICITA_BECA presenta serios problemas de diseño debido a las dependencias funcionales que no están alineadas con la clave primaria.
 - **Redundancia:** Los datos del estudiante (Nombre_E, Apellido, DNI, Dirección) se repiten por cada beca que el estudiante solicita. De manera similar, la información de la beca (Nombre_B, Requisito) se duplica si múltiples estudiantes solicitan la misma beca.
 - **Anomalías de Modificación:** Cambiar la Dirección de un estudiante requiere modificar múltiples tuplas, lo que puede llevar a **inconsistencias** si el cambio no se propaga a todas las ocurrencias.
 - **Anomalías de Inserción:** No se puede registrar un nuevo estudiante si aún no ha solicitado una beca, ni una beca si aún no ha sido solicitada por ningún estudiante, porque Cod_Estud y Cod_Beca son parte de la clave primaria y no pueden contener valores nulos.
 - **Anomalías de Borrado:** Si la última tupla de solicitud de un estudiante se elimina, se perderán todos los datos personales de ese estudiante de la base de datos (Ej: si "Luis García" solo tiene una entrada con "Cod_Beca B56784" y se elimina, sus datos se pierden).

2. Segunda Forma Normal (2FN)

- **Definición:** Una relación está en 2FN si está en 1FN y **ningún atributo no primo depende parcialmente de cualquier clave** de la relación. Un atributo no primo es aquel que no forma parte de ninguna clave candidata.



- **Análisis para ESTUDIANTE_SOLICITA_BECA:** La clave primaria de ESTUDIANTE_SOLICITA_BECA es {Cod_Estud, Cod_Beca}.
 - **Dependencia Parcial 1 (violación de 2FN):** Cod_Estud → Nombre_E, Apellido, DNI, Dirección. Los atributos Nombre_E, Apellido, DNI, Dirección son atributos no primos (no forman parte de ninguna clave candidata para la relación ESTUDIANTE_SOLICITA_BECA en su conjunto, solo son atributos del estudiante) y dependen solo de Cod_Estud, que es una **parte** de la clave primaria. Esto viola la 2FN.
 - **Dependencia Parcial 2 (violación de 2FN):** Cod_Beca → Nombre_B, Requisito. Los atributos Nombre_B y Requisito son no primos y dependen solo de Cod_Beca, que es también una **parte** de la clave primaria. Esto viola la 2FN.
- **Anomalías encontradas (debido a la violación de 2FN):** Las anomalías de redundancia, inserción, modificación y borrado mencionadas en la sección de 1FN son directamente causadas por estas dependencias parciales. Por ejemplo, la redundancia de datos del estudiante y de la beca, y la imposibilidad de insertar estudiantes o becas sin una solicitud activa.
- **Descomposición para 2FN:** Para eliminar estas dependencias parciales, descomponemos ESTUDIANTE_SOLICITA_BECA en tres relaciones, cada una con su propia clave primaria:
 - **R1: ESTUDIANTES (Cod_Estud, Nombre_E, Apellido, DNI, Dirección)**
 - **Clave Primaria:** Cod_Estud
 - **Dependencias Funcionales:**
 - Cod_Estud → Nombre_E, Apellido, DNI, Dirección
 - DNI → Cod_Estud, Nombre_E, Apellido, Dirección (DF de clave candidata alternativa)
 - **Propósito:** Almacena la información de cada estudiante, eliminando la redundancia de sus datos personales.
 - **R2: BECAS (Cod_Beca, Nombre_B, Requisito)**



- **Clave Primaria:** Cod_Beca
- **Dependencias Funcionales:**
 - Cod_Beca → Nombre_B, Requisito
- **Propósito:** Almacena la información de cada tipo de beca, eliminando la redundancia de sus detalles.
- **R3: SOLICITUDES (Cod_Estud, Cod_Beca, Fecha)**
 - **Clave Primaria:** {Cod_Estud, Cod_Beca}
 - **Dependencias Funcionales:**
 - {Cod_Estud, Cod_Beca} → Fecha
 - **Claves Foráneas:** Cod_Estud referencia a ESTUDIANTES, Cod_Beca referencia a BECAS.
 - **Propósito:** Relaciona a los estudiantes con las becas que solicitan y registra la fecha de la solicitud, manteniendo la integridad referencial.

Esta descomposición cumple con la **preservación de dependencias** y la **reunión sin pérdida de información**.

3. Tercera Forma Normal (3FN)

- **Definición:** Una relación está en 3FN si está en 2FN y **no existe ninguna dependencia funcional transitiva** en los atributos que no son clave. Una dependencia transitiva $X \rightarrow Z$ ocurre a través de Y si $X \rightarrow Y$ y $Y \rightarrow Z$, donde Y no es una superclave.
- **Análisis para las tablas obtenidas en 2FN:**
 - **Para R1: ESTUDIANTES (Cod_Estud, Nombre_E, Apellido, DNI, Dirección)**
 - **Claves Candidatas:** Cod_Estud y DNI. Ambos son atributos primos.
 - **Atributos No Primos:** Nombre_E, Apellido, Dirección.
 - **Dependencias a revisar:**
 - Cod_Estud → DNI y DNI → Dirección. A primera vista, esto parece una dependencia transitiva (Cod_Estud → DNI → Dirección). Sin embargo, para que sea una violación de 3FN,



el atributo intermedio (DNI) no debe ser una superclave. En este caso, DNI *sí es una superclave* de ESTUDIANTES (ya que determina Cod_Estud y, a través de él, todos los demás atributos). Por lo tanto, esta dependencia DNI → Dirección cumple la condición (a) de 3FN (DNI es superclave).

- ESTUDIANTES **está en 3FN**.

- **Para R2: BECAS (Cod_Beca, Nombre_B, Requisito)**

- **Clave Primaria:** Cod_Beca.
- **Atributos No Primos:** Nombre_B, Requisito.
- **Dependencias a revisar:**
 - Cod_Beca → Nombre_B y Nombre_B → Requisito (DF5). Esto forma una **dependencia transitiva**: Cod_Beca → Nombre_B → Requisito.
 - X = Nombre_B, A = Requisito.
 - ¿Es Nombre_B una superclave de BECAS? No, porque Nombre_B no determina Cod_Beca (pueden existir varias becas con el mismo nombre pero diferentes códigos, ej: "Beca de Excelencia 2023" y "Beca de Excelencia 2024").
 - ¿Es Requisito un atributo primo? No, no forma parte de ninguna clave candidata.
 - Dado que Nombre_B no es una superclave y Requisito no es un atributo primo, esta es una **violación de la 3FN**.

- **Para R3: SOLICITUDES (Cod_Estud, Cod_Beca, Fecha)**

- **Clave Primaria:** {Cod_Estud, Cod_Beca}.
- No tiene atributos no primos que dependan transitivamente de la clave.
- SOLICITUDES **está en 3FN**.

- **Anomalías encontradas (debido a la violación de 3FN en BECAS):**



- **Redundancia:** El Requisito para un Nombre_B específico se repite si ese Nombre_B aparece con diferentes Cod_Beca (lo cual es el caso si el nombre de la beca se repite anualmente, por ejemplo).
- **Anomalías de Modificación:** Si el Requisito de una Nombre_B cambia, se debe actualizar en todas las tuplas de la tabla BECAS que contengan ese Nombre_B, lo que puede llevar a inconsistencias.
- **Anomalías de Inserción:** No se puede registrar un Nombre_B y su Requisito si no hay un Cod_Beca asociado (Ej: no se puede listar un nuevo "tipo de beca" con sus requisitos hasta que se le asigne un código de beca).
- **Descomposición para 3FN para R2 (BECAS):** Descomponemos BECAS en dos relaciones para eliminar la dependencia transitiva:
 - **R2.1: BECAS_DETALLE (Cod_Beca, Nombre_B)**
 - **Clave Primaria:** Cod_Beca
 - **Dependencias Funcionales:** Cod_Beca → Nombre_B
 - **Propósito:** Contiene la información principal de la beca, vinculando el código con su nombre.
 - **R2.2: TIPOS_BECA (Nombre_B, Requisito)**
 - **Clave Primaria:** Nombre_B
 - **Dependencias Funcionales:** Nombre_B → Requisito
 - **Clave Foránea:** Nombre_B en BECAS_DETALLE referencia a TIPOS_BECA.
 - **Propósito:** Almacena los requisitos para cada tipo de beca, eliminando la dependencia transitiva.
- **Esquemas resultantes después de 3FN (final):**
 - **ESTUDIANTES (Cod_Estud, Nombre_E, Apellido, DNI, Dirección)**
 - **BECAS_DETALLE (Cod_Beca, Nombre_B)**
 - **TIPOS_BECA (Nombre_B, Requisito)**
 - **SOLICITUDES (Cod_Estud, Cod_Beca, Fecha)**

Esta descomposición conserva las dependencias y es sin pérdida de información.



4. Forma Normal de Boyce-Codd (FNBC)

- **Definición:** Una relación está en FNBC si **cada determinante (atributo o conjunto de atributos que determina a otro) es una clave candidata**. Se diferencia de 3FN en que no se permite que un determinante no clave determine a un atributo primo.
- **Análisis para las tablas obtenidas en 3FN:** Revisamos cada una de las relaciones finales:
 - **Para ESTUDIANTES (Cod_Estud, Nombre_E, Apellido, DNI, Dirección):**
 - **Determinantes:** Cod_Estud y DNI.
 - Cod_Estud es una clave candidata (y por lo tanto superclave).
 - DNI es una clave candidata (y por lo tanto superclave).
 - Dado que todos los determinantes (Cod_Estud y DNI) son claves candidatas, la relación **ESTUDIANTES está en FNBC**.
 - **Para BECAS_DETALLE (Cod_Beca, Nombre_B):**
 - **Determinante:** Cod_Beca.
 - Cod_Beca es la clave primaria y, por lo tanto, una clave candidata (superclave).
 - La relación **BECAS_DETALLE está en FNBC**.
 - **Para TIPOS_BECA (Nombre_B, Requisito):**
 - **Determinante:** Nombre_B.
 - Nombre_B es la clave primaria y, por lo tanto, una clave candidata (superclave).
 - La relación **TIPOS_BECA está en FNBC**.
 - **Para SOLICITUDES (Cod_Estud, Cod_Beca, Fecha):**
 - **Determinante:** {Cod_Estud, Cod_Beca}.
 - {Cod_Estud, Cod_Beca} es la clave primaria y, por lo tanto, una clave candidata (superclave).



- La relación **SOLICITUDES** está en **FNBC**.
- **Conclusión de FNBC:** Todas las relaciones resultantes de este proceso de normalización hasta 3FN cumplen con la Forma Normal de Boyce-Codd. Esto indica que hemos logrado un diseño de base de datos **eficiente e íntegro**, minimizando la redundancia y previniendo la mayoría de las anomalías de actualización.



Tema 5 Álgebra relacional

5.1 Operaciones fundamentales del álgebra relacional

El **Álgebra Relacional** es un **lenguaje de consulta procedimental** donde las consultas se expresan como la aplicación sucesiva de operaciones a las relaciones, resultando en una nueva relación. Fue definido por Codd en 1970. Las operaciones del álgebra relacional toman relaciones como operandos y devuelven relaciones como resultado, lo que permite anidar y combinar operadores para expresar consultas complejas.

Las **ocho operaciones básicas** se clasifican en monarias (operan sobre una relación) y binarias (operan sobre dos relaciones):

- **Operaciones Monarias:**

- **Selección (σ):** Permite **filtrar tuplas (filas)** de una relación que satisfacen una condición específica. Se denota con el símbolo σ (Sigma). Es **conmutativa**, lo que significa que el orden de aplicación de múltiples selecciones no altera el resultado.
- **Proyección (π):** Permite **extraer columnas (atributos)** de una relación. Se denota con el símbolo π (Pi). La operación elimina implícitamente cualquier tupla repetida del resultado. No es conmutativa.
- **Renombre (ρ):** Permite cambiar el nombre de una relación o de sus atributos.

- **Operaciones Binarias:**

- **Producto Cartesiano ($\times$):** Combina cada tupla de una relación con cada tupla de otra relación, generando una nueva relación con todos los atributos de ambas.
- **Unión ($\cup$):** Retorna el conjunto de tuplas que están en la primera relación, la segunda, o ambas. Las relaciones deben ser **compatibles**, es decir, tener el mismo número de atributos y dominios equivalentes.



- **Diferencia (-):** Retorna todas las tuplas que están en la primera relación pero **no** en la segunda. Las relaciones deben ser compatibles.

Además de las básicas, existen **operaciones derivadas o extendidas** que simplifican la definición de consultas complejas:

- **Intersección ($\cap$):** Retorna las tuplas comunes a dos relaciones. Puede expresarse en términos de diferencias.
- **Theta-Reunión (Join / $\bowtie$):** Combina tuplas de dos relaciones basándose en una condición. Es equivalente a una selección sobre el producto cartesiano de las relaciones.
- **Unión Natural (Natural Join / $\Join$):** Un tipo especial de Join que combina relaciones que tienen atributos comunes, eliminando las columnas duplicadas en el resultado.
- **División ($\div$):** Un operador complejo que devuelve las entidades de un tipo que se conectan, a través de una relación, con **todas** las entidades de otro tipo.
- **Proyección Generalizada ($\Pi F_1, F_2, \dots, F_n(E)$):** Extiende la operación de proyección permitiendo el uso de funciones aritméticas en la lista de proyección (ej. calcular bonos, totales de venta). Permite crear atributos derivados.
- **Funciones de Agregación (G):** Operaciones que toman una colección de valores y devuelven un único valor. Incluyen: **SUMA (SUM)**, **PROMEDIO (AVG)**, **CUENTA (COUNT)**, **MÁXIMO (MAX)** y **MÍNIMO (MIN)**. Pueden aplicarse a grupos de tuplas mediante un atributo de agrupación.
- **Reunión Externa (Outer Join):** Similar al Natural Join, pero **mantiene la información que se perdería** al no encontrar coincidencias, rellenando con valores nulos (NULL). Existen tres tipos: **Left Outer Join**, **Right Outer Join** y **Full Outer Join**.

Cálculo Relacional

El **Cálculo Relacional** es un **lenguaje de consulta declarativo** que describe la respuesta deseada de una base de datos sin especificar cómo obtenerla. Utiliza fórmulas bien



formadas, donde las variables se interpretan como variantes sobre las tuplas de las tablas. Se basa en el Cálculo de Predicados.

- **Tipos de Cálculo Relacional:**

- **Cálculo Relacional de Tuplas:** Las variables están asociadas a las tuplas de las tablas.
- **Cálculo Relacional de Dominios:** Utiliza variables de dominio que toman sus valores del dominio de un atributo.
- **Seguridad de las Expresiones:** Las expresiones deben ser "seguras", lo que significa que todos los valores que aparecen en el resultado deben provenir del dominio de la fórmula.
- **Equivalencia:** El cálculo relacional (restringido a expresiones seguras) es **equivalente en potencia expresiva al álgebra relacional básica**. Esto significa que cualquier consulta que pueda resolverse en un lenguaje puede resolverse en el otro, aunque sus enfoques sean diferentes.

Lenguajes Comerciales de Consulta: SQL (Structured Query Language)

SQL es el **lenguaje comercial estándar de consulta** con mayor influencia. Aunque a menudo se le denomina "lenguaje de consulta", es más que eso, ya que incluye capacidades para definir la estructura de los datos, manipularlos y especificar ligaduras de seguridad. SQL es tanto un **Lenguaje de Definición de Datos (DDL)** como un **Lenguaje de Manipulación de Datos (DML)**.

- **Sentencias DDL:** Proporcionan órdenes para definir o modificar esquemas de relación, eliminar relaciones y crear índices (ej. CREATE, DROP).
- **Sentencias DML:** Permiten realizar consultas y mantener los datos (insertar, suprimir, modificar) (ej. SELECT, INSERT, UPDATE, DELETE).
- **Tipos de Dominios en SQL:** Incluyen CHAR, VARCHAR, INTEGER, FLOAT, DATE, TIME y NULL.



- **Estructura Básica (SELECT-FROM-WHERE):** La forma fundamental de una consulta SQL para obtener datos, especificando qué atributos se desean, de qué tablas y bajo qué condiciones.
- **Operaciones sobre Conjuntos:** Permite combinar resultados de múltiples consultas: **UNION**, **INTERSECT** y **MINUS** (Diferencia).
- **Funciones de Agregación:** Se usan dentro de la cláusula SELECT para devolver un único valor sobre un grupo de registros. Incluyen **AVG** (promedio), **COUNT** (conteo), **SUM** (suma), **MAX** (valor más alto), y **MIN** (valor más bajo).
- **Valores Nulos (NULL):** Indican la ausencia de valor. Se comparan usando IS NULL o IS NOT NULL.
- **Subconsultas Anidadas:** Permiten usar los resultados de una consulta como parte de otra, mejorando la expresión natural y facilitando la descomposición de consultas complejas. Pueden aparecer en las cláusulas WHERE o HAVING.
- **Vistas:** Son tablas virtuales, derivadas de otras tablas o vistas. No existen físicamente. Se crean con CREATE VIEW y se eliminan con DROP VIEW.
- **Modificación de la Base de Datos:**
 - **INSERT:** Añade nuevas filas de datos a una tabla.
 - **DELETE:** Elimina filas seleccionadas de una tabla.
 - **UPDATE:** Actualiza (cambia) datos existentes en una o más columnas de filas seleccionadas.
- **Disparadores (Triggers):** Acciones asociadas que el SGBD debe ejecutar para cualquier evento que provoca un cambio en el conjunto de tablas (INSERT, DELETE, UPDATE).

Seguridad e Integridad de Bases de Datos

La **seguridad** en un sistema informático implica estar libre de peligros de accesos no permitidos que puedan causar daño. Más que de sistemas "seguros", se habla de sistemas



"fiables". Para mantener un sistema fiable, se deben garantizar tres aspectos fundamentales:

- **Confidencialidad:** Los objetos son accedidos únicamente por métodos permitidos.
- **Integridad:** Los objetos solo son modificados por elementos autorizados y de forma controlada.
- **Disponibilidad:** Los objetos del sistema permanecen accesibles a elementos autorizados.

Los elementos principales para proteger son el **software, hardware y datos**. Los **mecanismos de seguridad** se dividen en **prevención, detección y recuperación**. Es prioritario implementar mecanismos de prevención robustos para evitar ataques.

- **Especificación de un Seguro:** Se necesitan pautas mínimas para minimizar problemas, como cerrar aplicaciones innecesarias y asegurar las **claves (contraseñas)**. Una contraseña debe ser difícil de adivinar, no común, y no debe ser anotada ni compartida. Es crucial usar contraseñas diferentes para distintas cuentas.
- **Criptografía:** Es una rama de las matemáticas que provee herramientas para la autenticidad y confiabilidad en mensajes digitales, mediante encriptación y firma digital.
 - **Criptografía de clave privada o simétrica:** Las partes tienen la misma clave para cifrar y descifrar. Ejemplo: DES (Data Encryption Standard) y su evolución Triple-DES (TDES).
 - **Criptografía de clave pública o asimétrica:** Utiliza dos claves diferentes por usuario: una pública para cifrar y una privada para descifrar. Ejemplo: RSA. Permite la **firma digital** de documentos y la **autenticación**. Los **certificados digitales** establecen un vínculo entre un sujeto y su clave pública, firmados por una autoridad certificadora (AC).



- **Mantenimiento de la Integridad:** Protege la base de datos contra operaciones que introduzcan inconsistencias. Puede ser **semántica** (restricciones de dominio o atributos) u **operacional** (control de concurrencia en sistemas multiusuario).

5.2 Álgebra relacional extendida

El álgebra relacional extendida es una ampliación del álgebra relacional tradicional que incorpora operaciones adicionales para mejorar su expresividad y capacidad de manejo de datos en bases relacionales. Sus elementos claves incluyen:

- **Proyección generalizada:** Permite usar funciones aritméticas sobre los atributos en la lista de proyección, no solo seleccionar atributos. Esto facilita cálculos como restas, sumas o cualquier expresión aritmética directamente durante la proyección.
- **Operaciones aritméticas y lógicas:** Se habilitan dentro del álgebra relacional sobre los valores de los atributos, aumentando la capacidad para efectuar cálculos y evaluaciones dentro de las consultas.
- **Funciones de agregación:** Se introducen funciones que trabajan sobre conjuntos o multiconjuntos de valores y devuelven un valor único, tales como suma (sum), promedio (avg), conteo (count), mínimo (min) y máximo (max).
- **Operación de reunión externa:** Permite manejar valores nulos o información faltante en las relaciones, algo que el álgebra relacional tradicional no gestiona bien.
- **Renombramiento de atributos:** Como complemento a la proyección generalizada, para dar nombre a los resultados de expresiones aritméticas complejas.

En resumen, el álgebra relacional extendida integra estas extensiones para facilitar consultas más complejas, incluyendo cálculos directos, agregaciones y manipulación de



valores nulos, haciendo el lenguaje más potente y expresivo para bases de datos relacionales

Herramientas para el Aprendizaje del Álgebra Relacional

Dada la importancia del álgebra relacional para comprender las bases de datos relacionales y SQL, se han desarrollado **herramientas educativas**. Estas herramientas buscan facilitar el aprendizaje de los lenguajes de consulta formales.

- **Funcionalidades Clave:**

- **Introducción y manipulación de consultas:** Permiten al alumno practicar escribiendo y editando expresiones de álgebra relacional. Algunas permiten la introducción gráfica de operadores para mayor claridad.
- **Visualización y manipulación de árboles de expresiones:** Ayudan a comprender consultas no triviales y su estructura.
- **Ejecución de consultas:** Permiten ejecutar consultas sobre relaciones de ejemplo (manual o automáticamente) o sobre SGBD externos. Se puede ejecutar paso a paso para una mejor comprensión.
- **Optimización de consultas:** Aplican reglas del álgebra relacional y estadísticas de las relaciones para mejorar la eficiencia. Puede ser automática o interactiva (paso a paso).
- **Traducción a SQL:** Permiten al usuario entender la relación entre el álgebra relacional y SQL, el lenguaje estándar de consulta actual.
- **Definición de relaciones y tuplas:** Facilitan la creación de datos de ejemplo manual o automáticamente, o importándolos de archivos o bases de datos externas.

- **Ejemplos de Herramientas Existentes:**

- **Relational:** Carga/guarda relaciones, ejecuta consultas, optimización simple (sin estadísticas), sin traducción SQL ni árboles.
- **LEAP:** Orientada a la educación, basada en comandos (sin interfaz gráfica), permite obtener árboles de expresiones en modo texto, no optimiza.



- **WinRDBI:** Herramienta educacional con interfaz gráfica amigable, carga relaciones desde diferentes formatos, no optimiza.
- **RelationalQuery:** Herramienta educativa, permite elegir lenguaje (álgebra relacional, TRC, SQL), traductor de consultas entre lenguajes.

El desarrollo de nuevas herramientas busca integrar y mejorar estas funcionalidades para ofrecer una experiencia de aprendizaje más completa e intuitiva.

Complemento educacional CE. 5.1 Descubriendo el Álgebra Relacional y sus Operadores



Propósito: Esta actividad te permitirá comprender a fondo el concepto del álgebra relacional, sus operadores fundamentales y extendidos, y su papel en el manejo de bases de datos, sintetizando la información en un mapa conceptual.

Objetivos de aprendizaje: Al finalizar esta actividad, serás capaz de:

- **Definir** el álgebra relacional y explicar su importancia en el contexto de las bases de datos relacionales.
- **Identificar y describir** los operadores unarios del álgebra relacional (selección y proyección), incluyendo su notación y ejemplos.
- **Identificar y describir** los operadores binarios del álgebra relacional (producto cartesiano, unión, intersección, diferencia, Join o reunión, y división), incluyendo su notación y ejemplos.
- **Comprender** la necesidad y la funcionalidad de los operadores del álgebra relacional extendida, como la proyección generalizada y las funciones de agregación.
- **Elaborar** un mapa conceptual que organice jerárquicamente todos los conceptos investigados.

Recursos Necesarios:

- Las fuentes proporcionadas.



- Herramientas para la creación de mapas conceptuales (físicas o digitales).

Instrucciones de la Actividad:

Paso 1: Investigación del Concepto de Álgebra Relacional Investiga y recopila información sobre el **Álgebra Relacional** basándote en las fuentes. Asegúrate de incluir:

- **Definición:** ¿Qué es el álgebra relacional? Es un **lenguaje de consulta procedimental** que usa **operadores especializados** para transformar una o más relaciones de entrada en una relación de salida.
- **Propósito y Aplicación Principal:** ¿Para qué se utiliza y cuál es su principal aplicación?. Sirve como **base teórica para las bases de datos relacionales** y sus lenguajes de consulta, como SQL. También se usa para obtener una **versión más eficiente de una consulta**.
- **Características Clave:**
 - Toda consulta genera una **relación como resultado**.
 - Los operadores toman relaciones como **entrada** y producen relaciones como **salida**, permitiendo **anidar y combinar operadores**.
 - Se basa en la **Teoría de Conjuntos**.
 - Fue presentado por **Edgar F. Codd** en 1970.
 - Se diferencia del Cálculo Relacional, que es declarativo, en que el Álgebra Relacional es **procedural**, especificando *cómo* obtener el resultado.

Paso 2: Investigación de Operadores Unarios Investiga y describe los siguientes operadores unarios, incluyendo su notación, definición y ejemplos relevantes de las fuentes:

- **Selección (σ):**
 - **Definición:** Permite seleccionar un **subconjunto horizontal de tuplas** de una relación que cumplen una o más condiciones o un predicado dado.



- **Notación:** Se denota con la letra griega minúscula sigma (σ). La condición se anota al lado del operador, y la relación entre paréntesis.
- **Condiciones:** La condición de selección es una **expresión booleana** que puede usar operadores de comparación como $=$, $<$, $\leq$, $>$, $\geq$, $\neq$ y conectarse con operadores booleanos Y (AND), O (OR), NO (NOT).
- **Comutatividad:** Es conmutativa, es decir, el orden de aplicación de múltiples selecciones no altera el resultado.
- **Unario:** Se aplica a una sola relación y opera sobre cada tupla individualmente.
- **Ejemplos:** σ ND=4 (EMPLEADO), σ salario $>$ 30000 (EMPLEADO), σ Apellido=Gomez (Alumnos).
- **Proyección (Π):**
 - **Definición:** Permite seleccionar **ciertas columnas (atributos)** de una tabla, produciendo un **subconjunto vertical** de atributos.
 - **Eliminación de duplicados:** La operación PROYECTAR **elimina implícitamente cualquier tupla repetida**, asegurando que el resultado sea una relación válida (un conjunto de tuplas).
 - **Notación:** Se denota con la letra griega mayúscula pi (π). La lista de atributos a proyectar se especifica y la relación entre paréntesis.
 - **No conmutativa:** A diferencia de la selección, la proyección no es conmutativa.
 - **Unario:** Es una operación unaria.
 - **Ejemplos:** π APELLIDO NOMBRE, SALARIO (EMPLEADO), π SALARIO, SEXO (EMPLEADO), π Apellido, Semestre, NumeroControl (Alumnos).
- **Renombre (ρ):**
 - **Definición:** Permite **cambiar los nombres de los atributos** de una relación, resultando en una relación idéntica a la original pero con nuevos nombres para sus atributos.
 - **Notación:** Se denota con la letra griega minúscula rho (ρ).



- **Clasificación:** Algunas veces se omite de la lista de operaciones fundamentales o se clasifica como operación auxiliar.

Paso 3: Investigación de Operadores Binarios Investiga y describe los siguientes operadores binarios, incluyendo su notación, definición y ejemplos relevantes de las fuentes:

- **Producto Cartesiano (X):**
 - **Definición:** Combina **todas las tuplas posibles** de dos (o más) relaciones. Si R tiene n_1 tuplas y S tiene n_2 tuplas, el resultado tendrá $n_1 * n_2$ tuplas.
 - **Esquema:** El esquema de la relación resultante combina los atributos de ambas relaciones. Es necesario renombrar atributos si tienen nombres en común para evitar ambigüedades.
 - **Notación:** Se escribe como $R \times S$.
 - **Ejemplo:** Alumnos $\times$ Maestros.
- **Unión (U):**
 - **Definición:** Retorna el **conjunto de tuplas que están en la relación R, o en la relación S, o en ambas**.
 - **Compatibilidad:** Las relaciones R y S deben ser **compatibles a la unión**, lo que significa que deben tener el mismo número de atributos (grado) y los atributos equivalentes deben estar definidos en el mismo dominio.
 - **Notación:** Se denota por U.
 - **Ejemplo:** Proveedor UNION Proveedor1.
- **Intersección (n):**
 - **Definición:** Retorna el **conjunto de tuplas que están en R y también en S**.
 - **Compatibilidad:** Al igual que la unión, requiere que R y S sean **compatibles a la unión**.
 - **Notación:** Se denota por n.
 - **Expresión Derivada:** Puede expresarse en función de otros operadores básicos: $R \cap S = R - (R - S)$.



- **Ejemplo:** Proveedor INTERSECT Proveedor1.
- **Diferencia (-):**
 - **Definición:** Retorna **todas las tuplas que están en R, pero no en S**.
 - **Compatibilidad:** Las relaciones R y S deben ser **compatibles a la unión**.
 - **Notación:** Se denota por -.
 - **Ejemplo:** Proveedor MINUS Proveedor1.
- **Join ($\bowtie$ o $|><|$):**
 - **Definición:** Permite **combinar tuplas de dos relaciones** a través de una condición sobre sus atributos. Es equivalente a un producto cartesiano seguido de una selección.
 - **Tipos de Join:**
 - **Theta-Join ($\bowtie_{\theta}$ o $|><|$ Condición):** $R \bowtie_{\theta} S$ es equivalente a $\sigma_{\text{condición}}(R \times S)$, donde la condición θ es libre. Si θ es una igualdad, se denomina EquiJoin.
 - **Natural Join ($\bowtie$ o $./.$):** Es una operación muy utilizada que combina relaciones con atributos comunes **sin necesidad de especificar la condición de igualdad**; automáticamente realiza el join por atributos con el mismo nombre y **elimina las columnas duplicadas**. $R \bowtie S = \Pi_{A_1, A_2, \dots, A_n}(\sigma_{\theta}(R \times S))$.
 - **Outer Joins:** Similares al Natural Join, pero **mantienen la información que se habría perdido**, reemplazando los valores faltantes con NULL.
 - **Left Outer Join ($+ ./.$):** Incluye todas las combinaciones de tuplas que son iguales en atributos comunes, más las **tuplas de R que no tienen pareja en S**.
 - **Right Outer Join ($./+$):** Incluye todas las combinaciones de tuplas que son iguales en atributos comunes, más las **tuplas de S que no tienen pareja en R**.



- **Full Outer Join (+ ./ +):** Incluye todas las combinaciones de tuplas que son iguales en atributos comunes, más las **tuplas de S que no tienen pareja en R y las tuplas de R que no tienen pareja en S.**
- **Semijoin (B <):** Retorna solo las **tuplas de R para las cuales hay una tupla en S** que es igual en sus atributos comunes.
 - **Ejemplo:** Proveedor JOIN Suministra.
- **División ($\div$ o $/$):**
 - **Definición:** Obtiene aquellas entidades de tipo A que se conectan, a través de la relación, **con todas las entidades de tipo B** que hay en la relación divisor. Retorna los valores de x tales que para todo valor y en B existe una tupla $\langle x, y \rangle$ en A.
 - **Notación:** Se denota por $\div$ o $/$.
 - **Ejemplo:** $P, S (ventas) S (ciudad = „Londres” (proveedor))$ para obtener códigos de piezas suministradas por todos los proveedores de Londres.

Paso 4: Investigación de Operadores del Álgebra Relacional Extendida Investiga y describe las ampliaciones del álgebra relacional básica, explicando por qué son necesarias y cómo funcionan:

- **Ampliaciones Generales:** Las operaciones básicas del álgebra relacional se han ampliado para permitir operaciones aritméticas como parte de la proyección, **funciones de agregación** y la operación de **reunión externa** para trabajar con valores nulos.
- **Proyección Generalizada ($\Pi F_1, F_2, \dots, F_n (E)$):**
 - **Definición:** Amplía la operación de proyección permitiendo el uso de **funciones aritméticas** en la lista de proyección. Estas funciones pueden ser sumas, restas, cálculos complejos, o incluso la creación de **pseudo-columnas/atributos derivados**.



- **Notación:** Se usa el símbolo π seguido de una lista de expresiones aritméticas ($F_1, F_2, \dots, F_n$) y la expresión relacional (E).
- **Ejemplos:** Calcular un bono del 15% sobre el sueldo: $\pi \text{ NombreCompleto, Sueldo} * 0.15 \rightarrow \text{Bono (EMPLEADO)}$. Calcular el total por producto en una factura: $\pi \text{ NumeroFactura, Cantidad} * \text{Precio} \rightarrow \text{Total (DETALLE_VENTA)}$.
- **Funciones de Agregación (G):**
 - **Definición:** Son funciones que toman una **colección de valores** y **devuelven un único valor** como resultado.
 - **Colecciones:** Las colecciones pueden tener valores repetidos y el orden no importa. Es posible especificar **-distinct** para borrar duplicados antes del cálculo.
 - **Notación:** Se usa la letra griega minúscula gamma (G) para operaciones de agrupación o agregación. Puede incluir **atributos de agrupación** y una lista de pares (función, atributo).
 - **Funciones Comunes:**
 - **SUM:** Suma de los elementos de un conjunto.
 - **AVG:** Media aritmética de los valores.
 - **COUNT:** Cuenta el número de tuplas o elementos.
 - **MAX:** Devuelve el valor más alto.
 - **MIN:** Devuelve el valor más bajo.
 - **Agrupamiento:** Permite agrupar tuplas de una relación según el valor de algunos atributos y aplicar una función agregada a cada grupo.
 - **Ejemplos:**
 - $G_{sum(Sueldo)}(Empleado)$ para el total de la planilla.
 - $G_{ND F CUENTA NSS, PROMEDIO SALARIO}(EMPLEADO)$ para contar empleados y salario medio por departamento.
 - Obtener sueldo más alto o más bajo (MAX, MIN).
 - Contar sueldos diferentes: $G_{Sueldos F COUNT Sueldos}(EMPLEADO)$.



- Total a pagar por factura: NumeroFactura G SUM Total (DetalleVenta)
(después de una proyección generalizada para Total).

Paso 5: Elaboración del Mapa Conceptual

Una vez que hayas investigado y comprendido todos los conceptos anteriores, diseña un mapa conceptual que sintetice la información de manera clara y organizada.

El mapa conceptual debe:

- **Partir** del concepto central "Álgebra Relacional".
- **Ramificarse** para incluir los conceptos de "Operadores Unarios", "Operadores Binarios" y "Operadores Extendidos".
- **Para cada operador:**
 - Incluir su **nombre** y **símbolo/notación**.
 - Una **breve definición** o su función principal.
 - Al menos un **ejemplo** (puedes usar los de las fuentes o adaptarlos).
- **Establecer relaciones** entre los conceptos utilizando líneas y palabras de enlace.
- **Ser visualmente claro** y fácil de entender.

Paso 6: Reflexión (Opcional, pero recomendado) Reflexiona sobre la relación entre el Álgebra Relacional y SQL. ¿Cómo el conocimiento del Álgebra Relacional facilita la comprensión de SQL?.

Criterios de Evaluación:

- **Investigación completa:** Se evalúa si se han abordado todos los conceptos solicitados para cada operador (definición, notación, ejemplos) y para el álgebra relacional en general.
- **Precisión de la Información:** La exactitud de las definiciones y ejemplos, respaldados por las fuentes.



- **Calidad del Mapa Conceptual:** Claridad, organización lógica, uso adecuado de símbolos y palabras de enlace, y correcta representación de las relaciones jerárquicas y cruzadas entre los conceptos.
 - **Citación:** Correcta citación de todas las fuentes utilizadas para cada afirmación.
-



Complemento educativo CE. 5.2 Exploración y Aplicación del Álgebra Relacional

Objetivos de Aprendizaje: Al finalizar esta actividad, los estudiantes serán capaces de:

- **Comprender y aplicar** los operadores fundamentales y extendidos del Álgebra Relacional.
- **Analizar y predecir** el resultado de consultas expresadas en Álgebra Relacional sobre una base de datos dada.
- **Formular consultas equivalentes en SQL** a partir de expresiones en Álgebra Relacional, y viceversa.
- **Utilizar eficazmente herramientas computacionales** para la creación, ejecución, validación y optimización básica de consultas en Álgebra Relacional.
- **Colaborar en equipo** para resolver problemas complejos y comunicar sus hallazgos de manera clara y concisa.

Duración Estimada: 4 a 6 horas (divididas en sesiones de laboratorio o trabajo grupal).

Materiales Necesarios:

- **Acceso a Computadoras:** Con un entorno de desarrollo o herramientas de gestión de bases de datos.
- **Software de Bases de Datos Relacionales:** (Opcional, pero recomendado para SQL) PostgreSQL, MySQL, SQL Server, etc.



- **Herramienta de Aprendizaje de Álgebra Relacional:** Se recomienda utilizar una de las siguientes, ya que permiten la traducción a SQL y la visualización:
 - RelationalQuery.
 - Relational.
- **Esquema de Base de Datos y Datos de Ejemplo:** Se puede utilizar la base de datos de "PROVEEDORES, COMPONENTES, ARTICULOS y ENVIOS" proporcionada en los ejercicios resueltos de Álgebra Relacional, o el ejemplo de sistema bancario.
- **Acceso a los Materiales del Curso/Fuentes:** Documentos sobre Álgebra Relacional y SQL.

Metodología: Aprendizaje basado en equipos (trabajo grupal) y enfoque práctico.

Desarrollo de la Actividad:

Fase 1: Preparación Teórica Individual (1 hora)

1. **Revisión de Conceptos:** Cada estudiante debe repasar los fundamentos del Álgebra Relacional, incluyendo:
 - **Definición:** Qué es el Álgebra Relacional y su naturaleza procedimental.
 - **Operadores Fundamentales:**
 - **Selección (σ):** Permite obtener un subconjunto horizontal de tuplas que cumplen una condición. Es un operador unario.
 - **Proyección (π):** Permite extraer columnas (atributos) y elimina duplicados. Es un operador unario.
 - **Producto Cartesiano ($\times$):** Combina todas las tuplas de dos relaciones, resultando en una relación con un esquema que es la unión de los esquemas y una cardinalidad igual al producto de las cardinalidades. Es un operador binario.



- **Unión ($\cup$):** Retorna el conjunto de tuplas que están en una relación, en otra, o en ambas. Las relaciones deben ser compatibles (mismo grado y dominios equivalentes). Es un operador binario.
- **Diferencia (-):** Retorna las tuplas que están en la primera relación pero no en la segunda. Las relaciones deben ser compatibles. Es un operador binario.
- **Operadores Extendidos (o Derivados):**
 - **Intersección ($\cap$):** Retorna las tuplas comunes a dos relaciones compatibles.
 - **Reunión Natural (Natural Join, $\bowtie$):** Combina relaciones con atributos comunes, eliminando columnas redundantes y reconstruyendo tablas normalizadas.
 - **División ($\div$):** Obtiene entidades de un tipo (A) que se conectan con *todas* las entidades de otro tipo (B).
 - **Proyección Generalizada ($\Pi_{F1, F2, \dots, Fn}(E)$):** Permite el uso de funciones aritméticas en la lista de proyección para crear atributos derivados (ej., calcular un bono sobre el sueldo).
 - **Funciones de Agregación ($\mathcal{G}\{f(a)\}(R)$ o $\mathcal{G}\{G1, G2, \dots, Gn\}F1(a1), F2(a2), \dots, Fm(am)(E)$):** Toman una colección de valores y devuelven un único valor (SUMA, PROMEDIO, MÁXIMO, MÍNIMO, CUENTA). Pueden aplicarse a grupos de tuplas.
- 2. **Familiarización con el Esquema:** Entender las tablas y sus relaciones en la base de datos de ejemplo a utilizar.

Fase 2: Trabajo Grupal y Aplicación Práctica (2-3 horas)

1. **Formación de Grupos:** El instructor dividirá la clase en grupos de 3-4 estudiantes.



2. **Asignación de Operadores:** Cada grupo recibirá (o elegirá) un conjunto específico de 2-3 operadores del Álgebra Relacional (incluyendo al menos un operador básico y uno extendido si es posible) para enfocarse.
3. **Diseño de Consultas y Ejemplos:** Para cada operador asignado, el grupo debe:
 - **Plantear una pregunta en lenguaje natural** que resuelva una necesidad de información usando el operador.
 - **Escribir la expresión en Álgebra Relacional** que responda a la pregunta, utilizando la notación correcta.
 - **Analizar la consulta paso a paso**, explicando cómo el operador transforma los datos y **prediciendo manualmente el resultado** para la base de datos de ejemplo.
 - **Traducir la consulta de Álgebra Relacional a una sentencia SELECT de SQL.**
4. **Uso de Herramientas Computacionales:**
 - **Cargar la Base de Datos:** Importar el esquema de la base de datos de ejemplo en la herramienta de aprendizaje de Álgebra Relacional seleccionada.
 - **Introducir y Ejecutar Consultas:** Ingresar las expresiones de Álgebra Relacional diseñadas en la herramienta y **ejecutarlas**.
 - **Validar Resultados:** Comparar los resultados obtenidos por la herramienta con los resultados predichos manualmente. Si hay discrepancias, identificar el error (en la consulta o en la predicción) y corregirlo.
 - **Explorar la Traducción a SQL:** Utilizar la funcionalidad de la herramienta para traducir la expresión de Álgebra Relacional a SQL y comparar con la traducción manual realizada.
 - **Explorar la Optimización (Opcional, si la herramienta lo permite):** Si la herramienta permite la optimización de consultas (como la de la Universidad de Zaragoza), explorar cómo se reordenan los operadores o se utilizan estadísticas para mejorar la eficiencia.



Fase 3: Preparación de la Exposición (1 hora)

1. **Elaboración de la Presentación:** Cada grupo creará una presentación visual (ej. PowerPoint, Google Slides) que incluya:
 - Título de la actividad y miembros del grupo.
 - Una breve introducción a los operadores asignados.
 - Para cada ejemplo diseñado:
 - La pregunta en lenguaje natural.
 - La expresión en Álgebra Relacional (bien formateada).
 - Una explicación clara del funcionamiento del operador y el resultado esperado.
 - La sentencia SQL equivalente.
 - **Capturas de pantalla claras del uso de la herramienta:** mostrando la consulta ingresada, el resultado obtenido y la traducción a SQL. Si la herramienta lo permite, incluir capturas de la visualización del árbol de expresión o los pasos de optimización.
 - Conclusiones sobre los desafíos encontrados, la utilidad de los operadores, y cómo las herramientas computacionales facilitaron el proceso de aprendizaje.

Fase 4: Exposición y Retroalimentación (1-2 horas)

1. **Presentaciones Grupales:** Cada grupo presentará sus ejemplos y hallazgos al resto de la clase. Se recomienda un tiempo de 10-15 minutos por grupo para la exposición, seguido de 5 minutos para preguntas y discusión.
2. **Discusión en Clase:** Fomentar un diálogo abierto donde los estudiantes puedan preguntar, comparar enfoques y aprender de las experiencias de otros grupos.
3. **Retroalimentación del Instructor:** El instructor proporcionará retroalimentación constructiva sobre la precisión de las consultas, la claridad de las explicaciones y



la eficacia del uso de las herramientas. Se destacarán las mejores prácticas y se corregirán posibles malentendidos.

Criterios de Evaluación (Sugeridos):

- **Precisión de las Consultas (40%):** Correcta aplicación de los operadores de Álgebra Relacional y su traducción a SQL.
- **Análisis y Comprensión (30%):** Claridad en la explicación del funcionamiento de los operadores y la lógica de las consultas, y exactitud de los resultados predichos manualmente.
- **Uso de Herramientas (20%):** Demostración efectiva del uso de la herramienta de aprendizaje de Álgebra Relacional para validar y explorar las consultas.
- **Presentación y Colaboración (10%):** Calidad de la presentación, claridad en la comunicación y participación equitativa de los miembros del grupo.

Esta actividad proporcionará una experiencia de aprendizaje profunda y práctica, conectando la teoría del Álgebra Relacional con su aplicación en entornos computacionales reales.



Complemento educativo CE. 5.3 Resolución Práctica de Consultas con Álgebra Relacional (Básica y Extendida)

Objetivos de Aprendizaje: Al finalizar esta actividad, los estudiantes serán capaces de:

- **Identificar y aplicar** correctamente los operadores fundamentales y extendidos del Álgebra Relacional para resolver problemas específicos de consulta de datos.
 - **Analizar la semántica** de las expresiones del Álgebra Relacional y predecir los resultados esperados sobre instancias de bases de datos dadas.
-



- **Traducir** expresiones de Álgebra Relacional a sentencias SQL equivalentes y viceversa.
- **Utilizar herramientas computacionales** específicas para el álgebra relacional para la creación, ejecución y validación de consultas, así como para la observación de su traducción a SQL.
- **Colaborar eficazmente** en grupos para el análisis y la resolución de ejercicios complejos, y presentar sus soluciones de manera clara y comprensible.

Duración Estimada: 4 a 6 horas (divididas en sesiones de laboratorio o trabajo grupal).

Materiales Necesarios:

- **Acceso a Computadoras:** Con un entorno de desarrollo o herramientas de gestión de bases de datos.
- **Software de Bases de Datos Relacionales:** (Recomendado para la ejecución de SQL) PostgreSQL, MySQL, SQL Server, etc.
- **Herramientas de Aprendizaje de Álgebra Relacional:** Es **fundamental** el uso de una herramienta que permita la entrada de expresiones en álgebra relacional, su ejecución y, preferiblemente, su traducción a SQL. Se sugiere el uso de:
 - La **herramienta desarrollada por Roberto Yus Peirote en la Universidad de Zaragoza**, ya que permite introducir expresiones, ejecutar consultas, visualizar árboles, optimizar y traducir a SQL.
 - Alternativas como **WinRDBI** o **RelationalQuery**.
- **Esquemas de Base de Datos y Datos de Ejemplo:**
 - **Esquema Principal: PROVEEDORES, COMPONENTES, ARTICULOS y ENVIOS (P, C, A, E).** Este esquema es ideal porque se proporcionan numerosos ejercicios resueltos en álgebra relacional y SQL.
 - **PROVEEDORES (P#, PNOMBRE, CATEGORIA, CIUDAD)**
 - **COMPONENTES (C#, CNOMBRE, COLOR, PESO, CIUDAD)**
 - **ARTICULOS (T#, TNOMBRE, CIUDAD)**
 - **ENVIOS (P#, C#, T#, CANTIDAD)**



- **Esquema Adicional: EMPLEADO** (para operadores extendidos específicos como la proyección generalizada y funciones de agregación).
 - **EMPLEADO (NSS, NOMBRE, APELLIDO, SUELDO, ND, ...)**
(asumiendo los atributos relevantes de los ejemplos proporcionados).
- **Materiales del Curso/Fuentes:** Documentos sobre Álgebra Relacional y SQL (como los proporcionados en esta conversación) para consulta.

Metodología: Aprendizaje basado en equipos (trabajo grupal) con un enfoque práctico y el uso intensivo de herramientas computacionales.

Desarrollo de la Actividad:

Fase 1: Preparación Teórica Individual (1 hora)

1. **Revisión de Conceptos:** Cada estudiante debe repasar los fundamentos del Álgebra Relacional, incluyendo operadores fundamentales (Selección (σ), Proyección (π), Producto Cartesiano ($\times$), Unión ($\cup$), Diferencia (-)) y operadores extendidos (Intersección ($\cap$), Reunión Natural ($\bowtie$), División ($\div$), Proyección Generalizada ($\Pi_{F1, F2, \dots, Fn}$), Funciones de Agregación ($\mathcal{G}_{f(a)}$)).
2. **Familiarización con los Esquemas de Base de Datos:** Los estudiantes deben revisar y comprender la estructura y los datos de ejemplo de los esquemas **PROVEEDORES, COMPONENTES, ARTICULOS y ENVIOS y EMPLEADO** (para los ejercicios de proyección generalizada y agregación).

Fase 2: Trabajo Grupal y Aplicación Práctica (2-3 horas)

1. **Formación de Grupos:** El instructor dividirá la clase en grupos de 3-4 estudiantes.



2. **Asignación de Ejercicios:** Cada grupo recibirá un conjunto de ejercicios que cubrirán la aplicación de operadores básicos y extendidos. Se recomienda que cada grupo trabaje con al menos un ejemplo de cada tipo de operador principal.
3. **Resolución y Análisis de Ejercicios:** Para cada ejercicio asignado, el grupo deberá seguir los siguientes pasos:
 - **a) Pregunta en Lenguaje Natural:** Plantear claramente la necesidad de información que la consulta debe resolver.
 - **b) Expresión en Álgebra Relacional:** Escribir la expresión utilizando la notación correcta del álgebra relacional.
 - **c) Análisis Manual:** Explicar paso a paso cómo cada operador transforma los datos y **predecir manualmente el resultado** para las bases de datos de ejemplo.
 - **d) Sentencia SQL Equivalente:** Traducir la expresión de Álgebra Relacional a una sentencia SELECT de SQL.
 - **e) Uso de Herramientas Computacionales:**
 - **Cargar el esquema y los datos** de ejemplo en la herramienta de Álgebra Relacional seleccionada (e.g., la herramienta de la Universidad de Zaragoza o WinRDBI).
 - **Introducir y ejecutar** la expresión de Álgebra Relacional en la herramienta.
 - **Validar los resultados** comparando la salida de la herramienta con la predicción manual. Corregir cualquier discrepancia.
 - Utilizar la función de la herramienta para **traducir la expresión de Álgebra Relacional a SQL** y comparar con la traducción manual.
 - (Opcional): Si la herramienta lo permite, explorar las opciones de **optimización de la consulta** para entender cómo se podría mejorar su eficiencia.

Aquí se presentan ejemplos de ejercicios específicos, usando los esquemas mencionados, para guiar la práctica:



○ **Ejemplos de Aplicación de Operadores Básicos (utilizando el esquema PROVEEDORES, COMPONENTES, ARTICULOS y ENVIOS):**

▪ **Selección (σ):**

- **Pregunta:** Obtener todos los detalles de los artículos que se montan en 'CACERES'.

▪ **Álgebra**

Relacional:

$\sigma_{\{CIUDAD='CACERES'\}}(ARTICULOS)$.

- **SQL Equivalente:** SELECT * FROM articulos WHERE ciudad = 'CACERES';.

- *(Ejemplifica la extracción de un subconjunto horizontal de tuplas que cumplen una condición).*

▪ **Proyección (π):**

- **Pregunta:** Obtener la lista de pares únicos de (COLOR, CIUDAD) de la tabla COMPONENTES.

▪ **Álgebra**

Relacional:

$\pi_{\{COLOR, CIUDAD\}}(COMPONENTES)$.

- **SQL Equivalente:** SELECT DISTINCT color, ciudad FROM componentes;

- *(Muestra la extracción de columnas específicas y la eliminación de duplicados).*

▪ **Producto Cartesiano ($\times$):**

- **Pregunta:** Generar todas las posibles combinaciones entre proveedores y artículos (esto es, el producto cartesiano de ambas relaciones, como paso previo a una reunión o JOIN).

- **Álgebra Relacional:** $PROVEEDORES \times ARTICULOS$.

- **SQL Equivalente:** SELECT * FROM PROVEEDORES, ARTICULOS;

- *(Ilustra la combinación de todas las tuplas de dos relaciones).*

▪ **Unión ($\cup$):**



- **Pregunta:** Suponiendo que se tiene una tabla PROVEEDORES_NUEVOS con el mismo esquema que PROVEEDORES, **obtener una lista de todos los proveedores, combinando los de la tabla original con los de la nueva tabla.**
- **Álgebra Relacional:** $\$PROVEEDORES \cup PROVEEDORES_NUEVOS\$$.
- **SQL Equivalente:** SELECT * FROM PROVEEDORES UNION SELECT * FROM PROVEEDORES_NUEVOS;
- *(Enfatiza la compatibilidad de esquemas para esta operación).*
- **Diferencia (-):**
 - **Pregunta:** Usando las tablas PROVEEDORES y PROVEEDORES_NUEVOS (compatibles), **obtener los proveedores que están en la tabla original pero no en la nueva.**
 - **Álgebra Relacional:** $\$PROVEEDORES - PROVEEDORES_NUEVOS\$$.
 - **SQL Equivalente:** SELECT * FROM PROVEEDORES EXCEPT SELECT * FROM PROVEEDORES_NUEVOS; (o MINUS según el SGBD).
 - *(Muestra la exclusión de tuplas comunes entre relaciones compatibles).*
- **Ejemplos de Aplicación de Operadores Extendidos (utilizando los esquemas PROVEEDORES, COMPONENTES, ARTICULOS, ENVIOS y EMPLEADO):**
 - **Intersección ($\$ \cap \$$):**
 - **Pregunta:** Obtener los números de proveedor (P#) de aquellos proveedores que suministran tanto el artículo 'T1' como el artículo 'T2' [177, Ejercicio 9].



- **Álgebra Relacional:** $\pi_{P\#}(\sigma_{T\#='T1'}(ENVIOS)) \cap \pi_{P\#}(\sigma_{T\#='T2'}(ENVIOS))$.
- **SQL Equivalente:** SELECT P# FROM ENVIOS WHERE T#='T1' INTERSECT SELECT P# FROM ENVIOS WHERE T#='T2';.
- *(Permite encontrar tuplas comunes a dos relaciones compatibles).*
- **Reunión Natural (Natural Join, $\bowtie$):**
 - **Pregunta:** Listar el nombre del proveedor (PNOMBRE) y el nombre del componente (CNOMBRE) de todos los envíos realizados.
 - **Álgebra Relacional:** $\pi_{PNOMBRE, CNOMBRE}(PROVEEDORES \bowtie ENVIOS \bowtie COMPONENTES)$.
 - **SQL Equivalente:** SELECT P.PNOMBRE, C.CNOMBRE FROM PROVEEDORES P NATURAL JOIN ENVIOS E NATURAL JOIN COMPONENTES C;
 - *(Combina relaciones eliminando columnas redundantes y restaurando conexiones semánticas).*
- **División ($\div$):**
 - **Pregunta:** Obtener los identificadores de artículos (T#) para los cuales se ha realizado envío de *todos* los componentes existentes en la base de datos [176, Ejercicio 30].
 - **Álgebra Relacional:** $\pi_{T\#}(\sigma_{\pi_{C\#}(ENVIOS) \div \pi_{C\#}(COMPONENTES)})$.
 - **SQL Equivalente (ejemplo usando NOT EXISTS para simular la división):**
 - SELECT DISTINCT E1.T#
 - FROM ENVIOS E1
 - WHERE NOT EXISTS (



- SELECT C2.C#
- FROM COMPONENTES C2
- WHERE NOT EXISTS (
- SELECT *
- FROM ENVIOS E3
- WHERE E3.T# = E1.T# AND E3.C# = C2.C#
-)
-);
- *(Este operador es útil para consultas tipo "para todos").*
- **Proyección Generalizada ($\Pi_{\{F1, F2, \dots, Fn\}}(E)$) (utilizando el esquema EMPLEADO):**
 - **Pregunta:** Mostrar el nombre del empleado, su apellido y un nuevo atributo 'BONO' calculado como el 15% de su 'SUELDO'.
 - **Álgebra Relacional:** $\Pi_{\{NOMBRE, APELLIDO, (SUELDO * 0.15 \rightarrow BONO)\}}(EMPLEADO)$.
 - **SQL Equivalente:** SELECT NOMBRE, APELLIDO, (SUELDO * 0.15) AS BONO FROM EMPLEADO;
 - *(Permite crear atributos derivados utilizando funciones aritméticas).*
- **Funciones de Agregación ($\mathcal{G}\{f(a)\}(R)$ o $\mathcal{G}\{G1, G2, \dots, Gn\}F1(a1), F2(a2), \dots, Fm(am)(E)$) (utilizando el esquema ENVIOS y EMPLEADO):**
 - **Pregunta (sin agrupación):** Calcular el número total de suministros, el número de artículos distintos suministrados y la cantidad total de artículos suministrados por el proveedor 'P2' [179, Ejercicio 15].
 - **Álgebra Relacional:** $\mathcal{G}\{COUNT(*) \rightarrow NumSuministros, COUNT(DISTINCT T#) \rightarrow$



ArticulosDistintos, SUM(CANTIDAD) \rightarrow TotalCantidad \} (\sigma_{P\#='P2'}(ENVIOS))\\$.

- **SQL Equivalente:** SELECT COUNT(*) AS NumSuministros, COUNT (DISTINCT T#) AS ArticulosDistintos, SUM(cantidad) AS TotalCantidad FROM envios WHERE P#='P2';.
- *(Agregan valores de una colección a un único resultado).*
- **Pregunta (con agrupación): Para cada artículo y componente suministrado, obtener los valores de C#, T# y la cantidad total correspondiente [180, Ejercicio 16].**
- **Álgebra Relacional:** $\mathcal{G}_{\{C\#, T\}}$ SUM(CANTIDAD) \rightarrow TOTAL(ENVIOS)\$.
- **SQL Equivalente:** SELECT C#, T#, SUM(cantidad) AS Total FROM envios GROUP BY C#, T#;.
- *(Agregan valores por grupos de tuplas).*

Fase 3: Preparación de la Exposición (1 hora)

1. **Elaboración de la Presentación:** Cada grupo creará una presentación visual (ej. PowerPoint, Google Slides) que incluya:
 - Título de la actividad y miembros del grupo.
 - Una breve introducción a los operadores asignados.
 - Para cada ejercicio resuelto:
 - La pregunta en lenguaje natural.
 - La expresión en Álgebra Relacional (bien formateada).
 - Una explicación clara del funcionamiento del operador y el resultado esperado (predicción manual).
 - La sentencia SQL equivalente.
 - **Capturas de pantalla claras del uso de la herramienta:** mostrando la consulta ingresada, el resultado obtenido y la traducción a SQL. Si



la herramienta lo permite, incluir capturas de la visualización del árbol de expresión o los pasos de optimización.

- Conclusiones del grupo sobre los desafíos encontrados, la utilidad de los operadores, y cómo las herramientas computacionales facilitaron el proceso de aprendizaje.

Fase 4: Exposición y Retroalimentación (1-2 horas)

1. **Presentaciones Grupales:** Cada grupo presentará sus ejemplos y hallazgos al resto de la clase. Se recomienda un tiempo de 10-15 minutos por grupo para la exposición, seguido de 5 minutos para preguntas y discusión.
 2. **Discusión en Clase:** Fomentar un diálogo abierto donde los estudiantes puedan preguntar, comparar enfoques y aprender de las experiencias de otros grupos.
 3. **Retroalimentación del docente:** El docente proporcionará retroalimentación constructiva sobre la precisión de las consultas, la claridad de las explicaciones y la eficacia del uso de las herramientas. Se destacarán las mejores prácticas y se corregirán posibles malentendidos.
-



TEMA 6 Introducción al lenguaje sql

6.1 Características

Introducción a SQL: El Lenguaje Universal de Bases de Datos

SQL (Structured Query Language), o Lenguaje de Consulta Estructurado, es un **lenguaje de programación estándar diseñado específicamente para interactuar con bases de datos relacionales**. Es una herramienta fundamental para analistas, científicos de datos, equipos de marketing, ventas y finanzas, permitiendo una gestión eficiente de la información y desarrollando aplicaciones rápidas, independientes y eficaces. SQL es considerado una de las principales razones del éxito comercial de las bases de datos relacionales.

El objetivo principal de SQL es **permitir la manipulación de datos de diversas maneras**, como modificarlos, borrarlos, añadirlos o simplemente extraerlos para informes. Las bases de datos SQL funcionan con tablas de información a las que se les ejecutan consultas, clasificadas en diferentes subconjuntos de comandos. Si bien existen diferencias entre los distintos paquetes de sistemas de gestión de bases de datos (DBMS) relacionales comerciales, SQL se ha convertido en un estándar (como SQL/89, SQL/92, SQL/1999 y la actual SQL/2016), lo que facilita la migración y la intercomunicabilidad entre sistemas que lo soportan fielmente.

Una sentencia SQL es como una frase que indica qué se quiere obtener y de dónde. Todas las sentencias comienzan con un verbo (palabra reservada que indica la acción a realizar) seguido de cláusulas que completan la frase, algunas obligatorias y otras opcionales. La ejecución de una cláusula SQL implica una **fase de análisis y verificación** (para asegurar que la formulación es correcta y descomponer la expresión) y una **etapa de optimización**



(donde se examinan posibles caminos de acceso a los datos para escoger el más eficiente), finalmente generando un plan de ejecución.

Subconjuntos de SQL: DDL, DML y Otros

SQL se divide en varios subconjuntos de comandos, cada uno con un propósito específico:

- **DDL (Data Definition Language - Lenguaje de Definición de Datos).**
- **DML (Data Manipulation Language - Lenguaje de Manipulación de Datos).**
- **DQL (Data Query Language - Lenguaje de Consulta de Datos).**
- **DCL (Data Control Language - Lenguaje de Control de Datos).**
- **TCL (Transaction Control Language - Lenguaje de Control de Transacciones).**

6.2 Lenguaje de Definición de Datos (LDD)

El DDL es un conjunto de comandos y sintaxis que permite a los usuarios **definir y mantener la estructura de la base de datos**. Su propósito es crear y gestionar las estructuras de almacenamiento de los datos sin modificar la información contenida en ellos. El esquema de la base de datos, definido por DDL, actúa como un plano que describe la organización y el almacenamiento de los datos.

Los **comandos DDL principales** incluyen:

- **CREATE:** Permite la **creación de nuevos objetos** en la base de datos, como bases de datos, tablas, índices, vistas, procedimientos almacenados y funciones.
 - **Ejemplo de CREATE DATABASE:** CREATE DATABASE UCAB.
 - **Ejemplo de CREATE TABLE:** CREATE TABLE ESTUDIANTE (ID INT, NOMBRE VARCHAR(60), FNACIMIENTO DATE, INSCRITO BOOLEAN);. Esta sentencia también permite especificar tipos de datos para cada columna y definir restricciones como claves primarias.



- **Ejemplo de CREATE VIEW:** Se utiliza para especificar una tabla virtual que deriva de otras tablas. `CREATE VIEW TRABAJA AS SELECT Nombre, Apellido1, NombreProyecto, Horas FROM EMPLEADO, PROYECTO, TRABAJA_EN WHERE Dni = DniEmpleado AND NumProy = NumProyecto;`
- **ALTER:** Se utiliza para **modificar la estructura de objetos existentes**. Esto puede incluir agregar, eliminar o cambiar columnas de una tabla, o renombrar objetos.
 - **Ejemplo para añadir una columna:** `ALTER TABLE ESTUDIANTE ADD COLUMN FECHA_INSCRITO DATE;`
 - **Ejemplo para añadir una restricción FOREIGN KEY:** `ALTER TABLE employees ADD CONSTRAINT fk_department FOREIGN KEY (department) REFERENCES departments(name);`
 - **Ejemplo para modificar propiedades de columna:** `ALTER TABLE empleados MODIFY nombre VARCHAR(10) NULL;` (para permitir valores nulos).
- **DROP:** Sirve para **eliminar permanentemente objetos** de la base de datos, como tablas, índices o bases de datos. Esta acción es irreversible y elimina los datos asociados.
 - **Ejemplo:** `DROP TABLE ESTUDIANTE;`
 - Al usar `DROP TABLE`, se puede elegir `CASCADE` (elimina la tabla y todas sus dependencias como vistas o restricciones) o `RESTRICT` (elimina la tabla solo si no hay dependencias).
- **TRUNCATE:** Permite **eliminar rápidamente todo el contenido de una tabla**, manteniendo la estructura de esta. Es más rápido que `DROP` para borrar todo el contenido, especialmente en tablas grandes, pero no permite la cláusula `WHERE`. Se considera DDL porque internamente borra y recrea la tabla.



Otros comandos DDL incluyen COMMENT (para añadir descripciones), RENAME (para cambiar nombres de objetos), CREATE SCHEMA (para organizar objetos en grupos lógicos) y ALTER SEQUENCE (para modificar propiedades de secuencias).

Los **beneficios del DDL** incluyen su independencia de plataforma (el modelo relacional se mantiene consistente en diferentes proveedores como Oracle, MySQL y Microsoft SQL Server) y la capacidad de crear scripts que definen la estructura de una base de datos para fines de respaldo, configuración de sistemas de prueba o generación de plantillas. DDL también es crucial para **definir restricciones de integridad** como claves primarias (PRIMARY KEY), claves foráneas (FOREIGN KEY) y restricciones de verificación (CHECK), asegurando que la información almacenada en la base de datos sea fiable y precisa.

6.3 Lenguaje de Manipulación de Datos (DML)

El DML define las sentencias SQL que **modifican los datos dentro de las tablas**. Es crucial para que usuarios y aplicaciones interactúen eficazmente con los datos, impulsando la lógica de las aplicaciones, la generación de informes y las interacciones de los usuarios. DML permite realizar el **ciclo CRUD completo (Create, Read, Update, Delete)**.

Los **comandos DML más comunes** son:

- **INSERT:** Se utiliza para **añadir nuevas filas (registros) a una tabla**.
 - **Sintaxis básica:** INSERT INTO tabla (columna1, columna2, ...) VALUES (valor1, valor2, ...);. La cantidad de columnas y valores deben ser iguales.
 - **Sentencia acortada:** Si se especifican todos los valores de una tabla en el orden correcto, se puede omitir la lista de columnas: INSERT INTO tabla VALUES (valor1, valor2, ...);.
 - **Ejemplo:** INSERT INTO agenda_telefonica (nombre, numero) VALUES ('Roberto Jeldrez', 4886850);.



- También se puede insertar datos desde el resultado de otra consulta
SELECT: INSERT INTO INVENTARIO_CD_2 SELECT NOMBRE_CD,
EN_EXISTENCIA FROM INVENTARIO_CD;.
- **UPDATE:** Se utiliza para **modificar los valores de un conjunto de registros existentes** en una tabla.
 - **Sintaxis:** UPDATE tabla SET campo1 = 'nuevo valor campo1' WHERE campo2 = 'N';.
 - Las cláusulas UPDATE y SET son obligatorias, mientras que WHERE es opcional; si se omite WHERE, todos los registros de la tabla serán actualizados.
 - **Ejemplo:** UPDATE mi_tabla SET campo1 = 'nuevo valor campo1' WHERE campo2 = 'N';.
 - También se puede usar una sentencia SELECT en la cláusula SET para actualizar valores.
- **DELETE:** Se utiliza para **eliminar una o más filas que coincidan con una condición** de una tabla.
 - **Ejemplo:** DELETE FROM Employees WHERE EmployeeID = 1001;.
 - La eliminación puede propagarse a tuplas de otras relaciones si se han especificado acciones de activación referencial en las restricciones de integridad.
- **SELECT:** Aunque es el comando principal del DQL, también se considera parte del DML ya que recupera datos que luego pueden ser manipulados o presentados. Es el comando SQL más importante y la mayoría de las consultas comienzan con él.
 - **Sintaxis básica:** SELECT Campos FROM Tabla;. Permite seleccionar todos los campos (SELECT *) o campos específicos.
 - **FROM:** Especifica la tabla o tablas de donde se seleccionarán los registros.
 - **WHERE:** Se usa para **filtrar los registros** que aparecerán en los resultados de la instrucción SELECT. Debe ir después de FROM.
 - **Operadores lógicos:** AND, OR, NOT.



- **Operadores de comparación:** <, >, <>, <=, >=, BETWEEN, LIKE, IN.
- **BETWEEN:** Permite seleccionar registros dentro de un intervalo.
- **LIKE:** Utilizado para la **comparación de patrones** en cadenas de texto, usando % (cero o más caracteres) y _ (un solo carácter).
- **IN:** Devuelve registros cuyo campo coincide con alguno de los valores en una lista o en una subconsulta.
- **Funciones de agregado:** Se utilizan dentro de una cláusula SELECT para devolver un **único valor que se aplica a un grupo de registros**.
 - **AVG:** Calcula el promedio.
 - **COUNT:** Devuelve el número de registros.
 - **SUM:** Devuelve la suma de valores.
 - **MAX:** Devuelve el valor más alto.
 - **MIN:** Devuelve el valor más bajo.
 - **Ejemplo de MIN:** SELECT Min(Gastos) AS EIMin FROM Pedidos WHERE Pais = 'Costa Rica';.
- **GROUP BY:** Combina los registros con valores idénticos en la lista de campos especificados en un único registro, permitiendo aplicar funciones de agregado a esos grupos.
 - **Ejemplo:** SELECT NumProyecto, NombreProyecto, COUNT(*) FROM PROYECTO, TRABAJA_EN WHERE NumProyecto = NumProy GROUP BY NumProyecto, NombreProyecto;.
- **HAVING:** Similar a WHERE, pero se aplica a los **grupos de registros** creados por GROUP BY, no a filas individuales. Los grupos que satisfacen la condición son los que aparecen en el resultado.
 - **Ejemplo:** SELECT PRECIO, CATEGORIA, SUM (A_LA_MANO) AS TOTAL_A_LA_MANO FROM EXISTENCIA_DISCO_COMPACTO GROUP BY PRECIO, CATEGORIA HAVING SUM(A_LA_MANO) > 10;.



- **ORDER BY:** Se utiliza para **ordenar los registros seleccionados** de acuerdo con un orden específico. Puede ser ascendente (ASC, valor por defecto) o descendente (DESC) y permite ordenar por múltiples campos.
 - **Ejemplo:** SELECT CodigoPostal, Nombre, Telefono FROM Clientes ORDER BY CodigoPostal DESC , Nombre ASC;
- **Manejo de varias tablas:**
 - **JOIN:** Utilizado para combinar filas de dos o más tablas basándose en una columna relacionada entre ellas. Las fuentes mencionan JOIN (que suele ser INNER JOIN), LEFT JOIN y RIGHT JOIN.
 - **UNION y UNION ALL:** Combinan el resultado de varias sentencias SELECT. UNION no muestra resultados duplicados, mientras que UNION ALL sí los muestra.

DML moderno también incluye técnicas avanzadas como Expresiones de Tabla Comunes (CTE) para lógica compleja, operaciones en bloque (INSERT INTO ... SELECT, MERGE) para mejorar el rendimiento con grandes volúmenes de datos, y funciones de ventana para análisis fila por fila.

Las **mejores prácticas para usar DML** incluyen el uso de consultas parametrizadas para proteger contra inyecciones SQL, agrupar sentencias DML relacionadas en transacciones para asegurar la consistencia de los datos, implementar manejo de errores adecuado, optimizar consultas (usando índices y filtros), validar y sanear la entrada de datos, y utilizar controles de acceso apropiados siguiendo el principio de menor privilegio.

Lenguajes de Control de Datos (DCL) y Control de Transacciones (TCL)

- **DCL:** Se emplea para administrar derechos y permisos para controlar el sistema de la base de datos. Sus comandos son GRANT (otorgar permisos) y REVOKE (revocar permisos).



- **TCL:** Trabaja con grupos de tareas preestablecidas de ejecución conjunta, definiendo características de transacciones y permitiendo guardar puntos de guardado (SAVEPOINT), ejecutar (COMMIT) o revertir cambios (ROLLBACK).

Lenguaje de Consulta de Datos (DQL)

El DQL está compuesto por comandos cuyo propósito es **obtener información relacionada a la consulta** para trabajar con otras operaciones. El comando principal del DQL es SELECT.

Conceptos Fundamentales de Bases de Datos Relacionales

Una **base de datos** es una colección de datos lógicamente coherente y organizada en un formato estructurado, con algún tipo de significado inherente, diseñada y construida para un propósito específico. No es un surtido aleatorio de datos. La información descriptiva de la base de datos, conocida como **metadatos**, también se almacena en ella misma, en un catálogo o diccionario.

El **Modelo Relacional**, introducido por E. F. Codd en 1970, se basa en principios matemáticos de la teoría de conjuntos y lógica de predicados. Su núcleo es la **relación**, que se puede considerar como una matriz compuesta de filas (llamadas **tuplas**) y columnas (que representan **atributos**). Este modelo apoya la recuperación de datos sencilla, la integración de datos y proporciona una estructura de base de datos independiente de las aplicaciones.

Un **Sistema de Gestión de Bases de Datos (DBMS)** es una colección de programas que permite a los usuarios crear y mantener una base de datos, facilitando los procesos de definición, construcción, manipulación y compartición de datos entre varios usuarios y aplicaciones.

En el contexto de una base de datos:



- Una **entidad** es una "cosa" u "objeto" distinguible en el mundo real (como una persona o un libro), con un conjunto de propiedades que la identifican unívocamente.
- Los **atributos** son las propiedades que describen una entidad. Cada atributo tiene un conjunto de valores permitidos llamado **dominio**. Los atributos pueden ser **simples** (indivisibles) o **compuestos** (divididos en subpartes más pequeñas, como una dirección).
- Una **relación** es una asociación entre diferentes entidades.
- La **razón de cardinalidad** expresa el número de entidades a las que otra entidad puede estar asociada a través de un conjunto de relaciones, mientras que la **restricción de participación** especifica si la existencia de una entidad depende de su relación con otra.

SQL es un lenguaje robusto y esencial que proporciona las herramientas para definir la estructura de las bases de datos (DDL) y manipular los datos almacenados en ellas (DML), siendo un pilar fundamental en la gestión de la información en el mundo tecnológico actual.



Complemento educativo CE. 6.1 Fundamentos del Lenguaje SQL

Objetivo General: Al finalizar esta actividad, el estudiante será capaz de comprender las características fundamentales de SQL, clasificar sus sentencias principales y reconocer la sintaxis básica de los comandos más utilizados para la definición y manipulación de datos en bases de datos relacionales.

Público Objetivo: Estudiantes, desarrolladores o analistas de datos principiantes en SQL.

Duración Estimada: 2-3 horas.



Materiales: Las fuentes de estudio proporcionadas sobre SQL (excerpts from "Conceptos básicos de SQL", "DDL: Lenguaje de Definición de Datos", "DML: Lenguaje de Manipulación de Datos", "Diferencias entre DDL y DML en el modelado de datos", "Guía de ejercicios SQL para Sistemas de Bases de Datos I", "Introducción a SQL: Fundamentos y Aplicaciones Prácticas", "Los comandos SQL más importantes", "SQL: Qué es, para qué sirve, sus características y cómo funciona el lenguaje", y "U6 - E2 - DEMH - Lenguaje de Definición de Datos (DDL) en SQL").

Instrucciones de la Actividad:

1. Introducción a SQL (15 minutos):

- Investiga en las fuentes qué es SQL y cuál es su propósito principal en la gestión de bases de datos relacionales.
- Identifica por qué SQL es considerado un lenguaje estándar y cuáles organizaciones lo certifican.

2. Características de SQL (20 minutos):

- Explora las fuentes para identificar y listar las **características clave de SQL** que lo hacen tan popular y esencial en el mundo de la tecnología actual. Intenta explicar brevemente cada característica con tus propias palabras.

3. Clasificación de las Sentencias SQL (45 minutos):

- Lee las secciones pertinentes en las fuentes que clasifican los comandos de SQL.
- Identifica y describe los **subconjuntos principales de SQL**:
 - **DDL** (Lenguaje de Definición de Datos).
 - **DML** (Lenguaje de Manipulación de Datos).
 - **DQL** (Lenguaje de Consulta de Datos). (Nota: aunque a menudo se incluye SELECT dentro de DML, algunas fuentes lo presentan como un subconjunto separado de DQL).
 - **DCL** (Lenguaje de Control de Datos).



- **TCL** (Lenguaje de Control de Transacciones).
 - Para cada subconjunto, anota el **propósito fundamental** que cumple en la base de datos.

4. Diseño de la Tabla de Sentencias SQL (60-90 minutos):

- Utilizando la información recopilada y la sintaxis básica presentada en las fuentes, completa la siguiente tabla.
- Concéntrate en las sentencias más comunes y fundamentales de DDL, DML y DQL.
- La columna "Sintaxis Básica" debe mostrar un ejemplo general o la estructura más simplificada del comando.

Tabla de Sentencias SQL: Clasificación, Descripción y Sintaxis Básica

Clasificación	Nombre de la Sentencia	Descripción	Sintaxis Básica
DDL	CREATE TABLE	Se utiliza para crear nuevas tablas en la base de datos, definiendo sus columnas y tipos de datos.	CREATE TABLE nombre_tabla (columna1 tipo_dato, columna2 tipo_dato, ...);
DDL	ALTER TABLE	Permite modificar la estructura de una tabla existente , como añadir, eliminar o modificar columnas.	ALTER TABLE nombre_tabla ADD columna_nueva tipo_dato; ALTER TABLE nombre_tabla DROP COLUMN nombre_columna; ALTER TABLE nombre_tabla ALTER COLUMN nombre_columna tipo_dato_nuevo;



DDL	DROP TABLE	Se utiliza para eliminar permanentemente una tabla y todos sus datos de la base de datos.	DROP TABLE nombre_tabla;
DDL	TRUNCATE TABLE	Borra todo el contenido de una tabla de manera muy rápida , manteniendo la estructura de la tabla vacía. Es una operación DDL porque internamente recrea la tabla y no ejecuta transacciones.	TRUNCATE TABLE nombre_tabla;
DML / DQL	SELECT	Se utiliza para consultar y recuperar datos de una o más tablas en la base de datos.	SELECT columna1, columna2 FROM nombre_tabla; SELECT * FROM nombre_tabla;
DML	INSERT INTO	Se utiliza para añadir uno o más registros (filas) a una tabla en la base de datos.	INSERT INTO nombre_tabla (columna1, columna2) VALUES (valor1, valor2);
DML	UPDATE	Se utiliza para modificar los valores de registros existentes en una tabla.	UPDATE nombre_tabla SET columna1 = 'nuevo_valor' WHERE condicion;
DML	DELETE FROM	Se utiliza para eliminar uno o más registros existentes de una tabla.	DELETE FROM nombre_tabla WHERE condicion;
DQL	WHERE	Una cláusula que filtra los registros según una o más condiciones, aplicándose a	SELECT * FROM tabla WHERE condicion;



		sentencias SELECT, UPDATE o DELETE.	
DQL	ORDER BY	Se utiliza para ordenar el conjunto de resultados de una sentencia SELECT en orden ascendente (ASC) o descendente (DESC).	<code>`SELECT * FROM tabla ORDER BY columna ASC\`</code>
DQL	GROUP BY	Agrupar filas que tienen los mismos valores en una o más columnas, a menudo usado con funciones de agregado para resumir datos.	<code>SELECT columna_agrupacion, funcion_agregado(columna) FROM tabla GROUP BY columna_agrupacion;</code>
DQL	JOIN	Se utiliza para combinar filas de dos o más tablas basándose en una columna relacionada entre ellas. Existen varios tipos (INNER, LEFT, RIGHT, FULL OUTER).	<code>SELECT c.nombre, a.accion FROM clientes c JOIN acciones a ON c.cid = a.cid;</code>
DDL	CREATE VIEW	Crea una vista , que es una tabla virtual basada en el resultado de una consulta SQL, almacenando solo su definición y no los datos físicos.	<code>CREATE VIEW nombre_vista AS SELECT columnas FROM tabla WHERE condicion;</code>

Reflexión y Aplicación (Opcional):



- Piensa en un ejemplo de la vida real donde se necesite cada tipo de sentencia (DDL, DML, DQL).
- ¿Cómo crees que el conocimiento de estos comandos básicos facilita la interacción con los sistemas de gestión de bases de datos (SGBD) como MySQL, PostgreSQL, Oracle, etc.?



Complemento educativo CE. 6.2 Aplicación Práctica de Comandos SQL Esenciales (DDL y DML)

Objetivo General: Al finalizar esta actividad, el estudiante será capaz de diseñar, modificar y mantener la estructura de una base de datos utilizando sentencias DDL, así como manipular (insertar, actualizar, eliminar) y consultar datos de manera eficiente mediante sentencias DML/DQL, aplicando la sintaxis básica y avanzada de SQL.

Público Objetivo: Estudiantes o profesionales con conocimientos básicos de bases de datos relacionales que deseen aplicar comandos SQL.

Duración Estimada: 3-4 horas.

Materiales: Las fuentes de estudio proporcionadas: "Conceptos básicos de SQL", "DDL: Lenguaje de Definición de Datos", "DML: Lenguaje de Manipulación de Datos", "Diferencias entre DDL y DML en el modelado de datos", "Guía de ejercicios SQL para Sistemas de Bases de Datos I", "Introducción a SQL: Fundamentos y Aplicaciones Prácticas", "Los comandos SQL más importantes", "SQL: Qué es, para qué sirve, sus características y cómo funciona el lenguaje", y "U6 - E2 - DEMH - Lenguaje de Definición de Datos (DDL) en SQL".

Escenario de la Actividad: Usted es el nuevo administrador de bases de datos de una pequeña empresa en crecimiento. Su primera tarea es modelar y gestionar la información



de los **Departamentos** y los **Empleados** de la compañía, así como realizar operaciones de mantenimiento y consulta sobre estos datos.

Instrucciones Generales:

1. **Revisión de Fuentes:** Antes de comenzar, revise las secciones relevantes en las fuentes sobre DDL y DML, prestando especial atención a la sintaxis y los ejemplos de cada comando.
2. **Entorno SQL (Recomendación):** Aunque no se requiere ejecutar los comandos en un sistema real para esta actividad, se recomienda encarecidamente practicar en un SGBD (Sistema de Gestión de Bases de Datos) como MySQL, PostgreSQL o SQLite para verificar la ejecución y los resultados. Si no dispone de uno, puede escribir solo el código SQL y describir el resultado esperado.
3. **Documentación de Resultados:** Para cada tarea, deberá:
 - Escribir la(s) sentencia(s) SQL completa(s) utilizada(s).
 - Explicar brevemente qué hace cada sentencia o el cambio que realiza en la base de datos/datos.
 - Para las consultas (SELECT), describa el conjunto de resultados esperado.

Parte 1: Lenguaje de Definición de Datos (DDL)

En esta parte, se enfocará en definir y modificar la estructura de las tablas de la empresa. El DDL se encarga de crear, modificar y eliminar objetos de la base de datos.

Esquema Inicial Propuesto:

- **DEPARTAMENTOS:**
 - DEPT_NO (Numérico, Clave Primaria)
 - DNOMBRE (Cadena de Caracteres, Obligatorio)
 - LOC (Cadena de Caracteres)



- **EMPLEADOS:**

- EMP_NO (Numérico, Clave Primaria)
- APELLIDO (Cadena de Caracteres, Obligatorio)
- OFICIO (Cadena de Caracteres)
- SALARIO (Numérico Decimal)
- DEPT_NO (Numérico, Clave Foránea a DEPARTAMENTOS)

Tareas DDL:**1. Creación de Tablas Iniciales:**

- Escriba la sentencia CREATE TABLE para la tabla DEPARTAMENTOS, incluyendo la definición de sus columnas y la clave primaria.
 - *Sintaxis de referencia:* CREATE TABLE nombre_tabla (columna1 tipo_dato, columna2 tipo_dato, ...);.
- Escriba la sentencia CREATE TABLE para la tabla EMPLEADOS, incluyendo la definición de sus columnas y su clave primaria. **Aún no incluya la clave foránea..**

2. Modificación de Estructura de Tablas (ALTER TABLE):

- **Agregar una nueva columna:** Añada una columna llamada FECHA_CONTRATO de tipo DATE a la tabla EMPLEADOS.
 - *Sintaxis de referencia:* ALTER TABLE table_name ADD column_name datatype;.
- **Modificar el tipo de una columna:** Cambie el tipo de dato de la columna LOC en la tabla DEPARTAMENTOS a VARCHAR(20) para permitir nombres de localidad más largos.
 - *Sintaxis de referencia:* ALTER TABLE table_name ALTER COLUMN column_name datatype;.
- **Añadir una restricción de chequeo (CHECK):** Agregue una restricción a la columna SALARIO en EMPLEADOS para asegurar que el salario siempre sea un valor positivo (mayor que 0).



- *Sintaxis de referencia:* NumeroDpto NUMBER NOT NULL CHECK (NumeroDpto > 0 AND NumeroDpto < 21);.
- **Añadir una restricción de clave foránea (FOREIGN KEY):** Establezca la relación entre EMPLEADOS y DEPARTAMENTOS añadiendo la clave foránea a DEPT_NO en EMPLEADOS que referencia a DEPT_NO en DEPARTAMENTOS.
 - *Sintaxis de referencia:* ALTER TABLE employees ADD CONSTRAINT fk_department FOREIGN KEY (department) REFERENCES departments(name);.
- **Eliminar una columna:** Elimine la columna OFICIO de la tabla EMPLEADOS. (Esto es solo para fines de práctica; en un escenario real, elimine con precaución).
 - *Sintaxis de referencia:* ALTER TABLE 'NOMBRE_TABLA' DROP COLUMN NOMBRE_COLUMNNA;.

3. Eliminación de Objetos:

- **TRUNCATE TABLE:** Escriba la sentencia para **truncar** la tabla EMPLEADOS. Explique brevemente la diferencia entre TRUNCATE TABLE y DROP TABLE.
 - *Sintaxis de referencia:* TRUNCATE TABLE 'NOMBRE_TABLA';.
- **DROP TABLE:** Escriba la sentencia para **eliminar permanentemente** la tabla DEPARTAMENTOS (considere las dependencias).
 - *Sintaxis de referencia:* DROP TABLE nombre_tabla;.

Parte 2: Lenguaje de Manipulación de Datos (DML y DQL)

Para esta parte, asuma que las tablas DEPARTAMENTOS y EMPLEADOS ya existen con el siguiente esquema y contienen algunos datos para que pueda manipularlos y consultarlos.

Esquema y Datos (asumidos existentes):



- **DEPARTAMENTOS:**
 - DEPT_NO (INT, PK)
 - DNOMBRE (VARCHAR(30), NOT NULL)
 - LOC (VARCHAR(20))
- **EMPLEADOS:**
 - EMP_NO (INT, PK)
 - APELLIDO (VARCHAR(15), NOT NULL)
 - OFICIO (VARCHAR(10))
 - SALARIO (DECIMAL(10,2))
 - DEPT_NO (INT, FK)

Datos de Ejemplo para INSERT (puede usar estos para su tarea 2.1):

- **DEPARTAMENTOS:**
 - (10, 'CONTABILIDAD', 'CARACAS')
 - (20, 'INVESTIGACION', 'MADRID')
 - (30, 'VENTAS', 'BARCELONA')
- **EMPLEADOS:**
 - (7369, 'SANCHEZ', 'EMPLEADO', 800.00, 20)
 - (7499, 'ARROYO', 'VENDEDOR', 1600.00, 30)
 - (7566, 'JIMENEZ', 'DIRECTOR', 2975.00, 20)
 - (7654, 'MARTIN', 'VENDEDOR', 1250.00, 30)
 - (7782, 'CEREZO', 'DIRECTOR', 2450.00, 10)

Tareas DML/DQL:

1. Insertar Datos (INSERT INTO):

- Escriba las sentencias INSERT INTO para poblar las tablas DEPARTAMENTOS y EMPLEADOS con los datos de ejemplo proporcionados arriba.
 - *Sintaxis de referencia:* INSERT INTO tabla (columna1, columna2) VALUES (valor1, valor2);.

2. Consultar Datos (SELECT):



- **Consulta Básica:** Muestre todos los datos de la tabla EMPLEADOS.
 - *Sintaxis de referencia:* `SELECT * FROM table_name;`
- **Selección de Columnas Específicas:** Obtenga solo el APELLIDO y SALARIO de todos los empleados.
- **Filtrado con WHERE y operadores:**
 - Obtenga todos los datos de los empleados cuyo SALARIO sea mayor o igual a 2000.
 - Obtenga APELLIDO y OFICIO de los empleados cuyo APELLIDO comience con 'A' o termine con 'Z'.
 - Liste los empleados que trabajan en los departamentos 'CONTABILIDAD' o 'VENTAS' utilizando el operador IN.
 - Liste los empleados cuyo SALARIO esté entre 1000 y 2500 (inclusive) utilizando BETWEEN.
 - Encuentre los empleados cuyo OFICIO sea NULL (si tuviera alguno).
- **Ordenamiento (ORDER BY):** Obtenga APELLIDO, SALARIO de todos los empleados, ordenados por SALARIO de forma descendente.
 - *Sintaxis de referencia:* `SELECT column1, column2 FROM table_name ORDER BY column_name ASC|DESC;`
- **Funciones de Agregado:**
 - Calcule el SALARIO promedio de todos los empleados.
 - Cuente cuántos empleados hay en el departamento con DEPT_NO 20.
 - Encuentre el SALARIO máximo y mínimo de todos los empleados.
- **Agrupamiento (GROUP BY) y Filtrado de Grupos (HAVING):**
 - Para cada OFICIO, muestre el OFICIO y el número total de empleados que lo desempeñan.
 - Liste los OFICIOS y el SALARIO promedio de cada uno, pero solo para aquellos OFICIOS donde el SALARIO promedio sea mayor a 1500..



- **Uniones (JOIN):** Muestre el APELLIDO del empleado, su SALARIO, el DNOMBRE del departamento y la LOC del departamento. Combine las tablas EMPLEADOS y DEPARTAMENTOS.
 - *Sintaxis de referencia:* select nombre, telefono, accion, cantidad from clientes join acciones on clientes.cid=acciones.cid;.

3. Actualizar Datos (UPDATE):

- Incremente el SALARIO de todos los empleados con OFICIO 'VENDEDOR' en un 15%.
 - *Sintaxis de referencia:* UPDATE table_name SET column1 = 'nuevo_valor' WHERE condition;.
- Cambie la LOC del departamento 'CONTABILIDAD' a 'SEVILLA'.

4. Eliminar Datos (DELETE FROM):

- Elimine al empleado con EMP_NO 7369.
 - *Sintaxis de referencia:* DELETE FROM table_name WHERE condition;.
- Elimine todos los empleados cuyo OFICIO sea 'EMPLEADO'.

Documentación de Resultados (Ejemplo de Formato):

Parte 1: Lenguaje de Definición de Datos (DDL)

Tarea 1.1: Creación de Tablas Iniciales

1.a. Crear tabla DEPARTAMENTOS:

```sql

CREATE TABLE DEPARTAMENTOS (

DEPT\_NO INT PRIMARY KEY,



```
DNOMBRE VARCHAR(30) NOT NULL,

LOC VARCHAR(15)

);
```

*Descripción:* Esta sentencia crea la tabla DEPARTAMENTOS con tres columnas: DEPT\_NO como clave primaria numérica, DNOMBRE como una cadena de texto obligatoria, y LOC como una cadena de texto opcional.

#### **1.b. Crear tabla EMPLEADOS (sin FK):**

```
CREATE TABLE EMPLEADOS (

 EMP_NO INT PRIMARY KEY,

 APELLIDO VARCHAR(15) NOT NULL,

 OFICIO VARCHAR(10),

 SALARIO DECIMAL(10,2)

 -- DEPT_NO INT -- Se añade como FK en la tarea 1.2

);
```

*Descripción:* Esta sentencia crea la tabla EMPLEADOS con sus columnas y clave primaria, tal como se especifica en el esquema inicial, excepto por la clave foránea que se añadirá más adelante.

*(Continúe con este formato para todas las tareas de la actividad, incluyendo las sentencias SQL, su descripción y, para las consultas SELECT, el resultado esperado.)*

---



## Referencias

- Korth, H. y Silbertchatz, A. Fundamentos de Bases de datos. (5ª ed.). Ed. McGraw Hill.
- De Miguel, A. y Piattini, M. Fundamentos y modelos de bases de datos. (2ª ed.), Ed. Alfaomega Ramma.
- Rob, P. y Coronel, C. Sistemas de Base de Datos (Diseño, Implementación y Administración).(5ª ed.).Ed. Thomson.
- Kroenke, D. Procesamiento de Base de Datos –Fundamentos, diseño e implementación-.( 8ª ed). Ed. Pearson Prentice-Hall.
- De Miguel, S. et al. Diseño de Base de datos –Problemas resueltos-. Ed. Alfaomega & Ra-ma.
- Groff, J. y Weinberg, P. Manual de referencia SQL. Ed. McGraw Hill.
- Date, C. (2012) Introduction to Data Base Systems (7ª Ed.) Ed. Addison – Wesley Iberoamericana.
- De Miguel, S. y Piattini, M. Concepción y Diseño de Base de datos –Del modelo E-R al modelo relacional. Ed. Addison Wesley Iberoamericana, Ra-ma.



## Instrumentación Didáctica

|                                                                                  |                                                                                         |                             |
|----------------------------------------------------------------------------------|-----------------------------------------------------------------------------------------|-----------------------------|
|  | Instrumentación Didáctica para la Formación y Desarrollo de Competencias Profesionales. | Código: ITSLP-SAC-PG-001-02 |
|                                                                                  |                                                                                         | Revisión: 4                 |
|                                                                                  |                                                                                         | Página: 1 de 18             |

Tecnológico Nacional de México

Subdirección Académica

Instrumentación Didáctica para la Formación y Desarrollo de Competencias Profesionales

Nombre de la asignatura: FUNDAMENTOS DE BASES DE DATOS

Plan de estudios: AEF-1031

Clave de la asignatura: DBB4

Periodo: Enero-junio/2023

Horas teoría - horas prácticas - créditos: 2 - 3 - 5

Atributos de Egreso: P.I. Problemas ingenieriles-I,

## 1. Caracterización de la asignatura

Esta asignatura aporta al perfil del egresado la capacidad para analizar, diseñar y gestionar sistemas de bases de datos conforme a los requerimientos del entorno para garantizar la integridad, disponibilidad y confidencialidad de la información, así como para desarrollar e implementar sistemas de información para la gestión de procesos y apoyo en la toma de decisiones, utilizando metodologías basadas en estándares internacionales.

Es importante porque el estudiante adquiere las competencias en el análisis y el diseño de base de datos, que le permitirán desarrollar aplicaciones para sistemas de información robustos que ofrezcan garantía en el manejo de la información. Es conveniente mencionar que hoy en día la información forma parte del capital intangible de las organizaciones y cada vez se demandan sistemas de que garanticen la integridad y seguridad de la misma.

La asignatura propicia el dominio de modelos de diseño de base de datos basados en reglas de normalización, de integridad y de seguridad. Esta asignatura requiere como competencia previa que el estudiante comprenda y aplique los conceptos y propiedades de álgebra de conjuntos, relaciones y álgebra booleana adquiridas en matemáticas discretas. Se relaciona con asignaturas posteriores donde se apliquen bases de datos y desarrollen aplicaciones para el tratamiento de información.

## 2. Intención didáctica

|                                                                                  |                                                                                                |                                    |
|----------------------------------------------------------------------------------|------------------------------------------------------------------------------------------------|------------------------------------|
|  | <b>Instrumentación Didáctica para la Formación y Desarrollo de Competencias Profesionales.</b> | <b>Código: ITSLP-SAC-PG-001-02</b> |
|                                                                                  |                                                                                                | <b>Revisión: 4</b>                 |
|                                                                                  |                                                                                                | <b>Página: 2 de 18</b>             |

El programa de la asignatura se organiza en seis temas, en los cuales se incluyen aspectos teóricos y de aplicación relacionados con modelado lógico de base de datos y lenguajes de acceso a base de datos.

El tema uno proporciona al estudiante el sustento teórico de las bases de datos, como son los objetivos, los diferentes modelos, la clasificación, las áreas de aplicación y arquitecturas que sirven de fundamento para que el estudiante incursione en el área de conocimiento de base de datos. Se recomienda que, en el tema de Arquitectura de la base de datos, se aborden los temas de niveles de abstracción, tipos de usuarios y tipos de lenguajes.

En el tema dos se estudia el proceso de diseño conceptual de las bases de datos aplicando el modelo Entidad – Relación (E-R), como una herramienta para modelar los esquemas en una forma consistente y estandarizada. El docente debe promover que el estudiante elija problemas reales y efectúe un análisis de las reglas de negocio antes de elaborar los diagramas E-R.

La mayoría de los sistemas gestores de bases de datos (SGBD) están basados en el modelo relacional, por lo que en el tema tres, se aborda este modelo, en el que se conoce y comprende su estructura, elementos que lo conforman y sus reglas de integridad. El docente deberá propiciar que el estudiante identifique la relación que existe entre el modelo E-R y el modelo relacional.

En el tema cuatro se estudian las formas normales de base de datos que garantizan la integridad de la base de datos y evitan la redundancia de información, contando con la posibilidad de ahondar en otras formas normales como la cuarta y quinta. Se recomienda que el docente proponga ejemplos de entidades para aplicar las reglas de normalización y demostrar claramente la diferencia o diferencias de entidades no normalizadas y normalizadas.

En el tema cinco se conoce y comprende el uso y aplicación del álgebra relacional como lenguaje de consulta formal a base de datos, los operadores básicos y los operadores del álgebra relacional extendida. Se sugiere que el docente realice planteamientos de consulta a base de datos.

El tema seis contempla aplicar los comandos básicos del Lenguaje de Definición y de Manipulación de Datos, haciendo uso de las herramientas del Sistema Gestor de Base de Datos, entre los cuales se deben considerar la creación de base de datos, creación de tablas y definición de llaves primarias y foráneas, la manipulación y consulta de la base de datos por medio de las operaciones de inserción, eliminación, modificación y consulta de datos. Es importante que el profesor aborde este tema a nivel básico, ya que en la asignatura de Taller de Base de Datos se dará profundidad en la definición, manipulación y control de la base de datos.

Es relevante que el docente a partir del tema 2, trabaje con los estudiantes en la identificación de un problema real y promueva la aplicación gradual de las competencias adquiridas en cada tema, logrando integrar un proyecto de asignatura. Esto propicia que el estudiante desarrolle las competencias genéricas para el análisis y resolución de problemas reales y aplique los conocimientos en la práctica.

Se deben promover las actividades de investigación y exposición con la intención de desarrollar competencias genéricas como la capacidad de análisis y síntesis y la expresión oral.

El docente debe en todo momento desempeñar el papel de guía, con la finalidad de generar un ambiente propicio de aprendizaje.

### 3. Competencia de la asignatura

|                                                                                  |                                                                                                |                                    |
|----------------------------------------------------------------------------------|------------------------------------------------------------------------------------------------|------------------------------------|
|  | <b>Instrumentación Didáctica para la Formación y Desarrollo de Competencias Profesionales.</b> | <b>Código: ITSLP-SAC-PG-001-02</b> |
|                                                                                  |                                                                                                | <b>Revisión: 4</b>                 |
|                                                                                  |                                                                                                | <b>Página: 3 de 18</b>             |

Analiza requerimientos y diseña bases de datos para generar soluciones al tratamiento de información basándose en modelos y estándares.

#### 4. Análisis por competencias específicas

Competencia No. : 1 Introducción a las bases de datos

Descripción:

Conoce y comprende los conceptos básicos de base de datos para proponer soluciones en el tratamiento de información.

| Temas y subtemas para desarrollar la competencia específica                                                                                                                                                                                                         | Actividades de aprendizaje (estudiante)                                                                                                                                                                                                                                                                                                                                                                                                                                                 | Actividades de enseñanza (profesor)                                                                                                                                                                                                                                                                               | Desarrollo de competencias genéricas                                                                                                                                                                                                                  | Horas teórico-práctica |
|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------------------------|
| 1.1 Conceptos básicos<br>1.2 Objetivos de las Bases de Datos<br>1.3 Áreas de Aplicación de los Sistemas de Bases de datos<br>1.4 Modelos de bases de datos<br>1.5 Clasificación de Bases de Datos<br>1.6 Arquitectura de base de datos<br>1.7 Arquitectura del SGBD | <ul style="list-style-type: none"> <li>Investigar los conceptos fundamentales, objetivos, usos y aplicaciones de las bases de datos y elaborar un mapa conceptual.</li> <li>Identificar los modelos de base de datos a través de una investigación y elaborar una tabla comparativa que incluya su definición y representación de datos.</li> <li>Identificar la clasificación de bases de datos a través de una investigación y elaborar un cuadro sinóptico que incluya su</li> </ul> | Búsqueda de información, lectura, estudio y comprensión de los temas de la asignatura.<br>Comprensión de conceptos de manera teórica para poder comprender la asignatura.<br>Conceptos básicos<br>Objetivos de las Bases de Datos<br>Áreas de Aplicación de los Sistemas de Bases de datos<br>Modelos de bases de | <ul style="list-style-type: none"> <li>Habilidades para buscar, procesar y analizar información procedente de diversas fuentes.</li> <li>Capacidad de abstracción, análisis y síntesis.</li> <li>Capacidad de comunicación oral y escrita.</li> </ul> | 15                     |



|                                                                                  |                                                                                         |                             |
|----------------------------------------------------------------------------------|-----------------------------------------------------------------------------------------|-----------------------------|
|  | Instrumentación Didáctica para la Formación y Desarrollo de Competencias Profesionales. | Código: ITSLP-SAC-PG-001-02 |
|                                                                                  |                                                                                         | Revisión: 4                 |
|                                                                                  |                                                                                         | Página: 5 de 18             |

| Evidencia de aprendizaje                                                                                 | %   | A   | Evaluación formativa de la competencia | Atributos de Egreso |
|----------------------------------------------------------------------------------------------------------|-----|-----|----------------------------------------|---------------------|
| 1._ Reporte de la instalación de un DBMS                                                                 | 10  | 10  | Guía de observación                    |                     |
| 2._ Informe sobre búsqueda de información, lectura, estudio y comprensión de los temas de la asignatura. | 30  | 30  | Lista de cotejo                        |                     |
| 3._ Realiza una representación gráfica sobre los conceptos aprendidos                                    | 40  | 40  | Guía de observación                    |                     |
| 4._ Examen                                                                                               | 20  | 20  | Revisión de respuestas correctas       |                     |
| Total:                                                                                                   | 100 | 100 |                                        |                     |

Competencia No. : 2 Diseño de Bases de Datos con el modelo E-R.

Descripción:

Conoce y aplica el modelo E-R para el diseño conceptual de bases de datos con el fin de organizar la información y atender necesidades del entorno.

| Temas y subtemas para desarrollar la competencia específica                                                                                        | Actividades de aprendizaje (estudiante)                                                                                                                                                                                                                               | Actividades de enseñanza (profesor)                                                                                                                                                                                                                                                                | Desarrollo de competencias genéricas                                                                                                                                                                                                                                          | Horas teórico-práctica |
|----------------------------------------------------------------------------------------------------------------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------------------------|
| 2.1 El Proceso de Diseño<br>2.2 Modelo Entidad-Relación<br>2.3 Diseño con diagramas E-R<br>2.4 Modelo E-R extendido<br>2.5 La Notación E-R con UML | <ul style="list-style-type: none"> <li>Identificar los componentes del modelo ER y E-R extendido, sus restricciones de diseño, mediante una investigación y elaborar un reporte.</li> <li>Investigar diversas herramientas de software para el modelado de</li> </ul> | <ul style="list-style-type: none"> <li>Presentar los temas y guiar al estudiante para identificar los componentes del modelo ER y E-R extendido, sus restricciones de diseño, mediante una investigación y elaborar un reporte.</li> <li>Presentar los temas y guiar al estudiante para</li> </ul> | <ul style="list-style-type: none"> <li>Capacidad de abstracción, análisis y síntesis.</li> <li>Capacidad de aplicar los conocimientos en la práctica.</li> <li>Capacidad para identificar, plantear y resolver problemas.</li> <li>Capacidad de trabajo en equipo.</li> </ul> | 15                     |

|                                                                                  |                                                                                                |                                    |
|----------------------------------------------------------------------------------|------------------------------------------------------------------------------------------------|------------------------------------|
|  | <b>Instrumentación Didáctica para la Formación y Desarrollo de Competencias Profesionales.</b> | <b>Código: ITSLP-SAC-PG-001-02</b> |
|                                                                                  |                                                                                                | <b>Revisión: 4</b>                 |
|                                                                                  |                                                                                                | <b>Página: 6 de 18</b>             |

|  |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                        |                                                                                                                                                                                                                                                                                                                                                                                                      |  |  |
|--|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--|--|
|  | <p>bases de datos que le permitan conocer y elegir una de ellas.</p> <ul style="list-style-type: none"> <li>• Seleccionar ejemplos de aplicación del modelo E-R y E-R extendido, analizarlos en grupos de trabajo y exponerlos, haciendo uso de herramientas computacionales de diseño de base de datos disponibles.</li> <li>• Analizar diversas situaciones del contexto y diseñar bases de datos con diagramas E-R.</li> <li>• Elegir un problema del entorno, describirlo, identificar los objetos de información, relaciones entre ellos y diseñar la base de datos con el diagrama E-R, utilizando la notación básica</li> </ul> | <p>que investigue diversas herramientas de software para el modelado de bases de datos que le permitan conocer y elegir una de ellas.</p> <ul style="list-style-type: none"> <li>• Seleccionar ejemplos de aplicación del modelo E-R y E-R extendido, analizarlos en grupos de trabajo y exponerlos, haciendo uso de herramientas computacionales de diseño de base de datos disponibles.</li> </ul> |  |  |
|--|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--|--|

| Indicadores de alcance                                                             | Valor del indicador |
|------------------------------------------------------------------------------------|---------------------|
| A. Comprende el diseño del modelo Entidad relación                                 | 70                  |
| B. Comprende el diseño del modelo Entidad relación extendido y su modelado con UML | 30                  |

### Niveles de desempeño

| Desempeño | Nivel de desempeño | Indicadores de alcance | Valoración numérica |
|-----------|--------------------|------------------------|---------------------|
|-----------|--------------------|------------------------|---------------------|

|                                                                                  |                                                                                         |  |                             |
|----------------------------------------------------------------------------------|-----------------------------------------------------------------------------------------|--|-----------------------------|
|  | Instrumentación Didáctica para la Formación y Desarrollo de Competencias Profesionales. |  | Código: ITSLP-SAC-PG-001-02 |
|                                                                                  |                                                                                         |  | Revisión: 4                 |
|                                                                                  |                                                                                         |  | Página: 7 de 18             |

|                          |              |                                                                                                                                              |                   |
|--------------------------|--------------|----------------------------------------------------------------------------------------------------------------------------------------------|-------------------|
| Competencia alcanzada    | Excelente    | Cumple en su totalidad con los indicadores de alcance.                                                                                       | 95 - 100          |
|                          | Notable      | Cumple en su totalidad con los indicadores de alcance de manera parcial.                                                                     | 85 - 94           |
|                          | Bueno        | Cumple con al menos 60% de los indicadores con desempeño excelente.                                                                          | 75 - 84           |
|                          | Suficiente   | Cumple con al menos 60% de los indicadores con desempeño parcial.                                                                            | 70 - 74           |
| Competencia no alcanzada | Insuficiente | No se cumple con el 100% de evidencias conceptuales, procedimentales y actitudinales de los indicadores definidos en el desempeño excelente. | NA (No alcanzada) |

### Matriz de evaluación

| Evidencia de aprendizaje                                                                       | %   | A  | B  | Evaluación formativa de la competencia | Atributos de Egreso |
|------------------------------------------------------------------------------------------------|-----|----|----|----------------------------------------|---------------------|
| 1._ Estudia y elabora un reporte sobre la identificación los elementos para el diseño,         | 20  | 10 | 10 | Guía de observación                    |                     |
| 2._ Entrega reporte de la realización de 25 ejercicios completos con el modelado de los mismos | 40  | 20 | 20 | Rúbrica para ejercicios                |                     |
| 3._ Reporta las herramientas de modelado para su uso                                           | 10  | 10 |    | Lista de cotejo                        |                     |
| 4._ Examen                                                                                     | 30  | 30 |    | Revisión de respuestas correctas       |                     |
| Total:                                                                                         | 100 | 70 | 30 |                                        |                     |

Competencia No. : 3 Modelo relacional

Descripción:

Conoce y aplica el modelo relacional para la generación de esquemas de base de datos con el fin de organizar la información y atender necesidades del entorno.

|  |  |  |  |  |
|--|--|--|--|--|
|  |  |  |  |  |
|--|--|--|--|--|

|                                                                                  |                                                                                                |                                    |
|----------------------------------------------------------------------------------|------------------------------------------------------------------------------------------------|------------------------------------|
|  | <b>Instrumentación Didáctica para la Formación y Desarrollo de Competencias Profesionales.</b> | <b>Código: ITSLP-SAC-PG-001-02</b> |
|                                                                                  |                                                                                                | <b>Revisión: 4</b>                 |
|                                                                                  |                                                                                                | <b>Página: 8 de 18</b>             |

| Temas y subtemas para desarrollar la competencia específica                                                                                                                                                                                   | Actividades de aprendizaje (estudiante)                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                      | Actividades de enseñanza (profesor)                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                    | Desarrollo de competencias genéricas                                                                                                                                                                                                                                                  | Horas teórico-práctica |
|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------------------------|
| 3.1 Introducción al modelo relacional<br>3.2 Conversión de Modelo E-R a Modelo relacional<br>3.3 Esquema de la base de datos<br>3.4 Restricciones<br>3.3.1 Integridad de entidad<br>3.3.2 Integridad referencial<br>3.5 Integridad de dominio | <ul style="list-style-type: none"> <li>• Identificar los componentes y las reglas de diseño de base de datos del modelo relacional mediante una investigación y elaborar un reporte.</li> <li>• Analizar diversas situaciones del contexto, diseñar bases de datos y crear el esquema aplicando el modelo relacional.</li> <li>• Convertir el diagrama E-R del problema planteado en el tema anterior a modelo relacional que le permita identificar las coincidencias entre los objetos de información de la base de datos en ambos modelos aplicando las reglas de diseño de base de datos. Presentarlo y discutirlo en el grupo.</li> <li>• Crear el esquema de base de datos haciendo uso de las herramientas del gestor y generar el esquema de forma automática en la herramienta de modelado utilizada. Elaborar un reporte que incluya el</li> </ul> | <ul style="list-style-type: none"> <li>• Presentar los temas y guiar al estudiante para que identifique los componentes y las reglas de diseño de base de datos del modelo relacional mediante una investigación y elaborar un reporte.</li> <li>• Presentar los temas y guiar al estudiante para que analice diversas situaciones del contexto, diseñar bases de datos y crear el esquema aplicando el modelo relacional.</li> <li>• Presentar los temas y guiar al estudiante para que convierta el diagrama E-R del problema planteado en el tema anterior a modelo relacional que le permita identificar las coincidencias entre los objetos de información de la base de datos en ambos modelos aplicando las reglas de diseño de base de datos. Presentarlo y discutirlo en el grupo.</li> </ul> | <ul style="list-style-type: none"> <li>• Capacidad de abstracción, análisis y síntesis.</li> <li>• Capacidad de aplicar los conocimientos en la práctica.</li> <li>• Capacidad para identificar, plantear y resolver problemas.</li> <li>• Capacidad de trabajo en equipo.</li> </ul> | 12                     |

|                                                                                  |                                                                                         |                             |
|----------------------------------------------------------------------------------|-----------------------------------------------------------------------------------------|-----------------------------|
|  | Instrumentación Didáctica para la Formación y Desarrollo de Competencias Profesionales. | Código: ITSLP-SAC-PG-001-02 |
|                                                                                  |                                                                                         | Revisión: 4                 |
|                                                                                  |                                                                                         | Página: 9 de 18             |

|  |                                                                                                                        |  |  |  |
|--|------------------------------------------------------------------------------------------------------------------------|--|--|--|
|  | diagrama conceptual de base de datos, el código y los comentarios que demuestren la interpretación lógica del esquema. |  |  |  |
|--|------------------------------------------------------------------------------------------------------------------------|--|--|--|

| Indicadores de alcance                                       | Valor del indicador |
|--------------------------------------------------------------|---------------------|
| A. Comprende el modelo relacional                            | 60                  |
| B. Convierte el modelo entidad relación al modelo relacional | 40                  |

### Niveles de desempeño

| Desempeño                | Nivel de desempeño | Indicadores de alcance                                                                                                                       | Valoración numérica |
|--------------------------|--------------------|----------------------------------------------------------------------------------------------------------------------------------------------|---------------------|
| Competencia alcanzada    | Excelente          | Cumple en su totalidad con los indicadores de alcance.                                                                                       | 95 - 100            |
|                          | Notable            | Cumple en su totalidad con los indicadores de alcance de manera parcial.                                                                     | 85 - 94             |
|                          | Bueno              | Cumple con al menos 60% de los indicadores con desempeño excelente.                                                                          | 75 - 84             |
|                          | Suficiente         | Cumple con al menos 60% de los indicadores con desempeño parcial.                                                                            | 70 - 74             |
| Competencia no alcanzada | Insuficiente       | No se cumple con el 100% de evidencias conceptuales, procedimentales y actitudinales de los indicadores definidos en el desempeño excelente. | NA (No alcanzada)   |

### Matriz de evaluación

| Evidencia de aprendizaje | % | A | B | Evaluación formativa de la competencia | Atributos de Egreso |
|--------------------------|---|---|---|----------------------------------------|---------------------|
|                          |   |   |   |                                        |                     |

|                                                                                  |                                                                                         |  |  |                             |  |
|----------------------------------------------------------------------------------|-----------------------------------------------------------------------------------------|--|--|-----------------------------|--|
|  | Instrumentación Didáctica para la Formación y Desarrollo de Competencias Profesionales. |  |  | Código: ITSLP-SAC-PG-001-02 |  |
|                                                                                  |                                                                                         |  |  | Revisión: 4                 |  |
|                                                                                  |                                                                                         |  |  | Página: 10 de 18            |  |

|                                                        |     |    |    |                                              |  |
|--------------------------------------------------------|-----|----|----|----------------------------------------------|--|
| 1._ Reporte sobre 5 ejercicios                         | 20  | 20 |    | Rúbrica para ejercicios                      |  |
| 2._ Reporta la conversión del modelo E-R al relacional | 20  |    | 20 | Rúbrica para ejercicios                      |  |
| 3._ Presenta en plenaria el análisis de 1 ejercicio    | 30  | 10 | 20 | Rúbrica para calificación multidisciplinaria |  |
| 4._ Examen                                             | 30  | 30 |    |                                              |  |
| Total:                                                 | 100 | 60 | 40 |                                              |  |

Competencia No. : 4 Normalización de bases de datos

Descripción:

Aplica el proceso de normalización al diseño de los esquemas de bases de datos para detectar anomalías y garantizar la integridad de la información.

| Temas y subtemas para desarrollar la competencia específica                                                                                                                                                               | Actividades de aprendizaje (estudiante)                                                                                                                                                                                                                                                                                                                                                                               | Actividades de enseñanza (profesor)                                                                                                                                                                                                                                                                                                                                       | Desarrollo de competencias genéricas                                                                                                                                                                                                                                         | Horas teórico-práctica |
|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------------------------|
| 4.1 Conceptos básicos<br>4.2 Primera forma normal.<br>4.3 Dependencias funcionales y transitivas.<br>4.4 Segunda forma normal.<br>4.5 Tercera forma normal.<br>4.6 Forma normal Boyce-Codd.<br>4.7 Otras formas normales. | <ul style="list-style-type: none"> <li>Identificar mediante una investigación, el concepto de normalización, las reglas aplicables en cada una de las formas normales y las anomalías que resuelve cada una de ellas y elaborar un mapa conceptual.</li> <li>Aplicar el proceso de normalización de base de datos a los esquemas generados en las situaciones del contexto analizadas en el tema anterior.</li> </ul> | <ul style="list-style-type: none"> <li>Presentar los temas y guiar al estudiante para que identifique mediante una investigación, el concepto de normalización, las reglas aplicables en cada una de las formas normales y las anomalías que resuelve cada una de ellas y elaborar un mapa conceptual.</li> <li>Presentar los temas y guiar al estudiante para</li> </ul> | <ul style="list-style-type: none"> <li>Capacidad de abstracción, análisis y síntesis.</li> <li>Capacidad de aplicar los conocimientos en la práctica.</li> <li>Capacidad para identificar, plantear y resolver problemas.</li> <li>Capacidad de trabajo en equipo</li> </ul> | 13                     |

|                                                                                  |                                                                                         |                             |
|----------------------------------------------------------------------------------|-----------------------------------------------------------------------------------------|-----------------------------|
|  | Instrumentación Didáctica para la Formación y Desarrollo de Competencias Profesionales. | Código: ITSLP-SAC-PG-001-02 |
|                                                                                  |                                                                                         | Revisión: 4                 |
|                                                                                  |                                                                                         | Página: 11 de 18            |

|  |                                                                                                                                                                                                                                                                                                                                                  |                                                                                                                                                   |  |  |
|--|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------|--|--|
|  | <ul style="list-style-type: none"> <li>Aplicar el proceso de normalización al esquema de base de datos del problema planteado anteriormente y elaborar un documento donde se muestre paso a paso la transición del diseño al aplicar cada forma normal, describiendo las anomalías encontradas. Presentarlo y discutirlo en el grupo.</li> </ul> | que aplique el proceso de normalización de base de datos a los esquemas generados en las situaciones del contexto analizadas en el tema anterior. |  |  |
|--|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------|--|--|

| Indicadores de alcance                                                     | Valor del indicador |
|----------------------------------------------------------------------------|---------------------|
| A. Comprende el proceso de normalización en sus 3 primeras formas normales | 70                  |
| B. Aplica el proceso de normalización                                      | 30                  |

### Niveles de desempeño

| Desempeño                | Nivel de desempeño | Indicadores de alcance                                                                                                                       | Valoración numérica |
|--------------------------|--------------------|----------------------------------------------------------------------------------------------------------------------------------------------|---------------------|
| Competencia alcanzada    | Excelente          | Cumple en su totalidad con los indicadores de alcance.                                                                                       | 95 - 100            |
|                          | Notable            | Cumple en su totalidad con los indicadores de alcance de manera parcial.                                                                     | 85 - 94             |
|                          | Bueno              | Cumple con al menos 60% de los indicadores con desempeño excelente.                                                                          | 75 - 84             |
|                          | Suficiente         | Cumple con al menos 60% de los indicadores con desempeño parcial.                                                                            | 70 - 74             |
| Competencia no alcanzada | Insuficiente       | No se cumple con el 100% de evidencias conceptuales, procedimentales y actitudinales de los indicadores definidos en el desempeño excelente. | NA (No alcanzada)   |

|                                                                                  |                                                                                         |                             |
|----------------------------------------------------------------------------------|-----------------------------------------------------------------------------------------|-----------------------------|
|  | Instrumentación Didáctica para la Formación y Desarrollo de Competencias Profesionales. | Código: ITSLP-SAC-PG-001-02 |
|                                                                                  |                                                                                         | Revisión: 4                 |
|                                                                                  |                                                                                         | Página: 12 de 18            |

## Matriz de evaluación

| Evidencia de aprendizaje                                             | %   | A  | B  | Evaluación formativa de la competencia | Atributos de Egreso |
|----------------------------------------------------------------------|-----|----|----|----------------------------------------|---------------------|
| 1._ Reporte de la investigación sobre los conceptos de normalización | 30  | 20 | 10 | Guía de observación                    |                     |
| 2._ Reporte de los ejercicios normalizados en sus 3FN                | 40  | 30 | 10 | Rúbrica para la solución de ejercicios |                     |
| 3._ Examen escrito                                                   | 30  | 20 | 10 | Revisión de respuestas correctas       |                     |
| Total:                                                               | 100 | 70 | 30 |                                        |                     |

Competencia No. : 5 Álgebra relacional.

Descripción:

Aplica operadores de álgebra relacional básica y extendida para acceder a la información de base de datos.

| Temas y subtemas para desarrollar la competencia específica                                | Actividades de aprendizaje (estudiante)                                                                                                                                                                                                                                                          | Actividades de enseñanza (profesor)                                                                                                                                                                                                                                                                                                                                              | Desarrollo de competencias genéricas                                                                                                                                                                                                                                        | Horas teórico-práctica |
|--------------------------------------------------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------------------------|
| 5.1 Operaciones fundamentales del álgebra relacional.<br>5.2 Álgebra relacional extendida. | <ul style="list-style-type: none"> <li>Investigar el concepto de álgebra relacional, operadores unarios (selección y proyección), operadores binarios (producto cartesiano, join o reunión, unión, intersección, diferencia, división) y operadores del álgebra relacional extendida.</li> </ul> | <ul style="list-style-type: none"> <li>Presentar los temas y guiar al estudiante para que investigue el concepto de álgebra relacional, operadores unarios (selección y proyección), operadores binarios (producto cartesiano, join o reunión, unión, intersección, diferencia, división) y operadores del álgebra relacional extendida. Elaborar un mapa conceptual.</li> </ul> | <ul style="list-style-type: none"> <li>Capacidad de abstracción análisis y síntesis.</li> <li>Capacidad de aplicar los conocimientos en la práctica.</li> <li>Capacidad para identificar, plantear y resolver problemas.</li> <li>Capacidad de trabajo en equipo</li> </ul> | 15                     |

|                                                                                  |                                                                                         |                             |
|----------------------------------------------------------------------------------|-----------------------------------------------------------------------------------------|-----------------------------|
|  | Instrumentación Didáctica para la Formación y Desarrollo de Competencias Profesionales. | Código: ITSLP-SAC-PG-001-02 |
|                                                                                  |                                                                                         | Revisión: 4                 |
|                                                                                  |                                                                                         | Página: 13 de 18            |

|  |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             |                                                                                                                                                                                                                                                                                                                 |  |  |
|--|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--|--|
|  | <p>Elaborar un mapa conceptual.</p> <ul style="list-style-type: none"> <li>• Seleccionar ejemplos donde se apliquen los operadores del álgebra relacional básica y extendida, analizarlos en grupos de trabajo y exponerlo haciendo uso de las herramientas computacionales disponibles.</li> <li>• Resolver ejercicios de aplicación del álgebra relacional básica y extendida de acuerdo a los requerimientos solicitados sobre la información de los esquemas de base de datos creados en temas anteriores, según lo requiera el caso y elaborar documentación de resultados.</li> </ul> | <ul style="list-style-type: none"> <li>• Presentar los temas y guiar al estudiante para que resuelva ejemplos donde se apliquen los operadores del álgebra relacional básica y extendida, analizarlos en grupos de trabajo y exponerlo haciendo uso de las herramientas computacionales disponibles.</li> </ul> |  |  |
|--|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--|--|

| Indicadores de alcance                                             | Valor del indicador |
|--------------------------------------------------------------------|---------------------|
| A. Comprende las operaciones fundamentales del álgebra relacional. | 70                  |
| B. Aplica las operaciones fundamentales del álgebra relacional.    | 30                  |

### Niveles de desempeño

|  |  |  |  |
|--|--|--|--|
|  |  |  |  |
|--|--|--|--|

|                                                                                  |                                                                                         |  |                             |
|----------------------------------------------------------------------------------|-----------------------------------------------------------------------------------------|--|-----------------------------|
|  | Instrumentación Didáctica para la Formación y Desarrollo de Competencias Profesionales. |  | Código: ITSLP-SAC-PG-001-02 |
|                                                                                  |                                                                                         |  | Revisión: 4                 |
|                                                                                  |                                                                                         |  | Página: 14 de 18            |

| Desempeño                | Nivel de desempeño | Indicadores de alcance                                                                                                                       | Valoración numérica |
|--------------------------|--------------------|----------------------------------------------------------------------------------------------------------------------------------------------|---------------------|
| Competencia alcanzada    | Excelente          | Cumple en su totalidad con los indicadores de alcance.                                                                                       | 95 - 100            |
|                          | Notable            | Cumple en su totalidad con los indicadores de alcance de manera parcial.                                                                     | 85 - 94             |
|                          | Bueno              | Cumple con al menos 60% de los indicadores con desempeño excelente.                                                                          | 75 - 84             |
|                          | Suficiente         | Cumple con al menos 60% de los indicadores con desempeño parcial.                                                                            | 70 - 74             |
| Competencia no alcanzada | Insuficiente       | No se cumple con el 100% de evidencias conceptuales, procedimentales y actitudinales de los indicadores definidos en el desempeño excelente. | NA (No alcanzada)   |

### Matriz de evaluación

| Evidencia de aprendizaje                                                                | %   | A  | B  | Evaluación formativa de la competencia | Atributos de Egreso |
|-----------------------------------------------------------------------------------------|-----|----|----|----------------------------------------|---------------------|
| 1._ Estudia y elabora un reporte sobre las operaciones básicas del Álgebra relacional   | 40  | 40 |    | Guía de observación                    |                     |
| 2._ Modela y reporta los resultados de ejercicios y prácticas otorgadas por su profesor | 30  |    | 30 | Rúbrica para ejercicios                |                     |
| 3._ :xamen Escrito                                                                      | 30  | 30 |    | Revisión de respuestas correctas       |                     |
| Total:                                                                                  | 100 | 70 | 30 |                                        |                     |

Competencia No. : 6 Introducción al lenguaje SQL

Descripción:

Aplica los comandos básicos del lenguaje SQL para la definición y manipulación de bases de datos.

|                                                                                  |                                                                                                |                                    |
|----------------------------------------------------------------------------------|------------------------------------------------------------------------------------------------|------------------------------------|
|  | <b>Instrumentación Didáctica para la Formación y Desarrollo de Competencias Profesionales.</b> | <b>Código: ITSLP-SAC-PG-001-02</b> |
|                                                                                  |                                                                                                | <b>Revisión: 4</b>                 |
|                                                                                  |                                                                                                | <b>Página: 15 de 18</b>            |

| Temas y subtemas para desarrollar la competencia específica                                                     | Actividades de aprendizaje (estudiante)                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                               | Actividades de enseñanza (profesor)                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                | Desarrollo de competencias genéricas                                                                                                                                                                                                                                         | Horas teórico-práctica |
|-----------------------------------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------------------------|
| 6.1 Características<br>6.2 Lenguaje de Definición de Datos (LDD)<br>6.3 Lenguaje de Manipulación de Datos (LMD) | <ul style="list-style-type: none"> <li>Investigar las características, la clasificación y sintaxis básica de sentencias del lenguaje SQL. Elaborar una tabla que incluya la clasificación, el nombre de la sentencia, su descripción y sintaxis.</li> <li>Resolver ejercicios de aplicación de comandos básicos del LDD ( create database, create table, alter table, drop table,entre otros) y del LMD (insert, update, delete y select) de acuerdo a los requerimientos solicitados sobre la información de los esquemas de base de datos creados en temas anteriores, según lo requiera el caso y elaborar documentación de resultados.</li> </ul> | <ul style="list-style-type: none"> <li>Presentar temas y guiar al estudiante para que investiguelas características, la clasificación y sintaxis básica de sentencias del lenguaje SQL. Elaborar una tabla que incluya la clasificación, el nombre de la sentencia, su descripción y sintaxis.</li> <li>Presentar temas y guiar al estudiante para que resuelva ejercicios de aplicación de comandos básicos del LDD ( create database, create table, alter table, drop table,entre otros) y del LMD (insert, update, delete y select) de acuerdo a los requerimientos solicitados sobre la información de los esquemas de base de datos creados en temas anteriores, según lo requiera el caso y elaborar documentación de resultados.</li> </ul> | <ul style="list-style-type: none"> <li>Capacidad de abstracción análisis y síntesis.</li> <li>Capacidad de aplicar los conocimientos en la práctica.</li> <li>Capacidad para identificar, plantear y resolver problemas.</li> <li>Capacidad de trabajo en equipo,</li> </ul> | 12                     |

|                                                                                  |                                                                                         |                             |
|----------------------------------------------------------------------------------|-----------------------------------------------------------------------------------------|-----------------------------|
|  | Instrumentación Didáctica para la Formación y Desarrollo de Competencias Profesionales. | Código: ITSLP-SAC-PG-001-02 |
|                                                                                  |                                                                                         | Revisión: 4                 |
|                                                                                  |                                                                                         | Página: 16 de 18            |

| Indicadores de alcance                                  | Valor del indicador |
|---------------------------------------------------------|---------------------|
| A. Comprende el Lenguaje de Definición de Datos (LDD)   | 50                  |
| B. Comprende el Lenguaje de Manipulación de Datos (LMD) | 50                  |

### Niveles de desempeño

| Desempeño                | Nivel de desempeño | Indicadores de alcance                                                                                                                       | Valoración numérica |
|--------------------------|--------------------|----------------------------------------------------------------------------------------------------------------------------------------------|---------------------|
| Competencia alcanzada    | Excelente          | Cumple en su totalidad con los indicadores de alcance.                                                                                       | 95 - 100            |
|                          | Notable            | Cumple en su totalidad con los indicadores de alcance de manera parcial.                                                                     | 85 - 94             |
|                          | Bueno              | Cumple con al menos 60% de los indicadores con desempeño excelente.                                                                          | 75 - 84             |
|                          | Suficiente         | Cumple con al menos 60% de los indicadores con desempeño parcial.                                                                            | 70 - 74             |
| Competencia no alcanzada | Insuficiente       | No se cumple con el 100% de evidencias conceptuales, procedimentales y actitudinales de los indicadores definidos en el desempeño excelente. | NA (No alcanzada)   |

### Matriz de evaluación

| Evidencia de aprendizaje                                                                                      | %  | A  | B  | Evaluación formativa de la competencia | Atributos de Egreso |
|---------------------------------------------------------------------------------------------------------------|----|----|----|----------------------------------------|---------------------|
| 1._ Entrega reporte de ejercicios resueltos con las operaciones básicas del lenguaje de definición de datos   | 40 | 40 |    | Rúbrica para ejercicios resueltos      |                     |
| 2._ Entrega reporte de ejercicios resueltos con las operaciones básicas del lenguaje de manipulación de datos | 30 |    | 30 | Rúbrica para ejercicios resueltos      |                     |
| 3._ Resuelve examen                                                                                           | 30 | 10 | 20 | Revisión de respuestas                 |                     |

|                                                                                  |                                                                                         |                             |
|----------------------------------------------------------------------------------|-----------------------------------------------------------------------------------------|-----------------------------|
|  | Instrumentación Didáctica para la Formación y Desarrollo de Competencias Profesionales. | Código: ITSLP-SAC-PG-001-02 |
|                                                                                  |                                                                                         | Revisión: 4                 |
|                                                                                  |                                                                                         | Página: 17 de 18            |

|        |     |    |    |  |
|--------|-----|----|----|--|
| Total: | 100 | 50 | 50 |  |
|--------|-----|----|----|--|

## 5. Fuentes de información y Apoyos didácticos

| Fuentes de información:                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                       | Apoyos didácticos:                                                               |
|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------|
| <ol style="list-style-type: none"> <li>1. Korth, H. y Silbertchatz, A. Fundamentos de Bases de datos. (5ª ed.). Ed. McGraw Hill.</li> <li>2. De Miguel, A. y Piattini, M. Fundamentos y modelos de bases de datos. (2ª ed.), Ed. AlfaOmega Ramma.</li> <li>3. Rob, P. y Coronel, C. Sistemas de Base de Datos (Diseño, Implementación y Administración). (5ª ed.).Ed. Thomson.</li> <li>4. Kroenke, D. Procesamiento de Base de Datos –Fundamentos, diseño e</li> <li>5. implementación-.( 8ª ed). Ed. Pearson Prentice-Hall.</li> <li>6. De Miguel, S. et al. Diseño de Base de datos –Problemas resueltos-. Ed. Alfaomega &amp; Ra-ma.</li> <li>7. Groff, J. y Weinberg, P. Manual de referencia SQL. Ed. McGraw Hill.</li> <li>8. Date, C. (2012) Introduction to Data Base Systems (7ª Ed.) Ed. Addison – Wesley Iberoamericana.</li> <li>9. De Miguel, S. y Piattini, M. Concepción y Diseño de Base de datos –Del modelo E-R al modelo relacional. Ed. Addison Wesley Iberoamericana, Ra-ma.</li> </ol> | <p>Internet</p> <p>Computadora</p> <p>Libros, apuntes</p> <p>Videotutoriales</p> |

Fecha de elaboración: 25 /08/2025

|                                                                                   |                                                                                         |                             |
|-----------------------------------------------------------------------------------|-----------------------------------------------------------------------------------------|-----------------------------|
|  | Instrumentación Didáctica para la Formación y Desarrollo de Competencias Profesionales. | Código: ITSLP-SAC-PG-001-02 |
|                                                                                   |                                                                                         | Revisión: 4                 |
|                                                                                   |                                                                                         | Página: 18 de 18            |

DUBELZA BEATRIZ OLIVA GARZA

**Nombre de la profesora**

MC. JUAN ANTONIO ALVARADO CANO

**Nombre de la Jefe de Departamento Académico**



CARTA DE RECONOCIMIENTO DEL AUTOR DE LOS DERECHOS A FAVOR DEL TECN

Ciudad de México, 25/08/2025

TECNOLÓGICO NACIONAL DE MÉXICO.  
PRESENTE

Bajo protesta de decir verdad, Dubelza Beatriz Oliva Garza personal docente adscrito al Instituto Tecnológico de San Luis Potosí del Tecnológico Nacional de México, manifiesto que en cumplimiento de mis actividades relacionadas con el Año Sabático elaboré la obra titulada "Elaboración de apuntes de la asignatura Fundamentos de Base de datos clave AEF-1031"

Con base en lo anterior, y con fundamento en los artículos 83 de la Ley Federal del Derecho de Autor y 46 de su Reglamento, reconozco que el Tecnológico Nacional de México es titular de los derechos patrimoniales sobre la misma y le corresponden las facultades relativas a la divulgación, integridad de la obra y de colección, conservando el derecho a figurar como autor.

Asimismo, respondo por la autoría y originalidad de la citada obra; y relevo de toda responsabilidad al Tecnológico Nacional de México de cualquier demanda o reclamación que llegara a formular alguna persona física o moral que considere que con esta obra es afectado en alguno de los derechos protegidos por la Ley en cita, asumiendo todas las consecuencias legales y económicas.

ATENTAMENTE

AUTOR

Dubelza Beatriz Oliva Garza



**2025**  
Año de  
**La Mujer  
Indígena**

# Dubelza Beatriz Oliva Garza

## Fundamentos de Base de Datos

### Reporte final FBD-10-144

 actividad-1

#### Detalles del documento

Identificador de la entrega

trn:oid::2945:304709022

Fecha de entrega

26 ago 2025, 9:04 GMT-6

Fecha de descarga

26 ago 2025, 9:14 GMT-6

Nombre del archivo

Reporte final FBD-10-144.docx

Tamaño del archivo

1001.2 KB

135 páginas

26.656 palabras

149.979 caracteres

  
Dra. Irma Luz Torres Vidales  
Subdirectora académica del ITSLP  
INSTITUTO TECNOLÓGICO  
DE SAN LUIS POTOSÍ  
SUBDIRECCIÓN  
ACADÉMICA

## 8% Similitud general

El total combinado de todas las coincidencias, incluidas las fuentes superpuestas, para ca...

### Filtrado desde el informe

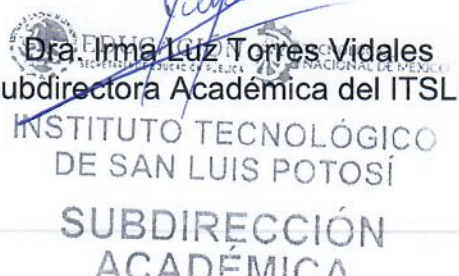
- Bibliografía
- Texto citado
- Coincidencias menores (menos de 15 palabras)

### Fuentes principales

- 8%  Fuentes de Internet
- 0%  Publicaciones
- 0%  Trabajos entregados (trabajos del estudiante)

### Marcas de integridad

N.º de alertas de integridad para revisión

  
  
Dra. Inma Luz Torres Vidales  
Subdirectora Académica del ITS LP  
INSTITUTO TECNOLÓGICO  
DE SAN LUIS POTOSÍ  
SUBDIRECCIÓN  
ACADÉMICA

Los algoritmos de nuestro sistema analizan un documento en profundidad para buscar inconsistencias que permitirían distinguirlo de una entrega normal. Si advertimos algo extraño, lo marcamos como una alerta para que pueda revisarlo.

Una marca de alerta no es necesariamente un indicador de problemas. Sin embargo, recomendamos que preste atención y la revise.