



UTPL

La Universidad Católica de Loja

Vicerrectorado de Modalidad Abierta y a Distancia

Fundamentos de Base de Datos

Guía didáctica





Facultad Ingenierías y Arquitectura

Fundamentos de Base de Datos

Guía didáctica

Carrera	PAO Nivel
Tecnologías de la Información	IV

Autor:

Juan Carlos Morocho Yunga



Universidad Técnica Particular de Loja

Fundamentos de Base de Datos

Guía didáctica

Juan Carlos Morocho Yunga

Diagramación y diseño digital

Ediloja Cía. Ltda.

Marcelino Champagnat s/n y París

edilocialtda@ediloja.com.ec

www.ediloja.com.ec

ISBN digital -978-9942-25-612-6

Año de edición: abril, 2020

Edición: primera edición reestructurada en febrero 2025 (con un cambio del 65%)

Loja-Ecuador



Los contenidos de este trabajo están sujetos a una licencia internacional Creative Commons **Reconocimiento-NoComercial-CompartirIgual** 4.0 (CC BY-NC-SA 4.0). Usted es libre de **Compartir** — copiar y redistribuir el material en cualquier medio o formato. Adaptar — remezclar, transformar y construir a partir del material citando la fuente, bajo los siguientes términos: Reconocimiento- debe dar crédito de manera adecuada, brindar un enlace a la licencia, e indicar si se han realizado cambios. Puede hacerlo en cualquier forma razonable, pero no de forma tal que sugiera que usted o su uso tienen el apoyo de la licenciante. No Comercial-no puede hacer uso del material con propósitos comerciales. Compartir igual-Si remezcla, transforma o crea a partir del material, debe distribuir su contribución bajo la misma licencia del original. No puede aplicar términos legales ni medidas tecnológicas que restrinjan legalmente a otras a hacer cualquier uso permitido por la licencia. <https://creativecommons.org/licenses/by-nc-sa/4.0/>



Índice

1. Datos de información	9
1.1 Presentación de la asignatura.....	9
1.2 Competencias genéricas de la UTPL.....	9
1.3 Competencias del perfil profesional	9
1.4 Problemática que aborda la asignatura	9
2. Metodología de aprendizaje	10
3. Orientaciones didácticas por resultados de aprendizaje.....	11
Primer bimestre	11
Resultados de aprendizaje 1 a 3:.....	11
Contenidos, recursos y actividades de aprendizaje recomendadas.....	11
Semana 1	12
Unidad 1. Introducción a los sistemas de bases de datos.....	12
1.1. Sistemas de gestión de base de datos (SGBD).	12
1.2. Evolución desde los sistemas basados en archivos.	14
1.3. Independencia y niveles de abstracción	15
1.4. Arquitecturas de los sistemas de gestión de bases de datos	17
Actividades de aprendizaje recomendadas	19
Autoevaluación 1	19
Resultado de aprendizaje 4:	23
Contenidos, recursos y actividades de aprendizaje recomendadas.....	23
Semana 2.....	23
Unidad 2. El modelo relacional	23
2.1. Estructura de las bases de datos relacionales.	24
2.2. Claves.	24
2.3. Esquemas de bases de datos relacionales y su representación.	25
2.4. Restricciones de integridad.	26
2.5. Operaciones relacionales.	26
2.6. Lenguajes de consulta relacional.	27



Actividades de aprendizaje recomendadas	28
Autoevaluación 2.....	29
Resultado de aprendizaje 5:	32
Contenidos, recursos y actividades de aprendizaje recomendadas.....	32
Semana 3	32
Unidad 3. Lenguaje SQL.....	32
3.1. Definición de datos en SQL	33
3.2. Consultas con SQL.....	35
Actividades de aprendizaje recomendadas	38
Contenidos, recursos y actividades de aprendizaje recomendadas.....	39
Semana 4	39
Unidad 3. Lenguaje SQL.....	39
3.3. Funciones de agregación	39
3.4. Subconsultas.....	40
3.5. Modificación de la base de datos	42
Actividades de aprendizaje recomendadas	44
Contenidos, recursos y actividades de aprendizaje recomendadas.....	44
Semana 5	44
Unidad 3. Lenguaje SQL.....	44
3.6. Combinación de tablas	45
3.7. Vistas	45
3.8. Transacciones	46
Actividad de aprendizaje recomendada	48
Contenidos, recursos y actividades de aprendizaje recomendadas.....	49
Semana 6	49
Unidad 3. Lenguaje SQL.....	49
3.9. Restricciones de integridad.....	49
3.10. Índices	51
3.11. Privilegios	53



Actividades de aprendizaje recomendadas 55

Autoevaluación 3..... 55

Resultados de aprendizaje 1 a 5:..... 59

Contenidos, recursos y actividades de aprendizaje recomendadas..... 59

Semana 7 60

 Actividades finales del bimestre 60

Contenidos, recursos y actividades de aprendizaje recomendadas..... 60

Semana 8 60

 Actividades finales del bimestre 60

Segundo bimestre..... 62

Resultado de aprendizaje 6: 62

Contenidos, recursos y actividades de aprendizaje recomendadas..... 62

Semana 9 62

 Unidad 4. Lenguajes de consulta formales 62

 4.1. Álgebra relacional 63

 4.2. Operaciones fundamentales 63

 Actividades de aprendizaje recomendadas 66

Contenidos, recursos y actividades de aprendizaje recomendadas..... 66

Semana 10 66

 Unidad 4. Lenguajes de consulta formales 66

 4.3. Cálculo relacional..... 66

 Actividades de aprendizaje recomendadas 68

 Autoevaluación 4..... 69

Resultado de aprendizaje 7: 72

Contenidos, recursos y actividades de aprendizaje recomendadas..... 72

Semana 11 72

 Unidad 5. Diseño conceptual de la base de datos 72

 5.1. Diseño de bases de datos 73

 5.2. Fases del diseño de bases de datos..... 73



Actividades de aprendizaje recomendadas	74
Contenidos, recursos y actividades de aprendizaje recomendadas.....	75
Semana 12.....	75
Unidad 5. Diseño conceptual de la base de datos	75
5.3. Diseño conceptual	75
5.4. Modelo E-R	75
5.5. Especialización-generalización	76
Actividades de aprendizaje recomendadas	77
Autoevaluación 5.....	77
Resultado de aprendizaje 8:	80
Contenidos, recursos y actividades de aprendizaje recomendadas.....	80
Semana 13.....	80
Unidad 6. Diseño lógico de bases de datos	80
6.1. Principios de diseño lógico de bases de datos.....	80
6.2. Técnica de normalización.....	81
6.3. Dominios atómicos y primera forma normal	81
Actividad de aprendizaje recomendada	82
Contenidos, recursos y actividades de aprendizaje recomendadas.....	83
Semana 14.....	83
Unidad 6. Diseño lógico de bases de datos	83
6.4. Dependencias funcionales	83
6.5. Segunda forma normal	83
6.6. Tercera forma normal	83
Actividades de aprendizaje recomendadas	84
Autoevaluación 6.....	85
Resultados de aprendizaje 6 a 8:.....	89
Contenidos, recursos y actividades de aprendizaje recomendadas.....	89
Semana 15.....	89
Actividades finales del bimestre	89



Contenidos, recursos y actividades de aprendizaje recomendadas..... 90

Semana 16..... 90

 Actividades finales del bimestre 90

4. Autoevaluaciones 91

5. Referencias bibliográficas 99





1. Datos de información

1.1 Presentación de la asignatura



1.2 Competencias genéricas de la UTPL

Organización y planificación del tiempo.

1.3 Competencias del perfil profesional

Construir modelos específicos de ciencias de la computación mediante esquemas matemáticos y estadísticos, para propiciar el uso y explotación eficiente de datos e información.

1.4 Problemática que aborda la asignatura

La asignatura “Fundamentos de bases de datos” cubre la parte del diseño e implementación de bases de datos, un componente fundamental para el correcto funcionamiento de sistemas de información computacionales.



2. Metodología de aprendizaje

En el estudio de esta asignatura, se ha optado por implementar la metodología de [aprendizaje basado en casos](#). Este enfoque pedagógico se seleccionó estratégicamente debido a su eficacia en la integración de conocimientos teóricos y prácticos a través de situaciones contextualizadas. Al presentar casos prácticos, los estudiantes se enfrentarán a escenarios reales que les permitirán aplicar los conceptos aprendidos en un entorno más cercano a los desafíos que podrían encontrar en su futura trayectoria profesional.

La adopción del aprendizaje basado en casos fomentará el pensamiento crítico, la resolución de problemas y la toma de decisiones informadas. Los estudiantes serán guiados a través de análisis detallados de casos, lo que no solo fortalecerá su comprensión de los conceptos fundamentales de la asignatura, sino que también cultivará habilidades prácticas esenciales. Esta metodología se alinea con la filosofía educativa de proporcionar a los estudiantes experiencias de aprendizaje significativas y aplicables, preparándolos para enfrentar los desafíos del mundo real en su campo de estudio.





3. Orientaciones didácticas por resultados de aprendizaje



Primer bimestre

Resultados de aprendizaje 1 a 3:

- Diferencia y utiliza los términos clave tales como: información, datos, base de datos, sistema de gestión de base de datos, metadatos, y minería de datos.
- Explica cómo el almacenamiento y recuperación de datos ha cambiado con el tiempo.
- Explica las ventajas de un enfoque de base de datos en comparación con el procesamiento de archivos tradicional.

Los resultados de aprendizaje se enfocarán en el reconocimiento de la importancia de la información como un recurso estratégico tanto para organizaciones como para individuos. Exploraremos específicamente la composición de un sistema de base de datos y su integración con los sistemas de información. La relevancia de este tema radica en que, en la actualidad, las bases de datos son omnipresentes en la vida cotidiana y desempeñan un papel esencial en los entornos informáticos modernos.

Para facilitar la comprensión, se proporcionarán lecturas comprensivas que serán guiadas por la estructura de la guía didáctica. Además, se propondrán actividades recomendadas para reforzar y aplicar los conceptos aprendidos.

Contenidos, recursos y actividades de aprendizaje recomendadas

Recuerde revisar de manera paralela los contenidos con las actividades de aprendizaje recomendadas y actividades de aprendizaje evaluadas.





Semana 1

Unidad 1. Introducción a los sistemas de bases de datos

Estimado estudiante bienvenido a la primera semana de estudios. Empezaremos nuestro trabajo identificando algunos hechos y conceptos relacionados con las bases de datos. Es necesario destacar que, en la actualidad, las bases de datos son parte de la vida cotidiana, y de igual manera forman parte esencial de los entornos informáticos modernos cuya aplicabilidad está presente en casi todas las actividades diarias.

1.1. Sistemas de gestión de base de datos (SGBD).

Los Sistemas de Gestión de Bases de Datos (SGBD) son programas diseñados para almacenar, organizar y gestionar datos estructurados bajo el modelo relacional, propuesto por E.F. Codd en 1970. El objetivo principal de un SGBD es proporcionar un entorno a los usuarios de la base de datos para las funciones de almacenamiento y recuperación de información.

Las funciones que cumple un SGBD se podrían resumir en las siguientes:

- Permitir a los usuarios, mediante un lenguaje de definición de datos (DDL, Data Definition Language), especificar las estructuras y tipos de datos y las restricciones aplicables a los datos que hay que almacenar en la base de datos.
- Permitir a los usuarios insertar, actualizar, borrar y extraer datos de la base de datos, usualmente mediante un lenguaje de manipulación de datos (DML, Data Manipulation Language).
- Controlar el acceso a los datos restringiendo el acceso a usuarios no autorizados haciendo más seguro el entorno donde se guardan los datos. Ayuda a mantener la coherencia de los datos y a gestionar el acceso concurrente de usuarios para evitar conflictos. En caso de fallos de hardware o software permite la recuperación de la base de datos y su restauración a un estado coherente. Permite que haya un catálogo accesible de metadatos que describen los datos almacenados.



Tanto DDL como DML son parte de un lenguaje que es estándar formal para los SGBD relacionales denominado SQL (Structured Query Language, lenguaje de consulta estructurado).

Un SGBD organiza la información en tablas (relaciones) compuestas por filas (tuplas) y columnas (atributos), que se relacionan entre sí mediante claves (primarias y foráneas) para evitar redundancia y garantizar integridad de los datos.

El SGBD es un intermediario entre el usuario y la propia base de datos, pues presenta algunas interfaces que los usuarios pueden utilizar:

- Interfaz basada en menú para clientes Web.
- Aplicación para dispositivos móviles.
- Interfaz basada en formularios.
- Interfaz gráfica de usuario.
- Búsqueda en bases de datos basadas en palabras clave.
- Interfaces parametrizadas para usuarios.
- Interfaces para DBA.

Algunos de los SGBD comerciales son MySQL^[1], PostgreSQL^[2], Oracle^[3], SQL Server entre otros. Se usan en sectores como finanzas, salud o redes sociales, donde la integridad y el acceso rápido a datos estructurados son críticos.

Aunque hoy coexisten con modelos NoSQL^[4] (para datos no estructurados), los SGBD relacionales siguen siendo la base de sistemas transaccionales o sistemas de información en muchas empresas gracias a su robustez y estandarización.



Revise el siguiente texto de Pulido Romero et al. (2019) [Base de datos](#), apartado 1.3, pp. 22 – 31, el cual le permitirá comprender mejor el concepto y funcionalidad de un Sistema de Gestión de Base de Datos SGBD.



1.2. Evolución desde los sistemas basados en archivos.

Antes de la aparición de las bases de datos relacionales, los sistemas basados en archivos eran la principal forma de almacenar y gestionar datos. Estos sistemas, de alguna manera, representaban la estructura de la organización debido a que cada departamento manejaba sus propios archivos de datos y los programas que gestionaban los archivos de datos. Además, cubrían las necesidades de manejo de información antes del apareamiento de las bases de datos relacionales, por lo que tenían gran acogida en las organizaciones.

Los sistemas basados en archivos almacenan los datos en archivos individuales. Cada archivo contiene datos relacionados y se organiza de manera que pueda ser leído y actualizado por programas específicos. Estos sistemas funcionaban en las décadas de 1960 y 1970.

Con la creciente cantidad de datos generados por las organizaciones, los sistemas basados en archivos comenzaron a presentar algunos problemas que se los enumera a continuación:

- **Redundancia e inconsistencia de datos:** La misma información se almacena en múltiples archivos y provoca el desperdicio del espacio de almacenamiento, pero también puede llevar a inconsistencias cuando los datos se actualizan en un archivo, pero no en otros donde aparece el mismo dato. Por ejemplo, la información de un cliente puede aparecer en el departamento de marketing y en el de facturación, si se llega a actualizar datos del cliente en el departamento de marketing, lo mismo debería pasar en el departamento de facturación para evitar el riesgo de inconsistencias.
- **Dificultad de acceso a los datos:** En los sistemas basados en archivos, acceder a los datos puede ser complicado y lento, especialmente si los archivos son grandes y no están indexados. Además, cada programa que accede a los datos maneja una estructura propia de archivo con lo que es complicado compartir la información.
- **Aislamiento de los datos:** Como en cada departamento posee sus propias aplicaciones que gestionan sus propios archivos, lo que genera duplicidad de información y dificulta compartir datos entre sistemas y mantener



coherencia. Por ejemplo, los datos de un cliente si se almacenan en archivos separados para ventas y facturación estarán aislados y generan duplicidad.

- **Problemas de integridad:** Mantener la integridad de los datos en los sistemas basados en archivos requiere mucho esfuerzo. No hay mecanismos para garantizar que las relaciones entre los datos se mantengan consistentes.
- **Problemas de atomicidad:** Las operaciones no se ejecutan como una unidad indivisible (todo o nada), como sucede en una transacción de base de datos, por lo que los datos pueden quedar en estado inconsistente o corruptos.
- **Anomalías en el acceso concurrente:** En ocasiones en las que múltiples usuarios modifican un mismo archivo simultáneamente, no hay mecanismos para coordinar los cambios, lo que puede generar sobrescritura de datos o resultados inconsistentes.
- **Problemas de seguridad:** La seguridad de los datos es limitada y no hay mecanismos integrados para controlar el acceso a los datos, lo que puede llevar a accesos no autorizados y a la manipulación de la información.



Revise los siguientes textos, los cuales le permitirán conocer qué había antes de las bases de datos y cómo ha evolucionado la tecnología en este campo:



- Pulido Romero et al. (2019) [Base de datos](#), apartado 1.1, pp. 2 - 17
- Martínez López & Gallegos Ruiz (2017) [Programación de bases de datos relacionales](#), apartado 1.1, pp. 16 - 19

1.3. Independencia y niveles de abstracción

Los sistemas de bases de datos buscan ofrecer una visión abstracta de los datos, ocultando detalles técnicos de almacenamiento y manipulación. Para esto, se utiliza la arquitectura ANSI-SPARC, que define tres niveles de abstracción:

Tabla 1
Niveles de abstracción

Nivel		Características
Nivel de Vistas (Esquema Externo)		<ul style="list-style-type: none">• Representa la perspectiva específica de cada usuario o aplicación.• Describe solo la parte de la base de datos relevante para un grupo de usuarios.
Nivel Lógico (Esquema Conceptual)		<ul style="list-style-type: none">• Define qué datos se almacenan y sus relaciones (entidades, atributos, claves).• Es independiente de las aplicaciones y del almacenamiento físico.
Nivel Físico (Esquema Interno)		<ul style="list-style-type: none">• Describe cómo se almacenan los datos en los dispositivos de almacenamiento (estructuras de archivos, índices, particiones).• Incluye detalles técnicos como formatos de almacenamiento o métodos de acceso.

Nota. Morocho, J., 2023.

Este enfoque asegura que los usuarios interactúen con una capa simplificada, mientras el SGBD gestionan la complejidad técnica de manera transparente.

Es importante mencionar también que los cambios en el nivel físico no deben afectar al nivel lógico ni a las vistas de usuario, garantizando flexibilidad y reduciendo costos de mantenimiento, esto se conoce como independencia física. Por ejemplo, el migrar datos a un nuevo tipo de disco SSD no debería afectar al nivel lógico ni a las vistas de usuario.





Revise el siguiente texto de Pulido Romero et al. (2019) [Base de datos](#), apartado 1.3.1, pp. 22 – 23 el cual le permitirá comprender con más detalle las características de los niveles de abstracción de la arquitectura ANSI-SPARC.

1.4. Arquitecturas de los sistemas de gestión de bases de datos

Los sistemas de gestión de bases de datos se implementan bajo tres arquitecturas principales, que han ido evolucionando según las necesidades de implementación y de procesamiento:

1. Arquitectura centralizada

Se basa en una computadora central de gran capacidad que gestiona todo el procesamiento y varios terminales^[5] envían solicitudes a la computadora central, que devuelve resultados tras procesarlas.

Características principales:

- Centraliza recursos y datos en un único servidor.
- Ideal para entornos con infraestructura limitada o sistemas heredados.

2. Arquitectura cliente/servidor en dos capas

En esta configuración tenemos:

- El Cliente gestiona la interfaz de usuario y envía solicitudes al servidor.
- El **Servidor** que procesa consultas SQL, valida y almacena datos.

En esta arquitectura las solicitudes de gestión de datos se envían del cliente al servidor y los resultados se transmiten del servidor al cliente, con lo que claramente se identifican dos capas.

3. Arquitectura cliente/servidor en tres capas



En esta configuración tenemos tres capas:

- **Cliente:** Que es la interfaz de usuario y puede ser un navegador o aplicación.
- **Servidor de aplicaciones:** Se encarga de ejecutar la lógica de negocio y procesamiento de datos.
- **Servidor de base de datos:** Se encarga de gestionar el acceso a los datos.

Este modelo de arquitectura ayuda a reducir la carga en clientes, que actualmente pueden ser dispositivos ligeros además, permite optimizar y escalar recursos en cada capa porque si se requiere mayor capacidad de almacenamiento se puede potenciar el servidor de base de datos para



Revise el siguiente texto de Pulido Romero et al. (2019) [Base de datos](#), apartado 1.3.1, pp. 22 – 23 el cual le permitirá comprender con más detalle las características de los niveles de abstracción de la arquitectura ANSI-SPARC.

En esta semana estudiaremos los SGBD, los sistemas de procesamiento de archivos, los niveles de independencia y niveles de abstracción en un SGBD, las diferentes arquitecturas del SGBD. También se complementa con una introducción a los lenguajes e interfaces de bases de datos para tener un contexto completo en el entorno de bases de datos.

Para una mayor comprensión de la unidad, a continuación, se presentan lecturas comprensivas y se proponen asimismo actividades recomendadas.

Recursos de aprendizaje

- Zorrilla Pantaleón y Duque Medina (2011). [Tema 01. Introducción a las BD Relacionales](#). OCW Universidad de Cantabria: Bases de Datos (p. 1-14).

Este material recomendado le permitirá conocer los detalles sobre la importancia de las bases de datos, algunos conceptos fundamentales y qué había antes de las bases de datos.



- Morocho-Y. & Romero-Peláez (2019). [Guía didáctica de Fundamentos de Bases de Datos](#). Loja, Ecuador: Editorial Universidad Técnica Particular de Loja.

En esta guía didáctica encontrará explicaciones ampliadas a las temáticas revisadas en la primera unidad.



Actividades de aprendizaje recomendadas

Es momento de aplicar sus conocimientos a través de las actividades que se han planteado a continuación:

1. Elabore un esquema gráfico que resuma las limitaciones de los sistemas de procesamiento de archivos, como resultado de un análisis de las circunstancias que las generan.
2. Identifique la diferencia entre los niveles de abstracción de la arquitectura ANSI-SPARC y diseñe un mapa mental. Si desea utilizar herramientas digitales en línea le sugerimos utilice Padlet^[6], puede compartir sus muros en el EVA.
3. Identifique las principales funciones de DML y DDL. Si desea utilizar herramientas digitales en línea le sugerimos utilice Padlet^[7], puede compartir sus muros en el EVA.
4. Revise el ejemplo de [pañales y cervezas](#), que es un ejemplo simpático y comprensible sobre posibles aplicaciones de minería de datos.
5. Una vez completada la presente unidad, le invitamos a desarrollar la siguiente autoevaluación que le permite verificar su aprendizaje sobre la primera unidad "Introducción a las bases de datos".



Autoevaluación 1

Seleccione la opción correcta.

1. Los sistemas basados en archivos mantienen los datos en varios archivos, generalmente en distintos formatos, y requieren igualmente,



varios programas de aplicación para manipular sus datos. Esta desventaja se conoce como

- a. Anomalías en el acceso concurrente.
- b. Dificultad en el acceso a los datos.
- c. Aislamiento de datos.

2. Una base de datos es el:

- a. *Software* que gestiona y controla el acceso a los datos.
- b. Conjunto de datos que contiene información relativa a una organización.
- c. Programa escrito por desarrolladores en respuesta a las necesidades de los usuarios.

3. El administrador de base de datos debe seleccionar los datos que se almacenan en la base de datos empleando el nivel de abstracción:

- a. Físico.
- b. Lógico.
- c. De vistas.

4. El lenguaje de definición de datos sirve para:

- a. Especificar el esquema de la base de datos.
- b. Expresar las consultas y modificaciones a la base de datos.
- c. Especificar los privilegios sobre elementos de la base de datos.

5. ¿Cuál de las siguientes sentencias no corresponde a características de una base de datos relacional?

- a. Emplea el lenguaje de manipulación de datos y de definición de datos.
- b. Utiliza un conjunto de tablas para representar datos y sus relaciones.
- c. Describe el diseño de la base de datos en el nivel físico.



6. Identifique el elemento que no corresponde a una aplicación de base de datos con una arquitectura de tres capas.
- a. Máquina cliente.
 - b. Servidor de aplicaciones.
 - c. Lógica de negocio.
 - d. Servidor de base de datos.
7. Minería de datos hace referencia a:
- a. El proceso de análisis semiautomático de grandes bases de datos para descubrir patrones útiles.
 - b. La creación de formularios e informes con un mínimo de esfuerzo de programación.
 - c. El método para añadir información de marcas a los documentos de texto para el intercambio de datos.
8. ¿Cuál de los siguientes enunciados es una actividad rutinaria de un DBA?
- a. Escribir aplicaciones de base de datos especializadas que no encajan en el marco tradicional del procesamiento de datos.
 - b. Asegurar disponibilidad del espacio en disco suficiente para la normal operación de la base de datos.
 - c. Crear el esquema original de la base de datos mediante la ejecución de comandos de definición de datos de SQL.
9. Los sistemas basados en archivos generaban problemas que fueron solucionados con la aparición de las bases de datos. Señale dos de esos problemas:
- a. Redundancia de datos.
 - b. Problemas de lenguajes de programación.
 - c. Problemas de seguridad.
 - d. Problemas de usuario final (errores humanos).



10. El esquema conceptual de una base de datos completamente desarrollado permite:

- a. Identificar los requisitos de datos de la organización.
- b. Especificar las características físicas de la base de datos.
- c. Indicar los requisitos funcionales de la empresa.

Ir al solucionario

[1] Página oficial <https://www.mysql.com/>

[2] Página oficial <https://www.postgresql.org/>

[3] Página oficial <https://www.oracle.com/>

[4] Qué es una base de datos NoSQL <https://www.ibm.com/es-es/topics/nosql-databases>

[5] Estaciones de trabajo sin capacidad de procesamiento

[6] Página oficial <https://padlet.com/>

[7] Página oficial <https://padlet.com/>



Resultado de aprendizaje 4:

Define la calidad, precisión y puntualidad, explica cómo su ausencia tendrá un impacto en las organizaciones.

En el presente resultado de aprendizaje, nos adentraremos en un tema crucial dentro del ámbito de las bases de datos: el modelo relacional. Este modelo es el fundamento de las bases de datos relacionales y su entendimiento y comprensión son esenciales en su formación como futuro ingeniero. Uno de los principales objetivos de la asignatura es equipar al estudiante con la capacidad de aplicar de manera efectiva el modelo relacional.

Para facilitar la comprensión, se proporcionarán lecturas comprensivas que serán guiadas por la estructura de la guía didáctica. Además, se propondrán actividades recomendadas para reforzar y aplicar los conceptos aprendidos.

Contenidos, recursos y actividades de aprendizaje recomendadas

Recuerde revisar de manera paralela los contenidos con las actividades de aprendizaje recomendadas y actividades de aprendizaje evaluadas.



Semana 2

Unidad 2. El modelo relacional

Estimado estudiante, en este momento se inicia la revisión de uno de los temas de vital importancia dentro de las bases de datos, debido a que sirve como fundamento teórico de la gestión y manipulación de datos en una base de datos. Uno de los objetivos principales de la asignatura es desarrollar en el estudiante la habilidad de aplicar el modelo relacional.



El **modelo de datos** es un conjunto de herramientas conceptuales para la descripción de los datos, las relaciones entre estos, su semántica y las restricciones de consistencia. El modelo de datos para bases de datos relacionales más ampliamente utilizado es el *modelo relacional*, que –por su simplicidad– facilita su comprensión.

Los temas a los que debe prestarle más atención son:

2.1. Estructura de las bases de datos relacionales.

Las bases de datos relacionales organizan los datos en relaciones que físicamente se representan como tablas. Cada tabla está formada por un conjunto de filas o tuplas y varias columnas o propiedades que describen a la relación y cada columna posee un dominio desde donde toma los posibles valores para dicha columna. Además, se puede distinguir claves primarias que permiten identificar de forma única cada tupla de la relación y claves foráneas que a través de un atributo común entre dos relaciones permiten crear un vínculo entre ellas a más de asegurar la integridad referencial. La cardinalidad está definida por el número de tuplas de la relación en un instante determinado y el grado de la relación lo define el número de atributos de la relación.

2.2. Claves.

Son atributos que se definen para garantizar unicidad, las cuales se muestran en la siguiente figura:



Figura 1

Claves

CLAVE PRIMARIA (PK)

Es un atributo único (no se repite) que identifica cada fila de la relación de forma única.



CLAVE FORÁNEA (FK)

Sirve para vincular dos tablas, referenciando una PK en otra tabla que generalmente se la conoce como tabla Padre.



CLAVE CANDIDATA

Son atributos que podrían ser elegidos para hacer de clave primaria PK porque identifican de forma única a cada tupla de la relación.



Nota. Morocho, J., 2023.

Revise los siguientes textos, los cuales le aportarán una visión más amplia de las generalidades del modelo relacional y los tipos de claves:

- Jiménez Capel (2024) [Bases de datos relacionales y modelado de datos](#) capítulo 2 apartados 1, 2 y 3, pp. 55 - 62.
- Valderrey Sanz (2015) [Gestión de bases de datos](#), apartado 2.2, pp. 37 – 38.

2.3. Esquemas de bases de datos relacionales y su representación.

Un esquema de base de datos es el diseño lógico de la base de datos en el cual se encuentran definidos varios elementos como tablas, atributos, tipos de datos y relaciones entre tablas. Generalmente, se representa mediante un Diagrama Entidad-Relación (ER) en el que gráficamente se representan las entidades y sus relaciones, de ahí su nombre.

Un ejemplo de un esquema de base de datos sería el siguiente:

Carreras (IdCarrera [PK], NombreC, Area)



Estudiantes (IdMatricula [PK], NombreE, IdCarrera[FK])

Calificaciones (IdCalificacion [PK], IdMatricula [FK], Nota)

2.4. Restricciones de integridad.

Dentro del modelo relacional existen algunas reglas para mantener la coherencia y precisión de los datos. A continuación, mencionamos algunas:

- Integridad de Entidad: Se define mediante una PK que por definición no puede ser nula ni se puede repetir.
- Integridad Referencial: Se refuerza mediante una FK que siempre debe referenciar a una PK en otra tabla. Por ejemplo, no se debería registrar una calificación para una matrícula que no existe.
- Restricciones de Dominio: Los posibles valores que tome la columna deben cumplir con el tipo de dato definido. Por ejemplo, un atributo fecha nacimiento no puede ser texto.

2.5. Operaciones relacionales.

Son operaciones básicas para manipular datos en el modelo relacional. Reciben como entrada relaciones y producen como resultado otra relación. Algunas de esas operaciones son las siguientes:

1. Selección ($\sigma_{\text{predicado}}(R)$): Esta operación permite filtrar filas según una condición o predicado.
2. Proyección ($\pi_{\text{col1}, \text{col2}, \dots, \text{coln}}(R)$): Esta operación selecciona columnas específicas que se desea recuperar. Pero se debe tener claro que recupera todas las tuplas de la tabla sin ninguna restricción.
3. Unión ($R \cup S$): Combinar filas de dos tablas similares siempre y cuando tengan el mismo número de atributos y los atributos se correspondan en tipo de dato.
4. Diferencia ($R - S$): Mediante esta operación se obtienen las filas de una tabla que no están en otra.



5. Producto Cartesiano ($R \times S$): Produce un producto entre todas las filas de dos tablas, es una especie de multiplicación.
6. Combinación ($R \bowtie S$): Mediante esta operación se puede unir tablas mediante una condición que se basa en columnas con valores similares de clave foránea.



Para una mejor comprensión de las operaciones relacionales que se ejecutan entre entidades de un modelo de base de datos revise el siguiente material de apoyo: [Bases de datos relacionales y modelado de datos](#), capítulo 2 apartados 1, 2 y 3, pp. 55 - 62

2.6. Lenguajes de consulta relacional.

Con el aparecimiento de las bases de datos relacionales apareció también un lenguaje para administrar los datos contenidos en la base de datos. El lenguaje en sus inicios se llamaba SEQUEL^[8] y fue desarrollado por IBM^[9] y luego pasó a llamarse SQL. Está compuesto por tres sublenguajes:

- DDL o Data Definition Language, permite definir las estructuras de almacenamiento y demás objetos de la base de datos.
- DML o Data Manipulation Language, facilita la consulta (SELECT), inserción (INSERT), modificación (UPDATE) y borrado (DELETE) de datos de las tablas de la base de datos.
- DCL o Data Control Language, mediante el cual se otorgan permisos y autorizaciones para controlar el acceso a los datos de la base de datos.

Estudiaremos más a fondo estas operaciones en la unidad 3 Lenguaje SQL y la unidad 4 Lenguajes de consulta formales.

Para una mayor comprensión de la unidad, a continuación, se presentan lecturas comprensivas y se proponen asimismo actividades recomendadas.



Recursos de aprendizaje

- Zorrilla Pantaleón, M. E., & Duque Medina, R. (2011). [Tema 1. Introducción a las BD Relacionales](#). OCW Universidad de Cantabria: Bases de Datos (p. 14-18).

Este material recomendado le permitirá conocer los conceptos fundamentales que se manejan en el modelo relacional.

- Morocho, J. y Romero, A. (2019). [Guía didáctica de Fundamentos de Bases de Datos](#). Loja, Ecuador: Editorial Universidad Técnica Particular de Loja.

En esta guía didáctica encontrará explicaciones ampliadas a las temáticas revisadas en la segunda unidad.



Actividades de aprendizaje recomendadas

Es hora de reforzar los conocimientos adquiridos resolviendo las siguientes actividades:

1. Identifique la diferencia entre DML procedimentales y DML declarativos (DML no procedimentales).
2. Basándonos en el siguiente esquema de Booking, identifique en cada relación, si es que hubiese: superclaves, claves candidatas, claves primarias y claves externas.

Hotel (hotel no, hotel name, city).

Room (room no, hotel no, type, price).

Guest (guest no, guest name, guest address).

Booking (hotel no, guest no, date from, date to, room no).



3. Una vez completada la presente unidad, le invito a desarrollar la siguiente autoevaluación que le permitirá verificar su aprendizaje en la unidad “Modelo relacional”.



Autoevaluación 2

Seleccione la opción correcta.

1. Una base de datos relacional se compone de un conjunto de:
 - a. Tablas que tienen nombres únicos.
 - b. Tuplas organizadas mediante una clave principal.
 - c. Columnas con nombres únicos.
2. Para referirse a una instancia específica de una relación se utiliza el término:
 - a. Esquema de relación.
 - b. Dominio atómico.
 - c. Ejemplar de relación.
3. Dentro del modelo relacional, una tupla se refiere a:
 - a. Una fila de la tabla.
 - b. La columna de una tabla.
 - c. Una instancia de una relación.
4. Dentro del modelo relacional, una relación se refiere a una:
 - a. Fila.
 - b. Tabla.
 - c. Columna.
5. Dentro del modelo relacional, un atributo se refiere a:
 - a. La columna de una tabla.
 - b. Una instancia de una relación.



c. Un ejemplar de una relación.

6. Las claves (sean superclaves, candidatas o primarias) son propiedades de:

- a. Toda la relación.
- b. Cada una de las tuplas.
- c. El esquema de la base de datos.

7. Una clave candidata es:

- a. La elegida por el diseñador de base de datos para identificar de forma única las tuplas de una relación.
- b. Aquella que podría identificar de forma única a cada dupla de una relación.
- c. Un conjunto de uno o varios atributos que considerados en conjunto identifican de forma única a cada dupla de una relación.

8. La operación relacional más frecuente es:

- a. Proyección.
- b. Selección.
- c. Producto cartesiano.

9. Cuando se aplica una consulta sobre una única relación y donde las tuplas tienen que cumplir un predicado particular (por ejemplo, `fecha_ingreso > "10/06/2015"`), el resultado es:

- a. Una vista que contiene las tuplas que satisfacen el predicado indicado.
- b. La misma relación original, pero que contiene solamente las tuplas que satisfacen el predicado.
- c. Una nueva relación, que es un subconjunto de la relación original.



10. Ciertos lenguajes relacionales cumplen estrictamente con la definición matemática de conjunto y:

- a. Eliminan los duplicados.
- b. Mantienen los duplicados.
- c. Eliminan o mantienen los duplicados de acuerdo con los recursos del servidor de base de datos.

[Ir al solucionario](#)

[8] Información ampliada en <https://es.wikipedia.org/wiki/SQL>

[9] Página oficial <https://www.ibm.com/mx-es/about>



Resultado de aprendizaje 5:

Describe los mecanismos de recolección de datos y sus implicaciones (recopilación de datos automatizada, formularios de entrada, fuentes).

En el presente resultado de aprendizaje, nos enfocaremos en el uso del Lenguaje de Consulta Estructurado (SQL), con énfasis en la capacidad para actualizar información en la base de datos y extraer datos mediante consultas tanto simples como complejas.

Asimismo, usted adquirirá habilidades fundamentales en el uso del SQL para manipular y definir datos, así como para realizar consultas que le permitan acceder y extraer información de manera efectiva.

Contenidos, recursos y actividades de aprendizaje recomendadas

Recuerde revisar de manera paralela los contenidos con las actividades de aprendizaje recomendadas y actividades de aprendizaje evaluadas.



Semana 3

Unidad 3. Lenguaje SQL

Estimado estudiante en la presente unidad vamos a revisar el Lenguaje de Consulta Estructurado (SQL^[10]). Centraremos la atención en la forma de construir sentencias SQL para *Manipulación de Datos* (DML) y para *Definición de Datos* (DDL), así como también los comandos para controlar el acceso a los datos.

El dominio del lenguaje SQL es una competencia importante en la gestión de base de datos y constituye una de las competencias más requeridas en todo perfil del profesional informático. SQL es la principal herramienta para la interacción del usuario con una base de datos, mediante la cual se envían solicitudes de datos al servidor que es el encargado de responder a dichas peticiones. SQL es un lenguaje relativamente fácil de aprender.





El lenguaje SQL, es considerado un estándar de la industria para interactuar con bases de datos relacionales. SQL está presente en todos los productos de base de datos comerciales y permite realizar operaciones como la creación de estructuras de datos u otros objetos de base de datos, así como consultas y manipulación de los datos almacenados.

SQL fue desarrollado en la década de 1970 por IBM, basado en el modelo relacional propuesto por Edgar F. Codd^[11]. Desde entonces, ha evolucionado y se ha convertido en el lenguaje de referencia para la gestión de bases de datos.

La importancia de SQL radica en que proporciona una forma eficiente y estandarizada de gestionar datos estructurados. Es utilizado en una amplia gama de aplicaciones, desde sistemas empresariales hasta aplicaciones web y móviles.

3.1. Definición de datos en SQL

Esta parte del lenguaje SQL permite crear las estructuras que almacenarán los datos y otros objetos de base de datos que son necesarios para su buen funcionamiento.

La principal estructura son las tablas, que están compuestas por columnas o atributos que tienen un tipo de dato y longitud (también llamado “dominio”). Los tipos de datos pueden variar de acuerdo con el producto de base de datos utilizado, pero contamos con unos tipos de datos básicos que ya debe haber estudiado en las asignaturas de programación.

El comando `CREATE` es muy usado para crear los objetos de base de datos que se requieren para implementar de forma física el diseño de base de datos creado. La sentencia `CREATE` va acompañada del tipo de objeto que se desea crear y pueden ser `CREATE TABLE` para crear tablas, `CREATE INDEX` para crear índices, y `CREATE VIEW` para crear vistas, entre otras.



Un ejemplo de creación de una tabla se muestra a continuación:

```
CREATE TABLE Empleados (  
    EmpleadoID NUMBER PRIMARY KEY,  
    Nombre VARCHAR2(100),  
    Apellido VARCHAR2(100),  
    FechaContratacion DATE,  
    Salario NUMBER (10, 2)  
);
```

En el ejemplo se crea la tabla Empleados conformada por las columnas EmpleadoID, Nombre, Apellido, FechaContratación y Salario, en donde el atributo EmpleadoID es la clave primaria en la que el propio SGBD asigna un nombre al índice pero también se puede crear a nivel de tabla como se muestra a continuación:

```
CREATE TABLE Empleados (  
    EmpleadoID NUMBER,  
    Nombre VARCHAR2(100),  
    Apellido VARCHAR2(100),  
    FechaContratacion DATE,  
    Salario NUMBER (10, 2),  
  
    CONSTRAINT empleados_pk PRIMARY KEY (EmpleadoID)  
);
```



Una vez creadas las estructuras de almacenamiento, suele requerirse también borrar estos elementos para lo cual podemos utilizar la sentencia `DROP` acompañada del tipo de objeto que se desea eliminar y pueden ser `DROP TABLE` para eliminar tablas, `DROP INDEX` para eliminar índices, y `DROP VIEW` para eliminar vistas, entre otras.

Por ejemplo, si se desea eliminar la tabla Empleados podemos utilizar la siguiente sentencia:

```
DROP TABLE Empleados;
```



Revise el siguiente texto de Martínez López & Gallegos Ruiz (2017) [Programación de bases de datos relacionales](#), capítulo 7 apartado 7.1, pp. 165 – 172, este le permitirá complementar su estudio de las sentencias DDL del lenguaje SQL.

3.2. Consultas con SQL

Debido a que una base de datos está diseñada para almacenar datos, es necesario que esos datos sean gestionados de alguna manera. Para acceder a esos datos se utilizan comandos DML (*Data Manipulation Language*) para realizar estas operaciones.

Selección de Datos: El comando `SELECT` se utiliza para recuperar datos de una o más tablas. Luego de la palabra reservada `SELECT` se debe especificar las columnas que se requieren separadas por comas, luego la palabra reservada `FROM` para especificar la o las tablas de donde se consultan los datos y con la palabra reservada `WHERE` aplicar filtros para obtener resultados específicos.

Por ejemplo, la siguiente consulta muestra los nombres y apellidos de los empleados cuyo salario es mayor a \$2000.

```
SELECT Nombre, Apellido
```

```
FROM Empleados
```



```
WHERE Salario > 2000;
```

Cabe mencionar que estas consultas son simples porque se ejecutan sobre una sola tabla, pero generalmente la información está distribuida en varias tablas, por lo que más adelante veremos consultas multitabla.

Ordenación de Datos: El comando `ORDER BY` se utiliza para ordenar los resultados de una consulta en orden ascendente o descendente. Por defecto el orden es ascendente y no hace falta especificar pero si se desea ordenar de forma descendente se agrega la palabra reservada `DESC`.

A continuación, un ejemplo de consulta en la que se selecciona los nombres y salarios de los empleados ordenados de mayor a menor por el salario.

```
SELECT Nombre, Salario
```

```
FROM Empleados
```

```
ORDER BY Salario DESC;
```

Agrupamiento de Datos: El comando `GROUP BY` se utiliza para agrupar filas que tienen valores iguales en columnas especificadas y permite realizar funciones de agregación en cada grupo, para obtener por ejemplo, subtotales por cada grupo formado. Para mejorar el resultado se puede aplicar alias de columna que coloca un título a la columna en donde se emplea la función de agregación.

En la consulta de ejemplo se hace un conteo del número de empleados por cada departamento.

```
SELECT Departamento, COUNT (*) total_emp
```

```
FROM Empleados
```

```
GROUP BY Departamento;
```



Otros apartados importantes que debe revisar la presente semana son operaciones básicas adicionales y los valores nulos.



Revise el siguiente texto de Nieto Bernal & Capacho Portilla (2017) [Diseño de base de datos](#), capítulo 4 apartado 4.5, pp. 204 - 206 para que conozca cómo recuperar información de la base de datos mediante consultas SQL.

Para facilitar una comprensión integral de la unidad, a continuación, se presentan lecturas comprensivas. Asimismo, se recomienda revisar los videos proporcionados para complementar los conceptos presentados.

Recursos de aprendizaje

- Considere el video colocado en el entorno virtual de aprendizaje sobre instalación de Oracle XE y Oracle SQLDeveloper. Para ello, revise el video: [Instalar OracleXE 11g en Windows](#) de (Morocho-Y., 2012).

Este video muestra cómo instalar las herramientas de base de datos y la interfaz gráfica para utilizar lenguaje SQL en el sistema operativo *Windows*. Si después de haber visto el video, aún tiene dudas o si se le presentó algún problema en el momento de la instalación, busque ayuda del tutor.

- Considere también el siguiente video colocado en el entorno virtual de aprendizaje sobre sentencias DDL de SQL. Para ello, revise el video [Lenguaje DDL](#) de (Morocho-Y., 2012b).

Este video muestra mediante un ejemplo cómo utilizar las sentencias de definición de datos (DDL) de SQL. Si después de haber visto el video aún tiene dudas o si se le presentó algún problema al momento de ejecutar alguna sentencia, busque ayuda del tutor.



Finalmente, le invito a revisar la siguiente guía didáctica donde encontrará explicaciones ampliadas a las temáticas revisadas en esta unidad:

- Morocho-Y. y Romero-Pelaez (2019). Guía didáctica de Fundamentos de Bases de Datos. Loja, Ecuador: Editorial Universidad Técnica Particular de Loja.



Actividades de aprendizaje recomendadas

Reforcemos el aprendizaje resolviendo las siguientes actividades:

1. Consulte en la Web, la sintaxis del comando *CREATE TABLE* de Oracle.

Luego, aplique la sintaxis de creación de una tabla que registra tareas y que tiene los atributos *Idtarea*, *tarea*, *detalle*, *fecha*, *hora_inicio* y *hora_fin*.

2. Consulte en internet sobre las formas de asignar alias a las columnas cuando necesitamos encabezados más elaborados que incluyen espacios en blanco entre palabras, y sobre la utilidad de la cláusula *Order by*.
3. Luego, aplique este conocimiento, en la tabla **tareas** creada anteriormente en otro apartado y cree una nueva tabla denominada **proyectos** con los atributos *idProyecto*, *proyecto*.
4. Relacione estas dos tablas y obtenga los resultados de las tareas asignadas a cada proyecto donde debe mostrar lo siguiente: nombre del proyecto, nombre de la tarea y fecha.

Nota. Por favor, complete las actividades en un cuaderno o documento Word

[10] Structured Query Language, por sus siglas en inglés.

[11] Historia de [Edgar F. Codd](#).





Semana 4

Unidad 3. Lenguaje SQL

Estimado estudiante, continuamos la Unidad 3. Lenguaje SQL, cuyo tema es amplio e importante para el perfil profesional del ingeniero en tecnologías de la información; por esta razón, es importante que haga énfasis en esta unidad.

La presente semana asegúrese que haya comprendido las funciones de agregación, subconsultas y modificación de la base de datos.

Estos temas se explican brevemente a continuación, aunque con mayor detalle se encuentran en la guía didáctica.

3.3. Funciones de agregación

¿Qué hace una función de agregación? Las funciones de agregación retornan un simple valor, calculado desde valores en una columna. SQL incorpora las cinco funciones de agregación: **SUM**, **AVG**, **COUNT**, **MAX** y **MIN**.

- **SUM**: Suma los valores de una columna numérica.
- **AVG**: Calcula el promedio de los valores de una columna numérica.
- **COUNT**: Cuenta el número de filas en una tabla o el número de valores no nulos en una columna.
- **MIN y MAX**: Encuentran el valor mínimo y máximo en una columna.

Por ejemplo, la siguiente consulta calcula el salario promedio de todos los empleados.

```
SELECT AVG(Salario)
```

```
FROM Empleados;
```



Revisemos otro ejemplo. Tomemos como base la tabla *Employees* del esquema de base de datos *HR* que viene por defecto al instalar *OracleXE*. La estructura de la tabla *Employees* es como sigue:

Employees (*employee_id*, *first_name*, *last_name*, *email*, *phone_number*, *hire_date*, *job_id*, *salary*, *commission_pct*)

```
SELECT avg (salary), --promedio salario.
```

```
MIN (salary), --salario más bajo.
```

```
MAX (salary), --salario más alto.
```

```
SUM (salary), --gasto en salarios.
```

```
COUNT (employee_id) --totalempleados.
```

```
FROM employees;
```



Le recomiendo ejecutar la consulta anterior en su base de datos. Esto para comprobar que, por cada función de agregación, la consulta devuelve un solo valor.

Por último, revise el siguiente el contenido de Nieto Bernal & Capacho Portilla, (2017), [Funciones de agregación](#), específicamente el capítulo 4 apartado 4.5.14, pp. 237 - 239 esto le permitirá reconocer la forma de aplicar las funciones de agregación.

Muy bien, ahora ya tiene clara la utilidad de las funciones de agregación, así que pasemos al siguiente tema

3.4. Subconsultas

Una subconsulta es una sentencia **SELECT**, que tiene la misma sintaxis que una sentencia normal. Una subconsulta aparece dentro de otra sentencia **SELECT**; dicho de otro modo, es una sentencia **SELECT** anidada dentro de otra.



La sintaxis de una subconsulta es la siguiente:

```
SELECT: nombre-columna, ...
```

```
FROM: nombre-tabla, ...
```

```
WHERE: nombre-columna operador (SELECT nombre-columna
```

```
FROM: nombre-tabla
```

```
WHERE: nombre-columna operador valor);
```

Cuando una subconsulta está presente en una sentencia SELECT, primero se ejecuta el SELECT más interior, ya que devolverá un valor que se requiere para completar la sentencia de mayor nivel y, finalmente, pueda ejecutarse.

La siguiente consulta muestra los nombres de los empleados cuyo salario es mayor que el salario promedio de todos los empleados.

```
SELECT Nombre
```

```
FROM Empleados
```

```
WHERE Salario > (SELECT AVG(Salario) FROM Empleados);
```



Revise el siguiente el contenido de Nieto Bernal & Capacho Portilla (2017), [Subconsultas](#), específicamente el capítulo 4 apartado 4.6.1, 4.6.2 y 4.6.3 , pp. 242 – 251. Esto le permitirá ampliar la explicación sobre las subconsultas en SQL.

Si todavía no tiene claro el funcionamiento de una subconsulta, consulte la guía didáctica en el apartado 3.7. Subconsultas, que le brinda una explicación más amplia del tema.



3.5. Modificación de la base de datos

Una parte del lenguaje SQL está dedicada a la manipulación de los datos (DML) de una base de datos. La base de datos es un activo de la organización y varios usuarios acceden al mismo tiempo; esto se denomina concurrencia. En una base de datos, varios usuarios a la vez estarán haciendo consultas, insertando datos en las tablas, actualizando datos y también borrando datos, por consiguiente, la base de datos está en continuo movimiento.

Las sentencias que se emplean en el lenguaje DML son:

Tabla 2
Sentencias

Sentencia	Sintaxis
INSERT	INSERT INTO <nombre-tabla> VALUES (valor1, valor2, valor3, ...);
UPDATE	UPDATE <nombre-tabla> SET columna1=valor1, columna2=valor2, ... WHERE alguna-columna operador valor;
DELETE	DELETE FROM <nombre-tabla> WHERE alguna-columna operador valor;

Nota. Morocho, J., 2023.

Veamos ejemplos de las sentencias de manipulación. Nos basamos en la tabla *Regions* (*región_id*, *region_name*) del esquema HR que viene en la instalación de *OracleXE*.

El comando `INSERT` se utiliza para agregar nuevas filas (tuplas) a una tabla. En el ejemplo, se agrega en la tabla `regions` una fila de datos para registrar Australia con el identificador 5.

```
INSERT INTO regions
VALUES (5, 'Australia'); --para insertar una nueva tupla.
```



El comando **UPDATE** se utiliza para modificar los datos existentes en una tabla. En el siguiente ejemplo, se actualiza dentro de la tabla regions la columna region_name con el valor 'australia' donde el identificador es igual a 5.

```
UPDATE regions.
```

```
SET region_name = 'australia'.
```

```
WHERE region_id = 5; --para actualizar el nombre de  
región.
```

El comando **DELETE** se utiliza para eliminar filas (tuplas) de una tabla. Cuando se combina con la sentencia **WHERE** se puede definir una condición que las filas deben cumplir para que sean borradas. Si no se especifica una cláusula **WHERE** se borrarán todas las tuplas de la tabla. En el ejemplo se eliminan los datos de la región cuyo **region_id** es igual a 5.

```
DELETE FROM regions.
```

```
WHERE region_id = 5; --para eliminar una tupla.
```



Revise el siguiente el contenido de Martínez Silverio & Tejada Betancourt (2019), [Sentencias de manipulación de datos](#), específicamente la Unidad IV, pp. 91 – 109, el cual le permitirá comprender desde otra perspectiva las operaciones de modificación de la base de datos.

Puede complementar el estudio de estos temas y puede revisar los recursos de aprendizaje que se indican a continuación.

Recurso de aprendizaje

En la siguiente guía didáctica encontrará explicaciones ampliadas a las temáticas revisadas en esta unidad.



Morocho, J. y Romero, A. (2019). [*Guía didáctica de fundamentos de bases de datos*](#). Loja, Ecuador: Editorial Universidad Técnica Particular de Loja.



Actividades de aprendizaje recomendadas

Es momento de aplicar sus conocimientos a través de las actividades que se han planteado a continuación:

1. Le sugerimos que complemente su estudio con el tutorial de SQL en línea disponible en el portal Web de [w3schools](https://www.w3schools.com).
2. Para evaluar el grado de asimilación de las sentencias SQL revisadas, le sugerimos desarrollar la prueba en línea disponible en el portal Web de [w3schools](https://www.w3schools.com).
3. Estimado estudiante para complementar su aprendizaje durante esta semana lo invito a desarrollar el siguiente crucigrama sobre funciones de agregación y subconsultas.

[Funciones de agregación y subconsultas](#)

Contenidos, recursos y actividades de aprendizaje recomendadas



Semana 5

Unidad 3. Lenguaje SQL

Estimado estudiante, hasta ahora hemos revisado consultas con el lenguaje SQL aplicadas a una sola tabla. En la mayoría de los casos de la vida real, la información está almacenada en más de una tabla, por lo que tenemos que aplicar combinación entre tablas.

La presente semana asegúrese de completar el estudio de combinación de tablas, vistas y transacciones. Estos temas se explican muy brevemente a continuación, aunque con mayor detalle se encuentran en la guía didáctica.



3.6. Combinación de tablas

La combinación entre tablas permite obtener información desde más de una tabla mediante las claves foráneas basadas en un atributo común. Por ejemplo, suponga que tiene una tabla que registra información de empleados y otra tabla con la información de los Departamentos en donde trabajan los primeros, si se requiere obtener un listado con los nombres de los empleados y el nombre del Departamento en donde trabajan, entonces se necesita combinar tablas. En SQL, esto se hace con las operaciones de JOIN.

En SQL tenemos algunos tipos de JOIN disponibles que los mencionamos a continuación:

- **INNER JOIN:** Combina filas de dos tablas solo si hay un atributo común entre las tablas.
- **LEFT JOIN / RIGHT JOIN:** Incluye todas las filas que coinciden entre ambas tablas más las filas de una tabla (izquierda o derecha), en las que no haya coincidencias.
- **FULL JOIN:** Es una combinación de LEFT y RIGHT JOIN juntas.

En la siguiente infografía se detalla algunos tipos de combinación:

[Combinación de tablas](#)

Revise el siguiente texto de Nieto Bernal & Capacho Portilla (2017) [Diseño de datos](#), capítulo 4, sección 4.6.4.1, 4.6.4.2, 4.6.4.3, 4.6.4.4, 4.6.5.1, 4.6.5.2, 4.6.5.3 pp. 222 – 237, para acceder a una explicación ampliada sobre consultas multitabla.

Para completar su estudio, consulte la guía didáctica en la sección 3.9, “Combinación de tablas”, donde puede revisar algunos ejemplos.


3.7. Vistas

¿Qué es una vista? Una vista es una relación virtual, que se define mediante una consulta SQL de cualquier tipo. Esta definición se almacena en la base de datos y se puede invocar como a cualquier tabla de la base de datos. Una vez



se invoque una vista, el SGBD ejecutará el código SQL almacenado en la definición, y la vista se creará como una relación que contenga todas las tuplas que devuelve la consulta. Se debe aclarar que las vistas no mejoran el rendimiento, sino que son una forma más ágil para ejecutar consultas un tanto complejas.

La sintaxis para crear una vista es la siguiente:



```
CREATE VIEW nombre_vista AS
SELECT nombre_columna(s)
FROM nombre_tabla
WHERE condiciones;
```

La llamada a una vista puede ser como sigue:

```
SELECT column_name(s)
FROM nombre_vista;
```

Revise el siguiente texto de Martínez López & Gallegos Ruiz (2017) [Programación de Bases de Datos Relacionales](#), capítulo 7, sección 7.1.3 pp. 169 – 170, para conocer un poco más sobre el funcionamiento de vistas de base de datos.

3.8. Transacciones

Una *transacción* es la unidad básica de procesamiento de una base de datos y refleja las operaciones que son activadas por eventos, como la compra de un boleto aéreo, el registro para un curso o realizar un depósito bancario. Se puede considerar a una transacción como un conjunto de operaciones que se tratan como una sola unidad. Si una operación falla, toda la transacción se cancela, asegurando así que los datos en la base de datos permanezcan consistentes.



Las transacciones en una base de datos ayudan a conservar las propiedades ACID (por sus siglas en inglés) de las transacciones:

- **Atomicidad:** Todas las operaciones dentro de una transacción se completan o ninguna se aplica.
- **Consistencia:** Las transacciones llevan la base de datos de un estado consistente a otro estado consistente.
- **Aislamiento:** Las transacciones concurrentes no interfieren entre sí.
- **Durabilidad:** Una vez que una transacción se ha completado, sus cambios son permanentes.

ANSI ha definido normas para el buen funcionamiento de las transacciones de una base de datos. En SQL se da soporte a las transacciones con dos comandos: `COMMIT` y `ROLLBACK`.

- **Commit:** hace que los cambios efectuados por la transacción se hagan permanentes en la base de datos. Luego de terminada una transacción, inmediatamente empieza otra.
- **Rollback:** ocasiona que todos los cambios realizados en la transacción se deshagan; por consiguiente, la base de datos vuelve a su estado antes de empezar la transacción y queda disponible para la ejecución de nuevas transacciones.

El siguiente ejemplo, muestra una transacción que realiza una transferencia de \$100 de una cuenta a otra.

```
BEGIN TRANSACTION;
```

```
UPDATE Cuentas SET Saldo = Saldo - 100 WHERE CuentaID  
= 1;
```

```
UPDATE Cuentas SET Saldo = Saldo + 100 WHERE CuentaID  
= 2;
```

```
END TRANSACTION;
```

```
COMMIT;
```



Si alguna de las operaciones falla, podemos revertir la transacción:

ROLLBACK;

Para una mayor comprensión de la unidad, se recomienda visitar la biblioteca virtual UTPL y revisar el siguiente material.



Revise el siguiente texto de Martínez López & Gallegos Ruiz (2017) [Manual de bases de datos](#), unidad 5, sección El control de transacción, pp. 131 - 134, para conocer un poco más sobre el funcionamiento de vistas de base de datos.

Recurso de aprendizaje

En la siguiente guía didáctica encontrará explicaciones ampliadas a las temáticas revisadas en esta unidad.

Morocho, J. y Romero, A. (2019). [Guía didáctica de Fundamentos de Bases de Datos](#). Loja, Ecuador: Editorial Universidad Técnica Particular de Loja.



Actividad de aprendizaje recomendada

Continuemos con el aprendizaje mediante su participación en la actividad que se describe a continuación:

Revise en la guía didáctica, los apartados sobre combinación de tablas, vistas y transacciones. Analice los ejemplos ahí presentados y una vez los haya entendido, realice las actividades sugeridas en la guía didáctica.

Nota. Por favor, complete la actividad en un cuaderno o documento Word





Semana 6

Unidad 3. Lenguaje SQL

Estimado estudiante continuamos con el estudio del lenguaje de bases de datos relacionales SQL. Como podrá darse cuenta, el lenguaje SQL es indispensable en su formación. En esta semana cerramos la unidad, abordando las restricciones de integridad, los índices y su utilidad y, finalmente, los privilegios sobre los objetos de base de datos. Estos temas se explican brevemente a continuación, aunque con mayor detalle se encuentran en la guía didáctica.

3.9. Restricciones de integridad

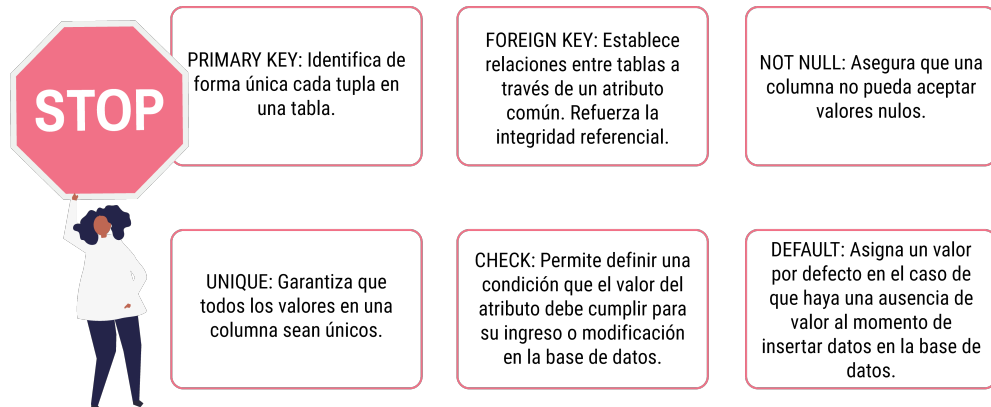
¿Qué es una restricción? Según la Real Academia de la Lengua, una *restricción* se define como la acción de restringir, es decir, “ceñir, circunscribir, reducir a menores límites”. En el contexto de bases de datos, una restricción ayuda a mantener los datos de forma consistente y coherente, lo que resulta beneficioso para la base de datos. Recordemos que una de las características de la base de datos es la consistencia, es decir, que la base de datos contenga datos que sean completos y tengan sentido.

En bases de datos relacionales, tenemos algunos tipos de restricciones entre las que podemos nombrar las siguientes: `primary key`, `foreign key`, `not null`, `unique`, `check` y `default`.



Figura 2

Tipos de restricciones



Nota. Morocho, J., 2023.

El siguiente ejemplo muestra la aplicación de algunos constraints:

```
CREATE TABLE Empleados (  
    EmpleadoID NUMBER,  
    Nombre VARCHAR2(100) NOT NULL,  
    Apellido VARCHAR2(100) NOT NULL,  
    FechaContratacion DATE,  
    Salario NUMBER(10, 2) CHECK (Salario>0),  
    CONSTRAINT empleados_pk PRIMARY KEY (EmpleadoID)  
);
```

Todas las restricciones apuntan a asegurar un nivel de calidad en los datos y ayudar a que la base de datos siempre sea coherente y consistente.

Revise el siguiente texto de Pulido Romero et al. (2019) [Base de Datos](#), capítulo 4 apartado 4.3, pp. 136 – 141, el cual le permitirá comprender con más detalle los tipos de restricciones en bases de datos.

3.10. Índices

¿Ha visto alguna vez el índice de un libro? Seguro que sí y también que los ha utilizado para ubicar de forma más exacta algún contenido que estaba buscando. Un concepto similar aplica para los índices de base de datos que se pueden crear con base en una o más columnas de una tabla. Los índices son estructuras adicionales a la tabla, por lo cual ocupan espacio en disco, y sirven para disminuir los tiempos de respuesta cuando se recupera información desde la base de datos debido a que no tienen que buscar secuencialmente a través de toda la tabla.

Por ejemplo, si se requiere consultar los datos de la tabla que contiene información de estudiantes, conviene que, a más de tener los datos indexados por la clave primaria, se indexe por nombres y/o apellidos de los estudiantes para que las búsquedas que se hagan por estos criterios sean más ágiles.



En lenguaje SQL, la sintaxis es la siguiente:

```
CREATE INDEX nombre_indice  
ON nombre_tabla (nombre_columna);
```

Para ejemplificar la sintaxis, vamos a crear un índice por la columna apellidos de la tabla estudiantes, lo cual acelera las búsquedas de estudiantes por el apellido:

```
CREATE INDEX apellidos_idx  
ON estudiantes (apellidos);
```

También se puede crear un índice único para asegurar que los valores en la columna indexada no se repitan:



```
CREATE UNIQUE INDEX correo_idx
```

```
ON Clientes (Correo);
```

En ocasiones una columna no basta para crear un índice por lo que se requiere crear un índice compuesto basado en dos o más columnas.

```
CREATE INDEX nombre_apellido_idx
```

```
ON Empleados (Nombre, Apellido);
```

A continuación, se muestran las ventajas y desventajas de los índices de base de datos:

Tabla 3

Ventajas y desventajas de los índices de datos

Ventajas	Desventajas
<ul style="list-style-type: none">• Mejora del Rendimiento: Aceleran las consultas y búsquedas.• Ordenación Rápida: Facilitan la ordenación de datos.	<ul style="list-style-type: none">• Espacio de Almacenamiento: Ocupan espacio adicional en la base de datos.• Impacto en las Operaciones de Escritura: Pueden hacer más lentas las operaciones de inserción, actualización y borrado debido a la necesidad de mantener el índice actualizado.

Nota. Morocho, J., 2025.

Revise el siguiente texto de Valderrey Sanz (2015) [Gestión de Base de Datos](#), capítulo 6 apartado 6.2.2, pp. 151, en donde encontrará información ampliada sobre la creación de índices de base de datos.



3.11. Privilegios

Los *privilegios* dentro de la temática de base de datos se orientan a permitir el acceso a objetos de base de datos desde los usuarios autorizados. El asignar privilegios es una estrategia de seguridad que permite evitar accesos indebidos a la base de datos, de tal manera que provoquen la alteración del contenido de la base de datos.

El usuario que es propietario del objeto de base de datos, por defecto es quien ha creado el objeto, es quien otorga los privilegios sobre ese objeto. Los privilegios a nivel de tablas de la base de datos pueden permitir consultar los datos (**SELECT**), insertar nuevos datos (**INSERT**), modificar los datos (**UPDATE**) o borrar los datos (**DELETE**). Entonces, un usuario A, propietario de la tabla XYZ, puede otorgar permisos de lectura, inserción, actualización o borrado, a un usuario B mediante el comando **GRANT** y también puede quitar esos privilegios a través del comando **REVOKE**.

El ejemplo a continuación muestra la asignación de los privilegios **SELECT** e **INSERT** sobre la tabla Empleados para el usuario usuarioA:

```
GRANT SELECT, INSERT ON Empleados TO usuarioA;
```

Como se sabe los privilegios son permisos y se pueden formar grupos de permisos predefinidos para simplificar la gestión, a estos grupos se los denomina Roles. Por ejemplo, un rol **Administrador** puede tener acceso de lectura y escritura, mientras que un rol **Analista** solo puede leer datos.

A continuación, se muestra un ejemplo en el que se crea un rol **gerente** al que se le asigna los privilegios **SELECT, INSERT, UPDATE** sobre la tabla Empleados y luego el rol gerente es asignado al usuario **usuarioA**.

```
CREATE ROLE gerente;
```

```
GRANT SELECT, INSERT, UPDATE ON Empleados TO gerente;
```

```
GRANT gerente TO usuarioA;
```



Finalmente, los privilegios otorgados pueden ser revocados utilizando el comando REVOKE.

```
REVOKE SELECT ON Empleados FROM usuarioA;
```

```
REVOKE ALL PRIVILEGES FROM usuarioA;
```

Para concluir con este apartado, lo invito a realizar la lectura del texto [Programación de Bases de Datos Relacionales](#), capítulo 7 apartado 7.11.3, pp. 204 – 206, para ampliar sus conocimientos acerca de la asignación y revocamiento de privilegios de base de datos.



Para revisar la sintaxis sobre privilegios, vaya a la sección 3.14. Privilegios, de su guía didáctica y analice los ejemplos ahí expuestos.

Recursos de aprendizaje

Considere la siguiente herramienta educativa colocada en el entorno virtual de aprendizaje para complementar el estudio de la presente semana:

En la siguiente guía didáctica encontrará explicaciones ampliadas a las temáticas revisadas en esta unidad.

Morocho, J. y Romero, A. (2019). [Guía didáctica de Fundamentos de Bases de Datos](#). Loja, Ecuador: Editorial Universidad Técnica Particular de Loja.





Actividades de aprendizaje recomendadas

Es hora de reforzar los conocimientos adquiridos resolviendo las siguientes actividades:

1. Ingrese al EVA y revise los anuncios académicos publicados por el tutor. Estas guías le ayudarán a comprender de mejor manera los contenidos.
2. Una vez completada la presente unidad sobre “Lenguaje SQL”, desarrolle la siguiente autoevaluación, con el objetivo de reafirmar lo estudiado de la unidad y que le permitirá medir el grado de asimilación de los contenidos revisados en la unidad y verificar su aprendizaje.



Autoevaluación 3

Seleccione la opción correcta.

A. Conteste Verdadero (V) o Falso (F) a las siguientes afirmaciones, según corresponda:

1. () En una instrucción SELECT la cláusula ORDER BY es obligatoria.
2. () En SQL las funciones de agregación calculan un único valor como resultado.
3. () La cláusula WHERE de los grupos es el HAVING.
4. () Una instrucción SELECT anidada se denomina subselección.
5. () Si las columnas de la tabla de resultados provienen de más de una tabla, debe utilizarse una combinación.
6. () La cláusula DELETE puede usarse para eliminar definitivamente una tabla.



7. () La reunión externa izquierda es simétrica a la reunión externa derecha.
8. () La operación de reunión externa completa se representa por LEFT-RIGHT OUTER JOIN.
9. () Una vista puede considerarse un método de seguridad.
10. () En una transacción, una vez aplicado un COMMIT, es posible aplicar un ROLLBACK para deshacer los cambios aplicados.
11. () Una vez creada una tabla, es posible añadir una restricción mediante la instrucción ALTER TABLE.

B. En las siguientes preguntas seleccione la respuesta correcta.

12. La operación relacional más frecuente es:

- a. Proyección.
- b. Selección
- c. Producto cartesiano

13. Examine la estructura de la tabla

EMP(emp Id, emp Nombre, emp Salario, comisión)

¿Qué sucedería al aplicar la siguiente sentencia?

ALTER TABLE emp MODIFY (salary DEFAULT 10000).

- a. Este genera un error debido a que las definiciones de columnas no pueden ser alteradas para añadir valores por defecto.
- b. Únicamente las subsecuentes inserciones donde EmpSalario no se especifique deberían almacenar el valor 10000.
- c. Generaría un error debido a que las definiciones de columna no pueden ser alteradas si la tabla ya contiene tuplas.
- d. Todas las tuplas existentes y las subsecuentes donde EmpSalario contenga un valor NULL se actualizarán a 10000.



14. Tenemos las tablas:

EMP(empId, empNombre, fechaIngreso, empSalario, depId) y,
DEPT(depId, depNombre, depLocalidad).

Se requiere generar un reporte que muestre todos los nombres de departamentos con la correspondiente cantidad que invierten en salarios.

¿Qué sentencia SQL ayudaría a cumplir el requerimiento?

- a. SELECT depNombre, SUM (empSalario) FROM emp e, dept d
WHERE e.depId = d.depId GROUP BY depNombre.
- b. SELECT depNombre, SUM(empSalario) FROM emp e, dept d
WHERE e.depId = d. depId GROUP BY depNombre, empSalario.
- c. SELECT depNombre, empSalario FROM emp e, dept d WHERE
e.depId = d. depId GROUP BY depNombre HAVING empSalario =
SUM(empSalario).
- d. SELECT depNombre, empSalario FROM emp e, dept d GROUP BY
depNombre HAVING empSalario = SUM(empSalario).

15. El lenguaje SQL se subdivide en:

- a. DML, DCL, TCL y FCL.
- b. DML, DDL, DCL y TTL.
- c. DML, DDL, DCL y XTL.
- d. DML, DDL, DCL y TCL.

16. Con el comando ALTER no se puede:

- a. Borrar una columna.
- b. Modificar el tipo de dato de una columna.
- c. Cambiarle el nombre a la tabla.
- d. Actualizar la tabla.



17. Con GRANT y REVOKE se puede:

- a. Otorgar permisos de SELECT a una o varias tablas.
- b. Otorgar permisos para borrar datos de una tabla.
- c. Retirar permisos para actualizar una tabla.
- d. Todas las anteriores.

18. Una transacción consiste:

- a. De una secuencia de instrucciones de consulta y de actualización.
- b. De una secuencia de instrucciones de consulta o de actualización
- c. De una instrucción de consulta.
- d. De una instrucción de actualización.

19. Los roles sirven para:

- a. Limitar los recursos que utiliza cada usuario.
- b. Agrupar un conjunto de permisos.
- c. Asignar permisos al administrador de la base de datos (DBA).
- d. Ninguna de las anteriores.

20. Para confirmar una transacción se emplea:

- a. El comando HAVING.
- b. La sentencia ROLLBACK.
- c. El comando COMMIT.
- d. La sentencia CHECK.

21. En una sentencia de reunión, se debe utilizar junto con JOIN la palabra reservada:

- a. ON.
- b. WHERE.
- c. HAVING.
- d. Cualquiera de las anteriores.

[Ir al solucionario](#)



Resultados de aprendizaje 1 a 5:

- Diferencia y utiliza los términos clave tales como: información, datos, base de datos, sistema de gestión de base de datos, metadatos, y minería de datos.
- Explica cómo el almacenamiento y recuperación de datos ha cambiado con el tiempo.
- Explica las ventajas de un enfoque de base de datos en comparación con el procesamiento de archivos tradicional.
- Define la calidad, precisión y puntualidad, explica cómo su ausencia tendrá un impacto en las organizaciones.
- Describe los mecanismos de recolección de datos y sus implicaciones (recopilación de datos automatizada, formularios de entrada, fuentes).

El estudiante diferencia y utiliza términos clave como información, datos, base de datos, sistema de gestión de base de datos, metadatos y minería de datos, comprendiendo su aplicación en la gestión de la información. Además, explica la evolución del almacenamiento y recuperación de datos, comparando el enfoque de base de datos con el procesamiento de archivos tradicional para identificar sus ventajas. También define la calidad, precisión y puntualidad de los datos, analizando su impacto en las organizaciones. Finalmente, describe los mecanismos de recolección de datos y sus implicaciones, considerando métodos como la recopilación automatizada, formularios de entrada y diversas fuentes de información.

Contenidos, recursos y actividades de aprendizaje recomendadas

Recuerde revisar de manera paralela los contenidos con las actividades de aprendizaje recomendadas y actividades de aprendizaje evaluadas.





Semana 7

Actividades finales del bimestre

Estimado estudiante, durante la presente semana es necesario empezar a hacer una revisión de lo abordado durante las semanas pasadas, con el fin de identificar aquellos temas en los que aún tenga dudas. Una vez haya identificado aquellos temas en los que requiere ayuda, comuníquese con el tutor solicitando la ayuda respectiva.

Recuerde que el tutor siempre tiene la disposición de ayudarle a cumplir sus objetivos de estudio. Asimismo, usted dispone de varios medios para comunicarse con el tutor: vía correo electrónico, mediante mensajes del EVA, vía telefónica o en el horario de tutorías.

Quiero recordarle también que durante las semanas 7 y 8, podrán participar en la actividad suplementaria que aplica para quienes no pudieron participar en el chat académico o para quienes deseen recuperar la nota de la actividad síncrona. A través del aula virtual, el tutor le proporcionará indicaciones más específicas para cumplir las actividades propuestas correspondientes al primer bimestre.

Contenidos, recursos y actividades de aprendizaje recomendadas



Semana 8

Actividades finales del bimestre

Estimado estudiante, estamos finalizando el primer bimestre. Durante esta última semana, podrá participar en la actividad suplementaria que aplica para quienes no pudieron participar en el chat académico o quienes deseen recuperar la nota de la actividad síncrona.



Además, le recuerdo que esta última semana pueden solicitar ayuda del tutor para resolver dudas que tenga antes de la evaluación presencial. Ahora, veamos las actividades que debe completar para cerrar el primer bimestre.
¡Éxitos!





Segundo bimestre

Resultado de aprendizaje 6:

Juzga si determinado material es información, datos o metadatos.

En el presente resultado de aprendizaje, nos adentraremos en la comprensión y aplicación de dos lenguajes formales para consultar bases de datos relacionales: el álgebra relacional y el cálculo relacional. Estos lenguajes proporcionan una estructura formal y precisa para expresar consultas que extraen información específica de las bases de datos. En el álgebra relacional, nos enfocaremos en operaciones fundamentales, como selección, proyección, unión, intersección y diferencia, basadas en la teoría de conjuntos.

Además, exploraremos el cálculo relacional, que se distingue por expresar qué información se busca sin especificar cómo obtener los resultados, adoptando así un enfoque más orientado a la búsqueda de patrones. Esta comprensión profunda de los lenguajes formales de consulta proporcionará a los estudiantes herramientas sólidas para expresar y ejecutar en entornos de bases de datos relacionales.

Contenidos, recursos y actividades de aprendizaje recomendadas

Recuerde revisar de manera paralela los contenidos con las actividades de aprendizaje recomendadas y actividades de aprendizaje evaluadas.



Semana 9

Unidad 4. Lenguajes de consulta formales

Estimado estudiante, iniciamos este segundo bimestre con ánimos renovados, sabiendo que estamos muy cerca de lograr nuestros objetivos trazados para el semestre. En esta unidad nos dedicaremos a revisar las operaciones de



álgebra relacional que se basan en la teoría de conjuntos y asimismo la formalidad del cálculo relacional. Estos temas se explican brevemente a continuación, aunque con mayor detalle se encuentran en la guía didáctica.

4.1. Álgebra relacional

Es un lenguaje formal que proporciona un conjunto de operaciones para manipular y consultar datos en una base de datos relacional. Fue introducido por Edgar F. Codd, el mismo científico que propuso el modelo relacional. El álgebra relacional es la base teórica sobre la cual se construyen los lenguajes de consulta como SQL.

El álgebra relacional incluye los mismos conceptos del modelo relacional de base de datos como son:

- Relación: Físicamente representada en una tabla de base de datos.
- Tupla: Representa una fila de una tabla.
- Atributo: Es una columna o propiedad de una tabla.

Las operaciones del álgebra relacional se basan en la teoría de conjuntos y reciben como entrada una o más relaciones, además de generar como resultado una nueva relación.

4.2. Operaciones fundamentales

Dentro de las operaciones fundamentales del álgebra relacional tenemos:

- **Selección (σ):** La operación de selección se utiliza para filtrar filas en una relación las cuales tienen que cumplir con una o varias condiciones.

$\sigma_{\text{Salario} > 500}(\text{Empleados})$

- **Proyección (π):** La operación de proyección se utiliza para seleccionar las columnas especificadas de una relación.

$\pi_{\text{Nombre, Apellido}}(\text{Empleados})$



- **Producto Cartesiano (\times):** La operación de producto cartesiano combina cada tupla de una relación con cada tupla de otra relación, por lo que el resultado puede ser muy extenso dependiendo del número de tuplas de cada relación.

`Empleados \times Departamentos`

- **Unión (\cup):** La operación de unión combina las tuplas de dos relaciones que tienen que ser compatibles con respecto a la unión, esto es ambas consultas deben tener el mismo número de atributos y cada atributo correspondiente debe coincidir en tipo de dato.

`Clientes \cup Proveedores`

- **Diferencia ($-$):** La operación de diferencia devuelve las tuplas que están en una relación, pero no en la otra.

`Clientes $-$ Proveedores`

Las operaciones descritas anteriormente, consideradas fundamentales, permiten realizar la mayoría de las operaciones de extracción de datos y están también las operaciones de combinación, intersección y división, que pueden expresarse en términos de las cinco operaciones básicas. Selección y proyección, son operaciones unitarias, debido a que se aplican a una sola relación para producir un resultado. Unión, diferencia y producto cartesiano también forman parte de las operaciones fundamentales y se denominan binarias, ya que reciben como entrada dos relaciones para producir un resultado.

Revise los siguientes textos:





- [Base de datos relacionales y modelado de datos](#), capítulo 2, apartado 4, pp. 62 - 68, que le servirá para complementar el estudio sobre las operaciones del álgebra relacional.
- Además, para conocer detalles sobre el funcionamiento de las operaciones fundamentales de álgebra relacional, revise el apartado 4.2. Operaciones fundamentales, en la guía didáctica.

Recursos de aprendizaje

Considere las siguientes herramientas educativas colocadas en el entorno virtual de aprendizaje para complementar el estudio de la presente semana:

- Morocho, J. y Romero, A. (2019). [Guía didáctica de Fundamentos de Bases de Datos](#). Loja, Ecuador: Editorial Universidad Técnica Particular de Loja.

En esta guía didáctica encontrará explicaciones ampliadas a las temáticas revisadas en esta unidad.

- Díaz, A. [Antonio FI UPM]. (2023, febrero 20). [Álgebra Relacional - operadores básicos](#). [Video].
- Díaz, A. [Antonio FI UPM]. (2023, febrero 20). [Álgebra Relacional - operadores derivados \(incluye división\)](#). [Video].

Este video le ayudará a conocer la estructura de las operaciones de álgebra relacional para que luego las apliquen en la resolución de ejercicios.





Actividades de aprendizaje recomendadas

Es momento de aplicar sus conocimientos a través de las actividades que se han planteado a continuación:

1. Con el objetivo de complementar lo estudiado, observe el video sobre álgebra relacional colocado en el entorno virtual de aprendizaje e identifique la forma de las operaciones de álgebra relacional que le ayudará en el desarrollo de ejercicios.
2. Además, ingrese al EVA y revise los anuncios académicos publicados por el tutor, cuyas guías le ayudarán a comprender de mejor manera los contenidos.

Contenidos, recursos y actividades de aprendizaje recomendadas



Semana 10

Unidad 4. Lenguajes de consulta formales

Estimado estudiante: continuamos con el estudio de esta unidad y esta semana nos dedicaremos a revisar el cálculo relacional como lenguaje de consulta formal de tipo procedimental. El tema se explica a continuación de forma breve, aunque con mayor detalle se encuentra en la guía didáctica.

4.3. Cálculo relacional

El cálculo relacional se fundamenta en el qué y no en el cómo. Se expresa qué es lo que se busca, pero –a diferencia del álgebra relacional– no se especifica cómo obtener los resultados; por ello es que se asemeja más a una búsqueda de patrones.

Asimismo, se distinguen dos tipos de cálculo relacional: de tuplas y de dominios.



4.3.1. Cálculo relacional de tuplas

Se basa en definir un conjunto de tuplas que satisfacen una fórmula lógica. Por lo que se debe describir las propiedades que deben tener las tuplas que interesan obtener como resultado. La sintaxis de una consulta tiene la siguiente forma:

$$\{T \mid P(T)\}$$

En donde:

T es una variable de tupla.

$P(T)$ es una fórmula lógica que describe las condiciones que debe cumplir T .

Por ejemplo, si se desea mostrar un listado de empleados cuyo salario es mayor a \$2000 se emplearía la siguiente fórmula lógica.

$$\{T.Nombre \mid Empleados(T) \wedge T.Salario > 2000\}$$

4.3.2. Cálculo relacional de dominios

Se basa en el uso de variables de dominio. Las consultas se expresan a través de fórmulas lógicas que describen las propiedades que deben cumplir los valores de los dominios (columnas) en el resultado. La sintaxis de una consulta tiene la siguiente forma:

$$\{ \langle d1, d2, \dots, dn \rangle \mid P(d1, d2, \dots, dn) \}$$

En donde:

$\langle d1, d2, \dots, dn \rangle$ son variables de dominio.

$P(d1, d2, \dots, dn)$ es una fórmula lógica que describe condiciones que deben cumplir los valores.



Tomamos el ejemplo anterior en donde se desea mostrar un listado de empleados cuyo salario es mayor a \$2000 pero representado ahora en cálculo relacional de dominios.

```
{<Nombre> | ∃ EmpleadoID, Salario (Empleados (EmpleadoID, Nombre, Salario) ∧ Salario > 2000)}
```

Para complementar el estudio del cálculo relacional, revise el apartado 4.3. Cálculo relacional, en la guía didáctica.



Revise el siguiente texto: [Base de datos relacionales y modelado de datos](#), capítulo 2, apartado 5 y 6, pp. 69 – 73, que le servirá para complementar el estudio sobre las operaciones del cálculo relacional.

Recurso de aprendizaje

Considere la siguiente herramienta educativa colocada en el entorno virtual de aprendizaje para complementar el estudio de la presente semana.

Morocho, J. y Romero, A. (2019). [Guía didáctica de Fundamentos de Bases de Datos](#). Loja, Ecuador: Editorial Universidad Técnica Particular de Loja.

En esta guía didáctica encontrará explicaciones ampliadas a las temáticas revisadas en esta unidad.



Actividades de aprendizaje recomendadas

Continuemos con el aprendizaje mediante su participación en las actividades que se describen a continuación:

1. Además, ingrese al EVA y revise los anuncios académicos publicados por el tutor, cuyas guías le ayudarán a comprender de mejor manera los contenidos.



2. Con el objetivo de reafirmar lo estudiado, desarrolle la autoevaluación de la unidad, que le permitirá medir el grado de asimilación de los contenidos revisados en la unidad “Lenguajes de consulta formales”.



Autoevaluación 4

Considere el siguiente modelo de base de datos relacional:

Empleado (nombre_persona, calle, ciudad).

Trabaja (nombre_persona, nombre_empresa, sueldo).

Empresa (nombre_empresa, ciudad).

Jefe (nombre_persona, nombre_jefe).

Responda los siguientes interrogantes.

1. ¿Cuáles de las seis operaciones básicas son binarias?
 - a. Unión.
 - b. Selección.
 - c. Diferencia.
 - d. Proyección.
 - e. Renombramiento.
 - f. Producto cartesiano.
2. Los operadores que toman una o dos relaciones como entrada, producen como salida:
 - a. El conjunto de tuplas que cumple la condición planteada en el WHERE.
 - b. Una nueva relación.
 - c. Una vista con las tuplas que cumplen el predicado.

3. Para seleccionar a todos los empleados que residen en Loja, ¿cuál de las siguientes sentencias emplearía?

- a. σ ciudad='Loja'(Empleado).
- b. π ciudad IN 'Loja'(Empleado).
- c. σ (Empleado ciudad>'Loja').

4. Se requiere presentar un listado con el nombre del empleado y el sueldo que gana, ¿cuál de las siguientes sentencias emplearía?

- a. σ nombre_persona, sueldo (trabaja).
- b. π nombre_persona, sueldo (empleado).
- c. π nombre_persona, sueldo (trabaja).

5. La operación producto cartesiano permite:

- a. Combinar información de dos relaciones cualesquiera.
- b. Encontrar tuplas que están en una relación, pero no en la otra.
- c. Encontrar el conjunto de tuplas que están en ambas relaciones eliminando las repetidas.

6. La operación de unión requiere dos condiciones para poder ejecutarse. Señale entonces cuál de las siguientes no corresponde:

- a. Las dos relaciones deben tener la misma cardinalidad, esto significa que deben tener el mismo número de atributos.
- b. Los dominios de los atributos i-ésimos de cada relación, se deben basar en el mismo dominio.
- c. Las dos relaciones deben tener un atributo común donde haya valores coincidentes.

7. Marque el enunciado correcto:

- a. El álgebra relacional es un lenguaje no procedimental y el cálculo relacional es procedimental.
- b. El álgebra relacional es un lenguaje procedimental y el cálculo relacional es no procedimental.



c. Tanto el cálculo relacional de tuplas como el cálculo relacional de dominios son lenguajes procedimentales.

8. La siguiente sentencia de cálculo relacional $\{S \mid \text{Empleado}(S) \wedge S.\text{ciudad} = \text{'Loja'}\}$, quiere decir que:

- a. Busque el nombre de una persona que es empleado y reside en Loja.
- b. Encuentre todas las tuplas de la relación empleado donde la ciudad sea Loja.
- c. No se puede ejecutar, ya que la variable S está mal declarada.

9. Existen dos tipos de cálculo relacional, señale entonces cuál de los siguientes tipos no corresponde:

- a. Tuplas.
- b. Conjuntos.
- c. Dominios.

10. El cálculo relacional de dominios sirve de base teórica al lenguaje:

- a. SQL.
- b. QBE.
- c. XML.

[Ir al solucionario](#)



Resultado de aprendizaje 7:

Explica los problemas básicos de la retención de datos, incluyendo la necesidad de retención, almacenamiento físico, la seguridad.

En el presente resultado de aprendizaje, se centrará en el diseño de bases de datos relacionales. El propósito de la fase de diseño de base de datos es la creación de un esquema que recoja todos los requerimientos de datos de la organización, incorporando claves primarias en cada tabla que permita identificación única de cada tupla, restricciones de integridad de entidades para mantener la consistencia interna y restricciones de integridad referencial para establecer conexiones entre las tablas. A través de ejercicios prácticos, se complementará el desarrollo de habilidades necesarias para el diseño de bases de datos.

Contenidos, recursos y actividades de aprendizaje recomendadas

Recuerde revisar de manera paralela los contenidos con las actividades de aprendizaje recomendadas y actividades de aprendizaje evaluadas.



Semana 11

Unidad 5. Diseño conceptual de la base de datos

Estimado estudiante durante las próximas dos semanas, revisaremos lo correspondiente acerca del proceso de diseño de bases de datos, tema importante para su formación como futuro ingeniero. El tema se explica a continuación de forma breve, aunque con mayor detalle se encuentra en la guía didáctica.



5.1. Diseño de bases de datos

Como en cualquier proyecto de desarrollo de software, el diseño de la base de datos empieza por obtener los requisitos de datos que requiere la organización. De igual forma, existe un proceso definido para llevar a cabo el diseño de la base de datos.

Entonces, el diseño de bases de datos es el proceso que busca crear un modelo para almacenar datos de manera eficiente y organizada. Este modelo permite que se pueda acceder, modificar y mantener los datos fácilmente. De manera análoga diremos que el diseño de la base de datos es cómo producir el plano de una casa antes de construirla.

5.2. Fases del diseño de bases de datos

Un proceso de diseño de base de datos empieza por identificar los requerimientos de datos de la organización para la cual se elabore el diseño de base de datos. Luego de ello se identifica tres fases bien definidas:

1. **Diseño conceptual**, donde se crea un modelo abstracto de la base de datos, generalmente utilizando el diagrama entidad-relación ER.
2. **Diseño lógico**, que es el proceso de traducir el modelo conceptual a un esquema lógico que pueda ser implementado en un sistema de gestión de bases de datos (SGBD).
3. **Diseño físico**, donde se implementa el modelo en un sistema de gestión de base de datos y una infraestructura determinada que sirve para especificar cómo se almacenarán los datos en el disco, incluyendo la organización de archivos, índices y otros detalles de implementación.

En esta asignatura solamente se abarcan las fases de diseño conceptual y lógico.





Revise el siguiente texto de Campos Monge et al. (2022) [Bases de Datos avanzadas e Ingeniería del Software](#), tema 40, apartado 40.2, pp. 45, que le servirá para introducirlo en el diseño de bases de datos relacionales.

Recursos de aprendizaje

Considere las siguientes herramientas educativas colocadas en el entorno virtual de aprendizaje para complementar el estudio de la presente semana:

- Morocho, J. y Romero, A. (2019). [Guía didáctica de Fundamentos de Bases de Datos](#). Loja, Ecuador: Editorial Universidad Técnica Particular de Loja.

En esta guía didáctica encontrará explicaciones ampliadas a las temáticas revisadas en esta unidad.

- Nettorius. (2022). [01 Por qué aprender el modelo ER \(Entidad-Relación\)](#). [Archivo de video].

Este video le ayudará a identificar las principales actividades por ejecutar durante la etapa de diseño de una base de datos.



Actividades de aprendizaje recomendadas

Es hora de reforzar los conocimientos adquiridos resolviendo las siguientes actividades:

1. Con el objetivo de complementar lo estudiado, observe el video sobre por qué aprender el modelo ER colocado en el entorno virtual de aprendizaje e identifique las características más importantes.
2. En la guía didáctica, revise los apartados de la unidad 5: 5.1 “Diseño de bases de datos” y 5.2. “Fases del diseño de bases de datos”; analice con más detalle las actividades de la fase de diseño conceptual





Unidad 5. Diseño conceptual de la base de datos

5.3. Diseño conceptual

Tal como indica su nombre, esta fase intenta construir un concepto acerca del problema de modelado a resolver. Por consiguiente, esta fase se centra en los requisitos de datos a representar en el modelo de datos.

¿Dónde encontramos los requisitos de datos? Generalmente, cuando se empieza un proyecto de diseño de software, una de las primeras tareas es la captura de requisitos y la generación del documento de requerimientos del sistema. Este documento sirve como entrada al proceso de diseño conceptual.

Para construir el modelo conceptual en forma de diagrama se emplea una herramienta gráfica, el modelo Entidad Relación (E-R). Este tema se explicará a continuación en forma breve, aunque con mayor detalle se encuentra en la guía didáctica.

5.4. Modelo E-R

¿Para qué sirve el modelo E-R? Es una herramienta que, de forma gráfica, permite hacer más entendible un modelo de datos. Los elementos más importantes del modelo E-R son las entidades y las relaciones, de ahí proviene su nombre. A más de entidades y las relaciones entre las entidades, se identifican los atributos, los dominios de los atributos, las claves primarias y las cardinalidades. El diagrama E-R constituye el producto de la fase de diseño conceptual y sirve como insumo para la fase de diseño lógico y a la vez forma parte de la documentación del proyecto.





Revise el siguiente texto de Nieto Bernal & Capacho Portilla (2017) [Bases de Datos avanzadas e Ingeniería del Software](#), capítulo 2, apartado 2.6.1, pp. 56 – 69, en el que encontrará a detalle el proceso de construcción de un diagrama E-R.

5.5. Especialización-generalización

¿Para qué sirve la técnica de especialización y de generalización? Estas técnicas buscan dotar de más semántica al diseño de base de datos. En ocasiones se requiere modelar con más detalle la relación entre entidades que muchas de las veces se encuentra abstraída. Es así que, para ir de un concepto más general a uno más específico, empleamos la especialización, por ejemplo, la entidad persona sería el concepto más general que se puede especializar en empleado, docente y alumno. La generalización sería el proceso inverso en el que se requiere ir de lo más específico hacia lo más general.



Revise el siguiente texto de Jiménez Capel (2024) [Bases de Datos avanzadas e Ingeniería del Software](#), capítulo 3, pp. 101 – 143, en el que encontrará información detallada sobre el proceso de diseño conceptual.

Recursos de aprendizaje

Considere las siguientes herramientas educativas colocadas en el entorno virtual de aprendizaje para complementar el estudio de la presente semana:

- Morocho, J. y Romero, A. (2019). [Guía didáctica de Fundamentos de Bases de Datos](#). Loja, Ecuador: Editorial Universidad Técnica Particular de Loja.

En esta guía didáctica encontrará explicaciones ampliadas a las temáticas revisadas en esta unidad.

- González, M. (2021, abril 13). Nivel 33 - Reto 3 - [Diseño conceptual Base Datos](#) - Modelo Entidad - Relación [Video]. YouTube.



Este video le ayudará a identificar las principales actividades por ejecutar durante la etapa de diseño conceptual.



Actividades de aprendizaje recomendadas

Es momento de aplicar sus conocimientos a través de las actividades que se han planteado a continuación:

1. Con el objetivo de complementar lo estudiado, observe el video sobre diseño conceptual colocado en el entorno virtual de aprendizaje e identifique las actividades de diseño de base de datos que se ejecutan en esta fase.
2. En la guía didáctica, revise los apartados de la unidad 5: 5.3. "Diseño conceptual", 5.4. "Modelo E-R" y 5.5. "Especialización/ Generalización"; analice los ejemplos ahí expuestos y relacione con los conceptos aprendidos.
3. Una vez completada la presente unidad, el siguiente cuestionario le permite verificar su aprendizaje en la unidad "diseño conceptual de bases de datos".



Autoevaluación 5

Seleccione la opción correcta.

1. Las necesidades de datos de los usuarios no son relevantes en la etapa de diseño conceptual.
 - a. Verdadero.
 - b. Falso.
2. En la fase de diseño conceptual la interacción con los usuarios es primordial.
 - a. Verdadero.
 - b. Falso.



3. El proceso de diseño de una base de datos es un proceso:
- a. Iterativo, que prácticamente no tiene fin.
 - b. Tedioso, pues se requieren muchos conocimientos.
 - c. Metodológico, ya que con base en este se define una metodología.
4. Uno de los insumos para el diseño de base de datos es:
- a. El diagrama de clases.
 - b. El diagrama de flujo del *software* por construir.
 - c. La especificación de requisitos de usuario.
5. Para transformar el diseño conceptual al diseño lógico nos basamos en las:
- a. Dependencias funcionales.
 - b. Cardinalidades de las relaciones.
 - c. Claves primarias y foráneas.
6. Una relación que tiene cardinalidad N:N, ¿cómo se transforma al diseño lógico?
- a. Se propaga la clave principal hacia la entidad que tenga cardinalidad máxima.
 - b. Se crea una tabla intermedia, cuyos atributos son las claves primarias de las entidades que está relacionando.
 - c. Se aplica normalización hasta reducir a una relación 1:N y aplicar propagación de clave primaria.
7. Para transformar al diseño lógico una relación expresada como especialización se debe:
- a. Generar una tabla por cada entidad de la especialización.
 - b. Generar solamente las subclases definidas en el modelo.
 - c. Generar solamente la superclase definida en el modelo.
 - d. Cualquiera de las anteriores.



8. Dentro del proceso de diseño de una base de datos, una técnica arriba-abajo es:

- a. Generalización.
- b. Especialización.
- c. Normalización.

9. ¿Cuál de los siguientes elementos no puede describir el modelo E- R?

- a. Conjuntos de entidades.
- b. Conjuntos de relaciones.
- c. Claves primarias.
- d. Claves foráneas.
- e. Restricciones.

10. Marque la respuesta incorrecta. La especialización y generalización definen una relación de inclusión entre:

- a. Superclases y subclases.
- b. Un conjunto de entidades fuertes y uno o más conjuntos de entidades débiles.
- c. Un conjunto de entidades de nivel superior y uno o más conjuntos de entidades de nivel inferior.

[Ir al solucionario](#)



Resultado de aprendizaje 8:

Explica el papel de los datos, la información y las bases de datos en las organizaciones.

El estudiante explica el papel de los datos, la información y las bases de datos en las organizaciones, comprendiendo cómo su gestión eficiente facilita la toma de decisiones, optimiza procesos y mejora la competitividad. Analiza la transformación de los datos en información valiosa y la importancia de las bases de datos para el almacenamiento, acceso y organización estructurada de la información dentro de una empresa.

Contenidos, recursos y actividades de aprendizaje recomendadas

Recuerde revisar de manera paralela los contenidos con las actividades de aprendizaje recomendadas y actividades de aprendizaje evaluadas.



Semana 13

Unidad 6. Diseño lógico de bases de datos

Estimado estudiante, continuamos con el proceso de diseño de bases de datos y vamos a revisar la fase de diseño lógico. El tema se explica a continuación en forma breve, aunque con mayor profundidad se encuentra en la guía didáctica.

6.1. Principios de diseño lógico de bases de datos

Una vez se haya generado el modelo E-R en la fase de diseño conceptual, se toma como base este insumo para generar el diseño lógico. Se toman como base las cardinalidades identificadas en el diseño conceptual para pasar del modelo E-R al modelo relacional, estas pueden ser relaciones 1:1, relaciones 1: No relaciones N:M y se aplica la regla correspondiente a cada caso.



Revise los siguientes videos, para complementar su aprendizaje sobre los elementos del diseño lógico.

- [Modelo Relacional para el diseño lógico de bases de datos](#)
- [Reglas de transformación en el Modelo Relacional](#)



Además, revise el contenido en [Bases de Datos avanzadas e Ingeniería del Software](#), de Campos Monge et al., (2022) Tema 40, apartado 40.4. Diseño Lógico, pp. 46 - 48.

6.2. Técnica de normalización

Una vez construido el modelo relacional, que forma parte de la documentación generada en el diseño lógico, se debe validar que se evite la redundancia al máximo. Una técnica para evitar la redundancia es la *normalización*, que se basa en la relación entre atributos denominados “dependencia funcional”. El proceso de normalización incluye algunas reglas conocidas como *formas normales*, las cuales se revisan a continuación.

6.3. Dominios atómicos y primera forma normal

Una de las primeras condiciones para iniciar el proceso de normalización es que las celdas de las tablas tengan valores atómicos, es decir, valores únicos que sean indivisibles. En la tabla 4 se puede observar que las columnas DNI_P y LibroTexto_As no tienen valores únicos.



Tabla 4*Ejemplo de tabla con celdas multivaluadas*

Cód_As	Nombre_As	Créditos_As	Facultad_As	DNI_P	LibroTexto_As
001	Base de Datos	6	Informática	11111111A	Libro 1
				22222222B	Libro 2
002	Redes	9	Arquitectura	22222222B	Libro 3
				33333333C	Libro 4

Nota. Tomado de *Programación de bases de datos relacionales* (p. 80), por Martínez, F. J., y Gallegos, A., 2017, RA-MA Editorial.

Para concluir lo invito a revisar el siguiente texto de Martínez López & Gallegos Ruiz (2017) [Programación de Bases de Datos Relacionales](#), capítulo 3, apartado 3.5, pp. 78 – 88, en el que encontrará una explicación sobre los elementos del diseño lógico.

Recurso de aprendizaje

Considere la siguiente herramienta educativa colocada en el entorno virtual de aprendizaje para complementar el estudio de la presente semana.

- Morocho, J. y Romero, A. (2019). [Guía didáctica de Fundamentos de Bases de Datos](#). Loja, Ecuador: Editorial Universidad Técnica Particular de Loja.

En esta guía didáctica encontrará explicaciones ampliadas a las temáticas revisadas en esta unidad.



Actividad de aprendizaje recomendada

Es hora de reforzar los conocimientos adquiridos resolviendo la siguiente actividad:

Ingrese al EVA y revise los anuncios académicos publicados por el tutor, cuyas guías le ayudarán a su proceso de aprendizaje.

Contenidos, recursos y actividades de aprendizaje recomendadas



Semana 14

Unidad 6. Diseño lógico de bases de datos

Estimado estudiante continuando con el estudio de la fase de diseño lógico, dentro del proceso de diseño de bases de datos, vamos a revisar las formas normales. El tema se explica a continuación en forma breve, aunque con mayor detalle se encuentra en la guía didáctica.

6.4. Dependencias funcionales

¿Qué son las *formas normales*? Es la relación que existe entre atributos de una relación que hace que se vayan agrupando y dando origen a las tablas del modelo de base de datos. Generalmente, se expresan de la forma $A \rightarrow B$, donde A es el atributo determinante y B es el atributo determinado.

6.5. Segunda forma normal

Se dice que una relación para estar en segunda forma normal requiere estar en primera forma normal y, además, que todo atributo que no forma parte de la clave principal depende funcionalmente de la clave principal en modo completo.

6.6. Tercera forma normal

Para que una relación esté en tercera forma normal, se requiere que esté en segunda forma normal y que, además, ningún atributo que no forma parte de la clave principal depende transitivamente de la clave principal.





Revise el siguiente texto de Martínez López & Gallegos Ruiz (2017) [Programación de Bases de Datos Relacionales](#), capítulo 3, apartado 3.5, pp. 78 – 88, en el que encontrará una explicación sobre los elementos del diseño lógico.

Recursos de aprendizaje

Considere las siguientes herramientas educativas colocadas en el entorno virtual de aprendizaje para complementar el estudio de la presente semana.

- Morocho, J. y Romero, A. (2019). [Guía didáctica de Fundamentos de Bases de Datos](#). Loja, Ecuador: Editorial Universidad Técnica Particular de Loja.

En esta guía didáctica encontrará explicaciones ampliadas a las temáticas revisadas en esta unidad.

- Díaz, A. [Antonio FI UPM]. (2023, marzo 2). [Formas normales en bases de datos](#) [Video]. YouTube.

Este video le ayudará a utilizar las formas normales para producir un modelo de datos con redundancia mínima



Actividades de aprendizaje recomendadas

Es hora de reforzar los conocimientos adquiridos resolviendo las siguientes actividades:

1. Con el objetivo de complementar lo estudiado, observe el video sobre el proceso de normalización colocado en el entorno virtual de aprendizaje; luego, aplique el proceso de normalización a un diseño de base de datos que usted haya generado.
2. Es momento de validar la asimilación de los contenidos de la unidad, por ello le invitamos a desarrollar la autoevaluación propuesta a continuación. Aunque el desarrollo de estas preguntas es opcional, se



recomienda resolverlas para consolidar sus conocimientos. Si tiene alguna duda, consulte con el docente-tutor.



Autoevaluación 6

Seleccione la opción correcta.

1. En la fase de diseño lógico se especifican las formas de organización de los archivos y las estructuras de almacenamiento.
 - a. Verdadero.
 - b. Falso.
2. La normalización busca construir un diseño de esquemas que se halle:
 - a. Al menos en 2FN.
 - b. Con cero redundancia.
 - c. En la forma normal adecuada.
3. Una relación sufre de anomalías de actualización cuando tiene:
 - a. Una clave primaria compuesta.
 - b. Redundancia en los datos.
 - c. Se halle en una de las formas normales no deseables.
4. Al identificar las dependencias funcionales, el atributo o grupo de atributos que está del lado izquierdo de la flecha se denomina:
 - a. Determinado.
 - b. Determinante.
 - c. Clave primaria.
5. Una tabla que contiene uno o más grupos repetitivos está en:
 - a. Forma no normalizada.
 - b. 1 FN.
 - c. 2 FN.



6. La normalización es un proceso formal que:

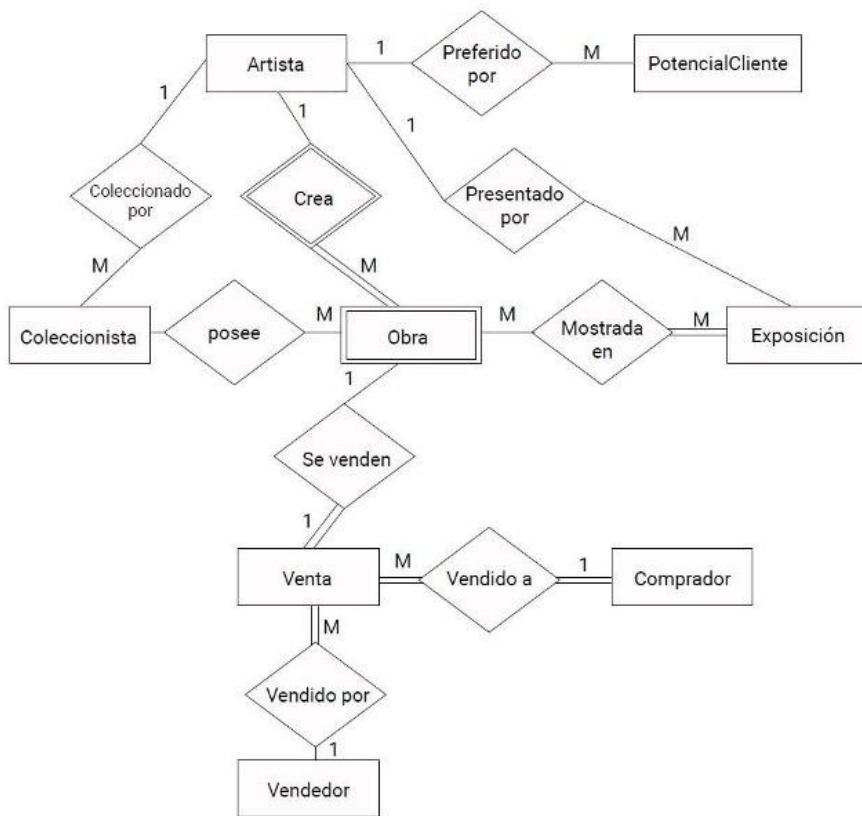
- a. Actúa como medio para eliminar totalmente la redundancia de un modelo de datos.
- b. Apoya en la identificación de relaciones, principalmente cuando tenemos esquemas grandes.
- c. Permite generar un conjunto de esquemas de relación a partir del diseño E-R.

7. Un dominio atómico se presenta cuando las relaciones están en:

- a. Forma no normalizada.
- b. 1 FN.
- c. 2 FN.

Con base en el esquema de exposiciones, responda las preguntas 8 a 10.





8. Para convertir al modelo relacional, la relación Artista- PresentadoPor-Exposición, el resultado sería:

- Se genera un solo conjunto de entidades, conformado por los atributos de las dos entidades que se relacionan.
- Se generan dos conjuntos de entidades: artista y exposición, y se propaga la clave principal de artista hacia exposición.
- Se generan tres conjuntos de: artista, presentado por y exposición, siendo presentado por una tabla intermedia.

9. Para convertir al modelo relacional, la relación obra— mostrada en exposición, el resultado sería:

- a. Se genera un solo conjunto de entidades, conformado por los atributos de las dos entidades que se relacionan.
- b. Se generan dos conjuntos de entidades: obra y exposición, y se propaga la clave principal de obra hacia exposición.
- c. Se generan tres conjuntos de entidades: obra, mostrada en y exposición, siendo mostrada en una tabla intermedia.

10. Para convertir al modelo relacional, la relación artista- coleccionado por-coleccionista, el resultado sería:

- a. Se generan dos conjuntos de entidades: artista y coleccionista, y se propaga la clave principal de artista hacia coleccionista.
- b. Se generan dos conjuntos de entidades: artista y coleccionista, y se propaga la clave principal de coleccionista hacia artista.
- c. Se generan tres conjuntos de entidades: artista, coleccionado por y coleccionista, siendo coleccionado por una tabla intermedia

[Ir al solucionario](#)



Resultados de aprendizaje 6 a 8:

- Juzga si determinado material es información, datos o metadatos.
- Explica los problemas básicos de la retención de datos, incluyendo la necesidad de retención, almacenamiento físico, la seguridad.
- Explica el papel de los datos, la información y las bases de datos en las organizaciones.

El estudiante juzga si un material corresponde a datos, información o metadatos, comprendiendo sus diferencias y aplicaciones. Además, explica los problemas básicos de la retención de datos, considerando la necesidad de almacenamiento, los desafíos del almacenamiento físico y la seguridad de la información. Finalmente, analiza el papel de los datos, la información y las bases de datos en las organizaciones, destacando su importancia en la gestión, toma de decisiones y optimización de procesos.

Contenidos, recursos y actividades de aprendizaje recomendadas

Recuerde revisar de manera paralela los contenidos con las actividades de aprendizaje recomendadas y actividades de aprendizaje evaluadas.



Semana 15

Actividades finales del bimestre

Estimado estudiante durante la presente semana, es necesario empezar a hacer una revisión de lo estudiado en las unidades 4, 5 y 6, con el fin de identificar aquellos temas en los que aún tenga dudas. Una vez haya identificado aquellos temas en los que requiere ayuda, comuníquese con el tutor solicitando la ayuda respectiva. Recuerde que el tutor siempre tiene la disposición de ayudarlo a cumplir sus objetivos de estudio. Asimismo, usted dispone de varios medios para comunicarse con el tutor: vía correo electrónico, mediante mensajes del EVA, vía telefónica o en el horario de tutorías.



Quiero recordarle también que durante las semanas 7 y 8 podrá participar en la actividad suplementaria, la cual aplica para quienes no pudieron participar en el chat académico o deseen recuperar la nota de la actividad síncrona.

A través del aula virtual, el tutor le proporcionará indicaciones más específicas para poder cumplir las actividades propuestas para la asignatura correspondientes al primer bimestre.

Contenidos, recursos y actividades de aprendizaje recomendadas



Semana 16

Actividades finales del bimestre

Estimado estudiante estamos finalizando el segundo bimestre. Durante esta última semana podrá participar en la actividad suplementaria, la cual aplica para quienes no pudieron participar en el chat académico o deseen recuperar la nota de la actividad síncrona.

Además, le recuerdo que esta última semana puede solicitar la ayuda del tutor para resolver dudas que tenga antes de la evaluación presencial.

¡Éxitos!





4. Autoevaluaciones

Autoevaluación 1

Pregunta	Respuesta	Retroalimentación
1	c	Al tener datos en varios archivos y no estar relacionados se dificulta la recuperación de datos que tengan algún tipo de relación.
2	b	Una base de datos representa en conjunto a la información que maneja o es propiedad de una persona u organización. El software que gestiona y controla el acceso a los datos es el SGBD.
3	b	El administrador de base de datos debe emplear el nivel lógico, ya que, debe analizar qué relaciones existen entre los datos que se van a almacenar en la base de datos.
4	a	El Lenguaje de Definición de Datos DDL se emplea para crear las estructuras (tablas) donde se almacenan los datos, así como otros objetos (vistas, índices, entre otros) de bases de datos.
5	c	El lenguaje DDL es el que describe el diseño de base de datos en el nivel físico, más no es una característica de la base de datos.
6	c	La lógica de negocio viene incorporada en el servidor de aplicaciones en lugar de estar distribuida entre múltiples clientes.
7	a	La minería de datos ayuda a encontrar información que antes no se podía ver o permanecía oculta entre grandes cantidades de datos.
8	b	El DBA debe permitir alta disponibilidad de la base de datos, lo que permite atender a los usuarios cuándo y dónde lo requieran.
9	a, c	Los sistemas basados en archivos tenían información duplicada lo que provocaba duplicidad. Además, al no controlar acceso concurrente tenían problemas de seguridad.
10	a, c	Un esquema conceptual permite validar que los requisitos de datos de la organización están contemplados en el esquema, así como también permite indicar los requisitos funcionales de la empresa. Recuerde que las características físicas de la base de datos se deben contemplar en etapas posteriores.



[Ir a la autoevaluación](#)



Autoevaluación 2

Pregunta	Respuesta	Retroalimentación
1	a	Una base de datos relacional es un conjunto de tablas que están relacionadas entre sí mediante claves foráneas y es precisamente esa característica por lo que se denomina relacional.
2	c	Un ejemplar de relación se denomina al estado en que se encuentra una relación en un instante determinado.
3	a	El término tupla es sinónimo de fila cuando hablamos de bases de datos relacionales.
4	b	El modelo relacional se basa en relaciones (de ahí su nombre), las cuales físicamente se representan a través de tablas.
5	a	El término atributo es sinónimo de columna cuando hablamos de base de datos relacionales.
6	a	Cada relación debe tener al menos una clave primaria que identifique las tuplas de forma única.
7	b	En cada relación, en el modelo relacional se deben definir claves candidatas de entre las cuales se selecciona la clave primaria.
8	b	La operación de selección es muy común; por lo general, las consultas de datos involucran el cumplimiento de alguna condición, por ejemplo, edad > 20.
9	c	El resultado de una consulta es un conjunto de tuplas que igualmente forman una relación que no tiene nombre.
10	a	Cuando hablamos de base de datos relacionales, una de sus características es que no debe tener valores duplicados.

[Ir a la autoevaluación](#)



Autoevaluación 3

Pregunta	Respuesta	Retroalimentación
1	F	En una instrucción <i>SELECT</i> , las cláusulas obligatorias son <i>SELECT</i> y <i>FROM</i> .
2	V	Una función de agregación puede ser <i>SUM()</i> . Esta función devuelve la sumatoria de una lista de valores; por consiguiente, un solo resultado.
3	V	La cláusula <i>HAVING</i> permite filtrar los grupos que se han formado mediante <i>GROUP BY</i> , para lo cual se evalúa una condición <i>HAVING</i> <condición>.
4	V	Una sentencia <i>SELECT</i> puede contener otras sentencias <i>SELECT</i> ; por ejemplo, en la parte del <i>WHERE</i> para completar una condición: <i>nom_departamento = (SELECT nom_depar FROM departamentos WHERE idDep = 10)</i> .
5	V	Cuando los datos que necesitamos presentar como resultado provienen de más de una tabla, se debe establecer un <i>JOIN</i> entre tablas, para lo cual deben tener un atributo común que permita dicha combinación.
6	F	La cláusula <i>DELETE</i> se usa para borrar tuplas de una tabla. En el caso de borrar una tabla empleamos <i>DROP TABLE</i> .
7	V	Son operaciones simétricas, lo único que cambia es que dependiendo si se aplica <i>LEFT OUTER JOIN</i> o <i>RIGHT OUTER JOIN</i> se recuperan los datos de la tabla que está a la derecha o a la izquierda del <i>JOIN</i> .
8	F	La operación de reunión externa completa se ejecuta mediante <i>FULL OUTER JOIN</i> .
9	V	Una vista puede ayudar a segmentar la información y solamente presentar una cierta cantidad, evitando mostrar todo el contenido de una o varias tablas.
10	F	La instrucción <i>COMMIT</i> confirma los cambios en los datos grabándolos en disco. Luego de un <i>COMMIT</i> ya no tendría efecto ejecutar inmediatamente un <i>ROLLBACK</i> , debido a que esta última sentencia deshace los cambios hasta antes de que se produzca un <i>COMMIT</i> .
11	V	Se puede alterar la tabla mediante <i>ALTER TABLE</i> ; por ejemplo, para añadir una restricción a nivel de columna que no permita aceptar valores nulos (<i>NOT NULL</i>).



Pregunta	Respuesta	Retroalimentación
12	b	Al alterar la tabla para incluir un valor por defecto para una columna, solamente se verán afectadas las siguientes inserciones a partir de la ejecución del <i>ALTER TABLE</i> .
13	a	Se debe aplicar la función <i>SUM()</i> para obtener los totales de inversión en salarios. Se requiere un <i>JOIN</i> entre tablas para obtener el nombre del departamento y agrupar mediante <i>GROUP BY</i> por el nombre del departamento.
14	d	SQL tiene comandos para definición de datos (DDL), manipulación de datos (DML), control de datos (DCL) y control de transacciones (TCL).
15	c	Para actualizar datos de una tabla, se requieren comandos DML.
16	d	Los comandos DCL: <i>GRANT</i> y <i>REVOKE</i> asignan privilegios para manipulación de datos sobre tablas.
17	b	Una transacción puede estar compuesta por una o más de una sentencia SQL que consulta o modifica datos en la base de datos.
18	b	Mediante un rol se pueden agrupar privilegios de acuerdo con las reglas del negocio, lo que provoca que la administración de usuarios sea más fácil.
19	c	<i>COMMIT</i> sirve para confirmar los cambios hechos a los datos, con lo cual las modificaciones se graban en disco.
20	a	<i>JOIN ON</i> es la sentencia que permite aplicar combinación (reunión) entre tablas mediante un atributo común.
21	a	<i>WHERE</i> se utiliza para filtrar datos y <i>HAVING</i> para filtrar grupos pero no se utilizan para el <i>JOIN</i> y por lo tanto la opción d se descartaría.

[Ir a la autoevaluación](#)



Autoevaluación 4

Pregunta	Respuesta	Retroalimentación
1	a, c, f	La unión requiere al menos dos relaciones para poder ejecutarse. La diferencia requiere al menos dos relaciones para funcionar y el producto cartesiano requiere dos relaciones para obtener un resultado.
2	b	Al ejecutar alguna operación de álgebra o cálculo relacional, el resultado es un conjunto de tuplas que forman una relación.
3	a	La operación de selección incluye restricciones (condiciones) que las filas tienen que cumplir para formar parte del resultado.
4	c	Se aplica una proyección debido a que se pide seleccionar solamente algunas de las columnas disponibles de la tabla y no incluye ninguna restricción.
5	a	El resultado es la combinación de cada una de las tuplas de la primera relación con todas y cada una de las tuplas de la segunda relación.
6	c	Cuando dos relaciones tienen un atributo en común con valores coincidentes, nos referimos a una clave foránea.
7	b	El álgebra relacional es procedimental porque debemos mencionar todas las operaciones que se deben realizar para obtener los datos solicitados. El cálculo relacional es no procedimental debido a que solamente se debe mencionar el patrón de búsqueda, más no el procedimiento por seguir para obtener los datos deseados.
8	b	S es una variable de tupla, la cual sirve para recuperar todas las tuplas de la relación Empleado que cumplen con la condición que el atributo ciudad sea igual a Loja.
9	b	El cálculo relacional de tuplas en el que el objetivo es encontrar las tuplas que cumplen una condición. El cálculo relacional de dominios se fundamenta en variables que actúan sobre todas las tuplas de la relación, pero toman sus valores de los dominios de los atributos.
10	b	QBE (<i>Query by Example</i> , consulta mediante ejemplo) es una técnica visual para elaborar consultas a base de datos que ayuda a comprender cómo obtener los datos de una base de datos.

[Ir a la autoevaluación](#)



Autoevaluación 5

Pregunta	Respuesta	Retroalimentación
1	b	En la etapa de diseño conceptual se construye el modelo ER que recoge los requerimientos de datos de la organización. Los requerimientos de datos provienen en su mayoría de los usuarios.
2	a	La activa participación de los usuarios en la etapa de diseño conceptual es crucial. Son ellos quienes aportan los requerimientos que necesitan ser modelados en la base de datos.
3	a	El diseño de una base de datos se dice que no tiene fin, debido a que siempre se puede mejorar o agregar alguna funcionalidad que requiere ser soportada por la base de datos. Sucede también que los cambios en los requerimientos suelen ser constantes por parte de los usuarios.
4	c	En un proceso formal de diseño de base de datos siempre se genera un documento de especificación de requisitos que es un insumo para el proceso de diseño conceptual de base de datos.
5	b	Las cardinalidades sirven de base para pasar del diseño conceptual al diseño lógico. Las cardinalidades se basan en el tipo de relación 1:n o n:n.
6	b	Un tipo de relación n:n siempre genera una tabla intermedia y sus atributos serán las claves primarias de las entidades que relaciona más los atributos que tenga la relación.
7	d	Todas las alternativas son válidas y se pueden implementar de acuerdo con las condiciones de modelado del problema por resolver.
8	b	La especialización, se dice una técnica de arriba-abajo, debido a que empieza en el concepto más general y se subdivide en conceptos más específicos. Por ejemplo, la entidad Persona se puede especializar en empleado, profesor y alumno, entre otros.
9	d	Las claves foráneas aparecen cuando pasamos del diseño conceptual al diseño físico, aplicando las reglas de transformación con base en las cardinalidades.
10	b	La especialización identifica una superclase de la que se derivan subclases más especializadas. También se dice que un subconjunto de entidades de nivel inferior se deriva de una entidad de nivel superior y también hereda sus atributos.

[Ir a la autoevaluación](#)



Autoevaluación 6

Pregunta	Respuesta	Retroalimentación
1	b	La especificación de las formas de organización de los archivos y las estructuras de almacenamiento se hacen en la fase de diseño físico.
2	c	Generalmente, el proceso de normalización llega hasta la 3FN, pero la redundancia no se puede eliminar sino minimizar.
3	b	La redundancia ocasiona anomalías de actualización. Esto quiere decir que, por ejemplo, cuando se actualiza algún dato, no se puede actualizar solamente en un sitio centralizado, sino que se debe actualizar en todas las ubicaciones donde exista ese dato, corriendo el riesgo que en algún sitio el dato no se actualice y, por consiguiente, los datos se vuelvan inconsistentes.
4	b	<i>Determinante</i> , pues determina al valor que se encuentra al lado derecho de la flecha.
5	a	Una forma no normalizada contiene grupos repetitivos de datos y es el punto desde donde se aplica la 1FN.
6	b	La normalización se basa en las dependencias funcionales y una dependencia funcional es la relación entre atributos; por tanto, al agrupar atributos estamos identificando las relaciones que componen la base de datos.
7	b	En 1FN ya no tenemos grupos repetitivos, sino un valor atómico en el dominio de cada atributo; esto quiere decir que son valores indivisibles.
8	b	La regla dice que cuando se presenta un tipo de relación 1:m, se propaga (duplica) la clave principal de la entidad que tiene cardinalidad mínima hacia la entidad que tiene cardinalidad máxima.
9	c	La regla dice que cuando se presenta un tipo de relación m:m, se debe crear una tabla intermedia formada por los atributos de clave primaria de las entidades que está relacionando.
10	a	La regla dice que cuando se presenta un tipo de relación 1:m, se propaga (duplica) la clave principal de la entidad que tiene cardinalidad mínima hacia la entidad que tiene cardinalidad máxima.

[Ir a la autoevaluación](#)





5. Referencias bibliográficas

- Campos Monge, M., Campos Monge, E. M., & López Querol, J. (2022). *Bases de datos avanzadas e ingeniería del software*. RA-MA Editorial. <https://elibro.net/es/lc/bibliotecaupl/titulos/230564>
- Jiménez Capel, M. Y. (2024). *Bases de datos relacionales y modelado de datos*. IFCT0310. IC Editorial. <https://elibro.net/es/lc/bibliotecaupl/titulos/279060>
- Martínez López, F. J., & Gallegos Ruiz, A. (2017). *Programación de bases de datos relacionales*. RA-MA Editorial. <https://elibro.net/es/lc/bibliotecaupl/titulos/106525>
- Martínez Silverio, D. A., & Tejada Betancourt, L. (2019). *Manual de bases de datos*. Universidad Abierta para Adultos (UAPA). <https://elibro.net/es/lc/bibliotecaupl/titulos/175897>
- Morocho-Y., J. C. (2012a). *Instalar OracleXE 11g en Windows*. https://www.youtube.com/watch?v=gjxhdEFIE_U
- Morocho-Y., J. C. (2012b). *Lenguaje DDL*. <https://www.youtube.com/watch?v=cq8UorDkiU4>
- Morocho-Y., J. C., & Romero-Peláez, A. (2019). *Guía didáctica de Fundamentos de Bases de Datos* (Universidad Técnica Particular de Loja, Ed.).
- Nieto Bernal, W., & Capacho Portilla, J. R. (2017). *Diseño de base de datos*. Universidad del Norte. <https://elibro.net/es/lc/bibliotecaupl/titulos/70030>



- Pulido Romero, E., Escobar Domínguez, O., & Núñez Pérez, J. A. (2019). *Base de datos*. Grupo Editorial Patria. <https://elibro.net/es/lc/bibliotecaupl/titulos/121283>
- Valderrey Sanz, P. (2015). *Gestión de bases de datos*. RA-MA Editorial. <https://elibro.net/es/lc/bibliotecaupl/titulos/62469>
- Zorrilla Pantaleón, M. E., & Duque Medina, R. (2011). Tema 01. Introducción a las BD Relacionales. En *Bases de Datos* (p. 34). <https://ocw.unican.es/pluginfile.php/1839/course/section/1447/Tema1.pdf>

