Bases de datos

Escuela de Código

Pilares Topiltzin

Temario

I.INTRODUCCIÓN A BASE DE DATOS
- **A. ¿Qué es un sistema gestor de bases de datos?**
- **B. Aplicaciones de los sistemas de bases de datos**
- **C. Propósito de los sistemas de bases de datos**
- **D. Bases de datos relacionadas**

II.MODELO RELACIONAL
- **A. La estructura de bases de datos relacionales**
- **B. El Modelo de entidad-relación**
- **C. Diagramas entidad-relación**
- **D. Aspectos del diseño entidad-relación**

III. LENGUAJE ESTRUCTURADO DE CONSULTA (SQL)
- **A. Características del lenguaje SQL. Utilidad del lenguaje**
- **B. Comando SQL. Grupo de comando**
- **C. Definición de datos**
- **D. Estructura básica de consultas SQL**

Temario

IV.ALMACÉN DE DATOS
- **A. Esquemas de almacenes de datos**
- **B. Diseño de almacenes de bases de datos práctica**

V. SOFTWARE DE DISEÑO DE BASE DE DATOS

VI. FUNCIONES EN SQL

VII. TRIGGERS EN SQL

VIII. CONSULTAS EN SQL
- **A. Consultas de Acción**
  - **a. Creación de tablas**
  - **b. Actualización**
  - **c. Eliminación**
  - **d. Inserción**
- **B. Consultas de selección**
  - **a. Consulta multitabla**
  - **b. Consulta de agrupación**

I. INTRODUCCIÓN A BASE DE DATOS

Objetivo:

Comprender qué es un sistema gestor de base de datos, así como la aplicación y propósitos, también presentar las características de las bases de datos relacionadas.

a) ¿Qué es un sistema gestor de bases de datos?

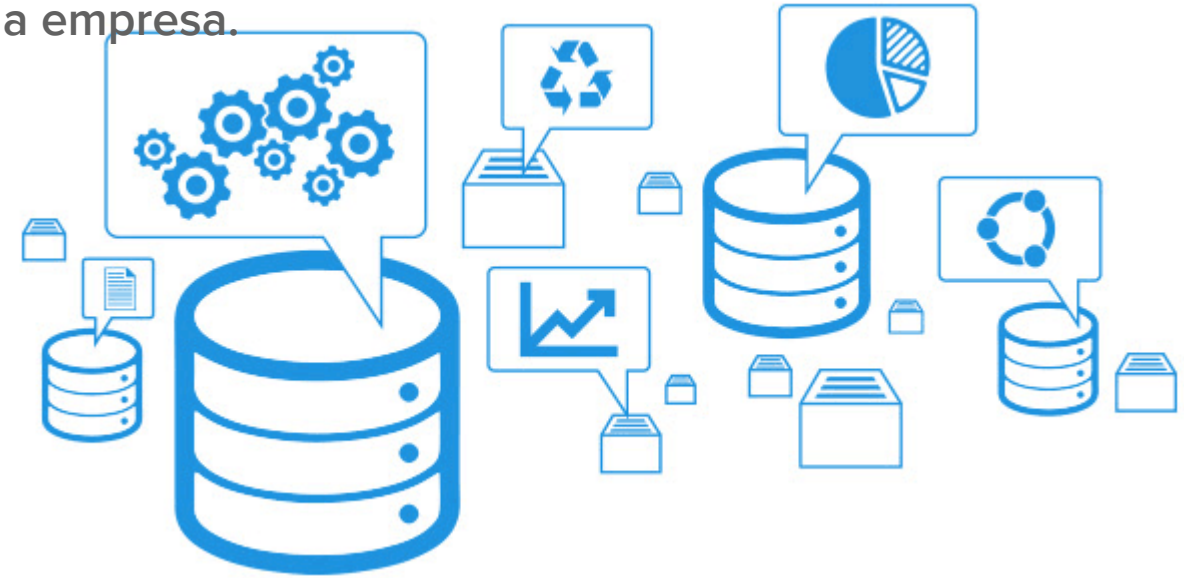
b) Aplicaciones de los sistemas de bases de datos

c) Propósito de los sistemas de bases de datos

d) Bases de datos relacionadas

¿Qué es un sistema gestor de bases de datos?

Un sistema gestor de bases de datos (SGBD) consiste en una colección de datos interrelacionados y un conjunto de programas para acceder a dichos datos. La colección de datos, normalmente denominada base de datos contiene información relevante para una empresa.



Objetivo

El objetivo principal de un SGBD es proporcionar una forma de **almacenar y recuperar** la **información de una base de datos** de manera que sea tanto **práctica** como **eficiente**. Los sistemas de bases de datos se diseñan para gestionar grandes **cantidades de información**. La gestión de los datos implica tanto la definición de estructuras para almacenar la información como la provisión de mecanismos para la manipulación de la información.


ORACLE


MySQL


SQLite


mongoDB


cassandra

En general

Los sistemas de bases de datos deben proporcionar las siguientes funciones:

- Instalar, configurar y gestionar bases de datos.
- Dar soporte al equipo de desarrollo, seguridad informática y redes.
- Definir el esquema del diccionario de datos.
- Especificar restricciones de integridad para asegurar los datos.
- Garantizar la alta disponibilidad de la base de datos.
- Contribuir a la creación de bases de datos más eficaces y consistentes.
- Determinar las estructuras de almacenamiento del sistema.
- Facilitar las búsquedas de datos de cualquier tipo y procedencia a los usuarios de negocio.
- Ayudar a mantener la integridad de los activos informacionales de la empresa.
- Introducir cambios en la información, si es requerido.
- Simplificar los procesos de consulta.
- Controlar los movimientos que se observan en la base de datos.

Entonces...
¿Qué actividades realiza un Administrador de Bases de Datos?

Asegurar el buen funcionamiento de las BBDD

Las bases de datos, en la gran mayoría de las ocasiones, almacenan una cantidad inmensa de datos, lo que puede disminuir su velocidad de búsqueda o ejecución, lo que disminuye enormemente su rendimiento.

Retención de información de la BBDD

La función más importante de las bases de datos es mantener guardados los datos para cualquier consulta.

Evitar pérdida de datos

Deben realizar periódicamente análisis de virus para detectar posibles amenazas y revisar que todos los componentes tanto externos como internos funcionan correctamente.

Solucionar incidencias y pérdidas de datos

En ocasiones no se puede evitar el desastre por un fallo del sistema y por tanto hay que pasar a un plan B de recuperación de datos.

¿Qué actividades realiza un Administrador de Bases de Datos?

Asegurar la seguridad de los datos

Es tarea del administrador de bases de datos establecer un sistema de verificación para poder acceder a los datos y a su vez proporcionar los acceso genéricos o unitarios a las personas que deben tener acceso a los datos. Así evitan que personal ajeno a la empresa acceda a esta información.

¿Qué operaciones se pueden realizar en una Base de datos con un SGBD?

Guardar datos (Agregar archivos nuevos a la Base de datos)

Borrar Datos

Insertar nuevos datos

Actualizar Datos

Obtener datos

Aplicaciones de los sistemas de bases de datos

- Banca: para información de los clientes, cuentas, préstamos y transacciones bancarias.
- Líneas aéreas: para reservas e información de horarios. Las líneas aéreas fueron de las primeras en usar las bases de datos de forma distribuida geográficamente.

Aplicaciones de los sistemas de bases de datos

- Universidades: para información de los estudiantes, matrículas en las asignaturas y cursos.
- Transacciones de tarjetas de crédito: para compras con tarjeta de crédito y la generación de los extractos mensuales.
- Telecomunicaciones: para guardar un registro de las llamadas realizadas, generar las facturas mensuales, mantener el saldo de las tarjetas telefónicas de prepago y para almacenar información sobre las redes de comunicaciones.
- Finanzas: para almacenar información sobre compañías, ventas y compras de productos financieros, como acciones y bonos; también para almacenar datos del mercado en tiempo real para permitir a los clientes la compraventa en línea y a la compañía la compraventa automática.

Aplicaciones de los sistemas de bases de datos

- Ventas: para información de clientes, productos y compras.
- Comercio en línea: para los datos de ventas ya mencionados y para el seguimiento de los pedidos Web, generación de listas de recomendaciones y mantenimiento de evaluaciones de productos en línea.
- Producción: para la gestión de la cadena de proveedores y para el seguimiento de la producción de artículos en las factorías, inventarios en los almacenes y pedidos.
- Recursos humanos: para información sobre los empleados, salarios, impuestos sobre los sueldos y prestaciones sociales, y para la generación de las nóminas.

En conclusión

Propósito de los sistemas de bases de datos

Una manera de guardar la información en la computadora es almacenarla en archivos del sistema operativo. Para permitir que los usuarios manipulen la información, el sistema tiene varios programas de aplicación que gestionan los archivos.

Los SO convencionales soportan este sistema de procesamiento de archivos típico. El sistema almacena los registros permanentes en varios archivos y necesita diferentes programas de aplicación para extraer y añadir a los archivos correspondientes.

Antes de la aparición de los sistemas gestores de bases de datos (SGBDs), las organizaciones normalmente almacenaban la información en sistemas de este tipo.

Desventajas

Guardar la información de la organización en un sistema de procesamiento de archivos tiene una serie de inconvenientes importantes:

- Redundancia e inconsistencia de los datos. Debido a que los archivos y programas de aplicación los crean diferentes programadores en el transcurso de un largo período de tiempo, es probable que los diversos archivos tengan estructuras diferentes y que los programas estén escritos en varios lenguajes de programación diferentes. Además, puede que la información esté duplicada en varios lugares (archivos).

- Dificultad en el acceso a los datos.
- Aislamiento de datos. Como los datos están dispersos en varios archivos, y los archivos pueden estar en diferentes formatos, es difícil escribir nuevos programas de aplicación para recuperar los datos correspondientes.
- Problemas de integridad. Los valores de los datos almacenados en la base de datos deben satisfacer ciertos tipos de restricciones de consistencia.

- Problemas de atomicidad. Los sistemas informáticos, como cualquier otro dispositivo mecánico eléctrico, está sujeto a fallos. En muchas aplicaciones es crucial asegurar que, si se produce algún fallo, los datos se restauren al estado consistente que existía antes del fallo.
- Anomalías en el acceso concurrente. Para aumentar el rendimiento global del sistema y obtener una respuesta más rápida, muchos sistemas permiten que varios usuarios actualicen los datos simultáneamente.
- Problemas de seguridad. No todos los usuarios de un sistema de bases de datos deben poder acceder a todos los datos.

Sistemas de información

Un Sistema de Información es un conjunto de componentes interrelacionados que trabajan juntos para recopilar, procesar, almacenar y difundir información para apoyar la toma de decisiones. Además apoyan la coordinación, control, análisis y visualización de una organización.

En general

Estas dificultades, entre otras, motivaron el desarrollo de los sistemas de bases de datos.

Bases de datos relacionadas

Una base de datos relacional es un tipo de base de datos que almacena y proporciona acceso a puntos de datos relacionados entre sí.

Las bases de datos relacionales se basan en el modelo relacional, una forma intuitiva y directa de representar datos en tablas.

En una base de datos relacional, cada fila en una tabla es un registro con una ID única, llamada clave. Las columnas de la tabla contienen los atributos de los datos y cada registro suele tener un valor para cada atributo, lo que simplifica la creación de relaciones entre los puntos de datos.

II. MODELO RELACIONAL

Objetivo:

Demostrar la estructura de las bases de datos relacionales, la representación de los diagramas de entidad-relación y los aspectos de su diseño.

a) La estructura de bases de datos relacionales

b) El Modelo de entidad-relación

c) Diagramas entidad-relación

d) Aspectos del diseño entidad-relación

Modelo de datos

¿Qué es un modelo?

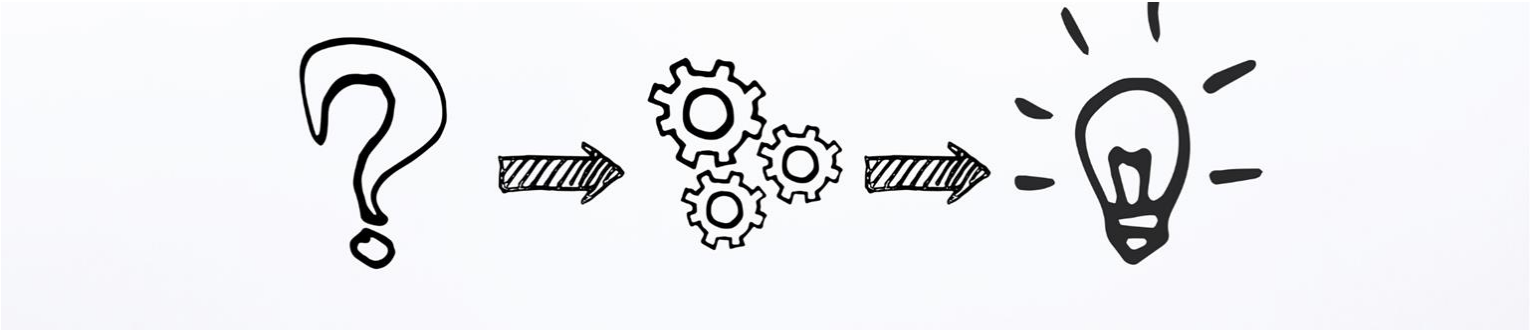
Un modelo es una representación abstracta, conceptual, gráfica o visual, física de fenómenos, sistemas o procesos a fin de analizar, describir, explicar, simular esos fenómenos o procesos.

Modelo de datos

Representación gráfica que describe la estructura de los datos y sus elementos de algún problema en particular. Abstracción de un problema a resolver.

Ventajas

- Visión unificada
- Entendimiento del contexto del problema a resolver
- Herramienta de comunicación entre participantes



Problema **Abstracción** **Desarrollo y Solución**

La estructura de bases de datos relacionales

Una base de datos relacional consiste en un conjunto de tablas, a las cuales se les asigna un nombre exclusivo. Cada fila de la tabla representa una relación entre un conjunto de valores. De manera informal, cada tabla es un conjunto de entidades, y cada fila es una entidad.

Ejemplo de una tabla.

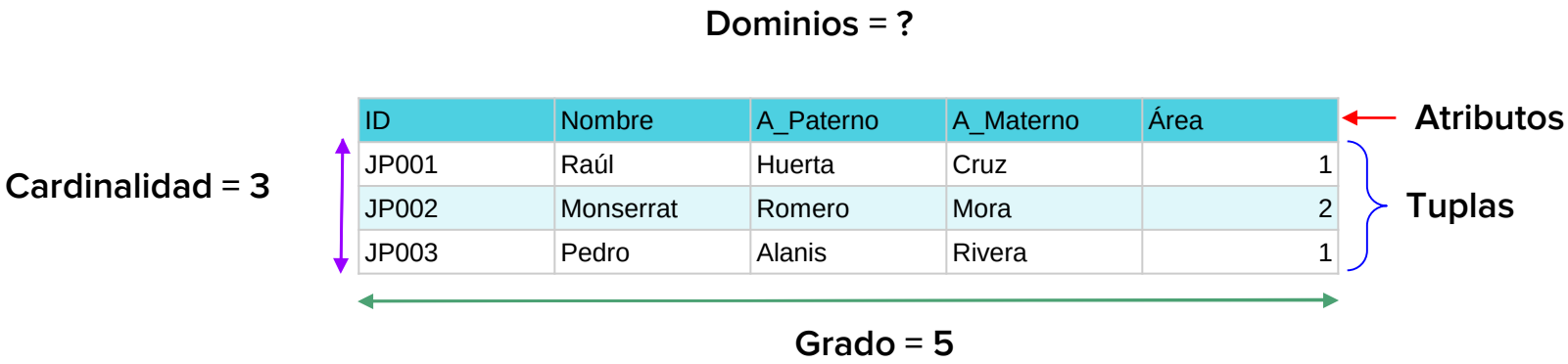
Modelo Relacional

Un modelo relacional consiste en representar datos por medio de tablas relacionadas cuyas filas se llaman tuplas y las columnas atributos, conformando así una base de datos

ID	Nombre	A_Paterno	A_Materno	Área
JP001	Raúl	Huerta	Cruz	1
JP002	Monserrat	Romero	Mora	2
JP003	Pedro	Alanis	Rivera	1

Tablas

- Tupla: Corresponde a cada fila de la tabla.
- Atributo: Corresponde a cada una de las columnas.
- Cardinalidad: Número de tuplas.
- Grado: Numéro de atributos de la tabla
- Dominio: Rango de valores que puede tomar un atributo.



El Modelo de entidad-relación

El modelo entidad-relación es el modelo conceptual más utilizado para el diseño de bases de datos. Fue introducido por Peter Pin-Shan Chen en 1976.

El modelo entidad-relación está formado por un conjunto de conceptos que permiten describir la realidad mediante un conjunto de representaciones gráficas y lingüísticas.

Conceptos generales

- Entidad: Cualquier cosa a partir de la cual se realiza obtención de datos.
- Atributo: Es una característica de una entidad. Normalmente las entidades se definen en términos de su lista de atributos.
- La entidades involucradas dentro de un diseño se relacionan entre sí. Para representar las relación se utiliza el símbolo:

Diseño conceptual (Diagrama)

- **Entidad**

Para representar una Entidad, se utiliza un rectángulo, con el nombre correspondiente.

- **Atributo**

Para representar un Atributo, se utiliza una elipse, con el nombre correspondiente.

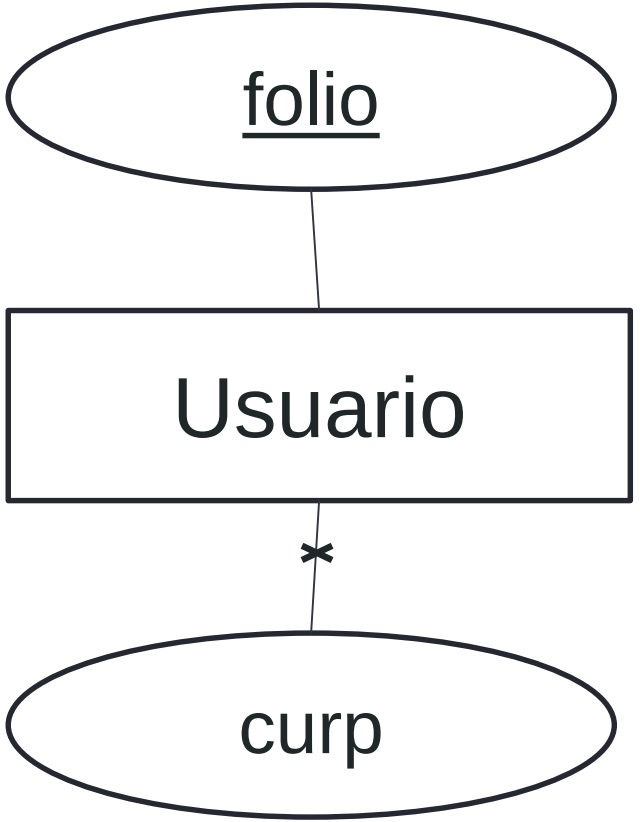
Diseño conceptual (Diagrama)

- ## Clave Principal (PK)

Una clave principal es un atributo cuyo valor es único para cada instancia de la entidad. Ayuda a identificar de forma única a cada una de las instancias de la Entidad.
Toda Entidad debe de tener su Clave primaria.

- ## Clave candidata

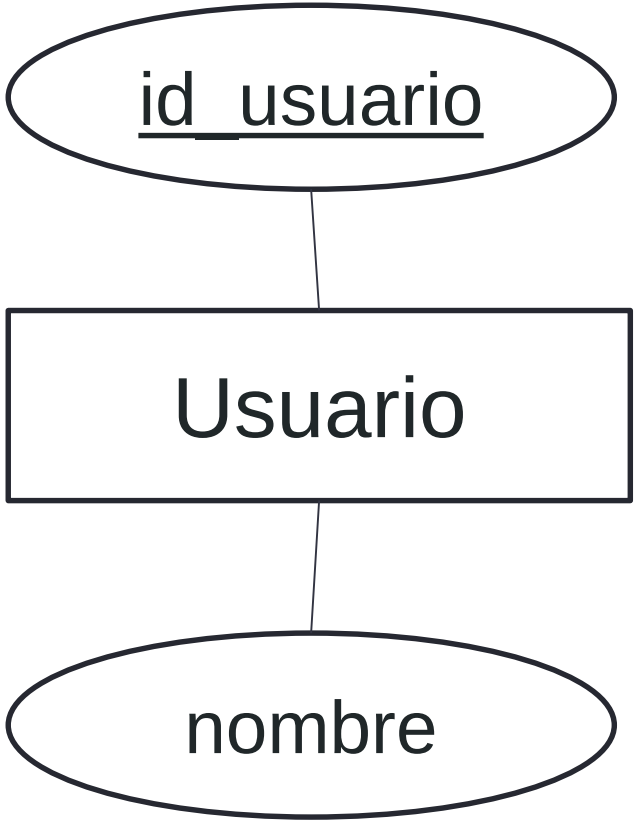
Una Entidad puede tener varias claves principales, con el mismo nivel jerárquico. En este caso se debe elegir una, y a las demás llamarlas claves candidatas.



Diseño conceptual (Diagrama)

- Clave Artificial (PK)

Cuando no existe una PK o no se encuentra definida, es necesario hacer uso de una Clave artificial. Para implementarla se puede usar el termino ID.



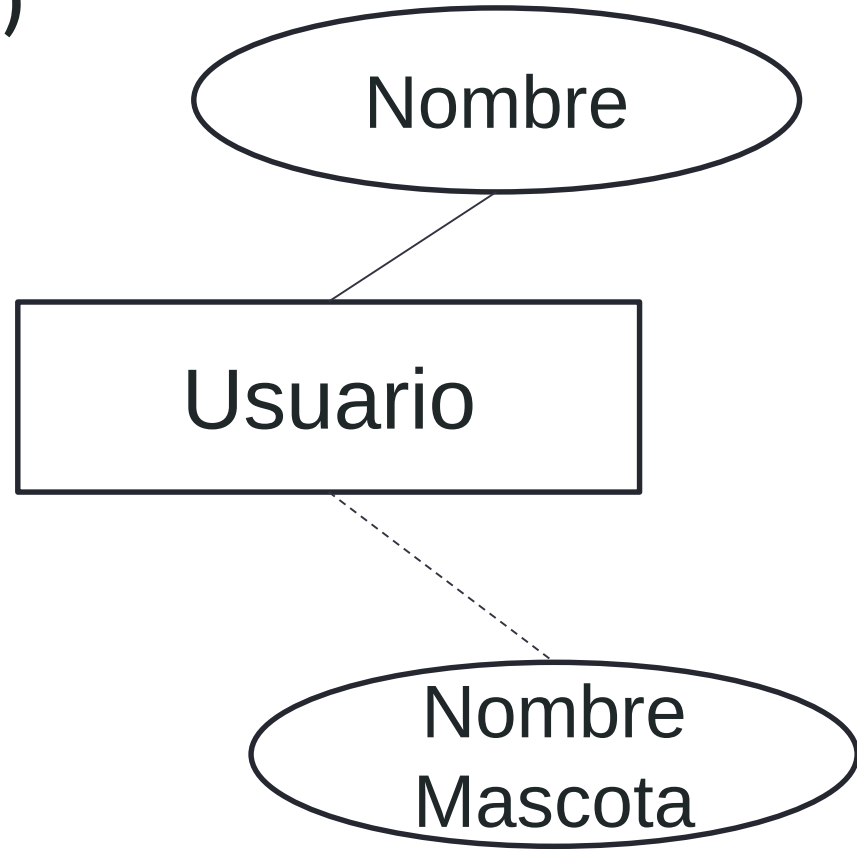
Diseño conceptual (Diagrama)

- **Atributo requerido**

Aquellos que no pueden se nulos, y debe estar presentes de forma obligatoria.

- **Atributo Opcional**

Son atributos que se pueden Omitir, se conocen también como atributos de valor nulo.



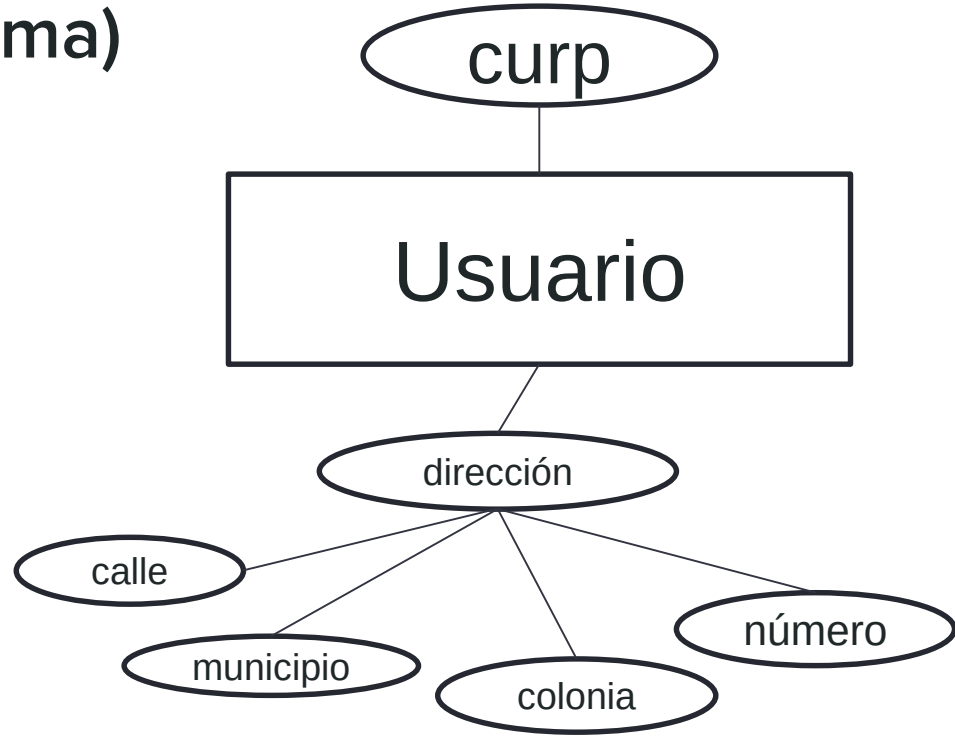
Diseño conceptual (Diagrama)

- **Atributo simple**

Aquellos que no pueden descomponerse en otros atributos.

- **Atributo Compuesto**

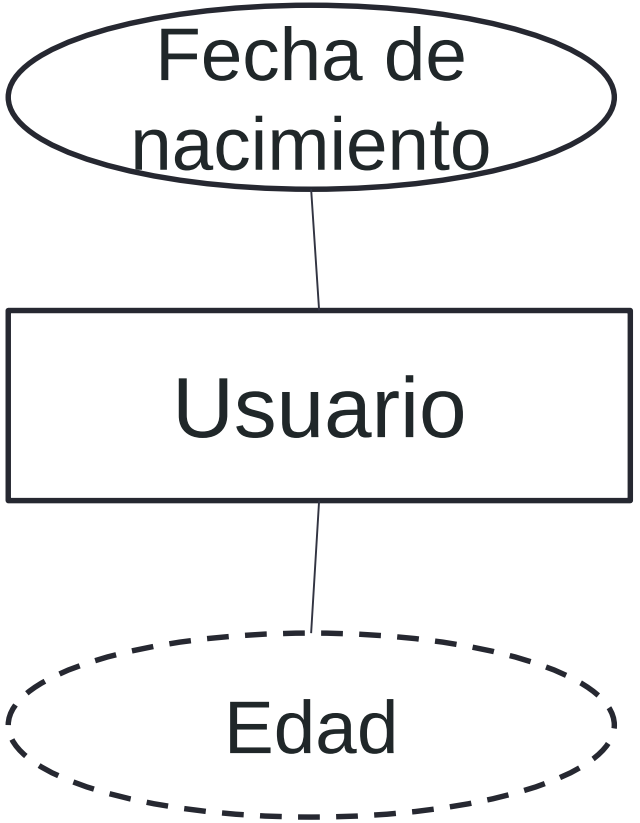
Aquellos que pueden dividirse en Sub Atributos.



Diseño conceptual (Diagrama)

- **Atributo derivado**

Es un atributo cuyo valor puede ser generado a partir de otro atributo, mediante un calculo, algoritmo, función etc.



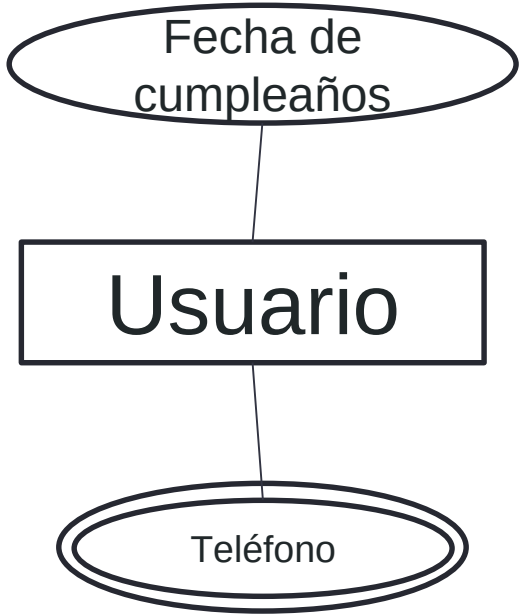
Diseño conceptual (Diagrama)

- **Atributo de valor simple**

Aquellos que solo pueden tener un valor por cada registro.

- **Atributo Multivalor**

Aquellos que pueden tener más de un valor para cada registro o instancia de una entidad



Representación de Relación entre Entidades

- Se utilizará un rombo, que contendrá la relación entre entidades.

Un usuario puede Asistir a un curso

Tipos Relación entre Entidades

- 1:1

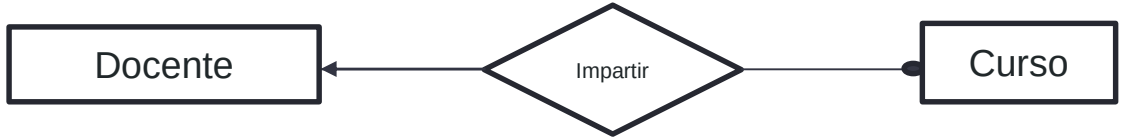
- 1:N

- N:M

Tipos Relación entre Entidades

- 1:1

- 1:N



- N:M

Cardinalidad

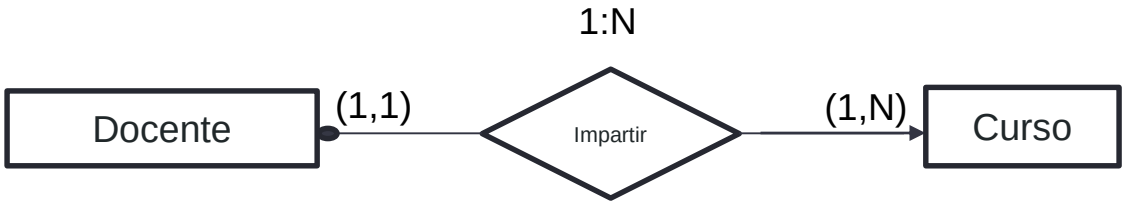
La cardinalidad expresa el número mínimo y máximo de ocurrencias de una entidad (instancias o registros) asociados con una ocurrencia (instancia o registro) de otra entidad.

(x,y)

X=Mínimo
Y=Máximo

Tipos Relación entre Entidades

- 1:N



Ejercicios

Ejercicio 1

Establecer el modelo Entidad-Relación de una Colonia y la Delegación.

Proponer los atributos de cada una de las entidades.

Ejercicio 2

Establecer la relación de un supervisor con un área de trabajo. De acuerdo a las siguientes restricciones.

- Un área es supervisada por un solo supervisor.
- Un supervisor sólo puede supervisar un área asignada.

Proponer los atributos de cada una de las entidades.

III. LENGUAJE ESTRUCTURADO DE CONSULTA (SQL)

Objetivo:

Presentar el lenguaje SQL, las utilidades, características, los comandos, cómo es que se realiza la definición de datos y la estructura básica de consultas mediante SQL.

A. Características del lenguaje SQL
   Utilidad del lenguaje

B. Comando SQL. Grupo de comando

C. Definición de datos

D. Estructura básica de consultas SQL

SQL

SQL es un acrónimo en inglés para **Structured Query Language**. Un **Lenguaje de Consulta Estructurado**. Un tipo de lenguaje de programación que permite manipular y descargar datos de una base de datos. Tiene capacidad de hacer cálculos avanzados y álgebra. Al día de hoy es el más utilizado en cualquier ámbito en el que se trabaje con bases de datos.

Se trata de un lenguaje que permite acceder, modificar o eliminar la información que se almacena en las bases de datos. Esta información está relacionada entre sí, por lo que debe ser estructurada y almacenada siguiendo un sistema. El lenguaje SQL permite interactuar con esa información.

Origen

En la década de los 70, en pleno desarrollo de las bases de datos, IBM creaba un lenguaje con el que gestionar los datos almacenados en el nuevo software System R. Era el **SEQUEL**, que más tarde pasaría a llamarse **SQL** (Structured Query Language).

En 1986, fue declarado estándar del Instituto Nacional Estadounidense de Estándares (ANSI) y, un año después, de la Organización Internacional de Normalización (ISO). Fue tal su impacto, que a partir de ese momento, varias compañías lanzaron su **propia versión**.

Oracle

Oracle PLSQL es un acrónimo para *“Procedural Language extensions to SQL”* y es una extensión de SQL usada en Oracle.

MySQL

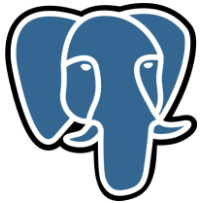
MySQL es considerada la base de datos de código abierto más popular del mundo.

Microsoft SQL Server

Microsoft SQL Server es un tipo de SQL desarrollado por Microsoft y tiene algunas características especiales como que soporta procedimientos almacenados.

PostgreSQL

PostgreSQL es otro tipo de base de datos SQL de código abierto. No es tan popular como MySQL pero está ganando terreno.



MySQL Workbench

MySQL Wordkbench es una herramienta visual unificada para arquitectos de bases de datos, desarrolladores y administradores de bases de datos,

MySQL Workbench proporciona modelado de datos, desarrollo de SQL y herramientas de administración integrales para la configuración del servidor, la administración de usuarios, la copia de seguridad y mucho más.

Plataformas: Windows, Linux y Mac OS X.

Aplicaciones del lenguaje SQL

Comandos SQL

Los comandos de SQL se dividen en 4 grupos:

DDL (Data Definition Language – Lenguaje Definición de Datos)	Para creación, modificación y eliminación de la estructura y objetos de la base de datos.	CREATE ALTER DROP TRUNCATE COMMENT RENAME
DML (Data Manipulation Language – Lenguaje Manipulación de Datos)	Para recuperar y trabajar con datos. En general permiten generar consultas para ordenar, filtrar y extraer datos de la base de datos.	SELECT INSERT UPDATE DELETE MERGE CALL EXPLAIN LOCK TABLE
DCL (Data Control Language -Lenguaje Control de Datos)	Para controlar el acceso a los datos.	GRANT REVOKE
TCL (Transaction Control Language -Lenguaje Control de Transacciones)	Para las transacciones de datos.	COMMIT ROLLBACK SAVEPOINT SET TRANSACTION

DDL

CREATE – Utilizado para crear objetos (tablas, vistas, índices) en la base de datos.

Estructura básica:

```
CREATE TABLE table_name (
           column1 datatype,
           column2 datatype,
           column3 datatype,
   ....
);
```

DDL

ALTER – Utilizado para modificar las tablas agregando o cambiando la definición de los campos.

Estructura básica:

```
ALTER TABLE table_name
ADD column_name datatype;
```

DDL

DROP – Utilizado para eliminar objetos en la base de datos

Estructura básica:

```
DROP TABLE table_name;
```

DDL

TRUNCATE – Borra todo el contenido de una tabla

Estructura básica:

```
TRUNCATE TABLE table_name;
```

DDL

COMMENT – Agregar comentarios al diccionario de datos sobre un objeto de la base de datos

Estructura básica:

```
COMMENT ON COLUMN table_name.column;
```

DDL

RENAME – Renombrar la tabla de la base de datos

Estructura básica:

```
ALTER TABLE table_name RENAME TO table_name_1
```

DML

SELECT – Utilizado para consultar registros de la base de datos que satisfagan un criterio determinado

Estructura básica:

```
SELECT column1, column2, ...
FROM table_name;

SELECT * FROM table_name;
```

DML

INSERT – Utilizado para cargar lotes de datos en la base de datos en una única operación

Estructura básica:

```
INSERT INTO table_name (column1, column2, column3, ...)
VALUES (value1, value2, value3, ...);
```

DML

UPDATE – Utilizado para modificar los valores de los campos y registros especificados

Estructura básica:

```
UPDATE table_name
SET column1 = value1, column2 = value2, ...
WHERE condition;
```

DML

DELETE – Utilizado para eliminar registros de una tabla de una base de datos

Estructura básica:

```
DELETE FROM table_name WHERE condition;
```

DELETE VS TRUNCATE

DELETE VS TRUNCATE

En la base de datos se lleva un registro de todas las transacciones que se realizan, llamado log de transacciones, cuando se ejecuta un DELETE se graban los IDs de los registros eliminados.

Un TRUNCATE no guarda los IDs, solo marca las páginas que contenían los datos para que se puedan volver a ocupar.

Nota:
Un *log* ("registro", en español) es un archivo de texto en el que constan cronológicamente los acontecimientos que han ido afectando a un sistema informático (programa, aplicación, servidor, etc.), así como el conjunto de cambios que estos han generado.

DCL

GRANT – Permite asignar permisos sobre el objeto de la base de datos. El usuario quien crea el objeto es el propietario y por defecto tiene todos los permisos, incluido el de dar permisos a otro usuario denominado permiso de concesión. El usuario propietario puede asignar los permisos que desee a otro usuario incluido el de concesión, si este usuario tiene el permiso de concesión puede asignar permisos a otro usuario.

REVOKE – Para quitar permisos.

TLC

COMMIT – Finaliza la transacción y realiza los cambios hechos durante la transacción. Las transacciones bloqueadas sobre la tabla quedan liberadas.

ROLLBACK – Rechaza la transacción y no aplica cambios, volviendo al estado antes de iniciarse la transacción.

SAVEPOINT– Crea un punto en la transacción que se pueda volver mediante ROLLBACK.

SET TRANSACTION – Inicializa una transacción en la base de datos, indicando si quiere que sea de solo lectura o lectura/escritura.

Restricciones (Constraints)

Restricción de Clave principal (PK)	Constraint PK
Restricción de referencia	Constraint FK
Restricción de integridad	Constraint CHECK Constraint NOT NULL Constraint UNIQUE
Restricción por trigger	TRIGGER

Primary Key (PK) Foreign Key (FK)

Una llave primaria o PRIMARY KEY es una columna o un grupo de columnas que identifica de forma exclusiva cada fila de una tabla.
Restricciones:

- Debe tener un valor siempre.
- Único e irrepetible.

Una FOREIGN KEY es una clave (campo de una columna) que sirve para relacionar dos tablas. El campo FOREIGN KEY se relaciona o vincula con la PRIMARY KEY de otra tabla de la bbdd

La tabla secundaria es la que contiene la FOREIGN KEY y la tabla principal contiene la PRIMARY KEY.

La FOREIGN KEY es una restricción que no permite que se agreguen o inserten datos no válidos en la columna de foreign key, ya que los valores que se van a insertar deben ser valores que se encuentren o ya estén en la tabla con la que se quiere relacionar. (Valores que coincidan con la PK de la tabla Padre.

Restricciones de integridad

Ayudan a mantener la consistencia e integridad de la base de datos, es decir verifica que los datos sean verídicos y correctos.

NOT NULL

La restricción **NOT NULL** sirve para especificar que una columna no acepta el valor NULL, es decir, que esa columna siempre tiene que tener algún valor, no puede estar vacía.

NULL

Es la ausencia del valor en un registro y columna.

Restricciones de integridad

Check

Es una restricción que utiliza una expresión booleana para validar que el valor a asignar a un atributo sea correcto o bien cumpla con las reglas preestablecidas.

```
CHECK (edad >= 17 AND edad <= 120)
```

UNIQUE

Es una restricción para indicar que los valores que puede tener un campo deben ser únicos.
No pueden existir dos registros con el mismo valor para las columnas con esta restricción.
A diferencia de una PK, los elementos restringidos con UNIQUE, pueden ser nulos.

Tipos de datos

- Cadenas
  - Cadenas de caracteres
    - De longitud fija
    - De longitud variable
- Cadenas binarias
- Numéricos
  - Exactos
  - Aproximados
- Tiempos y fechas
- Booleanos

Cadenas de caracteres

CHAR( )

Longitud fija (puede contener letras, números y caracteres especiales)
Al usar CHAR decimos que la memoria será reservada para la cantidad de bytes que estamos estableciendo. Por ejemplo, si declaramos un CHAR de longitud 20, y guardamos un “Hola” sólo se ocuparán 4 bytes y los otros 16 se llenarán con espacios.

Al recibir los datos, se les quitarán esos espacios. Pero siempre se ocupara la longitud que definió, se llene de datos o no.

CHAR es más rápido que VARCHAR en cuanto a rendimiento, ya que se sabe de antemano cuál es la longitud de los datos, no como VARCHAR que tiene una longitud variable.

El límite de CHAR es 255 bytes.

Cadenas de caracteres

VARCHAR( )

Longitud variable (puede contener letras, números y caracteres especiales)

Al declarar un tipo de dato como VARCHAR estamos indicando que será como CHAR pero con una **longitud variable**.

Lo que significa que sólo reservará memoria para los datos que pongamos, así aunque declaremos un VARCHAR de 1000 y pongamos un “hola”, sólo guardará los 4 caracteres y utilizará bytes extra que guardaran la longitud de los datos. Es decir se ocupara 5 bytes en total, en este ejemplo.

El límite para VARCHAR es **65,535** bytes.

¿CHAR VS VARCHAR?

- Si vamos a almacenar datos cuyo tamaño sabemos de antemano, es mejor usar CHAR.

- Si definimos una longitud fija y no la llenamos estaremos desperdiciando espacio en disco duro. Para estos casos es mejor utilizar VARCHAR( ).

Cadenas de caracteres de mayor extensión

CLOB( )
(Character Large Object)

Este tipo de dato nos ayudara a almacenar grandes cantidades de caracteres.
Como libros, textos, artículos etc.

No es necesario escribir la longitud máxima.

Cadenas binarias

BLOB( )
(Binary Large Object)

Este tipo de dato permite almacenar archivos binarios: Imágenes, videos, música, fotos, documentos, etc.

Al igual que los datos tipo CLOB, no requiere escribir la longitud máxima.

Numéricos

Exactos

- Int (-2000000000 , 2000000000)
- Integer
- Smallint (-32000 , 32000)
- Tinyint (0,255)
- Bigint (-9X10^18 , 9X10^18)

- Decimal
- Numeric

(Parte decimal FINITA)

- Money

Numéricos

Aproximados

Float ( )
Real ( )

Se utilizan cuando no se conoce con exactitud los valores de una columna.

Tiempo, Fechas

date

Representa una fecha en formato(Año, Mes, Día)

time

Representa tiempo en formato HH:MM:SS

timestamp

Formato: Año, Mes, Día, Hora, Minuto, Segundo, Milisegundo

**De forma interna una fecha se guarda como un dato numérico.

Booleano

Bool (True, False)

Boolean

Number(0,1)