

**CURSO DESARROLLO DE APLICACIONES MÓVILES ANDROID TRAINEE**

## **Módulo 2.**

# **Lenguaje de consultas a una base de datos (4 unidades)**

## **Temario**

# **Unidad 1**

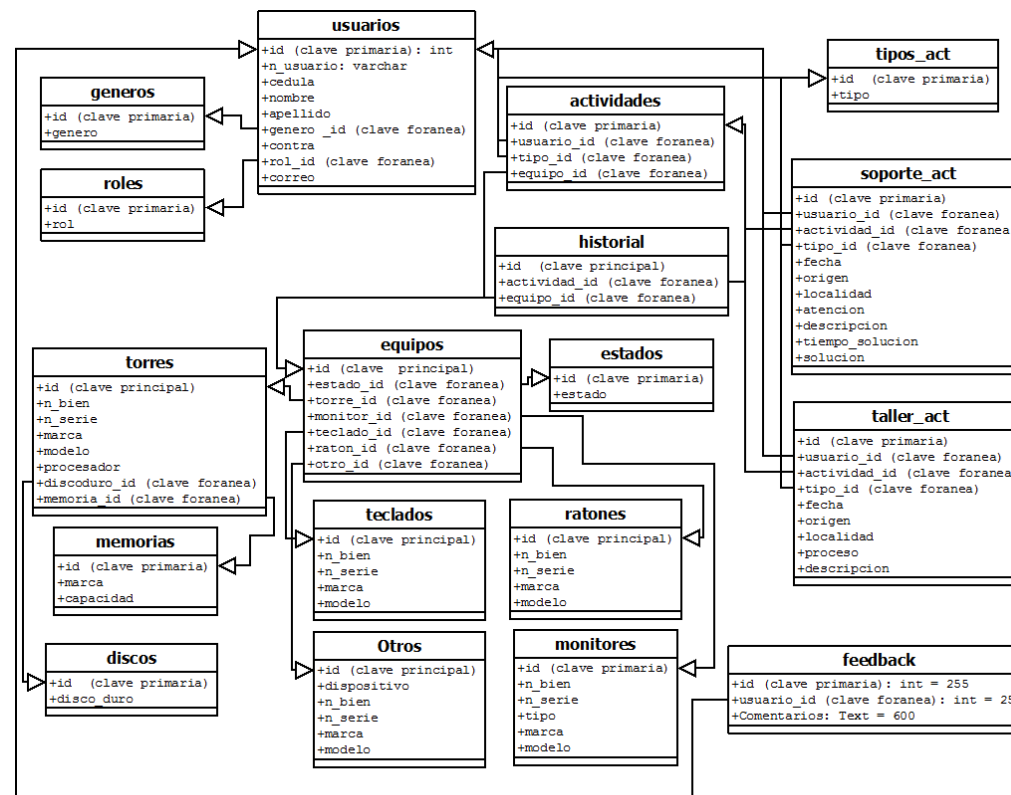
- a) Las bases de datos relacionales**
- b) Consultando información de una tabla**
- c) Consultando información relacionada en varias tablas**

## Unidad 1

# Las bases de datos relacionales

Una base de datos relacional es una base de datos que cumple con el modelo relacional, el cual es el modelo más utilizado en la actualidad para implementar bases de datos ya planificadas

Permiten establecer interconexiones (relaciones) entre los datos (que están guardados en tablas), y a través de dichas conexiones relacionar los datos de ambas tablas.



# Unidad 1

- Una base de datos relacional se compone de varias tablas o relaciones.
- No pueden existir dos tablas con el mismo nombre ni registro.
- Cada tabla es a su vez un conjunto de registros (filas y columnas).
- La relación entre una tabla padre y un hijo se lleva a cabo por medio de las llaves primarias y foráneas.
- Las llaves primarias son la llaves principal de un registro dentro de una tabla y éstas deben cumplir con la integridad de datos.
- Las llaves foráneas se colocan en la tabla hija, contienen el mismo valor que la llaves primaria del registro padre; por medio de éstas se hacen las relaciones.

## Unidad 1

# Consultando información de una tabla

Una consulta sirve para extraer información de una base de datos. Permite manipular datos; agregar, eliminar y cambiar. Nos referimos a este código como el lenguaje de consulta SQL:

Mediante el uso de SQL, se puede:

- Consultar, actualizar y reorganizar datos.
- Crear y modificar la estructura de un sistema de base de datos.
- Controlar el acceso a sus datos.

Las operaciones básicas de manipulación de datos que podemos realizar con SQL se les denomina **operaciones CRUD** (de **C**reate, **R**ead, **U**ppdate and **D**elelete)

Hay cuatro instrucciones para realizar estas tareas:

- **INSERT**: Inserta filas en una tabla. Se corresponde con la “C” de CRUD.
- **SELECT**: muestra información sobre los datos almacenados en la base de datos. Dicha información puede pertenecer a una o varias tablas. Es la “R”.
- **UPDATE**: Actualiza información de una tabla. Es, obviamente, la “U”.
- **DELETE**: Borra filas de una tabla. Se corresponde con la “D”.

## Unidad 1

# Consultando información relacionada en varias tablas

Los diseños de la base de datos están estrechamente relacionados con las relaciones de la base de datos, la asociación entre dos columnas en una o más tablas. Los tipos de relaciones no deben confundirse con los tipos de datos en SQL, siendo conceptos distintos.

Se crea un enlace entre dos tablas donde la clave principal de una tabla se asocia con la clave externa de otra tabla utilizando las relaciones de la base de datos.

# Material complementario de la unidad

Link a video relacionado

***“Base de Datos #1 | Conceptos Básicos”***

<https://youtu.be/yoeV4Ex8C8U>

Link a lectura complementaria

**Conceptos básicos de bases de datos**

<https://doc.4d.com/4Dv15/4D/15.6/Conceptos-basicos-de-bases-de-datos.300-3836654.es.html>



**Temario**

# **Unidad 2**

**a) Sentencias para la manipulación de datos**

**b) Transaccionalidad en las operaciones**

## Unidad 2

# Sentencias para la manipulación de datos

### Data Manipulación Lenguaje (DML)

Las sentencias de lenguaje de manipulación de datos (DML) son utilizadas para gestionar datos dentro de los schemas. Algunos ejemplos:

- SELECT - para obtener datos de una base de datos.
- INSERT - para insertar datos a una tabla.
- UPDATE - para modificar datos existentes dentro de una tabla.
- DELETE - elimina todos los registros de la tabla; no borra los espacios asignados a los registros.

## Actualizando la información de una tabla

Update es la instrucción del lenguaje SQL que nos sirve para modificar los registros de una tabla.

**UPDATE** nombre\_tabla

**SET** nombre\_campo1 = valor\_campo1, nombre\_campo2 = valor\_campo2,...

**WHERE** condiciones\_de\_selección

**\*NUNCA OLVIDAR EL WHERE\*** Si no hay condiciones\_de\_selección actualizaremos todos los registros de la tabla

## Borrando información de una tabla

La instrucción DELETE permite eliminar uno o múltiples registros. Incluso todos los registros de una tabla, dejándola vacía.

La sintaxis es de este tipo:

**DELETE** [**FROM**] nombre\_tabla

**WHERE** condiciones\_de\_selección

**\*NUNCA OLVIDAR EL WHERE\*** Si no hay condiciones\_de\_selección actualizaremos todos los registros de la tabla

## Ingresando información a una tabla

INSERT es una instrucción que permite la inserción de filas en una tabla determinada con valores recibidos o generados.

La sintaxis es de este tipo:

```
INSERT INTO nombre_tabla (nombre_campo1, nombre_campo2)  
VALUES (valor_campo1, valor_campo2);
```

## Unidad 2

# Transaccionalidad en las operaciones

Las **transacciones en SQL** son unidades o secuencias de trabajo realizadas de forma ordenada y separada en una base de datos. Normalmente representan cualquier cambio en la base de datos, y tienen dos objetivos principales:

- **Proporcionar** secuencias de trabajo fiables **que permitan poder** recuperarse fácilmente ante errores **y mantener una** base de datos consistente incluso frente a fallos del sistema.
- **Proporcionar** aislamiento entre programas accediendo a la vez **a la base de datos**.

## Propiedades de las transacciones

Las transacciones siguen **cuatro propiedades básicas**, bajo el acrónimo **ACID** (Atomicity, Consistency, Isolation, Dura-

### Garantías de ACID

- **Atomicidad:**
  - La ejecución de cada transacción es atómica:
    - Se realizan todas las acciones o no se realiza ninguna
- **Consistencia:**
  - Cada transacción debe preservar la integridad
    - La base de datos satisfacen todas las restricciones después de una transacción
- **Aislamiento (Isolation):**
  - Una transacción no puede afectar otra
- **Durabilidad:**
  - Una vez que haya un **COMMIT**, la base de datos debe persistir los cambios

bility):

- **Atomicidad:** aseguran que todas las operaciones dentro de la secuencia de trabajo **se completen satisfactoriamente**. Si no es así, la transacción se abandona en el punto del error y las operaciones previas retroceden a su estado inicial.
- **Consistencia:** aseguran que la base de datos **cambie estados** en una transacción exitosa.
- Aislamiento: permiten que las operaciones sean aisladas y transparentes unas de otras.
- **Durabilidad:** aseguran que el resultado o efecto de una transacción completada **permanezca en caso de error del sistema**.

# Material complementario de la unidad

Link a video relacionado

**” SQL Transacciones”**

<https://youtu.be/lrVY1Ctozl8>

Link a lectura complementaria

**”Transacciones en SQL”**

<https://diego.com.es/transacciones-en-sql>



**Temario**

# **Unidad 3**

**a) Sentencias para la definición de tablas**

**b) Creando un modelo de datos con integridad referencial**

## Unidad 3

# Sentencias para la definición de tablas

Un lenguaje de definición de datos (Data Definition Language, DDL por sus siglas en inglés) es un lenguaje proporcionado por el sistema de gestión de base de datos que permite a los usuarios de la misma llevar a cabo las tareas de definición de las estructuras que almacenarán los datos así como de los procedimientos o funciones que permitan consultarlos.

**CREATE TABLE** crea una tabla con el nombre dado. Se debe tener el permiso CREATE para la tabla.

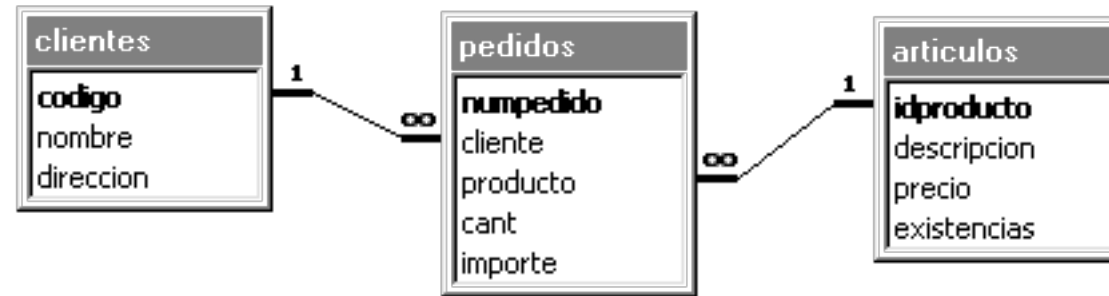
Por defecto, la tabla se crea en la base de datos actual. Ocurre un error si la tabla existe, si no hay base de datos actual o si la base de datos no existe.

La sintaxis es de este tipo:

```
CREATE TABLE "nombre_tabla"  
("columna 1" "tipo_de_datos_para_columna_1",  
"columna 2" "tipo_de_datos_para_columna_2",  
... );
```

# Creando un modelo de datos con integridad referencial

La integridad referencial es un sistema de reglas que utilizan la mayoría de las bases de datos relacionales para asegurarse que los registros de tablas relacionadas son válidos y que no se borren o cambien datos relacionados de forma accidental produciendo errores de integridad.



# Material complementario de la unidad

Link a video relacionado

***“Modelo Relacional e Integridad Referencial”***

<https://youtu.be/kNLUloAjWKg>

Link a lectura complementaria

***“Integridad Referencial”***

[https://es.wikipedia.org/wiki/Integridad\\_referencial](https://es.wikipedia.org/wiki/Integridad_referencial)

**Temario**

# **Unidad 4**

**a)El modelo Entidad-Relación**

**b)El modelo Relacional**

## Unidad 4

# El modelo Entidad-Relación

Es el modelo conceptual más utilizado para el diseño conceptual de bases de datos. Permite diseñar esquemas que posteriormente debemos de implementar en un gestor de BBDD (bases de datos). Este modelo se representa a través de diagramas y está formado por distintos elementos.

## Entidades

Tipo de objeto sobre el que se recoge información: cosa, persona, concepto abstracto o suceso (coches, casas, empleados, clientes, empresas, oficios, diseños de productos, conciertos, excursiones, etc.).

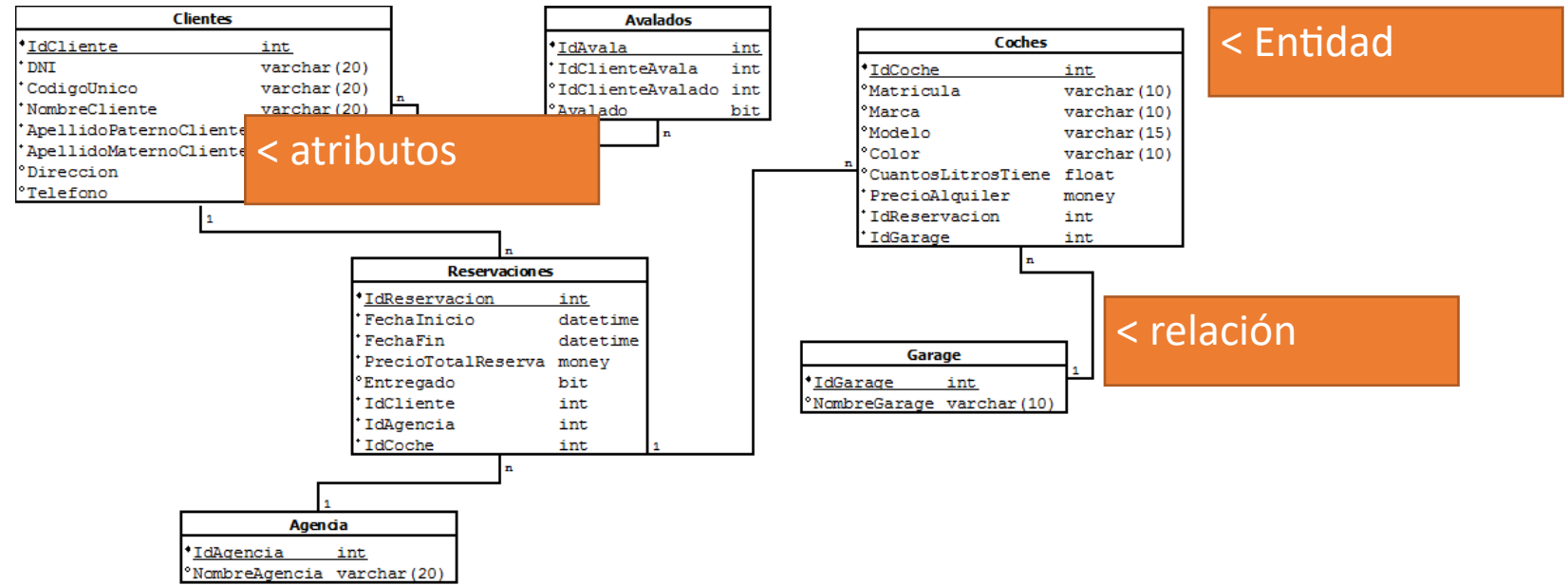
- Las entidades se representan gráficamente mediante rectángulos y su nombre aparece en el interior.
- Un nombre de entidad sólo puede aparecer una vez en el esquema.

## Atributos

Los atributos definen o identifican las características de entidad. Cada entidad contiene distintos atributos, que dan información sobre esta entidad. Estos atributos pueden ser de distintos tipos (numéricos, texto, fecha...).

## Relación entre entidades

La relación o interrelación es un elemento del modelo Entidad/Relación que permite relacionar datos entre sí. En una relación se asocia un elemento de una entidad con otro de otra entidad.





## Unidad 4

# El modelo Relacional

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

Fueron creados en 1970 por Edgar Frank Codd, empleado de IBM en San José (California). Su funcionamiento es realmente sencillo. Se basan en la teoría de conjuntos y la lógica de predicados.

Se representan con tablas de datos, de manera que las filas son los diferentes casos y las columnas los campos a estudiar. Es muy utilizado para administrar datos de forma dinámica.

# Términos del modelo relacional

Existen una serie de términos que se corresponden con expresiones informales. Conviene conocerlos para así familiarizarse con ellos. En la práctica suelen usarse las expresiones sencillas, más fáciles de entender.

- La **relación**, que es el término formal, tiene en la tabla su equivalente informal.
- La **tupla** no es más que un registro que se representa en las filas de la tabla y el atributo es una columna o campo.
- La **cardinalidad** se refiere al número de filas o registros y el grado es el número de columnas o campos.
- Por último, la **clave primaria** es un identificador único de cada caso.

## Funcionamiento del modelo relacional

Aunque lo veremos en el ejemplo final, vamos a mostrar cómo funciona. Hay que tener en cuenta que las propias tablas ya establecen relaciones entre los datos, de ahí el nombre. En realidad se rige por unas normas sencillas, que son las siguientes:

- Las tablas son el centro del modelo y los datos deben ser representados en ellas. También se usan tablas cuando se calculan resultados de otras.
- El orden de cada columna viene determinado por el tipo de consulta que se realice. Por tanto, no es necesario un orden inicial, ya que cada relación la conforma un conjunto único de datos.
- Como ya hemos mencionado, las filas son datos o casos, las columnas campos o variables. Por otro lado, cada celda es un registro que tiene dos dimensiones.
- Es imprescindible disponer de un identificador único (clave primaria) de cada registro. Este permite establecer relaciones entre dos o más tablas, usándolo como una clave externa.

## Ventajas e desventajas del modelo relacional

Este modelo relacional, como todo, tiene ventajas e inconvenientes.

Entre las ventajas podemos destacar las siguientes;

- Por un lado, tiene procesos que evitan la duplicidad de los datos.
- Por otro, garantiza la integridad referencial eliminando todo lo relacionado con un registro cuando sea necesario.
- Además, permite la normalización.

Por su parte, entre las desventajas o inconvenientes tenemos los siguientes:

- El primero es que tiene limitaciones en las representaciones gráficas o en los sistemas de geoposicionamiento.
- El segundo es que los bloques de texto no se procesan de forma eficiente.

# Material complementario de la unidad

Link a video relacionado

**“Base de Datos #2 | Modelo relacional”**

<https://youtu.be/MRmmPJld5-k>

Link a lectura complementaria

***“El Modelo de Datos Entidad-Relación”***

[http://basededatos.umh.es/e\\_r.htm](http://basededatos.umh.es/e_r.htm)

***“Fundamentos de diseño de bases de datos”***

<https://elvex.ugr.es/idbis/db/docs/intro/D%20Modelo%20relacional.pdf>