

## 2. MODELO RELACIONAL

C.I.F.P A CARBALLEIRA

Miguel Ángel Domínguez Llorente  
2025-2026

Base de datos

## Índice

1. Modelo de datos .....	2
2. Terminología del modelo relacional.....	4
2.1 – Relación o tabla. Tuplas. Dominios.....	4
2.2 Grado. Cardinalidad.....	4
2.4 Sinónimos.....	4
3. Relaciones de una tabla.....	4
3.1 Tipos de relaciones (tabla) .....	4
4. Tipos de datos .....	5
5. Claves.....	6
6. Índices. Características. ....	7
7. El valor NULL. Operadores con este valor .....	8
9. Usuario, Roles y Privilegios.....	9
10 SQL.....	10
10.1 Elementos del lenguaje. Normas de escritura .....	11
11. Lenguaje de descripción de datos (DDL) .....	12
11.1.- Creación de bases de datos. Objetos de la base de datos. ....	13
11.3 Restricciones .....	14
11.3.3 Restricciones PRIMARY KEY.....	15

## 1. Modelo de datos

Los modelos de datos son aquellos que nos permite describir elementos que intervienen en una realidad o en un problema dado y la forma en que se relacionan dichos elementos entre sí

Son como un conjunto de métodos y reglas que indican como se han de almacenar la información y como se han de manipular los datos. Estos modelos de datos se implementan con un lenguaje utilizado para la descripción de una base de datos.

Con un lenguaje nos permite: estructuras de datos, restricciones de integridad, las operaciones de manipulación de los datos (CRUD).

Y ese lenguaje es SQL.

SQL\* es un lenguaje de dominio específico utilizado en (programación ¿?), utilizado para administrar y recuperar información de sistemas de gestión de base de datos relacionales

Este lenguaje tiene dos sublenguajes:

Lenguajes de definición de datos o DDL (Data Definition Language)

cuya función es describir de una forma abstracta, las estructuras de datos y las restricciones de integridad.

Lenguajes de manipulación de datos o DML (Data Manipulation Language) sirve para demostrar las operaciones de manipulación de los datos.

Existen tres fases de modelado en el diseño de una base de datos basada en el nivel de abstracción.

- **Modelo de datos conceptual:** Es una representación de lo que será la base de datos basado en tablas sin añadir datos completos ni propiedades de los atributos que tendrá la tabla.
- **Modelo Lógico:** Determina los criterios de almacenamiento y los operadores de manipulación de datos dentro de un tipo de entorno informático.
- **Modelo físico:** Es la implementación física del modelo anterior. Son estructuras de datos a bajo nivel. Implementado dentro de un sistema de gestor de datos comercial.

## 2. Terminología del modelo relacional.

### 2.1 – Relación o tabla. Tuplas. Dominios.

propiedades de la tabla

Atributos

Tuplas

Definición de dominio

### 2.2 Grado. Cardinalidad

Grado: Tamaño de una tabla en base del número de atributos

Cardinalidad: Número de filas / tuplas de una relación o tabla

### 2.4 Sinónimos

Definiciones de lo que vemos en el temario

## 3. Relaciones de una tabla

[salto]

### 3.1 Tipos de relaciones (tabla)

[salto]

## 4. Tipos de datos

- Texto: almacena cadenas (conjunto) de caracteres: números con los que no vamos a realizar operaciones matemáticas, letras o símbolos.
  - Numérico: almacena números. Si dudamos entre numérico o texto tendremos en cuenta si vamos a realizar operaciones matemáticas con ellos, en cuyo caso será numérico.
  - Fecha/hora: almacena fechas y horas.
  - Sí/No: almacena datos que solo tienen dos posibilidades (verdadero/falso).
- 
- Autonumérico: se puede considerar un subtipo de Numérico ya que almacena valores numéricos secuenciales
    - que el SGBD incrementa de modo automático al añadir un registro (fila).
    - Memo: almacena texto largo (mayor que un tipo texto).
    - Moneda: pero con una característica especial, y es que los valores representan cantidades de dinero.
  - Objeto OLE: almacena gráficos, imágenes o textos creados por otras aplicaciones.

## 5. Claves

Útil para poder diferenciar valores de otros valores en una misma tabla

Clave primaria - Es un valor que solo va a existir en una única tabla que no se puede duplicar sus filas

Clave Candidata – Conjunto de atributos que identifiquen de manera única cada fila / tupla de la relación (tabla) para que esos valores no se repitan en ninguna otra fila de la tabla, cada tabla debe tener una clave candidata puede haber más de una.

Una clave candidata tiene los siguientes requisitos:

**Unicidad:** no puede haber dos tuplas (filas) con los mismos valores para esos atributos.

**Irreductibilidad:** si se elimina alguno de los atributos deja de ser única.

Para reconocer una clave candidata es clave identificar el significado real de los atributos para evitar los duplicados

Clave alternativa – son claves candidatas que no son escogidas como clave primaria.

### 5.2 Clave externa, ajena o secundaria

## 6. Índices. Características.

Es una estructura de datos que permite acceder a diferentes filas de una misma tabla a través de un campo o campos. Gracias a esta idea podemos acceder a estos datos de una manera más rápida. Muy útil cuando hacemos consultas de manera frecuente.

Los cambios en los datos de las tablas con CRUD son realizados de forma automática con los índices con transparencia total

Los índice son independientes a la lógica y físicamente de los datos , es por eso que pueden ser creados y eliminados en cualquier momentos sin afectar a otra tablas (relaciones)

No hay un límite para indexar

Si se elimina un índice , el acceso a los datos será más lento

El gestor de base de datos es quien utiliza índices para las claves primarias y ajenas, en el caso de la claves primarias serán índices únicos.

## 7. El valor NULL. Operadores con este valor

Es la indicación de ausencia de datos en una tupla , no es lo mismo valor nulo que cero ni espacio en blanco

## 8. Vistas

Son tablas virtuales cuyas filas y columnas se obtienen a partir de una o varias tablas que construyen nuestro modelo

Motivos por las cuales queremos crear vistas:

- Seguridad – Para acceder a una parte de la tabla o de la base de datos para los usuarios
- Comodidad -

## 9. Usuario, Roles y Privilegios

modos de acceso

usuario

administrador

Privilegios

De sistemas

Sobre objetos

--

Rol – agrupaciones de permisos de sistema y objetos

## 10 SQL

### Definición

Es un lenguaje de dominio específico utilizado en programación y diseñado para administrar y recuperar información de sistemas de gestión de base de datos relacionales. Es el lenguaje fundamental de los gestores de bases de datos relacionales.

### Características o funciones

Es un lenguaje declarativo, potente y versátil.

es fácil de aprender por su cercanía al lenguaje natural.

Es un lenguaje de cuarta generación.

aparte de ser un lenguaje de consultas también puede hacer:

definir estructuras de los datos DDL

la manipulación de los datos DML

la especificación de las conexiones seguras DCL

Formas de trabajar con sql:

Embebido – Las sentencias se trabajan dentro de un programa escrito en otros lenguaje como java , PHP etc.

Interpretado – Las sentencias se trabajan desde un entorno gráfico como el cmd de un sistema operativo o un SGDB

## 10.1 Elementos del lenguaje. Normas de escritura

Conceptos de escritura:

Comandos

Cláusulas

Operadores

Funciones

Literales

Normas de escritura

- Todas las instrucciones terminan con un signo de punto y coma.
- No se distingue entre mayúsculas y minúsculas.
- Cualquier comando puede ser partido con saltos de línea o espacios para facilitar su lectura y comprensión.
- Los comentarios comienzan por /\* y terminan con \*/ (excepto en algunos SGBD).

## 11. Lenguaje de descripción de datos (DDL)

Antes de poder almacenar y recuperar información debemos definir las estructuras para dar paso a las sentencias DDL

Este lenguaje es imprescindible para crear, modificar, y eliminar objetos de la base de datos

Las instrucciones DDL generan acciones que no se pueden deshacer, por eso es conveniente usarlas con precaución y tener copias de seguridad cuando manipulamos la base de datos.

## 11.1.- Creación de bases de datos. Objetos de la base de datos.

### Create Database

```
CREATE DATABASE NombredemiBasedeDatos;
```

Según los estándares, una **base de datos** es un conjunto de objetos que nos servirán para gestionar los datos. Para poder guardar los datos debemos definir los objetos que vamos a agrupar en la base de datos, estos objetos son las tablas, las vistas e índices

Antes de crear la **tabla** debemos tener en cuenta la siguiente información que viene desde el modelo lógico:

- Qué nombre le vamos a dar a la tabla.
- Qué nombre le vamos a dar a cada una de las columnas.
- Qué tipo y tamaño de datos vamos a almacenar en cada columna.
- Qué restricciones tenemos sobre los datos.
- Alguna otra información adicional que necesitemos.

Para el nombre de las tablas debemos cumplir las siguientes reglas:

- No podemos tener nombres de tablas duplicados en un mismo esquema (usuario).
- Deben comenzar por un carácter alfabético.
- Su longitud máxima es de 30 caracteres.
- Solo se permiten letras del alfabeto inglés, dígitos o el signo de guión bajo.
- No puede coincidir con las palabras reservadas de SQL (por ejemplo, no podemos llamar a una tabla WHERE).
- No se distingue entre mayúsculas y minúsculas.
- En el caso de que el nombre tenga espacios en blanco o caracteres nacionales (permitido sólo en algunas bases de datos), entonces se suele entrecerrar con comillas dobles. En el estándar SQL99 (respetado por Oracle) se pueden utilizar comillas dobles al poner el nombre de la tabla a fin de hacerla sensible a las mayúsculas (se diferenciará entre "USUARIOS" y "Usuarios").

Sintaxis básica del comando que permite crear una tabla:

```
CREATE TABLE [esquema.] nombredeTabla (columna1 Tipo_Dato,columna2 Tipo_Dato, ...
columnaN Tipo_Dato
```

Tabla USUARIOS con un solo campo de tipo VARCHAR:

```
CREATE TABLE USUARIOS (Nombre VARCHAR(25));
```

## 11.3 Restricciones

Son condiciones que deben cumplir obligatoriamente

Los tipos son los siguiente

### 11.3.1 Restricción NOT NULL

Con esta restricción obligamos que el campo o atributo de la tabla a que no puede estar nulo

Ejemplo

```
CREATE TABLE USUARIOS ( F_Nacimiento DATE CONSTRAINT Usu_Fnac_NN NOT NULL);
```

### 11.3.2 Restricción UNIQUE

Es una restricción que nos permite el valor en varias columnas, También podemos poner esta restricción a varios campos a la vez

Ejemplo

```
CREATE TABLE USUARIOS ( Login VARCHAR2 (25) CONSTRAINT Usu_Log_UK UNIQUE);
```

### 11.3.3 Restricciones PRIMARY KEY

En el modelo relacional , las tablas deben tener una clave primaria, solo puede haber una por tabla pero puede estar formada por varios campos

Esta restricción hace que los campos sean NOT NULL y UNIQUE

### Ejemplo

```
CREATE TABLE USUARIOS ( Login VARCHAR2 (25) PRIMARY KEY);
```

Si hay varios campos se hace de la siguiente

```
CREATE TABLE USUARIOS ( Nombre VARCHAR2 (25), Apellidos VARCHAR2 (30),
F_Nacimiento DATE, CONSTRAINT Usu_PK PRIMARY KEY(Nombre, Apellidos,
F_Nacimiento));
```

### 11.3.4 Restricciones REFERENCES, FOREING KEY

Recordamos que la clave foránea era la relación con la clave primaria

### Ejemplo

```
CREATE TABLE USUARIOS ( Cod_Partida NUMBER(8) CONSTRAINT Cod_Part_FK
REFERENCES PARTIDAS(Cod_Partida));
```

### 11.3.5 Restricción DEFAULT Y VALIDACIÓN

Asigna valores por defecto desde la sentencia

### ejemplo

```
CREATE TABLE USUARIOS ( Pais VARCHAR2(20) DEFAULT 'España' );
```

#### 11.4 Eliminación de tablas

Permite eliminar tablas cuando ya no sirven , ahorrando espacio

```
DROP TABLE NombreTabla [CASCADE CONSTRAINTS];
```

#### 11.5 Modificación de tablas

Cambiar el nombre de una tabla

```
RENAME NombreViejo TO NombreNuevo;
```

Alterar una tabla para añadir columnas al final de una tabla

```
ALTER TABLE NombreTabla ADD ( ColumnaNueva1 Tipo_Datos [Propiedades] [, ColumnaNueva2 Tipo_Datos [Propiedades] ... ]);
```

Eliminar columnas de una tabla

```
ALTER TABLE NombreTabla DROP COLUMN (Columna1 [, Columna2, ...]);
```

Modificar una columna de una tabla

```
ALTER TABLE NombreTabla MODIFY (Columna1 TipoDatos [propiedades] [, columna2 TipoDatos [propiedades] ...] );
```

Si queremos renombrar columnas de una tabla:

```
ALTER TABLE NombreTabla RENAME COLUMN NombreAntiguo TO NombreNuevo;
```

### 11.5.1 Modificación de tablas (II)

Para borrar restricciones

```
ALTER TABLA NombreTabla DROP CONSTRAINT NombreRestriccion;
```

Para cambiar el nombre de una restricción

```
ALTER TABLE NombreTabla RENAME CONSTRAINT NombreViejo TO NombreNuevo;
```

Si queremos activar o desactivar restricciones:

```
ALTER TABLE NombreTabla DISABLE CONSTRAINT NombreRestriccion [CASCADE];
```

La opción **CASCADE** desactiva las restricciones que depende de esta ,para volver activar de nuevo la restricción

```
ALTER TABLE NombreTabla ENABLE CONSTRAINT NombreRestriccion [CASCADE];
```

### 11.6 Creación y eliminación de índices

Un índice nos ayuda a localizar más rápida la información que contiene en una tabla

```
CREATE INDEX NombreIndice ON NombreTabla (Columna1 [, Columna2 ...]);
```

Advertencia sobre el uso de índices:

- No utilizar los índices sobre campos de tablas pequeñas o que se actualizan mucho
- No utilizar los índices sobre campos que usen con poca frecuencia
- Una mala elección ocasiona ineficacia y tiempo tardío

Eliminar un índice

```
DROP INDEX NombreIndice;
```

La mayoría de los índices se crean de manera implícita cuando ponemos las restricciones PRIMARY KEY, FOREIGN KEY o UNIQUE.

## 12. Lenguajes de control de datos (DCL)

Las claves de acceso se establecen cuando se crea el usuario y pueden ser modificables por el administrador o el propietario de dicha clave. Las bases de datos almacena de forma encriptada las claves en una tabla del diccionario llamada DBA\_USERS

Así se crea los usuarios

```
CREATE USER NombreUsuario IDENTIFIED BY ClaveAcceso [DEFAULT TABLESPACE  
tablespace] [TEMPORARY TABLESPACE tablespace] [QUOTA int {K | M} ON tablespace]  
[QUOTA UNLIMITED ON tablespace] [PROFILE perfil];
```

cada una de esta se :

Practica de forma limitada la creación de usuario que no tenga derecho ni a guardar datos ni crear objetos ( más adelante le daremos permisos )

```
CREATE USER UsuarioLimitado IDENTIFIED BY passworddemiusuariolimitado ;
```

Modificación del usuario mediante el comando **Alter User** :

```
ALTER USER NombreUsuario IDENTIFIED BY clave_acceso [DEFAULT TABLESPACE tablespace]  
[TEMPORARY TABLESPACE tablespace] [QUOTA int {K | M} ON tablespace] [QUOTA  
UNLIMITED ON tablespace] [PROFILE perfil];
```

Un usuario sin privilegios de administrador, únicamente solo puede cambiar su clave, la contraseña

```
DROP USER NombreUsuario [CASCADE];
```

La opción CASCADE borra todos los objetos del usuario antes de borrarlo. Sin esta opción no nos dejaría eliminar al usuario si éste tuviera tablas creadas.

### 12.1 Permisos (I)

Ningún usuario puede llevar a cabo una operación si antes no se le ha concedido el permiso para ello. Por ello vamos a indicar como se hace

Para poder acceder a los objetos de una clase de datos, necesitamos permisos, estos pueden agruparse por **roles**, lo que ayuda a la administración. Los roles pueden activarse, desactivarse o protegerse con una clave. Mediante los roles además podemos gestionar los comandos que puede utilizar los usuarios. Un permiso se puede asignar a un usuario o a un rol.

Un permiso o privilegio se concede con el comando GRANT (conceder)

Si se dan privilegios sobre los objetos

```
GRANT {privilegio_objeto [, privilegio_objeto]...|ALL|[PRIVILEGES]} ON [usuario.]objeto  
FROM {usuario1|rol1|PUBLIC} [, {usuario2|rol2|PUBLIC} ... [WITH GRANT OPTION];
```

Descripción:

Los privilegios de sistema son los que dan derecho a ejecutar comandos SQL o acciones sobre objetos de un tipo especificado. Existen gran cantidad de privilegios distintos

```
GRANT {Privilegio1 | rol1 } [, privilegio2 | rol2}, ...]  
TO {usuario1 | rol1| PUBLIC} [, usuario2 | rol2 | PUBLIC} ... ]  
[WITH ADMIN OPTION];
```

Descripción:

### 12.1.1 Permisos (II)

Remover permisos con REVOKE

Sobre objetos

```
REVOKE {privilegio_objeto [, privilegio_objeto]...|ALL|[PRIVILEGES]} ON [usuario.]objeto  
FROM {usuario|rol|PUBLIC} [,,{usuario|rol|PUBLIC}] ...;
```

Sobre sistemas o roles a usuarios:

```
REVOKE {privilegio_stma | rol} [, {privilegio_stma | rol}]...|ALL|[PRIVILEGES]} ON  
[usuario.]objeto FROM {usuario|rol|PUBLIC} [,,{usuario|rol|PUBLIC}] ...;
```

Y ya está

## Anexo I Elementos de Lenguaje SQL

Comandos:

Comandos DDL. Lenguaje de definición de datos.

Comando	Descripción
Create	Se utiliza para crear nuevas , tablas e índices
Drop	Se utiliza para eliminar tablas e índices
Alter	Se utiliza para modificar tablas

Comando DML. Lenguaje de manipulación de datos

Comando	Descripción
Select	Se utiliza para consultar filas
Insert	Se utiliza para cargar datos desde una única operación
Update	Se utiliza para modifica valores de campos y filas
Delete	Se utiliza para eliminar filas de una tabla

Comandos DCL. Lenguaje de control de datos

Comandos	Descripción
GRANT	Permite dar permisos a uno o varios usuarios o roles para realizar tareas determinadas
REVOKE	Permite eliminar permisos que previamente sean concedido con GRANT

### Cláusulas:

Llamado también condiciones o criterios, permiten modificar el funcionamiento de un comando para una ejecución específica

Cláusulas	Descripción
From	Se utiliza para especificar la tabla de la que se van a seleccionar las filas
Where	Se utiliza para especificar las condiciones que deben reunir las filas que se van a seleccionar
Group by	Se utiliza para separar las filas seleccionadas en grupos específicos.
Having	Se utiliza para expresar la condición que debe satisfacer cada grupo
Order by	Se utiliza para ordenar las filas seleccionadas de acuerdo a un orden específico

### Operadores

Operadores	Descripción
AND	Evalúa dos condiciones y devuelve un valor de verdad solo si ambas son ciertas
OR	Evalúa dos condiciones y devuelve un valor de verdad si alguna de las dos es cierta
NOT	Devuelve el valor contrario de la expresión

### Operadores de comparación

Operadores	Descripción
<	Menor que
>	Mayor que
<>	Distinto de
<=	Menor o igual
>=	Mayor o igual
=	Igual
BETWEEN	Se utiliza para especificar un intervalo de valores
LIKE	Se utiliza para comparar
IN	Se utiliza para especificar

### Funciones de agregado

Función	Descripción
AVG	Calcula el promedio de los valores de un campo determinado
COUNT	Devuelve el número de filas de selección
SUM	Devuelve la suma de todos los valores de un campo determinado
MAX	Devuelve el valor más alto de un campo
MIN	Devuelve el valor mínimo de un campo determinado

### Literales

Literales	Descripción
23/05/99	Literal fecha
María	Literal caracteres
5	Literal numero

Anexo IV interesante de ver