

# Resumen de Comandos SQL para Examen

## 1. Data Definition Language (DDL)

Los comandos DDL se utilizan para definir y estructurar la base de datos y sus objetos (tablas, índices, vistas, etc.). Estos comandos son:

### 1.1 CREATE

- **Función:** Crear **objetos** en la base de datos (como tablas, vistas, índices o sinónimos).
- **Ejemplos:**

```
CREATE TABLE empleados (  
    id NUMBER PRIMARY KEY,  
    nombre VARCHAR2(50),  
    salario NUMBER  
);  
  
CREATE VIEW vista_activos AS  
SELECT id, nombre FROM empleados WHERE salario > 0;  
  
CREATE INDEX idx_nombre ON empleados(nombre);  
  
CREATE SYNONYM alias_empleados FOR empleados;
```

### 1.2 ALTER

- **Función:** Modificar la estructura de objetos existentes.
  - **Ejemplos:**
- ```
ALTER TABLE empleados ADD email VARCHAR2(100);  
ALTER TABLE empleados DROP COLUMN email;  
ALTER TABLE empleados ADD CONSTRAINT salario_min CHECK (salario >= 1000);  
ALTER TABLE empleados DROP CONSTRAINT salario_min;
```

### 1.3 DROP

- **Función:** Eliminar objetos de la base de datos.
- **Ejemplos:**

```
DROP TABLE empleados;  
DROP VIEW vista_activos;  
DROP INDEX idx_nombre;
```

## 2. Data Manipulation Language (DML)

Los comandos DML permiten manipular los datos almacenados en las tablas. Estos son:

### 2.1 INSERT

- **Función:** Insertar nuevos datos en una tabla.

- **Ejemplos:**

```
INSERT INTO empleados (id, nombre, salario)
VALUES (1, 'Juan', 1200);
```

```
INSERT INTO empleados (id, nombre)
VALUES (2, 'Ana');
```

## 2.2 UPDATE

- **Función:** Modificar los datos existentes en una tabla.
- **Ejemplo:**

```
UPDATE empleados
SET salario = 1500
WHERE id = 1;
```

## 2.3 DELETE

- **Función:** Eliminar datos de una tabla.
- **Ejemplo:**

```
DELETE FROM empleados
WHERE id = 2;
```

## 3. Tipos de Datos en Oracle

| Tipo de Dato        | Descripción                                                 | Ejemplo práctico                                                                                                      |
|---------------------|-------------------------------------------------------------|-----------------------------------------------------------------------------------------------------------------------|
| <b>VARCHAR2(n)</b>  | Cadenas de longitud variable (máx. 4000 bytes).             | nombre VARCHAR2(50)                                                                                                   |
| <b>CHAR(n)</b>      | Cadenas de longitud fija (máx. 2000 bytes).                 | inicial CHAR(1)                                                                                                       |
| <b>NUMBER(p, s)</b> | Números con precisión (p) y escala (s).                     | precio NUMBER(8, 2)                                                                                                   |
| <b>LONG</b>         | Cadenas de longitud variable (máx. 2 GB).                   | descripcion LONG                                                                                                      |
| <b>DATE</b>         | Fechas que incluyen año, mes, día, hora, minutos, segundos. | fecha_nacimiento DATE, donde se puede insertar:<br>INSERT INTO tabla VALUES<br>(TO_DATE('1990-12-25', 'YYYY-MM-DD')); |
| <b>RAW(n)</b>       | Cadenas de bytes de longitud fija (máx. 2000 bytes).        | firma RAW(200)                                                                                                        |
| <b>LONG RAW</b>     | Datos binarios como gráficos o sonidos (máx. 2 GB).         | imagen LONG RAW                                                                                                       |
| <b>ROWID</b>        | Dirección hexadecimal única para cada fila en una tabla.    | No es declarable directamente, pero se usa internamente en consultas avanzadas.                                       |
| <b>CLOB/NCLOB</b>   | Almacena texto grande (máx. más de 2 GB).                   | documento CLOB                                                                                                        |

| Tipo de Dato | Descripción                                               | Ejemplo práctico                                                                          |
|--------------|-----------------------------------------------------------|-------------------------------------------------------------------------------------------|
| <b>BLOB</b>  | Almacena datos binarios grandes (imágenes, videos, etc.). | <b>video BLOB</b> , donde se puede insertar usando herramientas específicas como loaders. |

| Tipo de Dato        | Descripción                                                 |
|---------------------|-------------------------------------------------------------|
| <b>VARCHAR2(n)</b>  | Cadenas de longitud variable (máx. 4000 bytes).             |
| <b>CHAR(n)</b>      | Cadenas de longitud fija (máx. 2000 bytes).                 |
| <b>NUMBER(p, s)</b> | Números con precisión (p) y escala (s).                     |
| <b>LONG</b>         | Cadenas de longitud variable (máx. 2 GB).                   |
| <b>DATE</b>         | Fechas que incluyen año, mes, día, hora, minutos, segundos. |
| <b>RAW(n)</b>       | Cadenas de bytes de longitud fija (máx. 2000 bytes).        |
| <b>LONG RAW</b>     | Datos binarios como gráficos o sonidos (máx. 2 GB).         |
| <b>ROWID</b>        | Dirección hexadecimal única para cada fila en una tabla.    |
| <b>CLOB/NCLOB</b>   | Almacena texto grande (máx. más de 2 GB).                   |
| <b>BLOB</b>         | Almacena datos binarios grandes (imágenes, videos, etc.).   |

## 4. Restricciones (Constraints)

Aseguran la integridad de los datos en las tablas.

### 4.1 NOT NULL

- **Propósito:** Obliga a que una columna no acepte valores nulos.
- **Ejemplo:**

```
CREATE TABLE pedidos (
    cantidad NUMBER NOT NULL
);
```

### 4.2 UNIQUE

- **Propósito:** Garantiza que los valores en una columna o conjunto de columnas no se repitan.
- **Ejemplo:**

```
CREATE TABLE usuarios (
    login VARCHAR2(50) UNIQUE
);
```

### 4.3 PRIMARY KEY

- **Propósito:** Define una clave primaria única por tabla. Los campos son automáticamente NOT NULL y UNIQUE.
- **Ejemplo:**

```
CREATE TABLE personas (
    id NUMBER,
    lastName VARCHAR2(50),
    PRIMARY KEY (id, lastName)
);
```

## 4.4 FOREIGN KEY

- **Propósito:** Garantiza la integridad referencial entre tablas.
- **Opciones de integridad:**
  - **RESTRICT:** Impide modificar/eliminar registros relacionados.
  - **CASCADE:** Modifica/borra automáticamente los registros relacionados.
  - **SET NULL:** Asigna NULL a los registros relacionados.
- **Ejemplo:**

```
CREATE TABLE telefonos (  
    id_persona NUMBER,  
    FOREIGN KEY (id_persona) REFERENCES personas(id) ON DELETE CASCADE  
);
```

## 4.5 DEFAULT

- **Propósito:** Establece un valor por defecto para una columna.
- **Ejemplo:**

```
CREATE TABLE productos (  
    stock NUMBER DEFAULT 0  
);
```

## 4.6 CHECK

- **Propósito:** Define una condición que los valores en una columna deben cumplir.
- **Ejemplo:**

```
CREATE TABLE empleados (  
    salario NUMBER CHECK (salario > 0)  
);
```

# 5. Gestión de Usuarios y Permisos

## 5.1 Crear un usuario

```
CREATE USER nombre_usuario IDENTIFIED BY password_usuario;  
GRANT CONNECT, RESOURCE TO nombre_usuario;  
ALTER USER nombre_usuario QUOTA UNLIMITED ON USERS;
```

### ¿Qué significan los privilegios?

- **CONNECT:** Permite que el usuario pueda conectarse a la base de datos.
- **RESOURCE:** Otorga permisos para crear objetos como tablas, vistas o índices dentro del esquema del usuario.

## 5.1 Crear un usuario

```
CREATE USER nombre_usuario IDENTIFIED BY password_usuario;  
GRANT CONNECT, RESOURCE TO nombre_usuario;  
ALTER USER nombre_usuario QUOTA UNLIMITED ON USERS;
```

## 6. Motores de Almacenamiento (MySQL)

Aunque estás trabajando con Oracle, es importante conocer los motores de MySQL:

| Motor  | Características                                                |
|--------|----------------------------------------------------------------|
| InnoDB | Soporta transacciones seguras y claves foráneas.               |
| MyISAM | Optimizado para consultas rápidas. No soporta claves foráneas. |
| Otros  | BerkeleyDB, HEAP (MEMORY), ISAM, MERGE.                        |

Ejemplo:

```
CREATE TABLE datos (  
    id NUMBER  
) ENGINE = InnoDB;
```