Índice

[POSTGRES SQL 2](#_Toc181465749)

[Select 2](#_Toc181465750)

[Ejemplo WHERE 2](#_Toc181465751)

[Ejemplo LIKE 2](#_Toc181465752)

[Ejemplo INNER JOIN 2](#_Toc181465753)

[Update 2](#_Toc181465754)

[Create 2](#_Toc181465755)

[Insert 2](#_Toc181465756)

[Trigger 3](#_Toc181465757)

[Clave foránea 3](#_Toc181465758)

[Tipo de dato 4](#_Toc181465759)

[Definición 4](#_Toc181465760)

[Procedimiento 5](#_Toc181465761)

[Oracle 6](#_Toc181465762)

[Select 6](#_Toc181465763)

[Ejemplo con WHERE 6](#_Toc181465764)

[Ejemplo con LIKE 6](#_Toc181465765)

[Ejemplo con INNERJOIN 6](#_Toc181465766)

[Update 6](#_Toc181465767)

[Create 6](#_Toc181465768)

[INSERT 6](#_Toc181465769)

[Trigger 7](#_Toc181465770)

[Clave foránea 7](#_Toc181465771)

[Tipo de Datos 7](#_Toc181465772)

[Definición 8](#_Toc181465773)

[Procedimiento 8](#_Toc181465774)

# POSTGRES SQL

## Select

### Ejemplo WHERE

SELECT \* FROM usuarios WHERE id = 1;

### Ejemplo LIKE

SELECT \* FROM productos WHERE nombre LIKE 'A%';

### Ejemplo INNER JOIN

SELECT u.nombre, o.total

FROM usuarios u

INNER JOIN ordenes o ON u.id = o.usuario\_id;

## Update

UPDATE productos

SET precio = 20

WHERE id = 1;

## Create

CREATE TABLE clientes (

id SERIAL PRIMARY KEY,

nombre VARCHAR(50),

email VARCHAR(50) UNIQUE NOT NULL

);

## Insert

INSERT INTO clientes (nombre, email)

VALUES ('Juan Pérez', 'juan.perez@example.com');

## Trigger

-- Crear función para el trigger

CREATE OR REPLACE FUNCTION actualizar\_fecha\_actualizacion()

RETURNS TRIGGER AS $$

BEGIN

NEW.fecha\_actualizacion = NOW();

RETURN NEW;

END;

$$ LANGUAGE plpgsql;

-- Crear trigger

CREATE TRIGGER actualizar\_fecha

BEFORE UPDATE ON productos

FOR EACH ROW

EXECUTE FUNCTION actualizar\_fecha\_actualizacion();

## Clave foránea

CREATE TABLE clientes (

id SERIAL PRIMARY KEY,

nombre VARCHAR(100) NOT NULL

);

CREATE TABLE pedidos (

id SERIAL PRIMARY KEY,

cliente\_id INTEGER,

fecha DATE,

FOREIGN KEY (cliente\_id) REFERENCES clientes(id)

);

## Tipo de dato

CREATE TABLE ejemplo (

id SERIAL PRIMARY KEY,

nombre VARCHAR(50) NOT NULL,

descripcion TEXT,

precio NUMERIC(10, 2),

cantidad INT,

fecha\_creacion DATE DEFAULT CURRENT\_DATE,

hora\_actualizacion TIME,

fecha\_y\_hora TIMESTAMP DEFAULT CURRENT\_TIMESTAMP,

activo BOOLEAN DEFAULT TRUE,

archivo BYTEA

);

### Definición

 VARCHAR(n): Cadena de texto con longitud variable, hasta un máximo de n caracteres.

 CHAR(n): Cadena de texto de longitud fija de n caracteres. Si la cadena es más corta, se rellena con espacios.

 TEXT: Cadena de texto de longitud variable, sin límite de caracteres.

 INTEGER (o INT): Número entero.

 SERIAL: Entero autoincremental, útil para claves primarias.

 BIGINT: Número entero de gran tamaño.

 NUMERIC(p, s): Número decimal con precisión p y escala s.

 REAL y DOUBLE PRECISION: Números de punto flotante.

 BOOLEAN: Tipo de dato para valores TRUE o FALSE.

 DATE: Fecha en formato YYYY-MM-DD.

 TIMESTAMP: Fecha y hora combinadas.

 TIME: Hora (sin fecha).

 BYTEA: Datos binarios, para almacenar archivos binarios como imágenes o documentos.

## Procedimiento

CREATE OR REPLACE FUNCTION calcular\_descuento(precio NUMERIC, porcentaje NUMERIC)

RETURNS NUMERIC AS $$

BEGIN

RETURN precio - (precio \* porcentaje / 100);

END;

$$ LANGUAGE plpgsql;

# Oracle

## Select

### Ejemplo con WHERE

SELECT \* FROM usuarios WHERE id = 1;

### Ejemplo con LIKE

SELECT \* FROM productos WHERE nombre LIKE 'A%';

### Ejemplo con INNERJOIN

SELECT u.nombre, o.total

FROM usuarios u

INNER JOIN ordenes o ON u.id = o.usuario\_id;

## Update

UPDATE productos

SET precio = 20

WHERE id = 1;

## Create

CREATE TABLE clientes (

id NUMBER GENERATED BY DEFAULT AS IDENTITY PRIMARY KEY,

nombre VARCHAR2(50),

email VARCHAR2(50) UNIQUE NOT NULL

);

## INSERT

INSERT INTO clientes (nombre, email)

VALUES ('Juan Pérez', 'juan.perez@example.com');

## Trigger

CREATE OR REPLACE TRIGGER actualizar\_fecha

BEFORE UPDATE ON productos

FOR EACH ROW

BEGIN

:NEW.fecha\_actualizacion := SYSDATE;

END;

## Clave foránea

CREATE TABLE clientes (

id NUMBER GENERATED BY DEFAULT AS IDENTITY PRIMARY KEY,

nombre VARCHAR2(100) NOT NULL

);

CREATE TABLE pedidos (

id NUMBER GENERATED BY DEFAULT AS IDENTITY PRIMARY KEY,

cliente\_id NUMBER,

fecha DATE,

CONSTRAINT fk\_cliente FOREIGN KEY (cliente\_id) REFERENCES clientes(id)

);

## Tipo de Datos

CREATE TABLE ejemplo (

id NUMBER GENERATED BY DEFAULT AS IDENTITY PRIMARY KEY,

nombre VARCHAR2(50) NOT NULL,

descripcion CLOB,

precio NUMBER(10, 2),

cantidad INTEGER,

fecha\_creacion DATE DEFAULT SYSDATE,

hora\_actualizacion TIMESTAMP,

fecha\_y\_hora TIMESTAMP DEFAULT SYSTIMESTAMP,

activo CHAR(1) DEFAULT 'Y',

archivo BLOB

);

### Definición

 VARCHAR2(n): Cadena de texto de longitud variable, con un máximo de n caracteres.

 CHAR(n): Cadena de texto de longitud fija de n caracteres. Si la cadena es más corta, se completa con espacios.

 CLOB: Cadena de texto de gran tamaño, ideal para almacenar grandes cantidades de texto.

 NUMBER(p, s): Número con precisión p y escala s, para valores decimales.

 INTEGER: Alias para NUMBER, utilizado para números enteros.

 FLOAT: Número de punto flotante.

 DATE: Fecha en formato DD-MON-YYYY.

 TIMESTAMP: Fecha y hora con mayor precisión que DATE.

 TIMESTAMP WITH TIME ZONE: Fecha y hora con zona horaria.

 BLOB: Datos binarios de gran tamaño, para almacenar archivos como imágenes o documentos.

 RAW(n): Datos binarios de tamaño fijo, hasta un máximo de n bytes.

 LONG: Cadena de texto de hasta 2 GB (obsoleto en versiones recientes).

## Procedimiento