UNIVERSIDAD TECNICA
FEDERICO SANTA MARIA

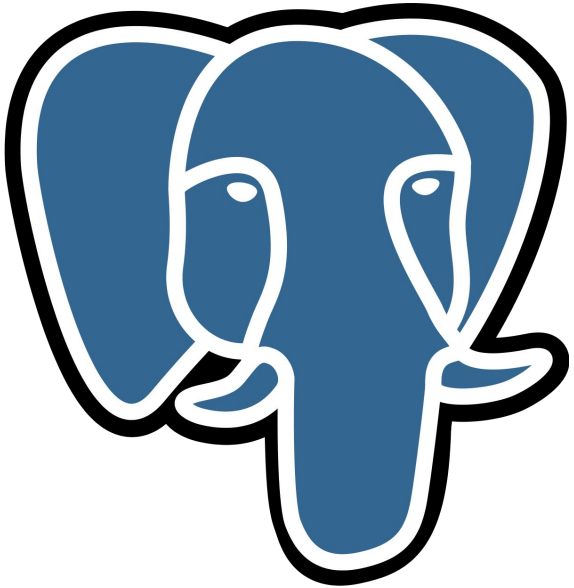
DEPARTAMENTO
DE INFORMÁTICA

INTRODUCCIÓN A SQL

V20230322

GUARANTEE POLICY

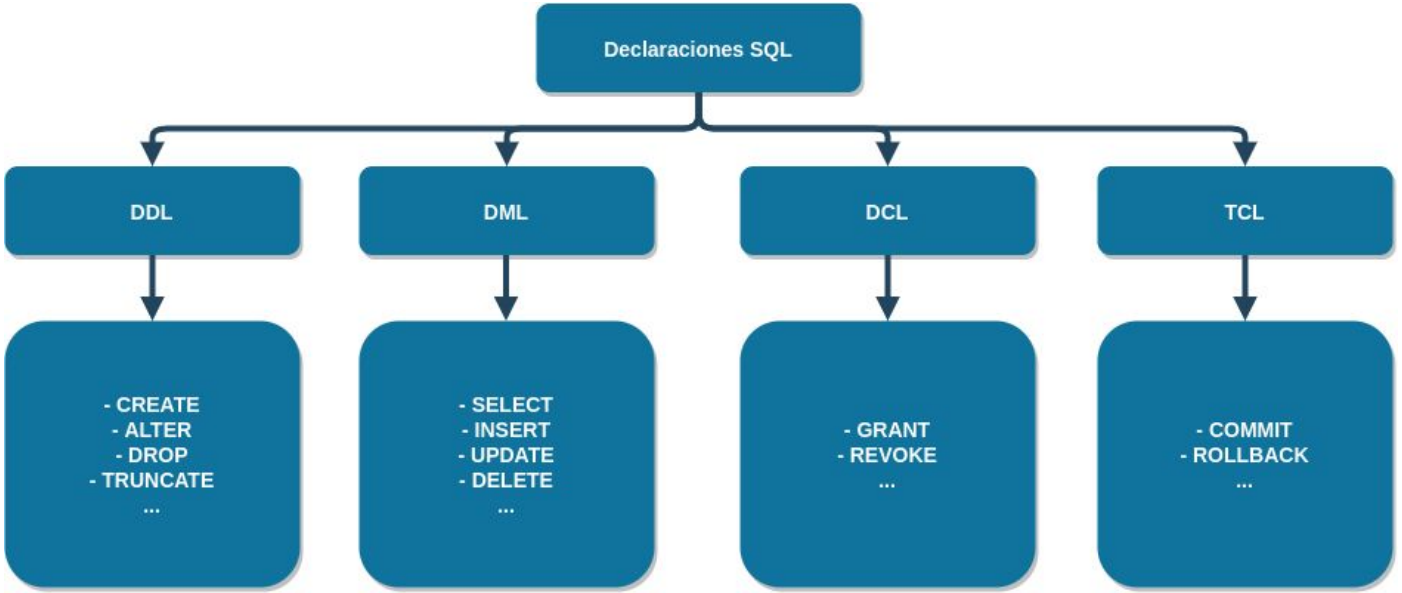
El dialecto SQL de la siguiente presentación corresponde a PostgreSQL 15 y es compatible con la mayoría de los RDBMS, pero en algunos casos se deben realizar modificaciones.



¿QUÉ ES SQL?

SQL (Structured Query Language o Lenguaje de consultas estructuradas) es un lenguaje de programación estandarizado que se utiliza para administrar bases de datos relacionales y realizar diversas operaciones con los datos que contienen.

DECLARACIONES SQL



DDL

Data Definition Language

CREATE

Crea una nueva base de datos:

```
CREATE DATABASE <nombre_bd>;
```

Crea una tabla en una base de datos:

```
CREATE TABLE <nombre_tabla> (
    <columna_1> DATA_TYPE CONSTRAINTS,
    <columna_2> DATA_TYPE CONSTRAINTS,
    <columna_3> DATA_TYPE CONSTRAINTS
);
```

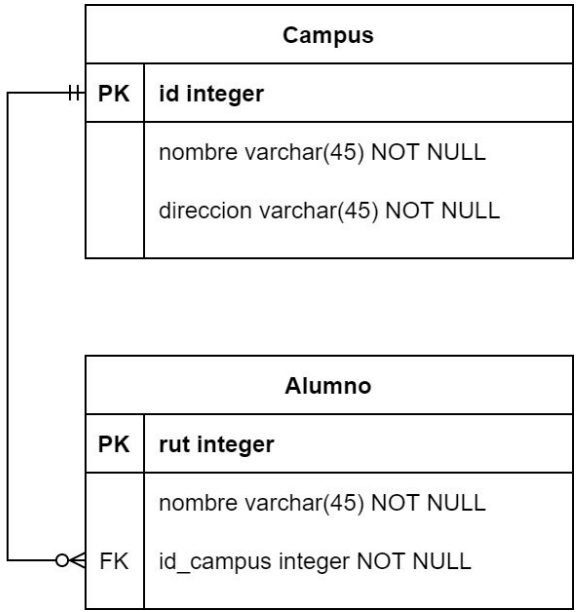
EJEMPLOS

```
CREATE DATABASE Universidad;
```

EJEMPLOS

```
CREATE TABLE Campus (
    id INTEGER,
    nombre VARCHAR(45) NOT NULL,
    direccion VARCHAR(45) NOT NULL,
    PRIMARY KEY (id)
);

CREATE TABLE Alumno (
    rut INTEGER,
    nombre VARCHAR(45) NOT NULL,
    id_campus INTEGER NOT NULL,
    PRIMARY KEY (rut),
    FOREIGN KEY (id_campus) REFERENCES Campus (id)
);
```



ALTER

Modifica la estructura de una base de datos:

```
ALTER DATABASE <nombre_bd> ...;
```

Modifica la estructura de una tabla en una base de datos:

```
ALTER TABLE <nombre_tabla> ...;
```

EJEMPLOS

```
ALTER DATABASE Universidad RENAME TO University;
```

```
ALTER TABLE Campus
ADD COLUMN nombre_director VARCHAR(45) NOT NULL;

ALTER TABLE Alumno
RENAME COLUMN nombre
TO nombre_apellido;

ALTER TABLE Alumno
DROP COLUMN nombre_apellido;
```

TRUNCATE Y DROP

Vacía una tabla sin borrar la estructura:

```
TRUNCATE TABLE <nombre_tabla>;
```

Borra la tabla:

```
DROP TABLE <nombre_tabla>;
```

ALGUNOS TIPOS DE DATOS

Tipo de dato	Descripción
INTEGER	Valor entero.
FLOAT / DOUBLE	Valor decimal en punto flotante / doble precisión.
NUMERIC(X, Y) / DECIMAL(X, Y)	Valor decimal con X dígitos enteros e Y decimales.
VARCHAR(X)	String de hasta X caracteres.
TEXT	String indefinidamente largo (para textos muy grandes).
BOOLEAN / TINYINT	Valor booleano, verdadero (1) o falso (0).
DATE	Fecha ("2021-09-21").
TIME	Hora ("15:24:00").
TIMESTAMP	Fecha y Hora.

ALGUNAS CONSTRAINTS

Constraints	**Descripción**
NOT NULL	La columna no puede tener valores nulos.
UNIQUE	Todos los valores en la columna deben ser distintos.
PRIMARY KEY (<columna>)	Específica claves primarias, NOT NULL + UNIQUE
REFERENCES <Tabla> (<columna>)	Específica claves foráneas.
CHECK <condición>	La columna solo admite valores que cumplan con la condición.

DML

Data Manipulation Language

INSERT

Inserta filas en una tabla:

```
INSERT INTO
    <tabla> (<columna_1>,<columna_2>,...)
VALUES
    (<valor_1a>, <valor_2a>,...),
    (<valor_1b>, <valor_2b>,...);
```

EJEMPLOS

```
INSERT INTO
    Campus (id, nombre, direccion)
VALUES
    (1, 'Casa Central', 'Av. España 1680, Valparaíso'),
    (2, 'San Joaquín', 'Vicuña Mackenna 3939, San Joaquín');
```

```
INSERT INTO
    Alumno (rut, nombre, id_campus)
VALUES
    (12345678, 'Rodrigo', 1);
```

UPDATE

Actualiza los datos de una o más filas (que cumpla con la condición) en una tabla:

```
UPDATE
    <tabla>
SET
    <columna_1> = <nuevo_valor_1>,
    <columna_2> = <nuevo_valor_2>, ...
WHERE
    <condición>;
```

EJEMPLOS

```
UPDATE
    Alumno
SET
    nombre = 'Juan'
WHERE
    rut = 12345678;
```

```
UPDATE
    Alumno
SET
    nombre = 'Lucas'
WHERE
    id_campus = 1 AND rut > 10000000;
```

DELETE

Elimina una o más filas (que cumpla con la condición) en una tabla:

```
DELETE FROM
    <tabla>
WHERE
    <condición>;
```

EJEMPLOS

```
DELETE FROM
    Alumno
WHERE
    nombre = 'Rodrigo' OR rut > 10000000;
```

OJOS

Si se omite la cláusula WHERE, las operaciones UPDATE, DELETE y SELECT afectarán a todas las filas.

```
DELETE FROM
    Alumno;
```

PALTA

Antes de realizar una operación DELETE, cambiala por un SELECT. De esta forma podrás chequear las filas antes de eliminarlas.

```
DELETE
    Alumno
WHERE
    rut > 10000000;
```

```
SELECT
    Alumno
WHERE
    rut > 10000000;
```

SELECT

Retorna una o más filas (que cumpla con la condición) en una tabla:

```
SELECT
    <columna_1>, <columna_2>, ..., <columna_n>
FROM
    <tabla>
WHERE
    <condición>;
```

EJEMPLOS

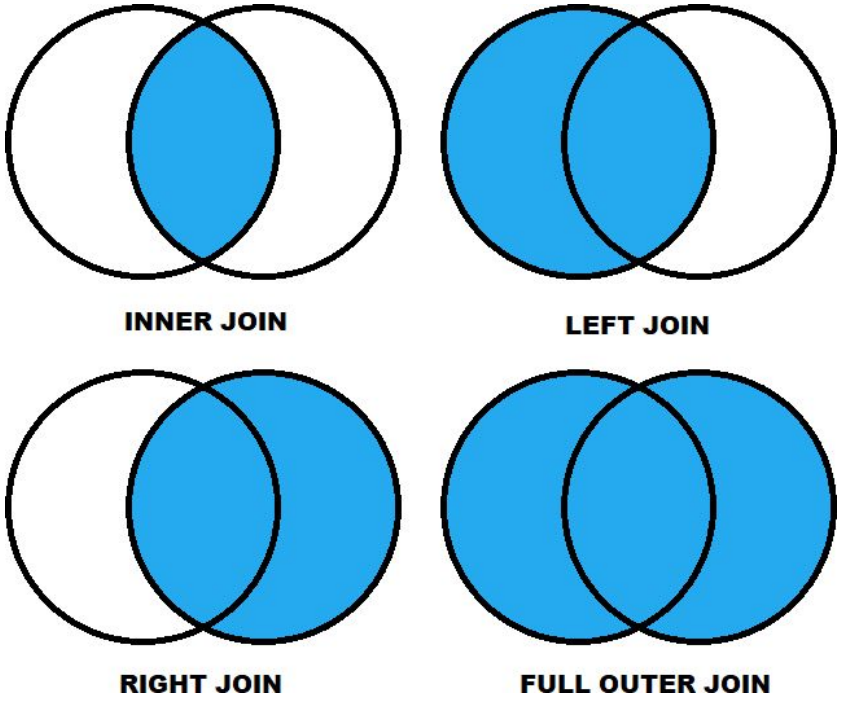
```
SELECT
    rut, nombre
FROM
    Alumno
WHERE
    id_campus = 1;
```

SELECT: JOIN

La operación SELECT permite hacer consultas bastante complejas. Por ejemplo, podemos obtener datos de varias tablas distintas al mismo tiempo. Para ese tipo de consultas, se utiliza la operación JOIN dentro de SELECT.

La operación JOIN junta columnas de 2 o más tablas horizontalmente, emparejando las filas según algún atributo a comparar.

TIPOS DE JOINS

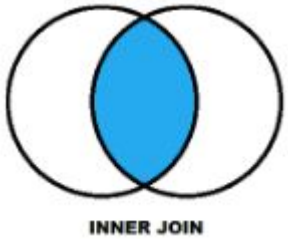


INNER JOIN

Alumno		
rut	nombre	id_campus
12345678	Rodrigo	1
20111222	Juan	1
15232323	Matilda	2
16898569	Lucas	

Campus		
id	nombre	direccion
1	Casa Central	Av. España 1680, Valparaíso
2	San Joaquín	Vicuña Mackenna 3939, San Joaquín
3	Vitacura	Sta. María 6400, Vitacura
4	José Miguel Carrera	Sta. María 6090, Viña del Mar

INNER JOIN			
Alumno.rut	Alumno.nombre	Campus.nombre	Campus.direccion
12345678	Rodrigo	Casa Central	Av. España 1680, Valparaíso
20111222	Juan	Casa Central	Av. España 1680, Valparaíso
15232323	Matilda	San Joaquín	Vicuña Mackenna 3939, San Joaquín

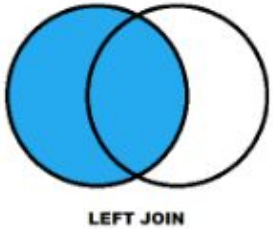


LEFT JOIN

Alumno		
rut	nombre	id_campus
12345678	Rodrigo	1
20111222	Juan	1
15232323	Matilda	2
16898569	Lucas	

Campus		
id	nombre	direccion
1	Casa Central	Av. España 1680, Valparaíso
2	San Joaquín	Vicuña Mackenna 3939, San Joaquín
3	Vitacura	Sta. María 6400, Vitacura
4	José Miguel Carrera	Sta. María 6090, Viña del Mar

LEFT JOIN			
Alumno.rut	Alumno.nombre	Campus.nombre	Campus.direccion
12345678	Rodrigo	Casa Central	Av. España 1680, Valparaíso
20111222	Juan	Casa Central	Av. España 1680, Valparaíso
15232323	Matilda	San Joaquín	Vicuña Mackenna 3939, San Joaquín
16898569	Lucas		

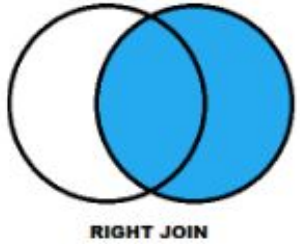


RIGHT JOIN

Alumno		
rut	nombre	id_campus
12345678	Rodrigo	1
20111222	Juan	1
15232323	Matilda	2
16898569	Lucas	

Campus		
id	nombre	direccion
1	Casa Central	Av. España 1680, Valparaíso
2	San Joaquín	Vicuña Mackenna 3939, San Joaquín
3	Vitacura	Sta. María 6400, Vitacura
4	José Miguel Carrera	Sta. María 6090, Viña del Mar

RIGHT JOIN			
Alumno.rut	Alumno.nombre	Campus.nombre	Campus.direccion
12345678	Rodrigo	Casa Central	Av. España 1680, Valparaíso
20111222	Juan	Casa Central	Av. España 1680, Valparaíso
15232323	Matilda	San Joaquín	Vicuña Mackenna 3939, San Joaquín
		Vitacura	Sta. María 6400, Vitacura
		José Miguel Carrera	Sta. María 6090, Viña del Mar

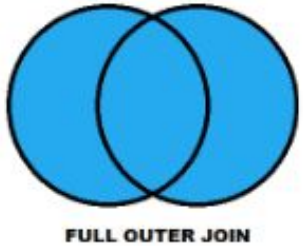


FULL OUTER JOIN

Alumno		
rut	nombre	id_campus
12345678	Rodrigo	1
20111222	Juan	1
15232323	Matilda	2
16898569	Lucas	

Campus		
id	nombre	direccion
1	Casa Central	Av. España 1680, Valparaíso
2	San Joaquín	Vicuña Mackenna 3939, San Joaquín
3	Vitacura	Sta. María 6400, Vitacura
4	José Miguel Carrera	Sta. María 6090, Viña del Mar

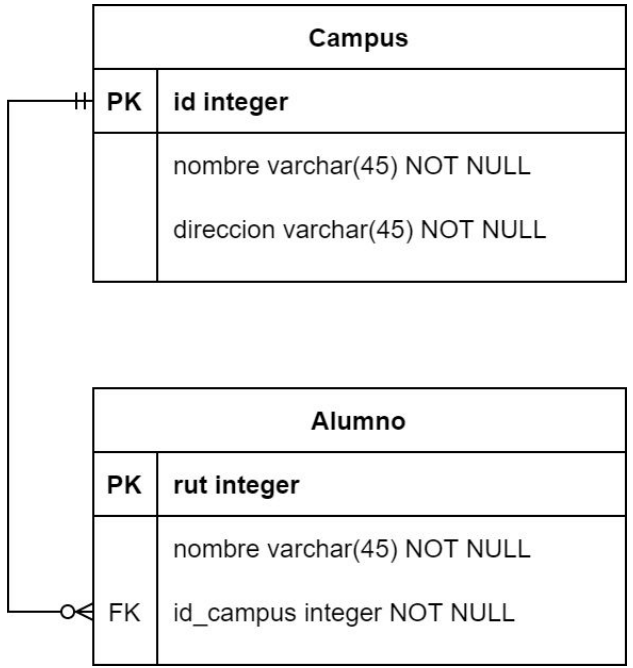
FULL OUTER JOIN			
Alumno.rut	Alumno.nombre	Campus.nombre	Campus.direccion
12345678	Rodrigo	Casa Central	Av. España 1680, Valparaíso
20111222	Juan	Casa Central	Av. España 1680, Valparaíso
15232323	Matilda	San Joaquín	Vicuña Mackenna 3939, San Joaquín
16898569	Lucas		
		Vitacura	Sta. María 6400, Vitacura
		José Miguel Carrera	Sta. María 6090, Viña del Mar



EJEMPLOS

```
SELECT
    Alumno.rut, Alumno.nombre, Campus.nombre,
Campus.direccion
FROM
    Alumno INNER JOIN Campus
ON
    Alumno.id_campus = Campus.id;
```

```
SELECT
    *
FROM
    Alumno LEFT JOIN Campus
ON
    Alumno.id_campus = Campus.id;
```



EJERCICIOS

SQL UTILIZADO EN EL EJERCICIO DE BIBLIOTECA

```sql
CREATE TABLE Editoriales(
    codigo INT PRIMARY KEY GENERATED ALWAYS AS IDENTITY,
    nombre VARCHAR(45) NOT NULL
);

CREATE TABLE Libros(
    id INT PRIMARY KEY GENERATED ALWAYS AS IDENTITY,
    nombre VARCHAR(45) NOT NULL,
    autor VARCHAR(45) NOT NULL,
    id_editorial INT NOT NULL,
    FOREIGN KEY(id_editorial) REFERENCES editoriales(codigo)
);
```

INSERTAR DATOS A LAS TABLAS

```
INSERT INTO editoriales(nombre)
VALUES('Planeta'),
('Emece'),
('Siglo XXI');

INSERT INTO libros(nombre,autor,id_editorial)
VALUES('El aleph','Borges',2),
('Martin Fierro','Jose Hernandez',1),
('Aprenda PHP','Mario Molina',3),
('Java en 10 minutos','Pedro',3);
```

CONSULTAS

```
SELECT *
FROM editoriales;
```

	codigo [PK] integer	nombre character varying (45)
1	1	Planeta
2	2	Emece
3	3	Siglo XXI

```
SELECT *
FROM editoriales
WHERE codigo = '2';
```

	codigo [PK] integer	nombre character varying (45)
1	2	Emece

```
SELECT Libros.id, Libros.nombre, editoriales.nombre
FROM libros INNER JOIN editoriales
    ON libros.id_editorial = editoriales.codigo
WHERE libros.id_editorial ='3';
```

	id integer	nombre character varying (45)	nombre character varying (45)
1	3	Aprenda PHP	Siglo XXI
2	4	Java en 10 minutos	Siglo XXI

LINKS DE INTERÉS

- S.Q.L or Sequel: How to Pronounce SQL? https://learnsql.com/blog/sql-or-sequel/
- https://www.rgomunita.cl