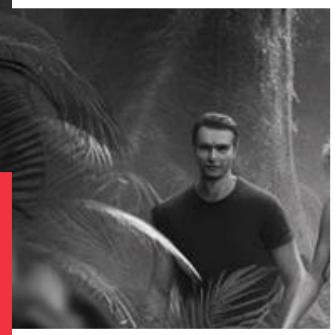
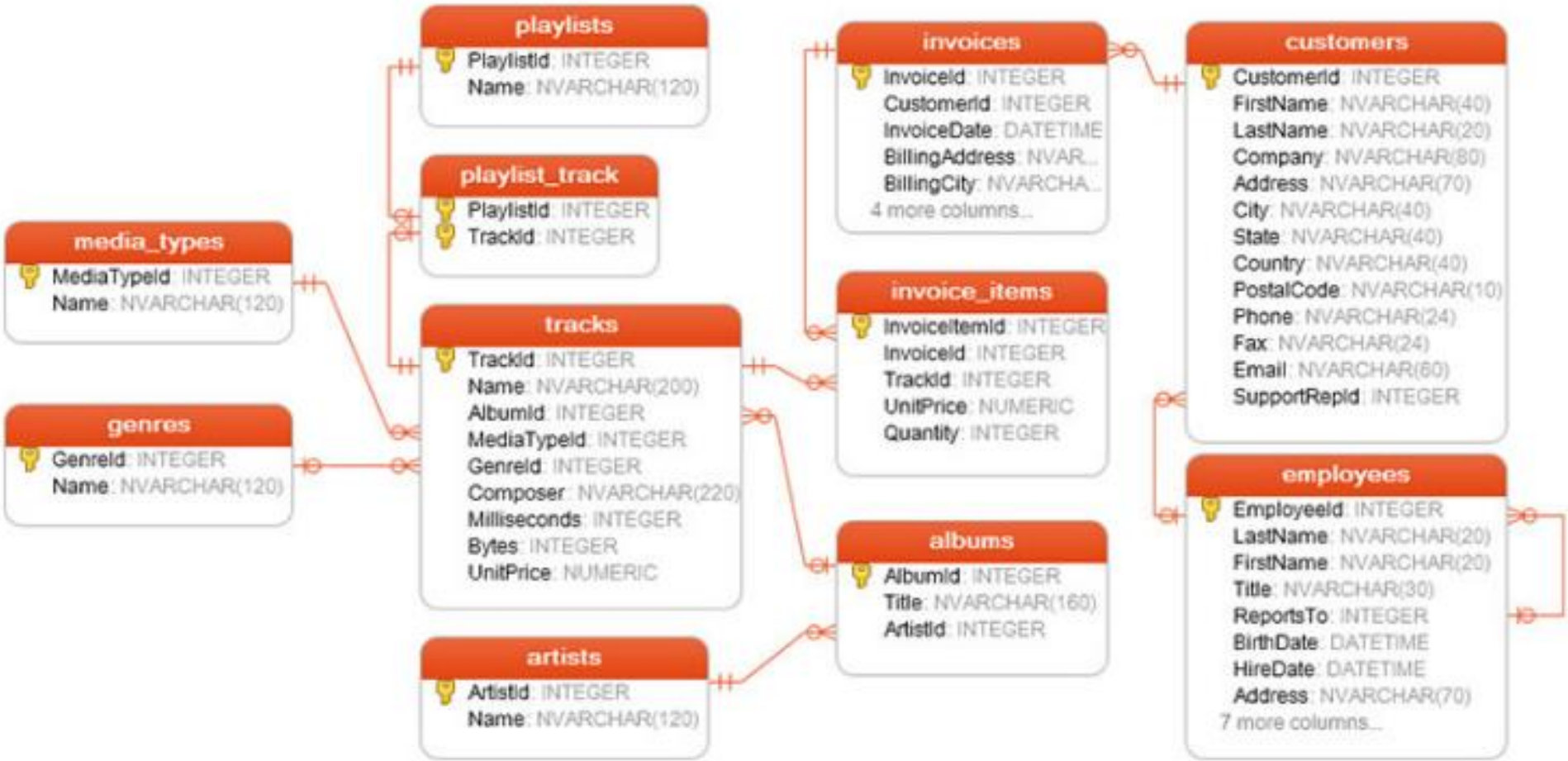
THE BRIDGE

SQL
Introducción



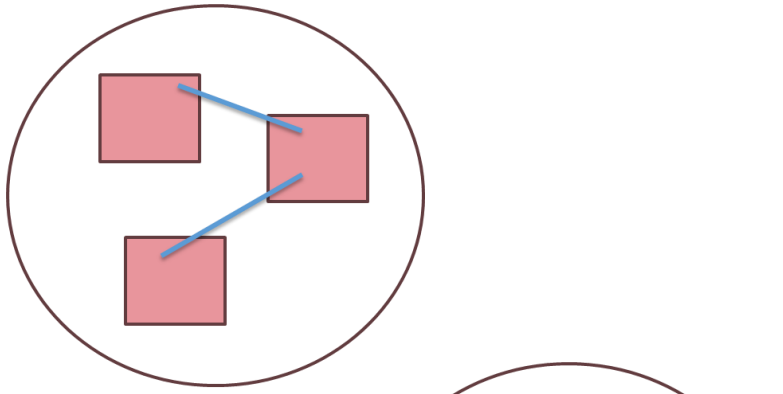
Recordatorio: Bases de Datos Relacionales



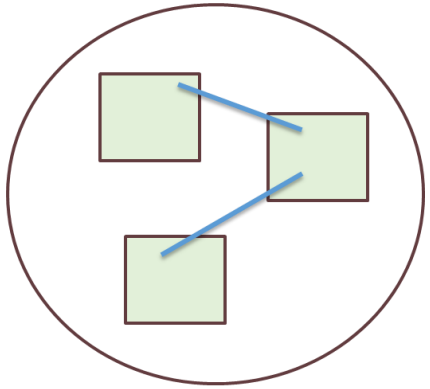
SQL: Structured Query Language

- **Lenguaje de Programación**: Sintáxis propia, interpretado por un ordenador para dar respuesta a unas instrucciones
- Creado en 1970, en IBM
- Es el estándar de facto para la interacción con bases de datos relacionales

SQL: Un poco de contexto

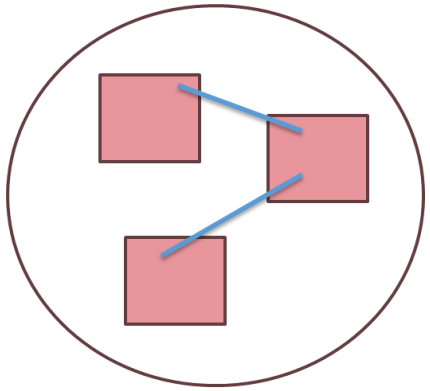


BBDD Relacional “A”

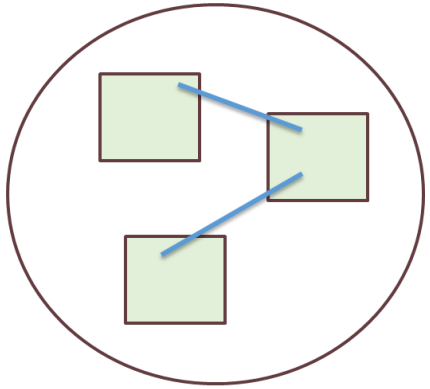


BBDD Relacional “B”

SQL: Un poco de contexto



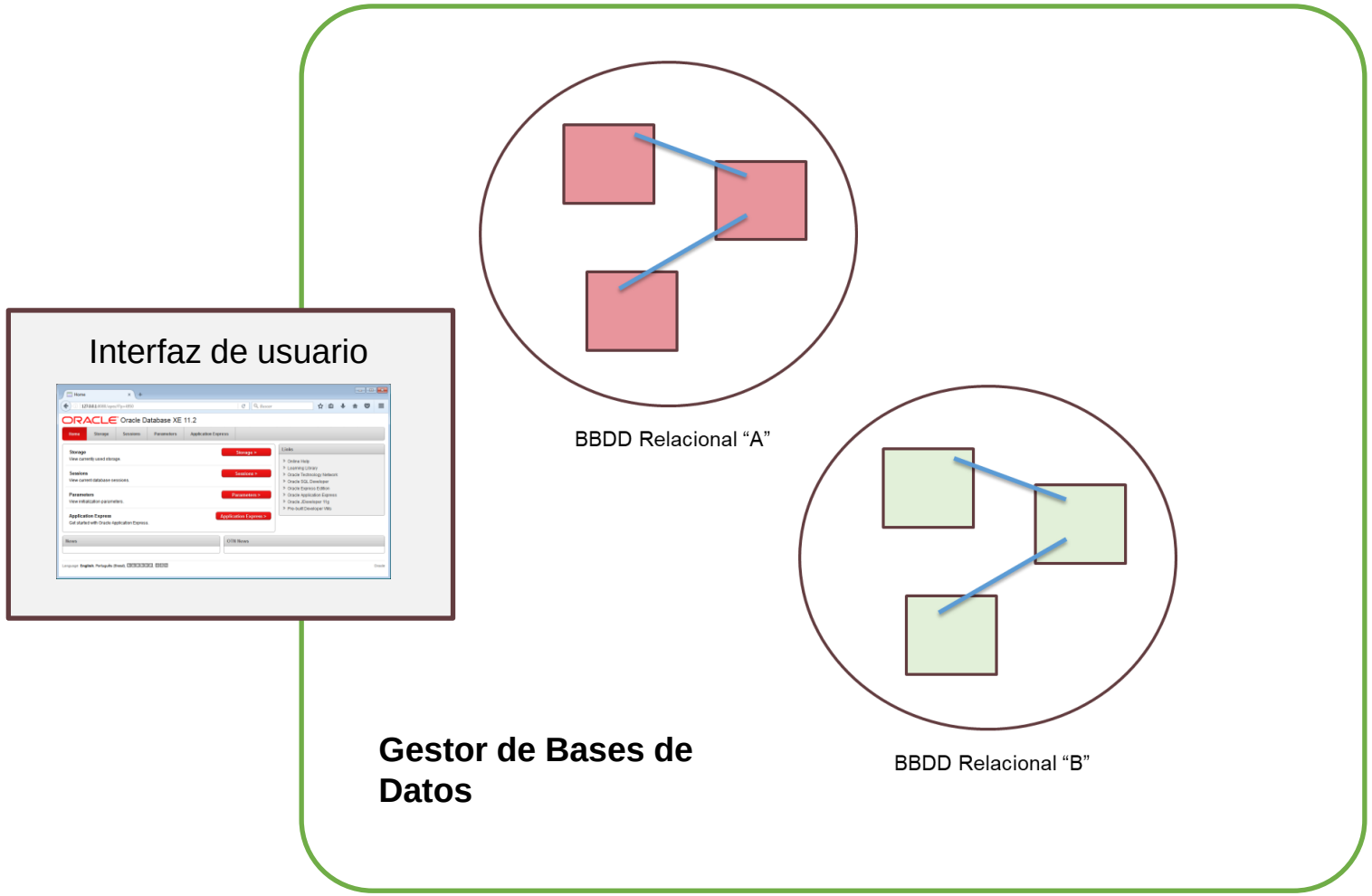
BBDD Relacional “A”



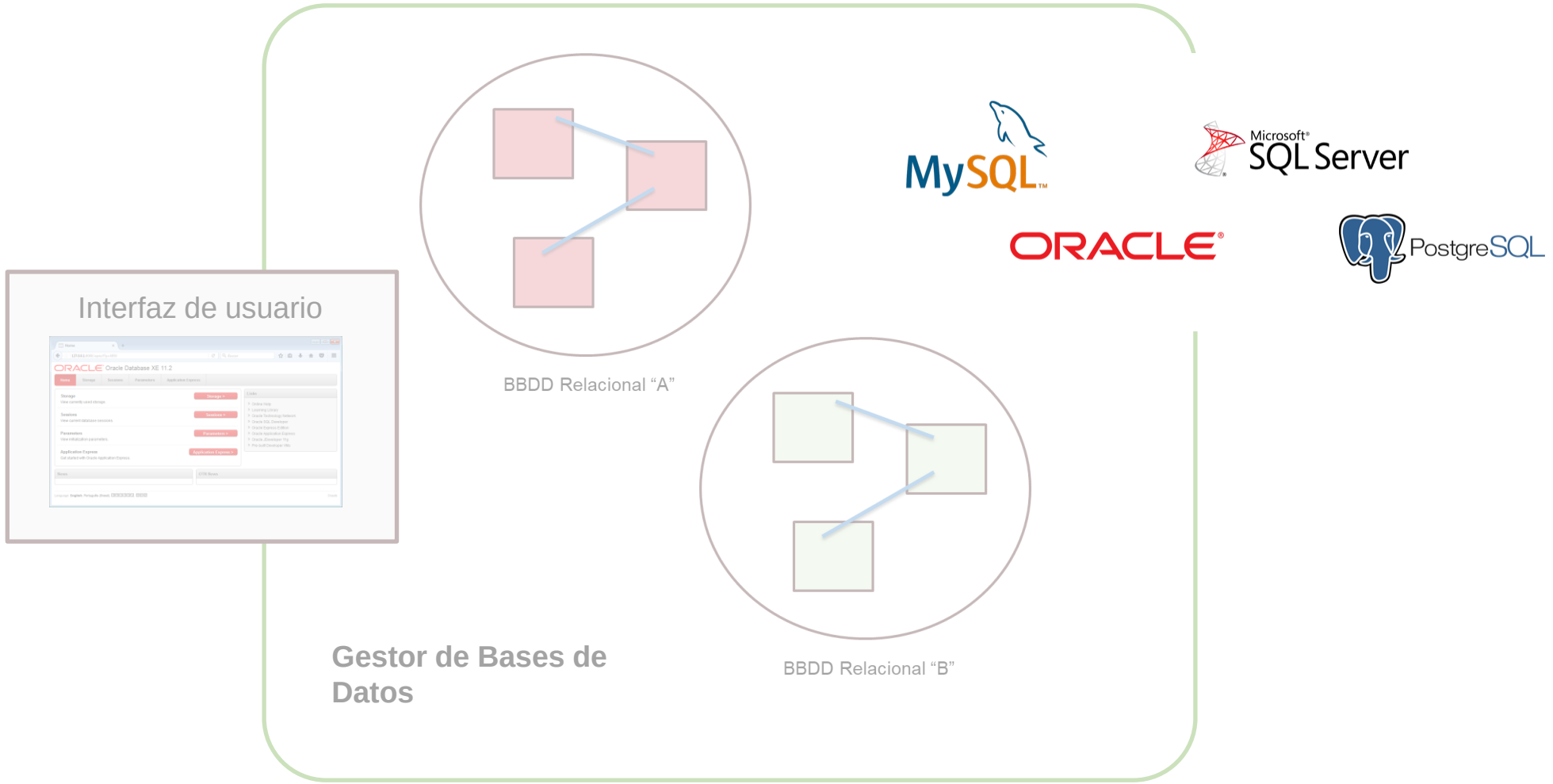
BBDD Relacional “B”

Gestor de Bases de Datos

SQL: Un poco de contexto

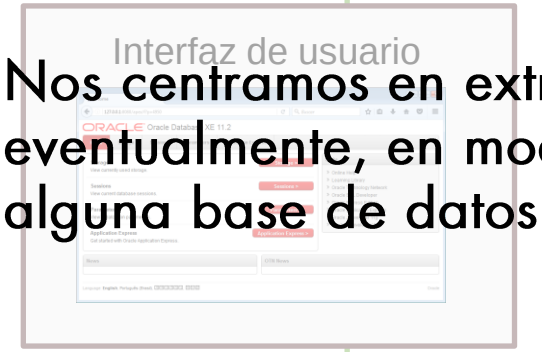


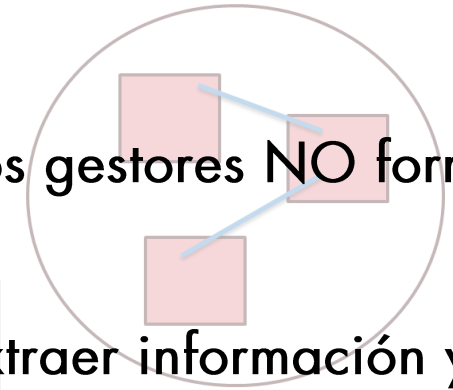
SQL: Un poco de contexto

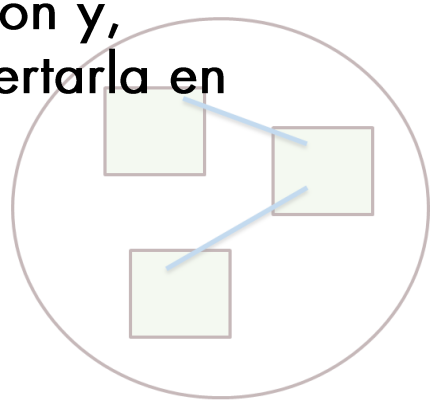


SQL: Un poco de contexto

- Aprender a usar estos gestores NO forma parte del bootcamp
- Nos centramos en extraer información y, eventualmente, en modificarla o insertarla en alguna base de datos







Gestor de Bases de Datos

BBDD Relacional "B"

MySQL

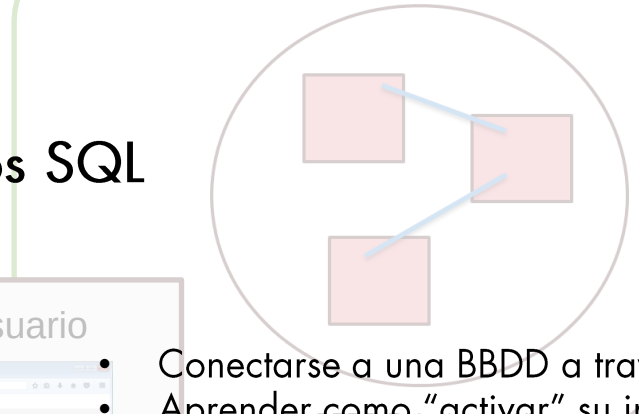
Microsoft SQL Server

ORACLE

PostgreSQL

SQL: Un poco de contexto

- Aprenderemos SQL



- Operativa:
  - Conectarse a una BBDD a través de un gestor.
  - Aprender como "activar" su interprete, usar SQL para extraer la información
  - Volcarla a nuestros Pandas
  - Trabajar con normalidad
  - Eventualmente: Guardar nuestros resultados (usando SQL…)

SQL: Un poco de contexto

- Probablemente tendrás que aprender alguna funcionalidad de algún gestor
- Se escapa del perímetro del bootcamp
- No te asustes, después del bootcamp aprender algo nuevo...

SQL: Cómo

- Python tiene librerías que permiten interactuar con las BBDD de esos gestores
- Veremos: Sqlite3 y MySQL (sqlite3 y pymysql)
- Les lanzaremos **consultas (queries)** en SQL y funcionaremos igual que si estuviéramos usando sus intérprertes

SQL: Componentes Clave

- SELECT
- INSERT
- UPDATE
- DELETE

SQL: Componentes Clave

- SELECT
- INSERT
- UPDATE
- DELETE

Tabla 1: Empleados

id_empleado	nombre	id_departamento
1	Ana	10
2	Carlos	20
3	Diana	10
4	Eduardo	30

SQL: SELECT

- SELECT
- INSERT
- UPDATE
- DELETE

Tabla 1: Empleados

id_empleado	nombre	id_departamento
1	Ana	10
2	Carlos	20
3	Diana	10
4	Eduardo	30

SELECT campo o campos a seleccionar
FROM tabla o tablas de las que seleccionarios
WHERE condiciones a aplicar
ORDER BY campo o campos por los que ordenar el resultado
LIMIT número de registros que quiero recuperar si no quiero todos los posibles

SQL: SELECT Ejempl

- SELECT
- INSERT
- UPDATE
- DELETE

SELECT * FROM Empleados;

Resultado:
Devuelve toda la tabla con sus campos

Id_Empleado	Nombre	Id_Departamento
1	Ana	10
2	Carlos	20
3	Diana	10
4	Eduardo	30

Tabla 1: Empleados

id_empleado	nombre	id_departamento
1	Ana	10
2	Carlos	20
3	Diana	10
4	Eduardo	30

SQL: SELECT Ejempl

- SELECT
- INSERT
- UPDATE
- DELETE

SELECT nombre,departamento
FROM Empleados;

Resultado:
Devuelve todas las filas pero solo con los campos nombre, departamento:

Nombre	Id_Departamento
Ana	10
Carlos	20
Diana	10
Eduardo	30

Tabla 1: Empleados

id_empleado	nombre	id_departamento
1	Ana	10
2	Carlos	20
3	Diana	10
4	Eduardo	30

SQL: SELECT Ejempl

- SELECT
- INSERT
- UPDATE
- DELETE

```
SELECT nombre,id_departamento
FROM Empleados
WHERE id_departamento = 10;
```

Tabla 1: Empleados

id_empleado	nombre	id_departamento
1	Ana	10
2	Carlos	20
3	Diana	10
4	Eduardo	30

Resultado:
Devuelve nombre e id_departamento de los empleados en el departamento 10:

Nombre	Id_Departamento
Ana	10
Diana	10

SQL: SELECT Ejempl

- SELECT
- INSERT
- UPDATE
- DELETE

```
SELECT nombre,id_departamento
FROM Empleados
WHERE id_departamento = 10
ORDER BY nombre DESC;
```

Resultado:
Devuelve nombre e id_departamento de los empleados en el departamento 10 pero ordenados por nombre de forma descendente:

Nombre	Id_Departamento
Diana	10
Ana	10

Tabla 1: Empleados

id_empleado	nombre	id_departamento
1	Ana	10
2	Carlos	20
3	Diana	10
4	Eduardo	30

SQL: SELECT Ejempl

- SELECT
- INSERT
- UPDATE
- DELETE

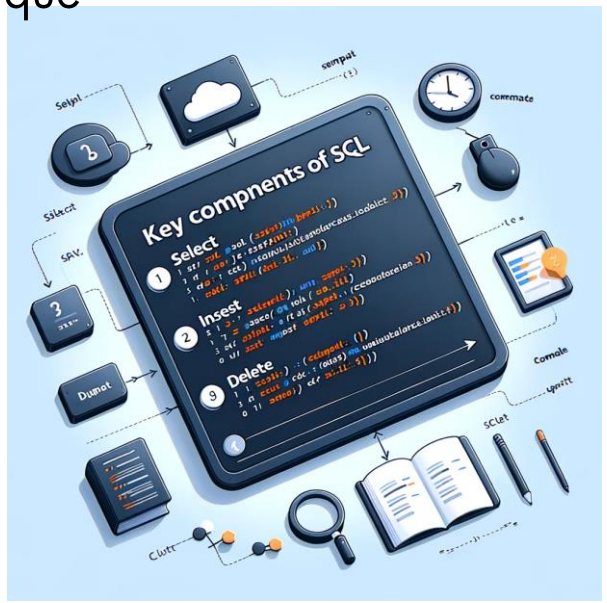
```
SELECT nombre,id_departamento
FROM Empleados
WHERE id_departamento != 30
ORDER BY nombre DESC LIMIT 2
```

Tabla 1: Empleados

id_empleado	nombre	id_departamento
1	Ana	10
2	Carlos	20
3	Diana	10
4	Eduardo	30

Resultado:
Devuelve nombre e id_departamento de los empleados que no son del departamento 30 ordenados por nombre de forma descendente pero sólo los dos primeros

Nombre	Id_Departamento
Diana	10
Carlos	20



SQL: INSERT

- SELECT
- INSERT
- UPDATE
- DELETE

INSERT INTO nombre_tabla(columnas)
VALUES (valores para cada columna)

SQL: INSERT ejemplo

- SELECT
- INSERT
- UPDATE
- DELETE

INSERT INTO Empleados (id_empleado, nombre, id_departamento) VALUES (5, 'Roberto', 30);

Tabla 1: Empleados

id_empleado	nombre	id_departamento
1	Ana	10
2	Carlos	20
3	Diana	10
4	Eduardo	30
5	Roberto	30

SQL: UPDATE

- SELECT
- INSERT
- UPDATE
- DELETE

```
UPDATE nombre_tabla
SET columna1 = valor1, columna2 = valor2, ...
WHERE condicion;
```

SQL: UPDATE ejemplo

- SELECT
- INSERT
- UPDATE
- DELETE

```
UPDATE empleados
SET id_departamento = 3
WHERE id_empleado = 3;
```

Tabla 1: Empleados

id_empleado	nombre	id_departamento
1	Ana	10
2	Carlos	20
3	Diana	3
4	Eduardo	30
5	Roberto	30

SQL: DELETE

- SELECT
- INSERT
- UPDATE
- DELETE

DELETE FROM nombre_tabla
WHERE condicion;

https://www.youtube.com/watch?v=mZWfdGR9ErQ


THE
BRIDGE