

## CURSO PROGRAMACIÓN WEB FULL STACK

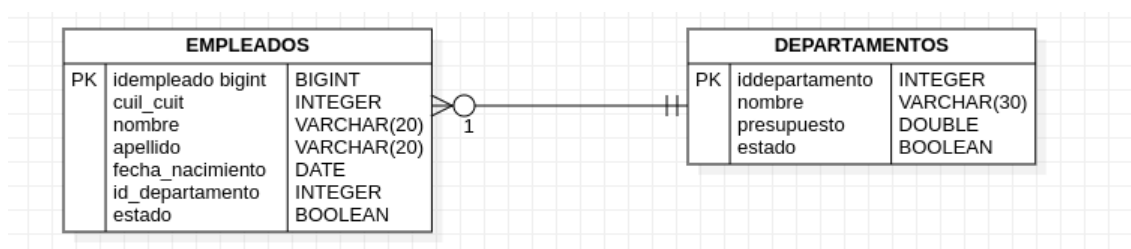
### NIVEL 2

#### Módulo 4 - Bases de datos

#### Ejercicio 1:

Partiendo del modelo entidad relación, ejecutar el script y resolver las consultas. Guardar en un archivo tp5-eje1.sql las consultas

#### Modelo ER Empleados - Departamentos



#### Script para mysql

```
DROP DATABASE IF EXISTS sistema;
CREATE DATABASE sistema CHARACTER SET utf8mb4;
USE sistema;

CREATE TABLE departamentos (
  iddepartamento INT UNSIGNED AUTO_INCREMENT PRIMARY KEY,
  nombre VARCHAR(100) NOT NULL,
  presupuesto DOUBLE UNSIGNED NOT NULL,
  estado boolean NOT NULL
);

CREATE TABLE empleados (
  idemplado INT UNSIGNED AUTO_INCREMENT PRIMARY KEY,
```

```
cuil_cuit VARCHAR(15) NOT NULL UNIQUE,  
nombre VARCHAR(100) NOT NULL,  
apellido VARCHAR(100) NOT NULL,  
id_departamento INT UNSIGNED,  
estado BOOLEAN,  
FOREIGN KEY (id_departamento) REFERENCES departamentos(iddepartamento)  
);
```

```
INSERT INTO departamentos VALUES(1, 'Desarrollo', 120000, true);  
INSERT INTO departamentos VALUES(2, 'Sistemas', 150000, true);  
INSERT INTO departamentos VALUES(3, 'Recursos Humanos', 280000, true);  
INSERT INTO departamentos VALUES(4, 'Contabilidad', 110000, true);  
INSERT INTO departamentos VALUES(5, 'I+D', 375000, true);  
INSERT INTO departamentos VALUES(6, 'Proyectos', 0,true );  
INSERT INTO departamentos VALUES(7, 'Publicidad', 0, true);  
INSERT INTO departamentos VALUES(8, 'Comercial', 0, false);
```

```
INSERT INTO empleados VALUES(1, '27-32481596-3', 'Aarón', 'Rivero', 1, true);  
INSERT INTO empleados VALUES(2, '27-29557532-1', 'Adela', 'Salas', 2, true);  
INSERT INTO empleados VALUES(3, '20-36970642-1', 'Adolfo', 'Rubio', 3, true);  
INSERT INTO empleados VALUES(4, '20-41705545-1', 'Adrián', 'Suárez', 4, true);  
INSERT INTO empleados VALUES(5, '20-17087203-3', 'Marcos', 'Loyola', 5, true);  
INSERT INTO empleados VALUES(6, '27-38382980-3', 'María', 'Santana', 1, true);  
INSERT INTO empleados VALUES(7, '23-80576669-1', 'Pilar', 'Ruiz', 2, true);  
INSERT INTO empleados VALUES(8, '24-71651431-3', 'Pepe', 'Ruiz', 3, true);  
INSERT INTO empleados VALUES(9, '25-36399183-3', 'Juan', 'Gómez', 2, true);  
INSERT INTO empleados VALUES(10, '20-34638446-3', 'Diego', 'Flores', 5, true);  
INSERT INTO empleados VALUES(11, '27-36738983-3', 'Marta', 'Herrera', 1, true);  
INSERT INTO empleados VALUES(12, '27-44123836-1', 'Irene', 'Salas', NULL, false);  
INSERT INTO empleados VALUES(13, '20-38265162-1', 'Juan Antonio', NULL, true);
```

### Resolver las siguientes consultas utilizando la sintaxis SQL

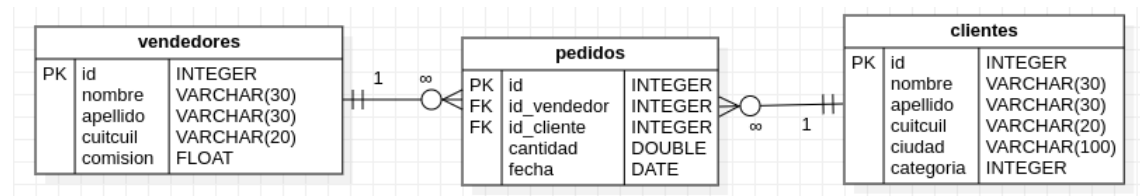
1. Lista el apellido de todos los empleados.
2. Lista el apellido de los empleados eliminando los apellidos que estén repetidos.
3. Lista todas las columnas de la tabla empleados.
4. Lista el nombre y apellido de todos los empleados.
5. Lista el cuil/cuil de los departamentos de los empleados que aparecen en la tabla empleados.
6. Lista el nombre y apellido de los empleados en una única columna.
7. Lista el nombre y apellido de los empleados en una única columna, convirtiendo todos los caracteres en mayúscula.
8. Lista el nombre y apellido de los empleados en una única columna, convirtiendo todos los caracteres en minúscula.

9. Lista el nombre de los departamentos y el valor del presupuesto actual ordenado de forma ascendente.
10. Lista el nombre de todos los departamentos ordenados de forma ascendente.
11. Lista el nombre de todos los departamentos ordenados de forma descendente.
12. Lista el apellido y el nombre de todos los empleados, ordenados de forma alfabética teniendo en cuenta en primer lugar su apellido y luego su nombre.
13. Devuelve una lista con el nombre y el presupuesto, de los 3 departamentos que tienen mayor presupuesto.
14. Devuelve una lista con el nombre y el presupuesto, de los 3 departamentos que tienen menor presupuesto.
15. Devuelve una lista con el nombre de los departamentos y el presupuesto, de aquellos que tienen un presupuesto mayor o igual a \$150000.
16. Devuelve una lista con el nombre de los departamentos y el presupuesto, de aquellos que tienen un presupuesto entre \$100000 y \$200000. Sin utilizar el operador BETWEEN.
17. Devuelve una lista con el nombre de los departamentos que **no** tienen un presupuesto entre \$100000 y \$200000. Sin utilizar el operador BETWEEN.
18. Devuelve una lista con el nombre de los departamentos que tienen un presupuesto entre \$100000 y \$200000. Utilizando el operador BETWEEN.
19. Devuelve una lista con el nombre de los departamentos que **no** tienen un presupuesto entre \$100000 y \$200000. Utilizando el operador BETWEEN.
20. Devuelve un listado con los empleados y los datos de los departamentos donde trabaja cada uno.
21. Devuelve un listado con los empleados y los datos de los departamentos donde trabaja cada uno. Ordena el resultado, en primer lugar por el nombre del departamento (en orden alfabético) y en segundo lugar por apellido y el nombre de los empleados.
22. Devuelve un listado con el código y el nombre del departamento, solamente de aquellos departamentos que tienen empleados.
23. Devuelve el nombre del departamento donde trabaja el empleado que tiene el cuit 27-38382980-3.
24. Devuelve el nombre del departamento donde trabaja el empleado Pepe Ruiz.
25. Devuelve un listado con los datos de los empleados que trabajan en el departamento de I+D. Ordena el resultado alfabéticamente.
26. Devuelve un listado con los datos de los empleados que trabajan en el departamento de Sistemas, Contabilidad o I+D. Ordena el resultado alfabéticamente.
27. Devuelve una lista con el nombre de los empleados que tienen los departamentos que **no** tienen un presupuesto entre \$100000 y \$200000.

## Ejercicio 2:

Partiendo del modelo entidad relación, ejecutar el script y resolver las consultas. Guardar en un archivo tp5-eje2.sql

### Modelo ER Gestion de Pedidos



### Script para mysql

```

DROP DATABASE IF EXISTS ventas;
CREATE DATABASE ventas CHARACTER SET utf8mb4;
USE ventas;

CREATE TABLE clientes (
  id INT UNSIGNED AUTO_INCREMENT PRIMARY KEY,
  nombre VARCHAR(30) NOT NULL,
  apellido VARCHAR(30) NOT NULL,
  cuitcuil VARCHAR(20),
  ciudad VARCHAR(100),
  categoria INT UNSIGNED
);

CREATE TABLE vendedores (
  id INT UNSIGNED AUTO_INCREMENT PRIMARY KEY,
  nombre VARCHAR(30) NOT NULL,
  apellido VARCHAR(30) NOT NULL,
  cuitcuil VARCHAR(20),
  comision FLOAT
);

CREATE TABLE pedidos (
  id INT UNSIGNED AUTO_INCREMENT PRIMARY KEY,
  cantidad DOUBLE NOT NULL,
  fecha DATE,
  id_cliente INT UNSIGNED NOT NULL,
  id_vendedor INT UNSIGNED NOT NULL
);
  
```

```
id_vendedor INT UNSIGNED NOT NULL,  
FOREIGN KEY (id_cliente) REFERENCES clientes(id),  
FOREIGN KEY (id_vendedor) REFERENCES vendedores(id)  
);  
  
INSERT INTO clientes VALUES(1, 'Aarón', 'Rivero', '20-4148122-1', 'Garuhape',  
100);  
INSERT INTO clientes VALUES(2, 'Adela', 'Salas', '27-42482596-3', 'Capiovi',  
200);  
INSERT INTO clientes VALUES(3, 'Adolfo', 'Rubio', '23-37928596-3', 'Posadas',  
NULL);  
INSERT INTO clientes VALUES(4, 'Adrián', 'Suárez', NULL, 'Jaén', 300);  
INSERT INTO clientes VALUES(5, 'Marcos', 'Loyola', '24-42481327-3', 'Posadas',  
200);  
INSERT INTO clientes VALUES(6, 'María', 'Santana', '27-38771523-1', 'Obera',  
100);  
INSERT INTO clientes VALUES(7, 'Pilar', 'Ruiz', NULL, 'Sevilla', 300);  
INSERT INTO clientes VALUES(8, 'Pepe', 'Ruiz', '20-27436152-2', 'Iguazu', 200);  
INSERT INTO clientes VALUES(9, 'Guillermo', 'López', '20-42481328-1',  
'Andresito', 225);  
INSERT INTO clientes VALUES(10, 'Daniel', 'Santana', '20-40159624-3',  
'Apostoles', 125);  
  
INSERT INTO vendedores VALUES(1, 'Daniel', 'Sáez', '24-38481296-3', 0.15);  
INSERT INTO vendedores VALUES(2, 'Juan', 'Gómez', '20-39231523-3', 0.13);  
INSERT INTO vendedores VALUES(3, 'Diego', 'Flores', '20-38481221-3', 0.11);  
INSERT INTO vendedores VALUES(4, 'Marta', 'Herrera', '27-40481736-1', 0.14);  
INSERT INTO vendedores VALUES(5, 'Antonia', 'Carretero', '20-31481370-7', 0.12);  
INSERT INTO vendedores VALUES(6, 'Manuel', 'Dominguez', '20-29876342-3', 0.13);  
INSERT INTO vendedores VALUES(7, 'Antonio', 'Vega', '20-32521596-3', 0.11);  
INSERT INTO vendedores VALUES(8, 'Alfredo', 'Ruiz', '20-42436477-3', 0.05);  
  
INSERT INTO pedidos VALUES(1, 150.5, '2020-10-05', 5, 2);  
INSERT INTO pedidos VALUES(2, 270.65, '2019-09-10', 1, 5);  
INSERT INTO pedidos VALUES(3, 65.26, '2022-10-05', 2, 1);  
INSERT INTO pedidos VALUES(4, 110.5, '2021-08-17', 8, 3);  
INSERT INTO pedidos VALUES(5, 948.5, '2021-09-10', 5, 2);  
INSERT INTO pedidos VALUES(6, 2400.6, '2021-07-27', 7, 1);  
INSERT INTO pedidos VALUES(7, 5760, '2018-09-10', 2, 1);  
INSERT INTO pedidos VALUES(8, 1983.43, '2021-10-10', 4, 6);  
INSERT INTO pedidos VALUES(9, 2480.4, '2019-10-10', 8, 3);
```

```
INSERT INTO pedidos VALUES(10, 250.45, '2018-06-27', 8, 2);  
INSERT INTO pedidos VALUES(11, 75.29, '2021-08-17', 3, 7);  
INSERT INTO pedidos VALUES(12, 3045.6, '2022-04-25', 2, 1);  
INSERT INTO pedidos VALUES(13, 545.75, '2022-01-25', 6, 1);  
INSERT INTO pedidos VALUES(14, 145.82, '2022-02-02', 6, 1);  
INSERT INTO pedidos VALUES(15, 370.85, '2021-03-11', 1, 5);  
INSERT INTO pedidos VALUES(16, 2389.23, '2021-03-11', 1, 5);  
INSERT INTO pedidos VALUES(17, 275.93, '2021-08-17', 2, 5);  
INSERT INTO pedidos VALUES(18, 328.34, '2021-08-17', 1, 3);  
INSERT INTO pedidos VALUES(19, 1838.66, '2020-03-19', 3, 4);  
INSERT INTO pedidos VALUES(20, 2598.23, '2020-05-21', 5, 1);
```

### Resolver las siguientes consultas utilizando la sintaxis SQL

1. Devuelve un listado con todos los pedidos que se han realizado. Los pedidos deben estar ordenados por la fecha de realización, mostrando en primer lugar los pedidos más recientes.
2. Devuelve todos los datos de los dos pedidos de mayor valor.
3. Devuelve un listado con los identificadores de los clientes que han realizado algún pedido. Tenga en cuenta que no debe mostrar identificadores que estén repetidos.
4. Devuelve un listado de todos los pedidos que se realizaron durante el año 2022, cuya cantidad total sea superior a \$500.
5. Devuelve un listado con el nombre y apellido de los vendedores que tienen una comisión entre 0.05 y 0.11.
6. Devuelve el valor de la comisión de mayor valor que existe en la tabla vendedores.
7. Devuelve el identificador, nombre y primer apellido de aquellos clientes cuyo cuitcuil no es NULL. El listado deberá estar ordenado alfabéticamente por apellido y nombre.
8. Devuelve un listado de los nombres de los clientes que empiezan por "A" y terminan por "n" y también los nombres que empiezan por "P". El listado deberá estar ordenado alfabéticamente.
9. Devuelve un listado de los nombres de los clientes que no empiezan por "A". El listado deberá estar ordenado alfabéticamente.
10. Devuelve un listado con los nombres de los vendedores que terminan por "el" o "o". Tenga en cuenta que se deberán eliminar los nombres repetidos.
11. Devuelve un listado con el identificador, nombre y apellido de todos los clientes que han realizado algún pedido. El listado debe estar ordenado alfabéticamente y se deben eliminar los elementos repetidos.
12. Devuelve un listado que muestre todos los pedidos que ha realizado cada cliente. El resultado debe mostrar todos los datos de los pedidos y del cliente. El listado debe mostrar los datos de los clientes ordenados alfabéticamente.
13. Devuelve un listado que muestre todos los pedidos en los que ha participado un vendedor. El resultado debe mostrar todos los datos de los pedidos y de los vendedores. El listado debe mostrar los datos de los vendedores ordenados alfabéticamente.

14. Devuelve un listado que muestre todos los clientes, con todos los pedidos que han realizado y con los datos de los vendedores asociados a cada pedido.
15. Devuelve un listado de todos los clientes que realizaron un pedido durante el año 2022, cuyo monto esté entre \$300 y \$1000.
16. Devuelve el nombre y apellido de todos los vendedores que han participado en algún pedido realizado por María Santana.
17. Devuelve el nombre de todos los clientes que han realizado algún pedido con el vendedor Daniel Sáez.
18. Devuelve un listado con todos los clientes junto con los datos de los pedidos que han realizado. Este listado también debe incluir los clientes que no han realizado ningún pedido. El listado debe estar ordenado alfabéticamente por el apellido y nombre de los clientes.
19. Devuelve un listado con todos los vendedores junto con los datos de los pedidos que han realizado. Este listado también debe incluir los vendedores que no han realizado ningún pedido. El listado debe estar ordenado alfabéticamente por el apellido y nombre de los vendedores.
20. Devuelve un listado que solamente muestre los clientes que no han realizado ningún pedido.
21. Devuelve un listado que solamente muestre los vendedores que no han realizado ningún pedido.
22. Devuelve un listado con los clientes que no han realizado ningún pedido y de los vendedores que no han participado en ningún pedido. Ordene el listado alfabéticamente por el apellido y el nombre. En el listado deberá diferenciar de algún modo los clientes y los vendedores.
23. Calcula la cantidad total que suman todos los pedidos que aparecen en la tabla pedido.
24. Calcula la cantidad media de todos los pedidos que aparecen en la tabla pedido.
25. Calcula el número total de vendedores distintos que aparecen en la tabla pedido.
26. Calcula el número total de clientes que aparecen en la tabla cliente.
27. Calcula cuál es la mayor cantidad que aparece en la tabla pedido.
28. Calcula cuál es la menor cantidad que aparece en la tabla pedido.
29. Calcula cuál es el valor máximo de categoría para cada una de las ciudades que aparece en la tabla cliente.
30. Calcula cuál es el máximo valor de los pedidos realizados durante el mismo día para cada uno de los clientes. Es decir, el mismo cliente puede haber realizado varios pedidos de diferentes cantidades el mismo día. Se pide que se calcule cuál es el pedido de máximo valor para cada uno de los días en los que un cliente ha realizado un pedido. Muestra el identificador del cliente, nombre, apellido, la fecha y el valor de la cantidad.
31. Calcula cuál es el máximo valor de los pedidos realizados durante el mismo día para cada uno de los clientes, teniendo en cuenta que sólo queremos mostrar aquellos pedidos que superen la cantidad de \$2000 .
32. Calcula el máximo valor de los pedidos realizados para cada uno de los vendedores durante la fecha 2021-08-17. Muestra el identificador del vendedor, nombre, apellido y total.
33. Devuelve un listado con el identificador de cliente, nombre y apellido y el número total de pedidos que ha realizado cada uno de los clientes. Tenga en cuenta que pueden existir clientes que no han realizado ningún pedido. Estos clientes también deben aparecer en el listado indicando que el número de pedidos realizados es 0.



34. Devuelve un listado con el identificador de cliente, nombre, apellido y el número total de pedidos que ha realizado cada uno de clientes durante el año 2020.
35. Devuelve cuál ha sido el pedido de máximo valor que se ha realizado cada año.
36. Devuelve el número total de pedidos que se han realizado cada año.
37. Devuelve un listado con todos los pedidos que ha realizado Adela Salas. (Sin utilizar INNER JOIN).
38. Devuelve el número de pedidos en los que ha participado el vendedor Daniel Sáe. (Sin utilizar INNER JOIN)
39. Devuelve los datos del cliente que realizó el pedido más caro en el año 2020. (Sin utilizar INNER JOIN)
40. Devuelve la fecha y la cantidad del pedido de menor valor realizado por el cliente Pepe Ruiz Santana.
41. Devuelve un listado de los clientes que no han realizado ningún pedido. (Utilizando IN o NOT IN).
42. Devuelve un listado de los vendedores que no han realizado ningún pedido. (Utilizando IN o NOT IN).
43. Devuelve un listado de los clientes que no han realizado ningún pedido. (Utilizando EXISTS o NOT EXISTS).
44. Devuelve un listado de los vendedores que no han realizado ningún pedido. (Utilizando EXISTS o NOT EXISTS).

### Ejercicio 3:

Subir al repositorio público apellido\_nombre las respuestas tp5-eje1.sql y tp5-eje2.sql.