

SGE_UD2_ACTIVIDAD1

IES Juan José Calvo Miguel

César Fernández Zapico

UD2. ACTIVIDAD 1

Pablo Herrero Sánchez

-

2º DAM



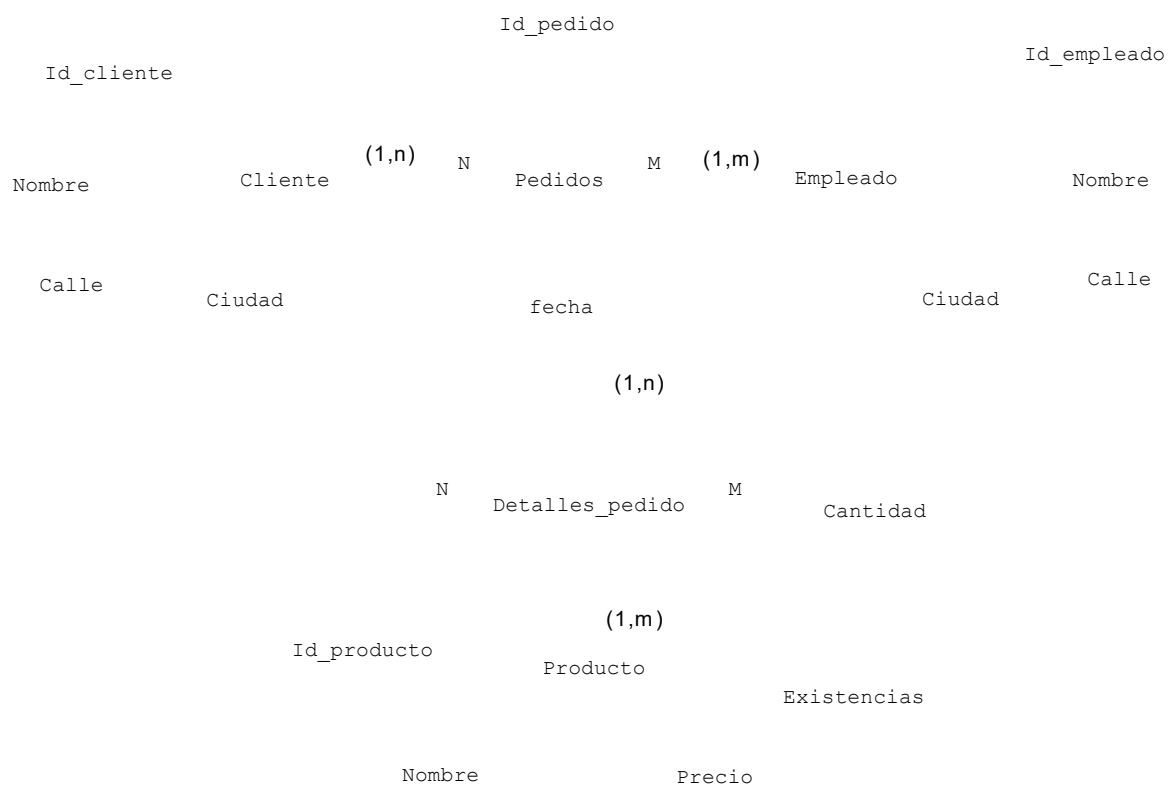
Entrega de la práctica

Para la entrega de esta práctica se deben subir dos documentos al aula virtual:

- Este mismo documento Word pasado a .pdf con las preguntas contestadas como se indica en cada caso.
- Un archivo .sql, que será la copia de la base de datos de la práctica, según se indica en el último punto.

Base de datos en PostgreSQL

Partiendo del siguiente diagrama Entidad – Relación, realizar las tareas indicadas mediante sentencias SQL:





- Crear una base de datos en PostgreSQL (también llamado Postgres) de nombre **pedidos**.

```
postgres=# CREATE DATABASE pedidos;  
CREATE DATABASE
```

- Crear las tablas correspondientes que se generan a partir del diagrama E-R, con todos sus campos, claves primarias y claves ajenas. En cuanto a los tipos de los campos, se escogen los que se considere, siempre que sean coherentes (por ejemplo, nombre puede ser varchar de 8, 13, un identificador puede ser integer o smallint, etc.)

```
postgres=# \connect pedidos  
You are now connected to database "pedidos" as user "postgres".  
pedidos=# create table cliente (id_cliente serial PRIMARY KEY not null,nombre VA  
RCHAR(50),calle VARCHAR(200),ciudad VARCHAR(50));  
CREATE TABLE  
pedidos=# create table empleado (id_empleado serial PRIMARY KEY not null, nombre  
VARCHAR(50), calle VARCHAR(200), ciudad VARCHAR(50));  
CREATE TABLE
```

```
pedidos=# create table pedidos (id_pedido serial PRIMARY KEY not null, id_cliente int  
references cliente(id_cliente), id_empleado int references empleado(id_empleado), fech  
a date default current_timestamp);  
CREATE TABLE  
pedidos=# create table producto (id_producto serial PRIMARY KEY not null, nombre VARCH  
AR(50), precio decimal(8,2), existencias int);  
CREATE TABLE
```

```
pedidos=# create table detalles_pedido (id_pedido int references pedidos(id_pedido), id_produ  
cto int references producto(id_producto), cantidad int, primary key(id_pedido, id_producto));  
CREATE TABLE
```

```
pedidos=# \dt  
List of relations  
Schema | Name | Type | Owner  
-----+-----+-----+-----  
public | cliente | table | postgres  
public | detalles_pedido | table | postgres  
public | empleado | table | postgres  
public | pedidos | table | postgres  
public | producto | table | postgres  
(5 rows)
```



- Insertar datos en cada una de las tablas, con valores a elegir. Como mínimo, insertar 5 clientes, 6 empleados, 5 productos, realizar 6 pedidos y dar detalles de al menos 3 pedidos. Se debe tener en cuenta que **en el mismo pedido** se pueden solicitar distintos productos y cantidades, lo cual va especificado en 'detalles_pedido'.

Clientes:

```
pedidos=# INSERT INTO cliente (nombre, calle, ciudad) VALUES ('Manuel García', 'Avda. Asturias 12, 2ºB', 'León');
INSERT 0 1
pedidos=# INSERT INTO cliente (nombre, calle, ciudad) VALUES ('Rosa Sánchez', 'Calle Uría 1, 6ºC', 'Oviedo');
INSERT 0 1
pedidos=# INSERT INTO cliente (nombre, calle, ciudad) VALUES ('Larraitz Urrikalde', 'Calle Loyola 10, bajo', 'San Sebastián');
INSERT 0 1
```

```
pedidos=# INSERT INTO cliente (nombre, calle, ciudad) VALUES ('Jesús Pérez', 'Avda del Cristo 1, 1ºA', 'Oviedo');
INSERT 0 1
pedidos=# INSERT INTO cliente (nombre, calle, ciudad) VALUES ('María Merayo', 'Calle Cervantes 10, 2ºD', 'Bembibre');
INSERT 0 1
```

id_cliente	nombre	calle	ciudad
1	Manuel García	Avda. Asturias 12, 2ºB	León
2	Rosa Sánchez	Calle Uría 1, 6ºC	Oviedo
3	Larraitz Urrikalde	Calle Loyola 10, bajo	San Sebastián
7	Jesús Pérez	Avda del Cristo 1, 1ºA	Oviedo
8	María Merayo	Calle Cervantes 10, 2ºD	Bembibre

(5 rows)



Empleados:

```
pedidos=# INSERT INTO empleado (nombre, calle, ciudad) VALUES ('Carlos López', 'Avenida de Java, 3, 4ºA', 'Avilés');
INSERT 0 1
pedidos=# INSERT INTO empleado (nombre, calle, ciudad) VALUES ('Jose García', 'Calle Salamanca 2, bajo', 'Luarca');
INSERT 0 1
pedidos=# INSERT INTO empleado (nombre, calle, ciudad) VALUES ('Laura Pérez', 'Avenida de Python, 26, bajo', 'Oviedo');
INSERT 0 1
pedidos=# INSERT INTO empleado (nombre, calle, ciudad) VALUES ('Fausto González', 'Avenida Libertad, 10, 10ºA', 'San Sebastián');
INSERT 0 1
pedidos=# INSERT INTO empleado (nombre, calle, ciudad) VALUES ('Álvaro Zubizarreta', 'Calle de las huertas, 10, bajo', 'León');
INSERT 0 1
pedidos=# INSERT INTO empleado (nombre, calle, ciudad) VALUES ('Juana Álvarez', 'Avenida Palanquinos, 2, 1ºC', 'Valencia de Don Juan');
INSERT 0 1
```

id_empleado	nombre	calle	ciudad
1	Carlos López	Avenida de Java, 3, 4ºA	Avilés
2	Jose García	Calle Salamanca 2, bajo	Luarca
3	Laura Pérez	Avenida de Python, 26, bajo	Oviedo
4	Fausto González	Avenida Libertad, 10, 10ºA	San Sebastián
5	Álvaro Zubizarreta	Calle de las huertas, 10, bajo	León
6	Juana Álvarez	Avenida Palanquinos, 2, 1ºC	Valencia de Don Juan

(6 rows)



Productos:

```
pedidos=# INSERT INTO producto (nombre, precio, existencias) VALUES ('Pera extra 1
kg', 2.49, 50);
INSERT 0 1
pedidos=# INSERT INTO producto (nombre, precio, existencias) VALUES ('Manzana Gold
en 1kg', 1.55, 80);
INSERT 0 1
pedidos=# INSERT INTO producto (nombre, precio, existencias) VALUES ('Kiwi Zespi 1
kg', 3.56, 30);
INSERT 0 1
pedidos=# INSERT INTO producto (nombre, precio, existencias) VALUES ('Plátano de C
anarias 1kg', 3.99, 80);
INSERT 0 1
pedidos=# INSERT INTO producto (nombre, precio, existencias) VALUES ('Naranja Prem
ium 1kg', 1.85, 200);
INSERT 0 1
```

id_producto	nombre	precio	existencias
1	Pera extra 1kg	2.49	50
2	Manzana Golden 1kg	1.55	80
3	Kiwi Zespi 1kg	3.56	30
4	Plátano de Canarias 1kg	3.99	80
5	Naranja Premium 1kg	1.85	200

(5 rows)

Pedidos:

```
pedidos=# INSERT INTO pedidos (id_cliente, id_empleado) VALUES (1,6);
INSERT 0 1
pedidos=# INSERT INTO pedidos (id_cliente, id_empleado) VALUES (2,5);
INSERT 0 1
pedidos=# INSERT INTO pedidos (id_cliente, id_empleado) VALUES (3,4);
INSERT 0 1
pedidos=# INSERT INTO pedidos (id_cliente, id_empleado) VALUES (7,3);
INSERT 0 1
pedidos=# INSERT INTO pedidos (id_cliente, id_empleado) VALUES (8,2);
INSERT 0 1
pedidos=# INSERT INTO pedidos (id_cliente, id_empleado) VALUES (7,1);
INSERT 0 1
```

id_pedido	id_cliente	id_empleado	fecha
1	1	6	2020-11-16
2	2	5	2020-11-16
3	3	4	2020-11-16
4	7	3	2020-11-16
5	8	2	2020-11-16
6	7	1	2020-11-16

(6 rows)



Detalles pedidos:

```
pedidos=# INSERT INTO detalles_pedido (id_pedido, id_producto, cantidad) VALUES (1, 1, 20);
INSERT 0 1
pedidos=# INSERT INTO detalles_pedido (id_pedido, id_producto, cantidad) VALUES (1, 2, 5);
INSERT 0 1
pedidos=# INSERT INTO detalles_pedido (id_pedido, id_producto, cantidad) VALUES (2, 4, 2);
INSERT 0 1
pedidos=# INSERT INTO detalles_pedido (id_pedido, id_producto, cantidad) VALUES (2, 5, 3);
INSERT 0 1
pedidos=# INSERT INTO detalles_pedido (id_pedido, id_producto, cantidad) VALUES (5, 2, 1);
INSERT 0 1
pedidos=# INSERT INTO detalles_pedido (id_pedido, id_producto, cantidad) VALUES (5, 5, 2);
INSERT 0 1
```

id_pedido	id_producto	cantidad
1	1	20
1	2	5
2	4	2
2	5	3
5	2	1
5	5	2

(6 rows)

- Indicar las sentencias necesarias para realizar las siguientes acciones, pegando también un pantallazo del resultado:

1. Mostrar todas las tablas de la base de datos **pedidos**.

```
pedidos=# \dt
          List of relations
Schema |      Name      | Type  | Owner
-----+-----+-----+-----
public | cliente        | table | postgres
public | detalles_pedido | table | postgres
public | empleado       | table | postgres
public | pedidos        | table | postgres
public | producto       | table | postgres
(5 rows)
```

2. Obtener el id_pedido y el nombre de cliente para todos los pedidos.

```
pedidos=# SELECT id_pedido, nombre FROM pedidos INNER JOIN cliente ON pedidos.id_cliente=cliente.id_cliente;
 id_pedido |      nombre
-----+-----
          1 | Manuel García
          2 | Rosa Sánchez
          3 | Larraitz Urrikalde
          4 | Jesús Pérez
          5 | María Merayo
          6 | Jesús Pérez
(6 rows)
```




3. Obtener el id_pedido, el nombre del cliente, el nombre de los productos y la cantidad de los mismos para todos los pedidos.

```
pedidos=# SELECT pedidos.id_pedido, cliente.nombre as cliente, producto.nombre as producto, detalles_pedido.cantidad FROM pedidos INNER JOIN cliente ON pedidos.id_cliente=cliente.id_cliente INNER JOIN detalles_pedido ON pedidos.id_pedido=detalles_pedido.id_pedido INNER JOIN producto ON detalles_pedido.id_producto=producto.id_producto ORDER BY id_pedido;
```

id_pedido	cliente	producto	cantidad
1	Manuel García	Pera extra 1kg	20
1	Manuel García	Manzana Golden 1kg	5
2	Rosa Sánchez	Plátano de Canarias 1kg	2
2	Rosa Sánchez	Naranja Premium 1kg	3
5	María Merayo	Manzana Golden 1kg	1
5	María Merayo	Naranja Premium 1kg	2

(6 rows)

4. Obtener los nombres de empleados o clientes que viven en una de las ciudades que hayas insertado (uso de 'union').

```
pedidos=# SELECT nombre as clientes_Y_Empleados_Oviedo FROM CLIENTE where lower(ciudad)='oviedo' UNION SELECT nombre FROM empleado where lower(ciudad)='oviedo';
```

clientes_y_empleados_oviedo
Jesús Pérez
Laura Pérez
Rosa Sánchez

(3 rows)

5. Obtener el número de pedidos atendidos por uno de los empleados que hayas insertado (uso de 'count'). Recuerda que discrimina tildes.

- Como no tengo ningún empleado con más de 1 pedido, **inserto un nuevo pedido para el empleado con id 1:**

```
pedidos=# INSERT INTO pedidos (id_cliente, id_empleado) VALUES (7,1);  
INSERT 0 1
```

- Ahora calculo el número de pedidos realizados por el empleado con id 1:

```
pedidos=# SELECT count(id_pedido) as pedidos_emple_1 FROM pedidos WHERE id_empleado='1';
```

pedidos_emple_1
2

(1 row)



6. Obtener el identificador de pedido, fecha del mismo, la cantidad total de productos que incluye y la media de unidades de cada producto (uso de sum, avg y group by).

- Vista de la sentencia en el editor de PgAdmin, donde se ve más clara. Uso subconsultas porque es la única forma que he encontrado para mostrar la media de cada producto dentro de cada pedido:

```
SELECT pedidos.id_pedido, fecha, sum(detalles_pedido.cantidad) as total_productos,
(SELECT round(avg(cantidad),2) from DETALLES_PEDIDO WHERE id_producto=1 AND pedidos.id_pedido=detalles_pedido.id_pedido) as media_peraExtra1kg,
(SELECT round(avg(cantidad),2) from DETALLES_PEDIDO WHERE id_producto=2 AND pedidos.id_pedido=detalles_pedido.id_pedido) as media_manzanaGolden1kg,
(SELECT round(avg(cantidad),2) from DETALLES_PEDIDO WHERE id_producto=3 AND pedidos.id_pedido=detalles_pedido.id_pedido) as media_kiwiZespilkg,
(SELECT round(avg(cantidad),2) from DETALLES_PEDIDO WHERE id_producto=4 AND pedidos.id_pedido=detalles_pedido.id_pedido) as media_platanoCanarias1kg,
(SELECT round(avg(cantidad),2) from DETALLES_PEDIDO WHERE id_producto=5 AND pedidos.id_pedido=detalles_pedido.id_pedido) as media_naranjaPremium1kg
FROM pedidos, detalles_pedido
GROUP BY pedidos.id_pedido, detalles_pedido.id_pedido HAVING pedidos.id_pedido=detalles_pedido.id_pedido;
```

- Ejecución y resultado en consola:

```
pedidos=# SELECT pedidos.id_pedido, fecha, sum(detalles_pedido.cantidad) as total_productos, (SELECT round(avg(cantidad),2) from DETALLES_PEDIDO WHERE id_producto=1 AND pedidos.id_pedido=detalles_pedido.id_pedido) as media_peraExtra1kg, (SELECT round(avg(cantidad),2) from DETALLES_PEDIDO WHERE id_producto=2 AND pedidos.id_pedido=detalles_pedido.id_pedido) as media_manzanaGolden1kg, (SELECT round(avg(cantidad),2) from DETALLES_PEDIDO WHERE id_producto=3 AND pedidos.id_pedido=detalles_pedido.id_pedido) as media_kiwiZespilkg, (SELECT round(avg(cantidad),2) from DETALLES_PEDIDO WHERE id_producto=4 AND pedidos.id_pedido=detalles_pedido.id_pedido) as media_platanoCanarias1kg, (SELECT round(avg(cantidad),2) from DETALLES_PEDIDO WHERE id_producto=5 AND pedidos.id_pedido=detalles_pedido.id_pedido) as media_naranjaPremium1kg FROM pedidos, detalles_pedido GROUP BY pedidos.id_pedido, detalles_pedido.id_pedido HAVING pedidos.id_pedido=detalles_pedido.id_pedido;
 id_pedido | fecha      | total_productos | media_peraextra1kg | media_manzanagolden1kg | media_kiwiZespilkg | media_platanocanarias1kg | media_naranjapremium1kg
-----+-----+-----+-----+-----+-----+-----+-----
1 | 2020-11-16 | 25 | 20.00 | 5.00 |  |  | 
2 | 2020-11-16 | 5 |  |  |  | 2.00 | 3.00
5 | 2020-11-16 | 3 |  | 1.00 |  |  | 2.00
(3 rows)
```

- Hacer una copia de la base de datos con **pg_dump** y adjuntarla a la entrega de la práctica. Para ello:

- Hacerlo desde el usuario 'postgres' con la orden:
`pg_dump base_de_datos > nombre_alumno.sql`

- Hago la copia sobre /tmp, en /home postgres no tiene permisos de escritura.

```
postgres@ubuntu-server:/$ cd /tmp
postgres@ubuntu-server:/tmp$ pg_dump pedidos > pablo_herrero.sql
postgres@ubuntu-server:/tmp$
```

- Se puede pasar a Windows el archivo .sql de copia para enviarlo utilizando **WinSCP**. Para saber la ruta del usuario postgres en el WinSCP, recuerda que se puede usar el comando **pwd** en el servidor.

