



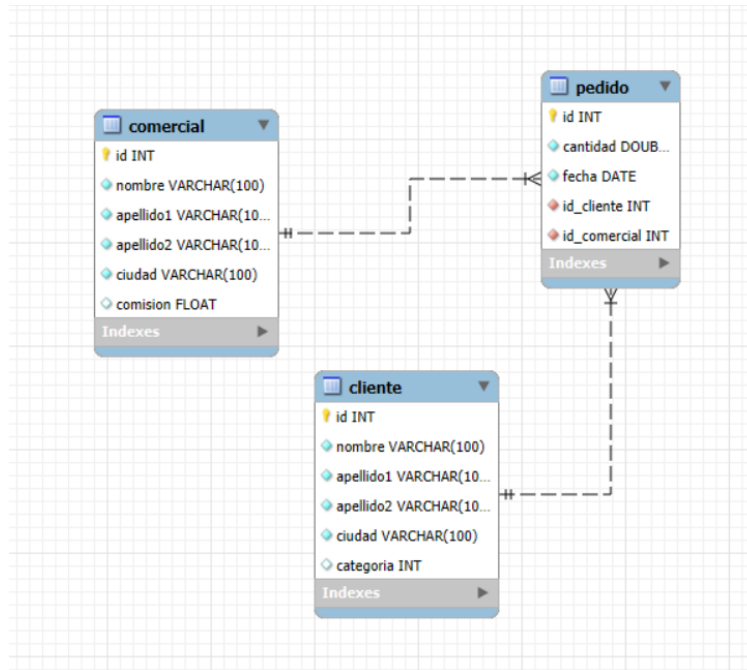
UNIVERSIDAD DE
CUNDINAMARCA
Generación Siglo 21

EJERCICIO BD

PRESENTADO POR:
NEISSY ALEJANDRA MEDINA CUBILLOS

PRESENTADO A:
WILLIAM ALEXANDER MATALLANA

UNIVERSIDAD DE CUNDINAMARCA
EXTENSION CHIA
INGENIERIA DE SISTEMAS
LINEA III
2025



1. Crear base de datos

```
mysql> create database Ejercicio_BD;
Query OK, 1 row affected (0.007 sec)
```

2. Crear tablas con las respectivas relaciones

```
mysql> create table comercial(id int auto_increment primary key,nombre varchar(100) not null, apellido1 varchar(100) not null , apellido2 varchar(100) not null,ciudad varchar(100) not null , comision float);
Query OK, 0 rows affected (0.046 sec)
```

```
mysql> create table cliente(id int auto_increment primary key,nombre varchar(100) not null, apellido1 varchar(100) not null , apellido2 varchar(100) not null,ciudad varchar(100) not null , categoria int(10));
Query OK, 0 rows affected, 1 warning (0.041 sec)

mysql>
```

```
mysql> create table pedido(id int auto_increment primary key, cantidad double not null, fecha date not null, id_cliente int(10) not null, foreign key(id_cliente) references cliente(id), id_comercial int(10) not null, foreign key(id_comercial) references comercial(id));
Query OK, 0 rows affected, 2 warnings (0.142 sec)

mysql>
```



```
mysql> describe cliente;
```

Field	Type	Null	Key	Default	Extra
id	int	NO	PRI	NULL	auto_increment
nombre	varchar(100)	NO		NULL	
apellido1	varchar(100)	NO		NULL	
apellido2	varchar(100)	NO		NULL	
ciudad	varchar(100)	NO		NULL	
categoria	int	YES		NULL	

```
6 rows in set (0.010 sec)
```

```
mysql> describe comercial;
```

Field	Type	Null	Key	Default	Extra
id	int	NO	PRI	NULL	auto_increment
nombre	varchar(100)	NO		NULL	
apellido1	varchar(100)	NO		NULL	
apellido2	varchar(100)	NO		NULL	
ciudad	varchar(100)	NO		NULL	
comision	float	YES		NULL	

```
6 rows in set (0.002 sec)
```

```
mysql> describe pedido;
```

Field	Type	Null	Key	Default	Extra
id	int	NO	PRI	NULL	auto_increment
cantidad	double	NO		NULL	
fecha	date	NO		NULL	
id_cliente	int	NO	MUL	NULL	
id_comercial	int	NO	MUL	NULL	

```
5 rows in set (0.012 sec)
```

```
mysql>
```



3. Insertar 10 registros en cada tabla

```
mysql> insert into comercial(nombre,apellido1,apellido2,ciudad,comision)values("Alejandra","Medina","Cubillos","Chia",28000);
Query OK, 1 row affected (0.083 sec)

mysql> insert into comercial(nombre,apellido1,apellido2,ciudad,comision)values("Daniel","Leon","Higuera","Chia",52000);
Query OK, 1 row affected (0.014 sec)

mysql> insert into comercial(nombre,apellido1,apellido2,ciudad,comision)values("Daniela","Velasquez","Lopez","Funza",1500);
Query OK, 1 row affected (0.014 sec)

mysql> insert into comercial(nombre,apellido1,apellido2,ciudad,comision)values("Nuri","Gonzalez","Miranda","Bogota",23300);
Query OK, 1 row affected (0.013 sec)

mysql> insert into comercial(nombre,apellido1,apellido2,ciudad,comision)values("Catalina","Castro","Ruiz","Tunja",5500);
Query OK, 1 row affected (0.013 sec)

mysql> insert into comercial(nombre,apellido1,apellido2,ciudad,comision)values("Sandra","Espinosa","Agudelo","Madrid",30500);
Query OK, 1 row affected (0.014 sec)

mysql> insert into comercial(nombre,apellido1,apellido2,ciudad,comision)values("Camilo","Gomez","Garcia","Mosquera",18250);
Query OK, 1 row affected (0.013 sec)

mysql> insert into comercial(nombre,apellido1,apellido2,ciudad,comision)values("Tatiana","Angarita","Morales","Cajica",366950);
Query OK, 1 row affected (0.014 sec)

mysql> insert into comercial(nombre,apellido1,apellido2,ciudad,comision)values("Nathalia","Rozo","Mendoza","Zipaquirá",24650);
Query OK, 1 row affected (0.014 sec)

mysql> insert into comercial(nombre,apellido1,apellido2,ciudad,comision)values("David","Bermudez","Angel","Facatativa",10200);
Query OK, 1 row affected (0.014 sec)

mysql>
```

```
mysql> insert into cliente(nombre,apellido1,apellido2,ciudad,categoria)values("Ivan","Bedoya","Velandia","Facatativa",1);
Query OK, 1 row affected (0.018 sec)

mysql> insert into cliente(nombre,apellido1,apellido2,ciudad,categoria)values("Dayana","Guzman","Casallas","Funza",2);
Query OK, 1 row affected (0.004 sec)

mysql> insert into cliente(nombre,apellido1,apellido2,ciudad,categoria)values("Amelia","Santos","Arevalo","Pereira",3);
Query OK, 1 row affected (0.004 sec)

mysql> insert into cliente(nombre,apellido1,apellido2,ciudad,categoria)values("Jeronimo","Duran","De la torre","Casanare",4);
Query OK, 1 row affected (0.003 sec)

mysql> insert into cliente(nombre,apellido1,apellido2,ciudad,categoria)values("Michell","Acevedo","Hernandez","Guajira",5);
Query OK, 1 row affected (0.003 sec)

mysql> insert into cliente(nombre,apellido1,apellido2,ciudad,categoria)values("Hugo","Blanco","Olivera","Neiva",6);
Query OK, 1 row affected (0.013 sec)

mysql> insert into cliente(nombre,apellido1,apellido2,ciudad,categoria)values("Juan","Bohorquez","Jaramillo","Tolima",7);
Query OK, 1 row affected (0.004 sec)

mysql> insert into cliente(nombre,apellido1,apellido2,ciudad,categoria)values("Marina","Coronado","Gil","Sogamoso",8);
Query OK, 1 row affected (0.014 sec)

mysql> insert into cliente(nombre,apellido1,apellido2,ciudad,categoria)values("Luana","Vargas","Gutierrez","Amazonas",9);
Query OK, 1 row affected (0.005 sec)

mysql> insert into cliente(nombre,apellido1,apellido2,ciudad,categoria)values("Felipe","Martinez","Yepes","Barranquilla",10);
Query OK, 1 row affected (0.014 sec)

mysql>
```



```
mysql> insert into pedido(cantidad,fecha,id_cliente,id_comercial)values(120,'2025-09-23',1,2);
Query OK, 1 row affected (0.003 sec)

mysql> insert into pedido(cantidad,fecha,id_cliente,id_comercial)values(2420,'2023-07-18',2,3);
Query OK, 1 row affected (0.003 sec)

mysql> insert into pedido(cantidad,fecha,id_cliente,id_comercial)values(580,'2020-06-27',3,4);
Query OK, 1 row affected (0.004 sec)

mysql> insert into pedido(cantidad,fecha,id_cliente,id_comercial)values(1056,'2000-11-17',4,5);
Query OK, 1 row affected (0.014 sec)

mysql> insert into pedido(cantidad,fecha,id_cliente,id_comercial)values(80,'2003-11-11',5,6);
Query OK, 1 row affected (0.014 sec)

mysql> insert into pedido(cantidad,fecha,id_cliente,id_comercial)values(170,'2010-10-31',6,7);
Query OK, 1 row affected (0.012 sec)

mysql> insert into pedido(cantidad,fecha,id_cliente,id_comercial)values(170,'2009-08-28',7,8);
Query OK, 1 row affected (0.013 sec)

mysql> insert into pedido(cantidad,fecha,id_cliente,id_comercial)values(52,'2022-07-05',8,9);
Query OK, 1 row affected (0.015 sec)

mysql> insert into pedido(cantidad,fecha,id_cliente,id_comercial)values(36,'2025-06-29',2,1);
Query OK, 1 row affected (0.003 sec)

mysql> insert into pedido(cantidad,fecha,id_cliente,id_comercial)values(106,'2024-10-05',1,3);
Query OK, 1 row affected (0.014 sec)

mysql>
```

4. 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.

```
mysql> SELECT*FROM pedido
-> ORDER BY fecha DESC;
+----+-----+-----+-----+-----+
| id | cantidad | fecha      | id_cliente | id_comercial |
+----+-----+-----+-----+-----+
| 1  | 120      | 2025-09-23 | 1          | 2            |
| 9  | 36       | 2025-06-29 | 2          | 1            |
| 10 | 106      | 2024-10-05 | 1          | 3            |
| 2  | 2420     | 2023-07-18 | 2          | 3            |
| 8  | 52       | 2022-07-05 | 8          | 9            |
| 3  | 580      | 2020-06-27 | 3          | 4            |
| 6  | 170      | 2010-10-31 | 6          | 7            |
| 7  | 170      | 2009-08-28 | 7          | 8            |
| 5  | 80       | 2003-11-11 | 5          | 6            |
| 4  | 1056     | 2000-11-17 | 4          | 5            |
+----+-----+-----+-----+-----+
10 rows in set (0.032 sec)

mysql>
```

5. Listar todos los datos de los dos pedidos de mayor valor.

```
mysql> SELECT
-> *
-> FROM pedido
-> ORDER BY
-> cantidad DESC
-> LIMIT 2;
+-----+-----+-----+-----+-----+
| id | cantidad | fecha      | id_cliente | id_comercial |
+-----+-----+-----+-----+-----+
| 2 | 2420 | 2023-07-18 | 2 | 3 |
| 4 | 1056 | 2000-11-17 | 4 | 5 |
+-----+-----+-----+-----+-----+
2 rows in set (0.001 sec)

mysql>
```

6. 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.

```
mysql> SELECT DISTINCT id_cliente
-> FROM pedido;
+-----+
| id_cliente |
+-----+
| 1 |
| 2 |
| 3 |
| 4 |
| 5 |
| 6 |
| 7 |
| 8 |
+-----+
8 rows in set (0.075 sec)

mysql>
```

7. Devuelve el valor de la comisión de mayor valor que existe en la tabla comercial

```
mysql> SELECT * FROM comercial WHERE comision = (SELECT MAX(comision) FROM comercial);
+-----+-----+-----+-----+-----+-----+
| id | nombre | apellido1 | apellido2 | ciudad | comision |
+-----+-----+-----+-----+-----+-----+
| 8 | Tatiana | Angarita | Morales | Cajica | 366950 |
+-----+-----+-----+-----+-----+-----+
1 row in set (0.101 sec)

mysql>
```

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.

```
mysql> SELECT nombre
-> FROM cliente
-> WHERE REGEXP_LIKE(nombre, '^(A.*N|P.*)$')
-> ORDER BY nombre;
Empty set (0.058 sec)

mysql> |
```

9. Devuelve un listado con el identificador, nombre y los apellidos de todos los clientes que han realizado algún pedido. El listado debe estar ordenado alfabéticamente y se deben eliminar los elementos repetidos

```
mysql> SELECT DISTINCT
-> c.id,
-> c.nombre,
-> c.apellido1,
-> c.apellido2
-> FROM cliente AS c
-> INNER JOIN pedido AS p
-> ON p.id_cliente = c.id -- usa aqu el nombre real de la FK
-> ORDER BY
-> c.nombre ASC,
-> c.apellido1 ASC,
-> c.apellido2 ASC,
-> c.id ASC;

+----+-----+-----+-----+
| id | nombre | apellido1 | apellido2 |
+----+-----+-----+-----+
| 3 | Amelia | Santos | Arevalo |
| 2 | Dayana | Guzman | Casallas |
| 6 | Hugo | Blanco | Olivera |
| 1 | Ivan | Bedoya | Velandia |
| 4 | Jeronimo | Duran | De la torre |
| 7 | Juan | Bohorquez | Jaramillo |
| 8 | Marina | Coronado | Gil |
| 5 | Michell | Acevedo | Hernandez |
+----+-----+-----+-----+
8 rows in set (0.038 sec)

mysql>
```

10. 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.

```
mysql> SELECT
-> c.*,
-> p.*,
-> FROM cliente AS c
-> INNER JOIN pedido AS p
-> ON p.id_cliente = c.id
-> ORDER BY
-> c.nombre ASC,
-> c.apellido1 ASC,
-> c.apellido2 ASC,
-> c.id ASC;
```

id	nombre	apellido1	apellido2	ciudad	categoria	id	cantidad	fecha	id_cliente	id_comercial
3	Amelia	Santos	Arevalo	Pereira	3	3	580	2020-06-27	3	4
2	Dayana	Guzman	Casallas	Funza	2	2	2420	2023-07-18	2	3
2	Dayana	Guzman	Casallas	Funza	2	9	36	2025-06-29	2	1
6	Hugo	Blanco	Olivera	Neiva	6	6	170	2010-10-31	6	7
1	Ivan	Bedoya	Velandia	Facatativa	1	1	120	2025-09-23	1	2
1	Ivan	Bedoya	Velandia	Facatativa	1	10	106	2024-10-05	1	3
4	Jeronimo	Duran	De la torre	Casanare	4	4	1056	2000-11-17	4	5
7	Juan	Bohorquez	Jaramillo	Tolima	7	7	170	2009-08-28	7	8
8	Marina	Coronado	Gil	Sogamoso	8	8	52	2022-07-05	8	9
5	Michell	Acevedo	Hernandez	Guajira	5	5	80	2003-11-11	5	6

10 rows in set (0.018 sec)

```
mysql>
```

11. Devuelve un listado que muestre todos los clientes, con todos los pedidos que han realizado y con los datos de los comerciales asociados a cada pedido.

```
mysql> SELECT
-> c.*,
-> p.*,
-> v.*,
-> FROM cliente AS c
-> LEFT JOIN pedido AS p
-> ON p.id_cliente = c.id
-> LEFT JOIN comercial AS v
-> ON v.id = p.id_comercial
-> ORDER BY
-> c.nombre ASC,
-> c.apellido1 ASC,
-> c.apellido2 ASC,
-> c.id ASC;
```

id	nombre	apellido1	apellido2	ciudad	categoria	id	cantidad	fecha	id_cliente	id_comercial	id	nombre	apellido1	apellido2	ciudad	comision
3	Amelia	Santos	Arevalo	Pereira	3	3	580	2020-06-27	3	4	4	Nuri	Gonzalez	Miranda	Bogota	23300
2	Dayana	Guzman	Casallas	Funza	2	2	2420	2023-07-18	2	3	3	Daniela	Velasquez	Lopez	Funza	1900
2	Dayana	Guzman	Casallas	Funza	2	9	36	2025-06-29	2	1	1	Alejandra	Medina	Cubillos	Chia	28000
10	Felipe	Martinez	Yepes	Barranquilla	10	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL
6	Hugo	Blanco	Olivera	Neiva	6	6	170	2010-10-31	6	7	7	Camilo	Gomez	Garcia	Mosquera	16250
1	Ivan	Bedoya	Velandia	Facatativa	1	1	120	2025-09-23	1	2	2	Daniel	Leon	Higuera	Chia	50000
1	Ivan	Bedoya	Velandia	Facatativa	1	10	106	2024-10-05	1	3	3	Daniela	Velasquez	Lopez	Funza	1500
4	Jeronimo	Duran	De la torre	Casanare	4	4	1056	2000-11-17	4	5	5	Catalina	Castro	Ruiz	Tunja	5500
7	Juan	Bohorquez	Jaramillo	Tolima	7	7	170	2009-08-28	7	8	8	Tatiana	Angarita	Morales	Cajica	366950
9	Luana	Vargas	Gutierrez	Amazonas	9	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL
8	Marina	Coronado	Gil	Sogamoso	8	8	52	2022-07-05	8	9	9	Nathalia	Rozo	Mendoza	Zipaquirá	24650
5	Michell	Acevedo	Hernandez	Guajira	5	5	80	2003-11-11	5	6	6	Sandra	Espinosa	Agudelo	Madrid	30500

12 rows in set (0.006 sec)

```
mysql>
```


12. 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 primer apellido, segundo apellido y nombre de los clientes.

```
mysql> SELECT
--> c.id, -- identificador del cliente
--> c.nombre, -- nombre del cliente
--> c.apellido1, -- primer apellido
--> c.apellido2, -- segundo apellido
--> c.ciudad, -- ciudad del cliente
--> c.categoria, -- categoria del cliente (puede ser NULL)
--> p.*, -- todas las columnas del pedido; sern NULL si no hay pedidos
--> FROM cliente AS c
--> -- LEFT JOIN asegura que se muestren TODOS los clientes,
--> -- aunque no tengan filas coincidentes en pedido.
--> LEFT JOIN pedido AS p
--> -- Reemplazar p.id_cliente por el nombre REAL de la FK hacia cliente.id
--> ON p.id_cliente = c.id
--> -- Ordena alfabeticamente por apellidos y nombre;
--> -- c.id se aade como criterio de desempate estable.
--> ORDER BY
--> c.apellido1 ASC,
--> c.apellido2 ASC,
--> c.nombre ASC,
--> c.id ASC;
```

id	nombre	apellido1	apellido2	ciudad	categoria	id	cantidad	fecha	id_cliente	id_comercial
5	Michell	Acevedo	Hernandez	Guajira	5	5	80	2003-11-11	5	6
1	Ivan	Bedoya	Velandia	Facatativa	1	1	120	2025-09-23	1	2
1	Ivan	Bedoya	Velandia	Facatativa	1	10	106	2024-10-05	1	3
6	Hugo	Blanco	Olivera	Neiva	6	6	170	2010-10-31	6	7
7	Juan	Bohorquez	Jaramillo	Tolima	7	7	170	2009-08-28	7	8
8	Marina	Coronado	Gil	Sogamoso	8	8	52	2022-07-05	8	9
4	Jeronimo	Duran	De la torre	Casanare	4	4	1056	2000-11-17	4	5
2	Dayana	Guzman	Casallas	Funza	2	2	2420	2023-07-18	2	3
2	Dayana	Guzman	Casallas	Funza	2	9	36	2025-06-29	2	1
10	Felipe	Martinez	Yepes	Barranquilla	10	NULL	NULL	NULL	NULL	NULL
3	Amelia	Santos	Arevalo	Pereira	3	3	580	2020-06-27	3	4
9	Luana	Vargas	Gutierrez	Amazonas	9	NULL	NULL	NULL	NULL	NULL

12 rows in set (0.002 sec)

```
mysql>
```

Documentado

13. Calcula la cantidad total que suman todos los pedidos que aparecen en la tabla pedido

```
mysql> SELECT COALESCE(SUM(cantidad), 0) AS total_pedidos
--> FROM pedido;
```

total_pedidos
4790

1 row in set (0.015 sec)

```
mysql>
```



14. Calcula la cantidad media de todos los pedidos que aparecen en la tabla pedido

```
mysql> SELECT AVG(cantidad) AS media_pedidos
-> FROM pedido;
+-----+
| media_pedidos |
+-----+
|          479 |
+-----+
1 row in set (0.003 sec)

mysql> |
```

15. Calcula cuál es la mayor cantidad que aparece en la tabla pedido

```
mysql> SELECT MAX(cantidad) AS mayor_cantidad
-> FROM pedido;
+-----+
| mayor_cantidad |
+-----+
|          2420 |
+-----+
1 row in set (0.015 sec)

mysql>
```

16. Calcula cuál es la menor cantidad que aparece en la tabla pedido

```
mysql> SELECT MIN(cantidad) AS menor_cantidad
-> FROM pedido;
+-----+
| menor_cantidad |
+-----+
|           36 |
+-----+
1 row in set (0.001 sec)

mysql>
```