

TECNOVENTAS S.A.




Escenario

TECNOVENTAS S.A. es una empresa dedicada a la comercialización de productos tecnológicos. Utiliza un sistema MySQL para gestionar clientes, productos, ventas y detalles de ventas.

1. Vistas previas de tablas disponibles




Clientes

```
1 • use ventas_db;
2   -- VISTAS PREVIAS
3   -- 1. Clientes
4 • select * from clientes;
5
```

Result Grid				
Filter Rows: <input type="text"/>				
Edit:    Exp				
cliente_id	nombre	correo	ciudad	
1	Paola Herrera	paola.herrera@tecnoventas.com	Manta	
2	Paola Pérez	paola.pérez@tecnoventas.com	Guayaquil	
3	Carlos Gómez	carlos.gómez@tecnoventas.com	Guayaquil	
4	Gabriel Ramírez	gabriel.ramírez@tecnoventas.com	Guayaquil	
5	Lucía Luna	lucía.luna@tecnoventas.com	Ambato	
6	Lucía Luna	lucía.luna@tecnoventas.com	Guayaquil	
7	María Pérez	maría.pérez@tecnoventas.com	Loja	

Productos

```
5   -- 2. Productos
6 • select * from productos;
```

Result Grid				
Filter Rows: <input type="text"/>				
Edit:   				
producto_id	nombre	precio	stock	
1	Memoria RAM 16GB Modelo 714	233.65	36	
2	Router TP-Link Modelo 150	488.65	10	
3	Cámara Web Full HD Modelo 753	439.80	17	
4	Impresora HP Modelo 727	705.36	39	
5	Memoria RAM 16GB Modelo 670	317.89	43	
6	Router TP-Link Modelo 393	771.00	19	
7	Laptop Lenovo Modelo 890	579.14	7	
8	Audífonos Bluetooth Modelo 453	568.57	38	
9	Router TP-Link Modelo 854	392.62	5	
10	Mouse Inalámbrico Modelo 403	614.98	23	
11	Audífonos Bluetooth Modelo 507	730.21	23	
12	Laptop Lenovo Modelo 735	498.12	20	
13	Cámara Web Full HD Modelo 977	797.81	37	
14	Laptop Lenovo Modelo 205	248.20	7	

Ventas

```
7      -- 3. Ventas
8 •    select * from ventas;
9      -- 4. Detalle ventas
```

Result Grid | Filter Rows:

	venta_id	cliente_id	fecha	total
1	89	2024-05-21	0.00	
2	56	2024-06-27	0.00	
3	34	2024-03-03	0.00	
4	14	2024-04-30	0.00	
5	77	2024-06-11	0.00	

Detalle_ventas

```
9      -- 4. Detalle_ventas
10 •   select * from detalle_ventas;
```

Result Grid | Filter Rows: Edit:

	detalle_id	venta_id	producto_id	cantidad	subtotal
1	296	1	4	1398.00	
2	337	48	3	143.79	
3	51	50	5	878.70	
4	53	26	3	1002.75	
5	347	22	5	1289.75	

2. Consultas (4)

1. Mostrar todos los productos con stock menor a 20.

```
4
3      -- CONSULTAS
4      -- 1. Mostrar todos los productos con stock menor a 20.
5 •    select * from productos -- Se muestran todos los campos de la tabla productos
6      where stock<20; -- Se pone como condicion que solo se muestren aquellos cuyo stock sea menor a 20
7
```

Result Grid | Filter Rows: Edit: Export/Import: Wrap Cell Content:

	producto_id	nombre	precio	stock
▶	2	Router TP-Link Modelo 150	488.65	10
	3	Cámara Web Full HD Modelo 753	439.80	17
	6	Router TP-Link Modelo 393	771.00	19
	7	Laptop Lenovo Modelo 890	579.14	7
	9	Router TP-Link Modelo 854	392.62	5
	14	Laptop Lenovo Modelo 205	248.20	7
	20	Impresora HP Modelo 101	628.54	9
	30	Monitor LG 24" Modelo 605	60.83	17
	33	Audífonos Bluetooth Modelo 832	624.09	8
	35	Router TP-Link Modelo 400	396.69	8
	36	Cámara Web Full HD Modelo 133	308.10	8
	37	Audífonos Bluetooth Modelo 659	617.09	15
	45	Disco SSD 1TB Modelo 386	207.17	18
	52	Router TP-Link Modelo 150	488.65	10
	53	Cámara Web Full HD Modelo 753	439.80	17
	56	Router TP-Link Modelo 393	771.00	19
	57	Laptop Lenovo Modelo 890	579.14	7
	59	Router TP-Link Modelo 854	392.62	5
	64	Laptop Lenovo Modelo 205	248.20	7

productos 8 x

- Forma 1

```
8 -- 2. Listar las ciudades de origen de los clientes sin duplicar.
9 • select ciudad from clientes -- Se selecciona el campo que se mostrar en la tabla y la tabla del que proviene
10 group by ciudad; -- Se agrupa por ciudad para evitar que se muestren duplicados
11
```

Result Grid	Filter Rows:	Export:	Wrap Cell Content:
ciudad			
▶ Manta			
Guayaquil			
Ambato			
Loja			
Riobamba			
Cuenca			
Portoviejo			
Quito			

- Forma 2

```
12 -- Otra forma de hacer la pregunta 2
13 • select distinct ciudad -- Se seleccionan valores unicos de (no repetidos) del campo ciudad
14 from clientes; -- Se selecciona la tabla de la que seextraeran los datos
15
```

Result Grid	Filter Rows:	Export:	Wrap Cell Content:
ciudad			
▶ Manta			
Guayaquil			
Ambato			
Loja			
Riobamba			
Cuenca			
Portoviejo			
Quito			

3. Mostrar la cantidad total de productos vendidos.

```
16 -- 3. Mostrar la cantidad total de productos vendidos.
17 • select sum(cantidad) as TOTAL_VENDIDO from detalle_ventas;
18 -- Se realiza una suma con de la cantidad de productos vendidos de la tabla detalle_ventas
19
```

Result Grid	Filter Rows:	Export:	Wrap Cell Content:
TOTAL_VENDIDO			
▶ 5958			

4. Mostrar los nombres de clientes y la fecha de cada venta.

```
20 -- 4. Mostrar los nombres de clientes y la fecha de cada venta.
21 • select c.nombre as CLIENTE, v.fecha as FECHA_VENTA -- Se seleccionan las columnas de las diferentes tablas
22 from clientes c -- Se establece la tabla principal
23 join ventas v on c.cliente_id=v.cliente_id; -- Se relaciona los IDs de cliente en ambas tablas
24
```

	CLIENTE	FECHA_VENTA
▶	Ana Mendoza	2024-05-21
	María Torres	2024-06-27
	Carlos Torres	2024-03-03
	María Luna	2024-04-30
	Diego Mendoza	2024-06-11
	Carlos Vargas	2024-05-02
	Fernanda Mendoza	2024-04-30
	Gabriel Gómez	2024-04-29
	Jorge Torres	2024-01-28
	Luis Torres	2024-06-26
	Paola Herrera	2024-04-13
	Luis Gómez	2024-04-12
	Carlos Luna	2024-06-18
	Gabriel Ramírez	2024-06-06
	Jorge Herrera	2024-06-24
	Carlos Torres	2024-05-25
	Fernanda Luna	2024-03-30
	Lucía Torres	2024-02-15
	Luis Herrera	2024-01-11

3. Subconsultas (4)

1. Mostrar los clientes que han comprado más que el promedio general de ventas.

```
37 -- SUBCONSULTAS
38 -- 1. Mostrar los clientes que han comprado más que el promedio general de ventas.
39 -- forma 1
40 • select c.nombre
41 from clientes c
42 join ventas v on c.cliente_id = v.cliente_id
43 join detalle_ventas dv on v.venta_id=dv.venta_id
44 group by c.nombre
45 having sum(dv.cantidad) > (select avg(cantidad) as PROMEDIO from detalle_ventas);
46
47 -- forma 2
48 • select c.nombre
49 from clientes c
```

	nombre
▶	Ana Mendoza
	Ana Torres
	Carlos Chávez
	Carlos Gómez
	Carlos Luna

```

47 -- forma 2
48 • select c.nombre
49 from clientes c
50 join ventas v on c.cliente_id = v.cliente_id
51 join detalle_ventas dv on v.venta_id=dv.venta_id
52 group by c.nombre
53 having sum(dv.subtotal) > (select avg(subtotal) as SUBTOTAL from detalle_ventas);
54
55 -- 2. Mostrar productos cuyo precio esté por encima del promedio de todos los productos.

```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

nombre
Ana Mendoza
Ana Torres
Carlos Chávez
Carlos Gómez
Carlos Luna

2. Mostrar productos cuyo precio está por encima del promedio de todos los productos.

```

30 -- 2. Mostrar productos cuyo precio está por encima del promedio de todos los productos.
31 • select nombre, precio from productos -- Se seleccionan las columnas que aparecerán en las tablas
32 -- Filtra solo productos donde el precio es mayor el promedio de precios calculado sobre todos los productos
33 where precio > (select avg(precio) from productos);

```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

nombre	precio
Router TP-Link Modelo 150	488.65
Impresora HP Modelo 727	705.36
Router TP-Link Modelo 393	771.00
Laptop Lenovo Modelo 890	579.14
Audífonos Bluetooth Modelo 453	568.57
Mouse Inalámbrico Modelo 403	614.98
Audífonos Bluetooth Modelo 507	730.21
Laptop Lenovo Modelo 735	498.12
Cámara Web Full HD Modelo 977	797.81
Router TP-Link Modelo 937	730.40
Audífonos Bluetooth Modelo 558	727.83
Impresora HP Modelo 101	628.54
Mouse Inalámbrico Modelo 890	579.14

3. Mostrar el cliente con la mayor suma en compras.

```

35 -- 3. Mostrar el cliente con la mayor suma en compras.
36 • select c.nombre as CLIENTE, sum(dv.subtotal) as total_compras -- Se realiza una suma de los subtotales de la tabla detalle_ventas
37 from ventas v -- Tabla principal
38 join clientes c on v.cliente_id=c.cliente_id -- Relación con la tabla de clientes
39 join detalle_ventas dv on v.venta_id = dv.venta_id -- Relación con el detalle de ventas
40 group by c.cliente_id -- Agrupación por ID de cliente
41 order by total_compras desc limit 1; -- Ordena de mayor a menor monto comprado

```

Result Grid | Filter Rows: | Export: | Wrap Cell Content: | Fetch rows:

CLIENTE	total_compras
Carlos Gómez	39865.22

4. Mostrar nombres de productos que aparecen en más de una venta.

```
43 -- 4. Mostrar nombres de productos que aparecen en más de una venta.
44 • select p.nombre, count(distinct dv.venta_id) as VENTA_TOTAL -- Se cuentan las ventas unicas
45 from productos p
46 join detalle_ventas dv on p.producto_id=dv.producto_id -- Se obtienen solo productos que tienen registros en detalle_ventas
47 group by p.producto_id -- Agrupa por ID de producto
48 having count(distinct dv.venta_id)>1; -- Filtra para mostrar solo productos con más de una venta
49
```

nombre	VENTA_TOTAL
Audifonos Bluetooth Modelo 659	17
Disco SSD 1TB Modelo 731	20
Disco SSD 1TB Modelo 611	15
Memoria RAM 16GB Modelo 495	17
Memoria RAM 16GB Modelo 214	18
Router TP-Link Modelo 202	21
Mouse Inalámbrico Modelo 536	28
Impresora HP Modelo 771	17
Disco SSD 1TB Modelo 386	19
Teclado Mecánico Modelo 364	14
Teclado Mecánico Modelo 952	24
Router TP-Link Modelo 438	28

```
51 -- JOINS
```

```
52 -- 1. INNER JOIN entre clientes y ventas
```

```
53 • select c.nombre as CLIENTES, v.venta_id -- Se seleccionan las columnas que apareceran en las tablas
54 from clientes c -- Tabla principal
55 join ventas v on c.cliente_id = v.cliente_id; -- Relacion a traves de los ids
56
```

CLIENTES	venta_id
Fernanda Luna	934
Luis Vargas	97
Luis Vargas	123
Luis Vargas	176
Luis Vargas	364
Luis Vargas	597
Luis Vargas	623
Luis Vargas	676
Luis Vargas	864
Ana Torres	26

2. JOIN entre ventas y detalle_ventas

```
58 -- 2. JOIN entre ventas y detalle_ventas
59 • select v.venta_id, v.fecha, dv.cantidad, dv.subtotal -- Se seleccionan las columnas que apareceran en las tablas
60 from ventas v -- Tabla principal
61 join detalle_ventas dv on v.venta_id=dv.venta_id; -- Relacion a traves de los ids
```

venta_id	fecha	cantidad	subtotal
296	2024-03-27	4	1398.00
337	2024-04-06	3	143.79
51	2024-04-29	5	878.70
53	2024-01-16	3	1002.75
347	2024-05-08	5	1289.75
379	2024-04-07	1	77.15
498	2024-06-16	5	799.15
410	2024-06-02	2	161.94
231	2024-05-02	1	383.64
381	2024-02-21	5	1671.25
141	2024-06-25	5	239.65
278	2024-03-29	5	142.10
23	2024-01-16	1	183.74
44	2024-04-18	5	1683.95

3. JOIN entre detalle_ventas y productos

```
64 -- 3. JOIN entre detalle_ventas y productos
65 • select dv.detalle_id, p.nombre as producto , dv.cantidad, p.precio, p.stock -- Se seleccionan las columnas que ap
66 from detalle_ventas dv -- Tabla principal
67 join productos p on dv.producto_id = p.producto_id; -- Relacion a traves de los ids
```

Result Grid	Filter Rows:	Export:	Wrap Cell Content:	Fetch rows:
detalle_id	producto	cantidad	precio	stock
1	Memoria RAM 16GB Modelo 714	4	233.65	36
2	Router TP-Link Modelo 438	3	581.45	38
3	Impresora HP Modelo 918	5	779.19	38
4	Impresora HP Modelo 291	3	796.27	27
5	Laptop Lenovo Modelo 317	5	133.06	48
6	Impresora HP Modelo 101	1	628.54	9
7	Disco SSD 1TB Modelo 731	5	733.64	23
8	Laptop Lenovo Modelo 735	2	498.12	20
9	Memoria RAM 16GB Modelo 670	1	317.89	43
10	Impresora HP Modelo 291	5	796.27	27
11	Router TP-Link Modelo 438	5	581.45	38
12	Memoria RAM 16GB Modelo 214	5	112.92	24
13	Router TP-Link Modelo 400	1	396.69	8
14	Router TP-Link Modelo 202	5	605.22	32

4. JOIN entre las 4 tablas

```
70 -- 4. JOIN entre las 4 tablas
71 • select c.nombre as CLIENTE, c.ciudad,
72 v.venta_id, v.fecha, p.nombre as PRODUCTO, dv.cantidad, p.stock -- Se seleccionan las columnas que aparecieran en la
73 from clientes c -- Tabla principal
74 join ventas v on c.cliente_id = v.cliente_id -- Relacion entre ventas y clientes
75 join detalle_ventas dv on v.venta_id = dv.venta_id -- Relacion entre detalle_ventas y ventas
76 join productos p on dv.producto_id = p.producto_id; -- Relacion entre productos y detalle_ventas
77
```

Result Grid

Filter Rows:

Export:

Wrap Cell Content:

Fetch rows:

	CLIENTE	ciudad	venta_id	fecha	PRODUCTO	cantidad	stock
▶	Diego Pérez	Manta	296	2024-03-27	Memoria RAM 16GB Modelo 714	4	36
	Fernanda Torres	Manta	337	2024-04-06	Router TP-Link Modelo 438	3	38
	Lucía Castillo	Loja	51	2024-04-29	Impresora HP Modelo 918	5	38
	Carlos Gómez	Riobamba	53	2024-01-16	Impresora HP Modelo 291	3	27
	Carlos Gómez	Riobamba	347	2024-05-08	Laptop Lenovo Modelo 317	5	48
	Paola Ramírez	Riobamba	379	2024-04-07	Impresora HP Modelo 101	1	9
	Carlos Gómez	Riobamba	498	2024-06-16	Disco SSD 1TB Modelo 731	5	23
	Lucía Torres	Manta	410	2024-06-02	Laptop Lenovo Modelo 735	2	20
	Paola Ramírez	Riobamba	231	2024-05-02	Memoria RAM 16GB Modelo 670	1	43
	Fernanda Gómez	Guayaquil	381	2024-02-21	Impresora HP Modelo 291	5	27
	Paola Mendoza	Loja	141	2024-06-25	Router TP-Link Modelo 438	5	38
	María Ramírez	Portoviejo	278	2024-03-29	Memoria RAM 16GB Modelo 214	5	24
	Carlos Torres	Riobamba	23	2024-01-16	Router TP-Link Modelo 400	1	8
	Jorge Gómez	Cuenca	44	2024-04-18	Router TP-Link Modelo 202	5	32

5. LEFT JOIN para mostrar clientes sin ventas

- Primera forma

```
78
79 -- 5. LEFT JOIN para mostrar clientes sin ventas
80 • select c.nombre as CLIENTE, v.venta_id, v.fecha as FECHA_VENTA -- Columnas que aparecieran en la tabla
81 from clientes c -- Tabla izquierda de la que se mostrara todos los registros
82 left join ventas v on c.cliente_id = v.cliente_id; -- Relacion entre ventas y clientes
83
```

Result Grid	Filter Rows:	Export:	Wrap Cell Content:	Fetch rows:
-------------	--------------	---------	--------------------	-------------

- Segunda forma

```
84
85 -- Segunda forma
86 • select c.nombre as CLIENTE, v.venta_id, v.fecha as FECHA_VENTA -- Columnas que aparecieran en la tabla
87 from clientes c -- Tabla izquierda de la que se mostrara todos los registros
88 left join ventas v on c.cliente_id = v.cliente_id -- Relacion entre ventas y clientes
89 where v.venta_id is null; -- Se establece que solo se muestren aquellos clientes que no tienen registros correspondientes en
```

Result Grid	Filter Rows:	Export:	Wrap Cell Content:	Result Grid	Form Editor	Field Types
CLIENTE	venta_id	FECHA_VENTA				
Jorge Chávez	NULL	NULL				
Ana Ramírez	NULL	NULL				
Fernanda Torres	NULL	NULL				
Paola Herrera	NULL	NULL				
Paola Pérez	NULL	NULL				
Carlos Gómez	NULL	NULL				
Gabriel Ramírez	NULL	NULL				
Lucía Luna	NULL	NULL				
Lucía Luna	NULL	NULL				

```
92 -- 6. RIGHT JOIN entre ventas y detalle_ventas
93 • select v.venta_id, v.fecha as fecha_venta, dv.cantidad, dv.subtotal
94 from detalle_ventas dv -- Tabla secundaria 'dv' de la cual se mostrarn solo las coincidencias y si no hay coincidencias se
95 -- Tabla principal 'ventas' de la cual se mostraran todos los registros
96 right join ventas v on dv.venta_id=v.venta_id; -- Relacion entre ventas y detalle_ventas
```

Result Grid	Filter Rows:	Export:	Wrap Cell Content:	Fetch rows:	Result Grid	Form Editor	Field Types	Query Stats
venta_id	fecha_venta	cantidad	subtotal					
62	2024-03-21	4	583.88					
62	2024-03-21	1	364.19					
62	2024-03-21	2	699.00					
63	2024-02-21	1	28.42					
63	2024-02-21	4	194.16					
63	2024-02-21	1	28.42					
63	2024-02-21	4	194.16					
64	2024-03-22	2	801.64					
64	2024-03-22	1	411.87					
64	2024-03-22	5	404.85					
64	2024-03-22	2	801.64					
64	2024-03-22	1	411.87					
64	2024-03-22	5	404.85					
65	2024-01-26	NULL	NULL					

7. JOIN para mostrar total vendido por producto

```
99 -- 7. JOIN para mostrar total vendido por producto
100 • select p.nombre as PRODUCTO, p.producto_id,
101 sum(dv.cantidad) as TOTAL_VENDIDO -- Suma la cantidad vendida de productos
102 from productos p -- Tabla principal
103 join detalle_ventas dv on p.producto_id = dv.producto_id -- Relacion entre la tabla detalle_ventas y productos
104 group by p.nombre, p.producto_id; -- Se agrupan los prodcutos por su nombre e id
105
```

Result Grid	Filter Rows:	Export:	Wrap Cell Content:	PRODUCTO	producto_id	TOTAL_VENDIDO
				Memoria RAM 16GB Modelo 714	1	96

8. JOIN para ventas con fecha específica

```
--
104 -- 8. JOIN para ventas con fecha específica
105 -- Mostrar los clientes que compraron en '2024-03-22'
106 • select c.nombre as CLIENTE, v.fecha as FECHA_COMPRA -- Columnas que aparecerán en la tabla
107 from clientes c -- Tabla principal
108 join ventas v on c.cliente_id=v.cliente_id -- Relacion entre la tabla ventas y clientes
109 where v.fecha='2024-03-22'; -- Se filtra para que solo aparezcan los clientes que realizaron una compra en 2024-03-22
```

CLIENTE	FECHA_COMPRA
Luis Ramírez	2024-03-22
Paola Ramírez	2024-03-22
Paola Herrera	2024-03-22
Fernanda Herrera	2024-03-22
María Chávez	2024-03-22
Diego Pérez	2024-03-22
Luis Ramírez	2024-03-22
Paola Ramírez	2024-03-22
Paola Herrera	2024-03-22
Fernanda Herrera	2024-03-22

5. Transacciones (4)

1. Venta válida con COMMIT.

```
116 -- Transacciones (4)
117 -- 1. Venta válida con COMMIT.
118 • start transaction; -- Se inicia una transaccion
119 • update ventas set total=30 where venta_id=1; -- Se actualiza el total de la venta cuyo id es 1
120 • commit; -- Para hacer los cambios permanentes en la base de datos
```

#	Time	Action	Message	Duration / Fetch
26	17:26:01	update ventas set total=30 where venta_id=1	1 row(s) affected Rows matched: 1 Changed: 1 Warnings: 0	0.016 sec
27	17:26:01	commit	0 row(s) affected	0.000 sec

```
1 • SELECT * FROM ventas_db.ventas;
```

venta_id	cliente_id	fecha	total
1	89	2024-05-21	30.00
2	56	2024-06-27	0.00
3	34	2024-03-03	0.00
4	14	2024-04-30	0.00
5	77	2024-06-11	0.00
6	100	2024-05-02	0.00
7	82	2024-04-30	0.00
8	96	2024-04-29	0.00
9	23	2024-01-28	0.00
10	39	2024-06-26	0.00

2. Error de stock, aplicar ROLLBACK.

```
123 -- 2. Error de stock, aplicar ROLLBACK.
124 • start transaction; -- Se inicia una transaccion
125 • update productos set stock= stock - 100 where producto_id=1; -- Se actualiza el stock del producto cuyo id es 1
126 • rollback; -- Se revierte todo de forma intencional
```

Output

Action Output

#	Time	Action	Message	Duration / Fetch
✓ 37	17:43:26	start transaction	0 row(s) affected	0.000 sec
✓ 38	17:43:26	update productos set stock= stock - 100 where producto_id=1	1 row(s) affected Rows matched: 1 Changed: 1 Warnings: 0	0.015 sec
✓ 39	17:43:34	SELECT * FROM ventas_db.productos LIMIT 0, 1000	100 row(s) returned	0.000 sec / 0.000 sec

```
1 • SELECT * FROM ventas_db.productos;
```

Result Grid



Filter Rows:

Edit:



Export/Import:



Wrap Cel

	producto_id	nombre	precio	stock
▶	1	Memoria RAM 16GB Modelo 714	233.65	-64
	2	Router TP-Link Modelo 150	488.65	10
	3	Cámara Web Full HD Modelo 753	439.80	17
	4	Impresora HP Modelo 727	705.36	39
	5	Memoria RAM 16GB Modelo 670	317.89	43
	6	Router TP-Link Modelo 393	771.00	19
	7	Laptop Lenovo Modelo 890	579.14	7
	8	Audífonos Bluetooth Modelo 453	568.57	38
	9	Router TP-Link Modelo 854	392.62	5
	10	Mouse Inalámbrico Modelo 403	614.98	23





```
---
123 -- 2. Error de stock, aplicar ROLLBACK.
124 • start transaction; -- Se inicia una transaccion
125 • update productos set stock= stock - 100 where producto_id=1; -- Se actualiza el stock del producto cuyo id es 1
126 • rollback; -- Se revierte todo de forma intencional
```

Output

Action Output

#	Time	Action	Message	Duration / Fetch
✓ 39	17:43:34	SELECT * FROM ventas_db.productos LIMIT 0, 1000	100 row(s) returned	0.000 sec / 0.000 sec
✓ 40	17:44:26	rollback	0 row(s) affected	0.000 sec
✓ 41	17:44:31	SELECT * FROM ventas_db.productos LIMIT 0, 1000	100 row(s) returned	0.000 sec / 0.000 sec

```
1 • SELECT * FROM ventas_db.productos;
```







Result Grid				
Filter Rows: <input type="text"/>				
Edit:    				
	producto_id	nombre	precio	stock
▶	1	Memoria RAM 16GB Modelo 714	233.65	36
	2	Router TP-Link Modelo 150	488.65	10
	3	Cámara Web Full HD Modelo 753	439.80	17
	4	Impresora HP Modelo 727	705.36	39
	5	Memoria RAM 16GB Modelo 670	317.89	43
	6	Router TP-Link Modelo 393	771.00	19
	7	Laptop Lenovo Modelo 890	579.14	7
	8	Audífonos Bluetooth Modelo 453	568.57	38
	9	Router TP-Link Modelo 854	392.62	5
	10	Mouse Inalámbrico Modelo 403	614.98	23

productos 1 x

3. Venta eliminando producto con COMMIT (irreversible).

```
128
129 -- 3. Venta eliminando producto con COMMIT (irreversible).
130 • start transaction; -- Se inicia una transaccion
131 • insert into detalle_ventas(detalle_id, venta_id, producto_id, cantidad, subtotal)
132 values (3000, 99, 94, 99, 100.50); -- Se agrega un nuevo registro en detalle_ventas
133 -- Se elimina de forma irreversible el producto
134 • delete from detalle_ventas where producto_id=94;
135 • delete from productos where producto_id=94;
136 • commit;
```

Output				
Action Output				
#	Time	Action	Message	
✓ 60	18:07:36	delete from detalle_ventas where producto_id=94	1 row(s) affected	
✓ 61	18:09:13	delete from productos where producto_id=94	1 row(s) affected	
✓ 62	18:09:18	commit	0 row(s) affected	

Result Grid				
Filter Rows: <input type="text"/>				
Edit:      				
	producto_id	nombre	precio	stock
	93	Mouse Inalámbrico Modelo 536	139.55	20
	94	Impresora HP Modelo 771	765.26	35
	95	Disco SSD 1TB Modelo 386	207.17	18
	96	Teclado Mecánico Modelo 364	356.99	24
	97	Teclado Mecánico Modelo 952	203.24	47
	98	Router TP-Link Modelo 438	581.45	38
	99	Disco SSD 1TB Modelo 130	298.67	47
	100	Impresora HP Modelo 918	779.19	38

productos 1 x Apply

Limit to 1000 rows				
1 • <code>SELECT * FROM ventas_db.productos;</code>				
Result Grid Filter Rows: Edit: Export/Import:				
	producto_id	nombre	precio	stock
	92	Router TP-Link Modelo 202	605.22	32
	93	Mouse Inalámbrico Modelo 536	139.55	20
	95	Disco SSD 1TB Modelo 386	207.17	18
	96	Teclado Mecánico Modelo 364	356.99	24
	97	Teclado Mecánico Modelo 952	203.24	47
	98	Router TP-Link Modelo 438	581.45	38
	99	Disco SSD 1TB Modelo 130	298.67	47
	100	Impresora HP Modelo 918	779.19	38
	NULL	NULL	NULL	NULL

```

139 -- 4. Uso de SAVEPOINT y ROLLBACK parcial.
140 • start transaction;
141 • insert into clientes(cliente_id, nombre, correo, ciudad) -- Se inserta un nuevo cliente
142 values (201, 'Liam Vega', 'liam@gmail.com', 'Quito');
143 • savepoint insertar_cliente; -- Se guarda un punto de recuperacion
144 • update productos set stock = stock - 100 where producto_id = 1; -- Se intenta restar 100 del stock del producto cuyo id es :
145 • rollback to savepoint insertar_cliente; -- Se revierten los cambios hasta el punto de recuperacion
146 • commit; -- Se confirman los cambios realizados hasta el punto de recuperacion
147

```

#	Time	Action	Message	Duration / Fetch
66	18:44:34	start transaction	0 row(s) affected	0.000 sec
67	18:44:34	insert into clientes(cliente_id, nombre, correo, ciudad) values (201, 'Lia...	1 row(s) affected	0.000 sec
68	18:44:34	savepoint insertar_cliente	0 row(s) affected	0.000 sec
69	18:44:34	update productos set stock = stock - 100 where producto_id = 1	1 row(s) affected Rows matched: 1 Changed: 1 Warnings: 0	0.000 sec
70	18:44:34	rollback to savepoint insertar_cliente	0 row(s) affected	0.000 sec
71	18:44:34	commit	0 row(s) affected	0.000 sec

uso_join*	cursoonline_cursos	leccionescompletadas	repaso	clientes	clientes	productos	productos x
Limit to 1000 rows							
1 • <code>SELECT * FROM ventas_db.productos;</code>							
Result Grid Filter Rows: Edit: Export/Import: Wrap Cell Content: Result Grid Form Editor							
	producto_id	nombre	precio	stock			
▶ 1	1	Memoria RAM 16GB Modelo 714	233.65	-64			
2	2	Router TP-Link Modelo 150	488.65	10			
3	3	Cámara Web Full HD Modelo 753	439.80	17			
4	4	Impresora HP Modelo 727	705.36	39			
5	5	Memoria RAM 16GB Modelo 670	317.89	43			
6	6	Router TP-Link Modelo 393	771.00	19			
7	7	Laptop Lenovo Modelo 890	579.14	7			
8	8	Audifonos Bluetooth Modelo 453	568.57	38			
9	9	Router TP-Link Modelo 854	392.62	5			

uso_join* cursosonline_cursos leccionescompletadas repaso* clientes clientes x

Limit to 1000 rows

1 • `SELECT * FROM ventas_db.clientes;`

Result Grid Filter Rows: Edit: Export/Import: Wrap Cell Content:

	cliente_id	nombre	correo	ciudad
	194	Lucía Castillo	lucia.castillo@tecnoventas.com	Cuenca
	195	Fernanda Gómez	fernanda.gómez@tecnoventas.com	Guayaquil
	196	Gabriel Gómez	gabriel.gómez@tecnoventas.com	Cuenca
	197	Lucía Mendoza	lucia.mendoza@tecnoventas.com	Portoviejo
	198	Gabriel Luna	gabriel.luna@tecnoventas.com	Guayaquil
	199	Lucía Castillo	lucia.castillo@tecnoventas.com	Loja
	200	Carlos Vargas	carlos.vargas@tecnoventas.com	Riobamba
	201	Liam Vega	liam@gmail.com	Quito
*	NULL	NULL	NULL	NULL

clientes 1 x Apply

uso_join* cursosonline_cursos leccionescompletadas repaso clientes clientes productos x

Limit to 1000 rows

1 • `SELECT * FROM ventas_db.productos;`

Result Grid Filter Rows: Edit: Export/Import: Wrap Cell Content:

	producto_id	nombre	precio	stock
▶	1	Memoria RAM 16GB Modelo 714	233.65	36
	2	Router TP-Link Modelo 150	488.65	10
	3	Cámara Web Full HD Modelo 753	439.80	17
	4	Impresora HP Modelo 727	705.36	39
	5	Memoria RAM 16GB Modelo 670	317.89	43
	6	Router TP-Link Modelo 393	771.00	19
	7	Laptop Lenovo Modelo 890	579.14	7
	8	Audífonos Bluetooth Modelo 453	568.57	38
	9	Router TP-Link Modelo 854	392.62	5

productos 1 x Apply

6. Problemas con Vistas

1. Crear vista vista_detalle_clientes con cliente, producto, cantidad y total.

```
173 -- PROBLEMAS CON VISTAS
174 -- 1. Crear vista vista_detalle_clientes con cliente, producto, cantidad y total.
175 -- Como nos pide que en la vista pongamos el total, y anteriormente el total ya lo calculamos, pero no es una columna hacemos:
176 -- crear una vista del total vendido para luego poder indicarlo en la vista_detalle_clientes
177
178 • create view vista_total_vendido as
179 select p.nombre as PRODUCTO, p.producto_id,
180 sum(dv.cantidad) as TOTAL_VENDIDO -- Suma la cantidad vendida de productos
181 from productos p -- Tabla principal
182 join detalle_ventas dv on p.producto_id = dv.producto_id -- Relacion entre la tabla detalle_ventas y productos
183 group by p.nombre, p.producto_id;
184
185 -- Con lo anterior ya podemos realizar vista_detalle_cliente
186 • create view vista_detalle_clientes as
187 select c.nombre as NOMBRE, p.nombre as PRODUCTO, dv.cantidad as CANTIDAD, vtv.TOTAL_VENDIDO as TOTAL
188 from clientes c -- indicar primero la tabla de clientes
189 join ventas v on c.cliente_id=v.cliente_id
190 join detalle_ventas dv on v.venta_id= dv.venta_id
191 join productos p on dv.producto_id= p.producto_id
192 join vista_total_vendido vtv on p.producto_id= vtv.producto_id; -- hacemos un join con la vista que anteriormente creamos, para evitar repetir proces
```

```
1 • SELECT * FROM ventas_db.vista_detalle_clientes;
```

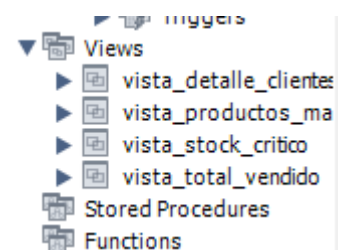
	NOMBRE	PRODUCTO	CANTIDAD	TOTAL
▶	Paola Herrera	Memoria RAM 16GB Modelo 670	4	70
	Paola Herrera	Impresora HP Modelo 918	1	75
	Paola Herrera	Monitor LG 24" Modelo 600	2	64
	Paola Herrera	Memoria RAM 16GB Modelo 214	1	66
	Paola Herrera	Router TP-Link Modelo 937	4	68
	Paola Herrera	Memoria RAM 16GB Modelo 156	3	64
	Paola Herrera	Router TP-Link Modelo 854	2	59
	Paola Herrera	Router TP-Link Modelo 150	5	45
	Paola Herrera	Router TP-Link Modelo 393	1	49

2. Crear vista vista_stock_critico para productos con stock < 15.

```
-- 2. Crear vista vista_stock_critico para productos con stock < 15.
create view vista_stock_critico as
select nombre,stock from productos
where stock<15;
```

```
1 • SELECT * FROM ventas_db.vista_stock_critico;
```

	nombre	stock
▶	Router TP-Link Modelo 150	10
	Laptop Lenovo Modelo 890	7
	Router TP-Link Modelo 854	5
	Laptop Lenovo Modelo 205	7
	Impresora HP Modelo 101	9
	Audifonos Bluetooth Modelo 832	8
	Router TP-Link Modelo 400	8
	Cámara Web Full HD Modelo 133	8



7. Problemas con Transacciones

1. Compra múltiple con dos productos (usar COMMIT).

```
199      -- PROBLEMAS CON TRANSACCIONES
200      -- 1. Compra múltiple con dos productos (usar COMMIT).
201      -- iniciamos la transacción de una compra múltiple
202 •    start transaction;
203
204      -- Registrar una nueva venta
205 •    INSERT INTO clientes (nombre, correo, ciudad)
206      VALUES
207          ('Juan Torres', 'juan@gmail.com', 'Quito'),
208          ('José Palacios', 'jose@gmail.com', 'Guayaquil');
209      -- confirmar la transaccion
210 •    commit;
```

Output		
Action Output		
#	Time	Action
✓ 2	22:57:57	INSERT INTO clientes (nombre, correo, ciudad) VALUES ('Juan Torres', juar
✓ 3	22:58:01	commit

2. Venta con total mal ingresado, aplicar ROLLBACK.

```
213      -- 2. Venta con total mal ingresado, aplicar ROLLBACK.
214      -- Insertar venta con total incorrecto (ejemplo)
215 •    start transaction;
216 •    insert into ventas (cliente_id, fecha, total)
217      values (1, CURDATE(), 9999.99); -- total incorrecto
218
219      -- Insertar detalles de venta (subtotal no suma 9999.99)
220 •    insert into detalle_ventas (venta_id, producto_id, cantidad, subtotal)values
221      (LAST_INSERT_ID(), 1, 2, 30.00),
222      (LAST_INSERT_ID(), 2, 1, 15.00);
223
224 •    ROLLBACK;
```

Output			
Action Output			
#	Time	Action	Message
7	23:02:11	insert into detalle_ventas (venta_id, producto_id, cantidad, subtotal)values (LAST_INSERT_...	2 row(s) affected
8	23:02:15	ROLLBACK	0 row(s) affected

1 • `SELECT * FROM ventas_db.ventas;`

Result Grid Filter Rows: Edit: Export/Import

	venta_id	cliente_id	fecha	total
	491	88	2024-02-13	0.00
	492	5	2024-04-21	0.00
	493	89	2024-03-02	0.00
	494	63	2024-04-17	0.00
	495	98	2024-05-04	0.00
	496	74	2024-04-23	0.00
	497	46	2024-01-14	0.00
	498	50	2024-06-16	0.00
	499	78	2024-01-22	0.00
	500	100	2024-06-12	0.00
	501	1	2025-06-09	60.00
*	NULL	NULL	NULL	NULL

ventas 1 x

Output

Action Output

	#	Time	Action
✓	8	23:02:15	ROLLBACK
✓	9	23:04:54	SELECT * FROM ventas_db.ventas LIMIT 0, 1000

8. Creación de usuarios

Usuario 1: administrador (admin_tventas)

```
-- CREACIÓN DE USUARIOS
-- 1. Usuario 1: administrador (admin_tventas)
create user 'admin_tventas'@'localhost' identified by '1234';
grant all privileges on ventas_db to 'admin_tventas'@'localhost';
flush privileges;
```







Usuario 2: consulta (consulta)

```
-- 2. Usuario 2: consulta (consulta)
create user 'consulta'@'localhost' identified by '2345';
grant select on ventas_db to 'consulta'@'localhost';
flush privileges;
```

on Output

	Time	Action
:	23:13:45	flush privileges
'	23:13:50	flush privileges






MySQL Connections

Local  root  127.0.0.1:3306	administrador_ventas  admin_tventas  127.0.0.1:3306	consultas  consulta  127.0.0.1:3306
--	--	--

9. Actividades sugeridas



1. Vista de productos más vendidos.

```
239 -- ACTIVIDADES SUGERIDAS
240 -- 1. Vista de productos más vendidos.
241 • create view vista_productos_mas_vendidos as -- creamos la vista
242 select p.nombre as PRODUCTO, sum(dv.cantidad) as TOTAL_VENDIDO -- seleccionamos los campos y como los vamos a llamar
243 from productos p-- visualizaremos primero la tabla productos
244 join detalle_ventas dv on p.producto_id=dv.producto_id -- realizamos la relación entre las tablas
245 group by p.producto_id,p.nombre -- agrupamos
246 order by TOTAL_VENDIDO desc; -- ordenamos de manera descendente
247 -- Después de creada la vista, solo visualizaremos los 5 productos más vendidos
248 • select * from vista_productos_mas_vendidos
249 limit 5;
```

Result Grid   Filter Rows: <input type="text"/>		Export:  Wrap Cell Content:  Fetch rows: 
PRODUCTO	TOTAL_VENDIDO	
Mouse Inalámbrico Modelo 403	94	
Impresora HP Modelo 499	87	
Mouse Inalámbrico Modelo 536	84	
Router TP-Link Modelo 400	82	
Router TP-Link Modelo 438	82	

2. Subconsulta para producto más costoso.

```
251 -- 2. Subconsulta para producto más costoso.
252 • select nombre, precio from productos
253 where precio=(select max(precio) from productos);
```

Result Grid   Filter Rows: <input type="text"/>		Export:  Wrap Cell C
nombre	precio	
Cámara Web Full HD Modelo 977	797.81	

3. Transacción fallida por stock insuficiente.

```
255 -- 3. Transacción fallida por stock insuficiente.
256 • select p.nombre, dv.cantidad, p.stock
257 from productos p
258 join detalle_ventas dv on p.producto_id = dv.producto_id
259 where dv.cantidad > p.stock;
```

Result Grid

Filter Rows:

Export:

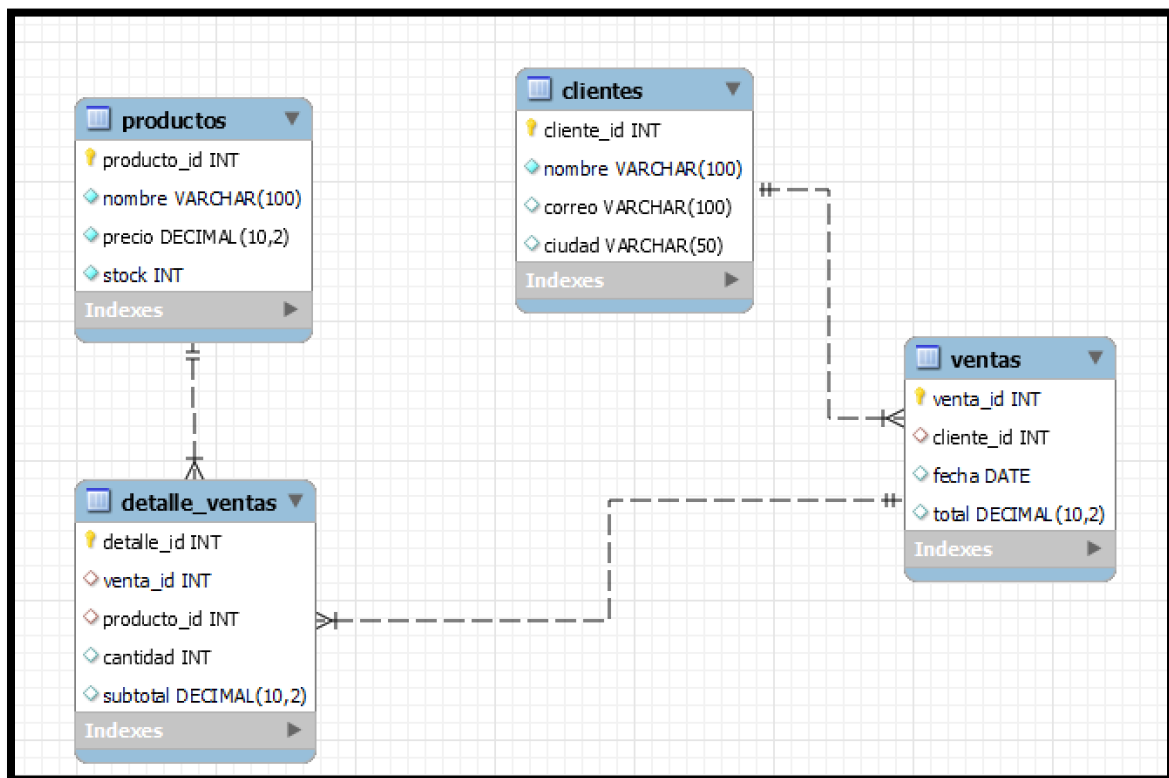
Wrap Cell Content:

venta_id	nombre	cantidad	stock	estado
----------	--------	----------	-------	--------

4. JOIN completo: cliente, producto, subtotal y fecha.

```
261 -- 4. JOIN completo: cliente, producto, subtotal y fecha.
262 • select c.nombre as CLIENTE, p.nombre as PRODUCTO, dv.subtotal as SUBTOTAL, v.fecha as FECHA
263 from clientes c
264 join ventas v on c.cliente_id=v.cliente_id
265 join detalle_ventas dv on v.venta_id= dv.venta_id
266 join productos p on dv.producto_id= p.producto_id;
```

	CLIENTE	PRODUCTO	SUBTOTAL	FECHA
▶	Paola Herrera	Memoria RAM 16GB Modelo 670	1534.56	2024-05-05
	Paola Herrera	Impresora HP Modelo 918	175.74	2024-05-05
	Paola Herrera	Monitor LG 24" Modelo 600	187.28	2024-05-05
	Paola Herrera	Memoria RAM 16GB Modelo 214	28.42	2024-02-21
	Paola Herrera	Router TP-Link Modelo 937	194.16	2024-02-21



ENTREGA

- Subir evidencia del trabajo completo en una carpeta compartida (puede ser en GitHub, Drive o Moodle).
- Incluir capturas de pantalla con consultas ejecutadas, vistas creadas y pruebas de transacciones.