

# ACTIVIDAD 2 - CREACIÓN DE VISTAS PERSONALIZADAS

Cristóbal Suárez Abad

ADMINISTRACIÓN DE SISTEMAS GESTORES DE BASES DE DATOS - 2º ASIR

## Contenido

<b>Actividad 2 - Creación de vistas personalizadas.....</b>	<b>2</b>
<b>    Creación de tablas y datos base .....</b>	<b>2</b>
<b>    Asignación de permisos a usuarios .....</b>	<b>8</b>
<b>    Ampliación.....</b>	<b>12</b>

## Actividad 2 - Creación de vistas personalizadas

### Creación de tablas y datos base

- Crea una base de datos para el área de ventas llamada ventas\_db

```
CREATE DATABASE ventas_db;
```

Y nos conectamos a ella: **\c ventas\_db;**

```
postgres=# CREATE DATABASE ventas_db;
CREATE DATABASE
postgres=# \c ventas_db;
You are now connected to database "ventas_db" as user "postgres".
ventas_db=# |
```

- Crea las tablas e inserta algunos registros de ejemplo:

- clientes (id, nombre, dni, telefono, email, saldo )

```
CREATE TABLE clientes (
```

```
    id SERIAL PRIMARY KEY,
    nombre VARCHAR(100) NOT NULL,
    dni VARCHAR(15) UNIQUE,
    telefono VARCHAR(20),
    email VARCHAR(100),
    saldo NUMERIC(10, 2) DEFAULT 0.00
```

```
);
```

```
ventas_db=# CREATE TABLE clientes (
        id SERIAL PRIMARY KEY,
        nombre VARCHAR(100) NOT NULL,
        dni VARCHAR(15) UNIQUE,
        telefono VARCHAR(20),
        email VARCHAR(100),
        saldo NUMERIC(10, 2) DEFAULT 0.00
```

```
);
```

```
CREATE TABLE
```

```
ventas_db=# |
```

- o pedidos (id, id\_cliente, fecha, total, estado )

```
CREATE TABLE pedidos (
    id SERIAL PRIMARY KEY,
    id_cliente INTEGER REFERENCES clientes(id),
    fecha DATE DEFAULT CURRENT_DATE,
    total NUMERIC(10, 2) NOT NULL,
    estado VARCHAR(50)
```

);

```
ventas_db=# CREATE TABLE pedidos (
    id SERIAL PRIMARY KEY,
    id_cliente INTEGER REFERENCES clientes(id),
    fecha DATE DEFAULT CURRENT_DATE,
    total NUMERIC(10, 2) NOT NULL,
    estado VARCHAR(50)
);
CREATE TABLE
ventas_db=# |
```

3. Inserta algunos registros de ejemplo:

-- Inserción en clientes

```
INSERT INTO clientes (nombre, dni, telefono, email, saldo) VALUES
('Perro Sanchez', '12345678A', '600111222', 'perrete@psoe.com', -9999999.00),
('Donald Trumpino', '98765432B', '600333444', 'trump@usa.com', 5000000.00),
('Vladimiro Putinino', '11223344C', '600555666', 'ervladi@ruski.com', 1.00);
```

```
ventas_db=# INSERT INTO clientes (nombre, dni, telefono, email, saldo) VALUES
('Perro Sanchez', '12345678A', '600111222', 'perrete@psoe.com', -9999999.00),
('Donald Trumpino', '98765432B', '600333444', 'trump@usa.com', 5000000.00),
('Vladimiro Putinino', '11223344C', '600555666', 'ervladi@ruski.com', 1.00);
INSERT 0 3
ventas_db=# |
```

-- Inserción en pedidos

```
INSERT INTO pedidos (id_cliente, total, estado) VALUES
(1, 150.00, 'Entregado'),
(1, 350.50, 'En Proceso'),
(2, 75.25, 'Pendiente'),
(3, 1200.00, 'Entregado');
```

```
INSERT 0 3
ventas_db=# INSERT INTO pedidos (id_cliente, total, estado) VALUES
(1, 150.00, 'Entregado'),
(1, 350.50, 'En Proceso'),
(2, 75.25, 'Pendiente'),
(3, 1200.00, 'Entregado');
INSERT 0 4
```

4. Verifica el contenido:

```
SELECT * FROM clientes;
```

```
SELECT * FROM pedidos;
```

ventas_db=# SELECT * FROM clientes;	id	nombre	dni	telefono	email	saldo
	1	Perro Sanchez	12345678A	600111222	perrete@psoe.com	-9999999.00
	2	Donaldo Trumpino	98765432B	600333444	trump@usa.com	5000000.00
	3	Vladimiro Putinino	11223344C	600555666	ervladi@ruski.com	1.00
	(3 rows)					

  

ventas_db=# SELECT * FROM pedidos;	id	id_cliente	fecha	total	estado
	1	1	2025-11-13	150.00	Entregado
	2	1	2025-11-13	350.50	En Proceso
	3	2	2025-11-13	75.25	Pendiente
	4	3	2025-11-13	1200.00	Entregado
	(4 rows)				

## Creación de vistas personalizadas

5. Los administradores deben tener acceso total a los datos de clientes y pedidos, con el número de pedidos y total de todos los pedidos

**CREATE VIEW vista\_admin\_ventas AS**

**SELECT**

```
c.id AS cliente_id,
c.nombre,
c.dni,
c.telefono,
c.email,
c.saldo,
COUNT(p.id) AS numero_pedidos,
COALESCE(SUM(p.total), 0) AS total_gastado
```

**FROM**

**clientes c**

**LEFT JOIN**

**pedidos p ON c.id = p.id\_cliente**

**GROUP BY**

**c.id, c.nombre, c.dni, c.telefono, c.email, c.saldo;**

```
ventas_db=# CREATE VIEW vista_admin_ventas AS
SELECT
    c.id AS cliente_id,
    c.nombre,
    c.dni,
    c.telefono,
    c.email,
    c.saldo,
    COUNT(p.id) AS numero_pedidos,
    COALESCE(SUM(p.total), 0) AS total_gastado
FROM
    clientes c
LEFT JOIN
    pedidos p ON c.id = p.id_cliente
GROUP BY
    c.id, c.nombre, c.dni, c.telefono, c.email, c.saldo;
CREATE VIEW
ventas_db=# |
```

clientes 1										
Text										
	123 cliente_id	AZ nombre	AZ dni	AZ telefono	AZ email	123 saldo	123 numero_pedidos	123 total_gastado		
1	3	Vladimiro Putinino	11223344C	600555666	ervladi@ruski.com	1	1	1.200		
2	2	Donald Trumpino	98765432B	600333444	trump@usa.com	5.000.000	1	75,25		
3	1	Perro Sanchez	12345678A	600111222	perrete@psoe.com	-9.999.999	2	500,5		

6. Los empleados solo deben ver información de contacto y saldo, sin DNI ni email.

**CREATE VIEW vista\_empleado\_ventas AS**

**SELECT**

*id,  
nombre,  
telefono,  
saldo*

**FROM**

*clientes;*

```
ventas_db=# CREATE VIEW vista_empleado_ventas AS
SELECT
    id,
    nombre,
    telefono,
    saldo
FROM
    clientes;
CREATE VIEW
```

↳ T SELECT id, nombre, telefono, saldo FROM clientes | ↵ ↴ Enter a SQL expression to filter results

Grilla	123 id	A-Z nombre	A-Z telefono	123 saldo
Grilla	1	Perro Sanchez	600111222	-9.999.999
Texto	2	Donaldo Trumpino	600333444	5.000.000
Texto	3	Vladimiro Putinino	600555666	1

7. El auditor puede consultar datos, pero sin información personal identifiable.

```
CREATE VIEW vista_auditor AS
SELECT
    c.id AS cliente_id,
    c.saldo,
    p.id AS pedido_id,
    p.fecha,
    p.total,
    p.estado
FROM
    clientes c
LEFT JOIN
    pedidos p ON c.id = p.id_cliente;
```

```
ventas_db=# CREATE VIEW vista_auditor AS
SELECT
    c.id AS cliente_id,
    c.saldo,
    p.id AS pedido_id,
    p.fecha,
    p.total,
    p.estado
FROM
    clientes c
LEFT JOIN
    pedidos p ON c.id = p.id_cliente;
CREATE VIEW
```

SELECT c.id AS cliente\_id, c.saldo, p.id AS pedido\_id, p.fecha | Enter a SQL expression to filter results (use Ctrl+Space)

Grilla	cliente_id	saldo	pedido_id	fecha	total	estado
Texto	1	-9.999.999	1	2025-11-13	150	Entregado
	2	-9.999.999	2	2025-11-13	350,5	En Proceso
	3	5.000.000	3	2025-11-13	75,25	Pendiente
	4		1 4	2025-11-13	1.200	Entregado

## Asignación de permisos a usuarios

8. Concede permisos de lectura sobre las vistas a cada role o usuarios:

Asignar la vista de administración al usuario admin\_ventas

```
GRANT SELECT ON vista_admin_ventas TO admin_ventas;
```

```
ventas_db=# GRANT SELECT ON vista_admin_ventas TO admin_ventas;
GRANT
ventas_db=#[
```

Asignar la vista de empleados al usuario empleado\_ventas.

```
GRANT SELECT ON vista_empleado_ventas TO empleado_ventas;
```

```
ventas_db=# GRANT SELECT ON vista_empleado_ventas TO empleado_ventas;
GRANT
ventas_db=#[
```

Asignar la vista de auditoría al usuario auditor

```
GRANT SELECT ON vista_auditor TO auditor;
```

```
GRANT
ventas_db=# GRANT SELECT ON vista_auditor TO auditor;
GRANT
ventas_db=#[
```

9. Revoca permisos directos sobre las tablas base, para que solo puedan acceder a través de las vistas

```
REVOKE ALL ON ALL TABLES IN SCHEMA public FROM PUBLIC;
```

```
GRANT
ventas_db=# REVOKE ALL ON ALL TABLES IN SCHEMA public FROM PUBLIC;
REVOKE
ventas_db=#[
```

Por si acaso, también podemos usar:

```
REVOKE ALL ON clientes FROM ventas_grupo;
REVOKE ALL ON pedidos FROM ventas_grupo;
REVOKE ALL ON clientes FROM auditor;
REVOKE ALL ON pedidos FROM auditor;
```

10. Comprueba que cada usuario solo puede acceder a su vista correspondiente:

"admin\_ventas": `SELECT * from vista_admin_ventas;`

The screenshot shows the MySQL Workbench interface. On the left, the 'Navegador de Bases de Datos' (Database Navigator) displays various database connections and the current connection to 'ventas-admin\_ventas'. Under 'Bases de Datos', the 'ventas\_db' schema is selected, and within it, the 'public' schema is expanded to show tables like 'vista\_admin\_ventas', 'vista\_auditor', and 'vista\_empleado\_ventas'. A script editor window titled 'Script-7' contains the SQL query: `SELECT * from vista_admin_ventas;`. Below the script, a results grid titled 'vista\_admin\_ventas 1' displays three rows of data:

cliente_id	nombre	dni	telefono	email	saldo	numero_pedidos	total_gastado
1	Vladimiro Putinino	11223344C	600555666	ervladi@ruski.com	1	1	1.200
2	Donald Trumpino	98765432B	600333444	trump@usa.com	5.000.000	1	75,25
3	Perro Sanchez	12345678A	600111222	perrete@psoe.com	-9.999.999	2	500,5

"auditor": select \* from vista\_auditor;

The screenshot shows the MySQL Workbench interface. On the left, the 'Navegador de Bases de Datos' (Database Navigator) lists various database connections and objects. The 'ventas\_db-auditor' connection is selected. Within this connection, the 'Bases de Datos' section is expanded, showing the 'ventas\_db' database. Under 'ventas\_db', the 'public' schema is selected, and its 'Tablas' (Tables) section is shown, listing 'clientes', 'pedidos', 'Foreign Tables', 'Vistas', 'Vistas Materializadas', 'Indices', 'Funciones', 'Secuencias', 'Tipos de datos', and 'Aggregate functions'. Other sections like 'Event Triggers', 'Extensiones', 'Storage', 'Info del Sistema', and 'Roles' are also listed. On the right, the main workspace displays the results of the SQL query 'select \* from vista\_auditor;'. The results are presented in a grid titled 'vista\_auditor 1'. The columns are labeled: cliente\_id, saldo, pedido\_id, fecha, total, and estado. The data shows four rows of information:

cliente_id	saldo	pedido_id	fecha	total	estado
1	-9.999.999	1	2025-11-13	150	Entregado
2	-9.999.999	2	2025-11-13	350,5	En Proceso
3	5.000.000	3	2025-11-13	75,25	Pendiente
4	1	4	2025-11-13	1.200	Entregado

"empleado\_ventas": select \* from vista\_empleado\_ventas;

The screenshot shows the MySQL Workbench interface. On the left, the 'Navegador de Bases de Datos' (Database Navigator) lists various database connections and projects. A connection named 'ventas\_db-empleado\_ventas' is selected, highlighted with a grey bar. The main workspace consists of two panes: a 'SQL' pane at the top containing the query 'select \* from vista\_empleado\_ventas;' and a 'Results' pane below it displaying the output of the query.

**SQL Pane:**

```
select * from vista_empleado_ventas;
```

**Results Pane:**

id	nombre	telefono	saldo
1	Perro Sanchez	600111222	-9.999.999
2	Donaldo Trumpino	600333444	5.000.000
3	Vladimiro Putinino	600555666	1

## Ampliación

11. Crea una nueva vista `vista_clientes_negativos` que muestre solo clientes con saldo menor que 0.

***CREATE VIEW vista\_clientes\_negativos AS***

***SELECT***

```
    id,
    nombre,
    telefono,
    saldo
FROM
    clientes
WHERE
    saldo < 0;
```

```
ventas_db=# CREATE VIEW vista_clientes_negativos AS
SELECT
    id,
    nombre,
    telefono,
    saldo
FROM
    clientes
WHERE
    saldo < 0;
CREATE VIEW
```

12. Asigna esta vista al rol `ventas_acceso` para que todos los usuarios de ventas puedan consultarla.

***GRANT SELECT ON vista\_clientes\_negativos TO ventas\_acceso;***

```
ventas_db=# GRANT SELECT ON vista_clientes_negativos TO ventas_acceso;
GRANT
ventas_db=# |
```

