

ACTIVIDAD 2 - CREACIÓN DE VISTAS PERSONALIZADAS

Cristóbal Suárez Abad

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

Contenido

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

Actividad 2 - Creación de vistas personalizadas

Creación de tablas y datos base

1. 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=# |
```

2. Crea las tablas e inserta algunos registros de ejemplo:
 - o 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 | Donald 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

- 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 X									
SELECT c.id AS cliente_id, c.nombre, c.dni, c.telefono, c.en Enter a SQL expression to filter results (use Ctrl+Space)									
	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	Donaldo 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
```

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

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

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

```
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, p.total, p.estado

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

```
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;

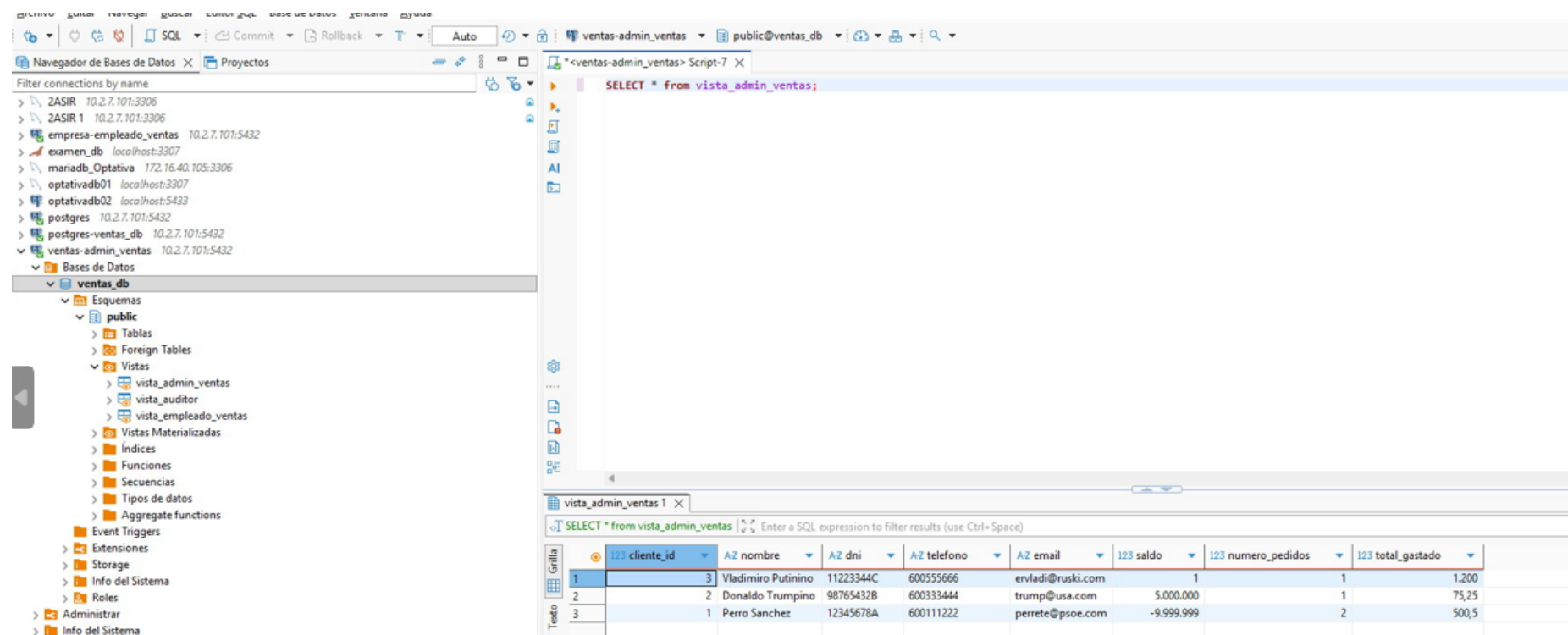
```
ventas_db=# REVOKE ALL ON ALL TABLES IN SCHEMA public FROM PUBLIC;
REVOKE
ventas_db=# |
```

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 a database management tool interface. On the left, a tree view displays the database structure, including a schema named 'ventas_db' with a view 'vista_admin_ventas'. The main window shows a SQL query: `SELECT * from vista_admin_ventas;`. Below the query, the results are displayed in a table with 10 columns: cliente_id, nombre, dni, telefono, email, saldo, numero_pedidos, total_gastado, and an additional column for the total. The results show three rows of data.

	cliente_id	nombre	dni	telefono	email	saldo	numero_pedidos	total_gastado	
1	3	Vladimiro Putinino	11223344C	600555666	ervladi@ruski.com	1	1	1,200	
2	2	Donaldo 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	

“auditor”: `select * from vista_auditor;`

The screenshot shows a database management interface with a left sidebar for the database structure and a main area for SQL queries and results.

Database Structure (Left Sidebar):

- Filter connections by name
- 2ASIR 10.2.7.101:3306
- 2ASIR 1 10.2.7.101:3306
- empresa-empleado_ventas 10.2.7.101:5432
- examen_db localhost:3307
- mariadb_Optativa 172.16.40.105:3306
- optivadb01 localhost:3307
- optivadb02 localhost:5433
- postgres 10.2.7.101:5432
- postgres-ventas_db 10.2.7.101:5432
- ventas_db-admin_ventas 10.2.7.101:5432
- ventas_db-auditor 10.2.7.101:5432**
 - Bases de Datos
 - ventas_db
 - Esquemas
 - public
 - Tablas
 - clientes 40K
 - pedidos 24K
 - Foreign Tables
 - Vistas
 - Vistas Materializadas
 - Índices
 - Funciones
 - Secuencias
 - Tipos de datos
 - Aggregate functions
 - Event Triggers
 - Extensiones
 - Storage
 - Info del Sistema
 - Roles
 - Administrar
 - Info del Sistema

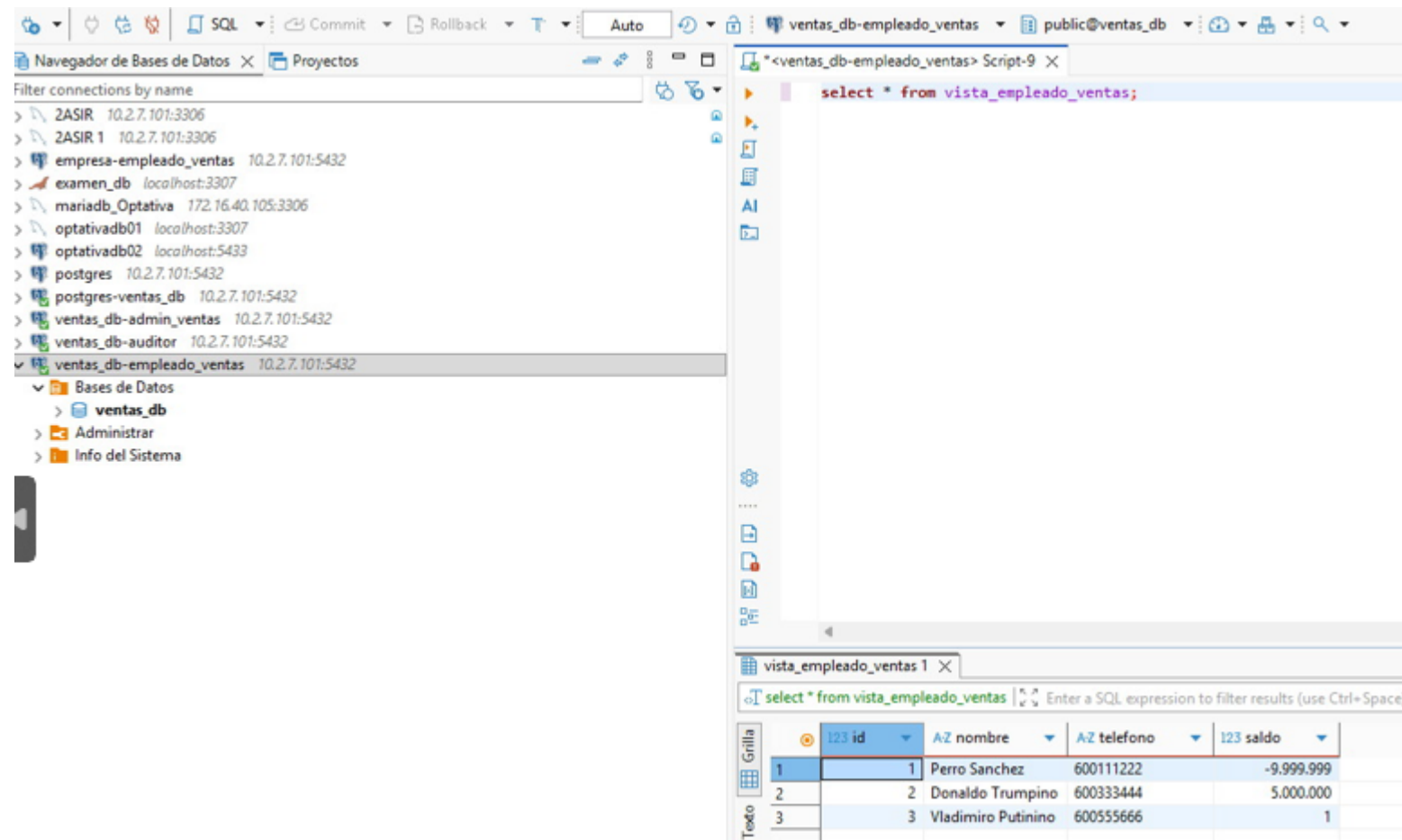
SQL Query (Script-8):

```
select * from vista_auditor;
```

Query Results (vista_auditor 1):

	cliente_id	saldo	pedido_id	fecha	total	estado
1	1	-9.999.999	1	2025-11-13	150	Entregado
2	1	-9.999.999	2	2025-11-13	350,5	En Proceso
3	2	5.000.000	3	2025-11-13	75,25	Pendiente
4	3	1	4	2025-11-13	1,200	Entregado

“empleado_ventas”: **select * from vista_empleado_ventas;**



The screenshot shows a database management interface with a left sidebar containing a tree view of database connections. The main window displays a SQL query in a script editor, and the bottom pane shows the results of the query in a table format.

Database Connections (Left Sidebar):

- 2ASIR 10.2.7.101:3306
- 2ASIR 1 10.2.7.101:3306
- empresa-empleado_ventas 10.2.7.101:5432
- examen_db localhost:3307
- mariadb_Optativa 172.16.40.105:3306
- optativadb01 localhost:3307
- optativadb02 localhost:5433
- postgres 10.2.7.101:5432
- postgres-ventas_db 10.2.7.101:5432
- ventas_db-admin_ventas 10.2.7.101:5432
- ventas_db-auditor 10.2.7.101:5432
- ventas_db-empleado_ventas 10.2.7.101:5432**
- Bases de Datos
 - ventas_db
 - Administrar
 - Info del Sistema

SQL Query (Script Editor):

```
select * from vista_empleado_ventas;
```

Query Results (Table):

	id	nombre	telefono	saldo
1	1	Perro Sanchez	600111222	-9.999.999
2	2	Donaldo Trumpino	600333444	5.000.000
3	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=#
```

