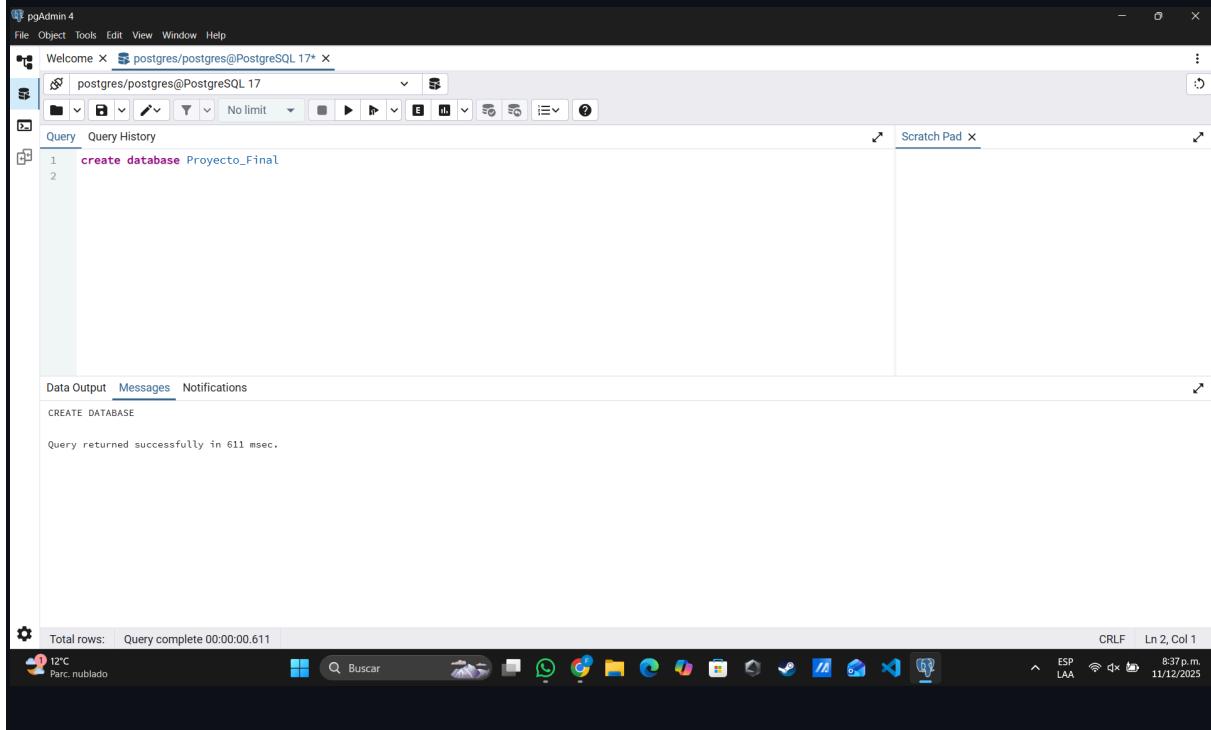


/sql/create_tables.sql



pgAdmin 4

Welcome × postgres/postgres@PostgreSQL 17*

Query Query History

```
1 create database Proyecto_Final
```

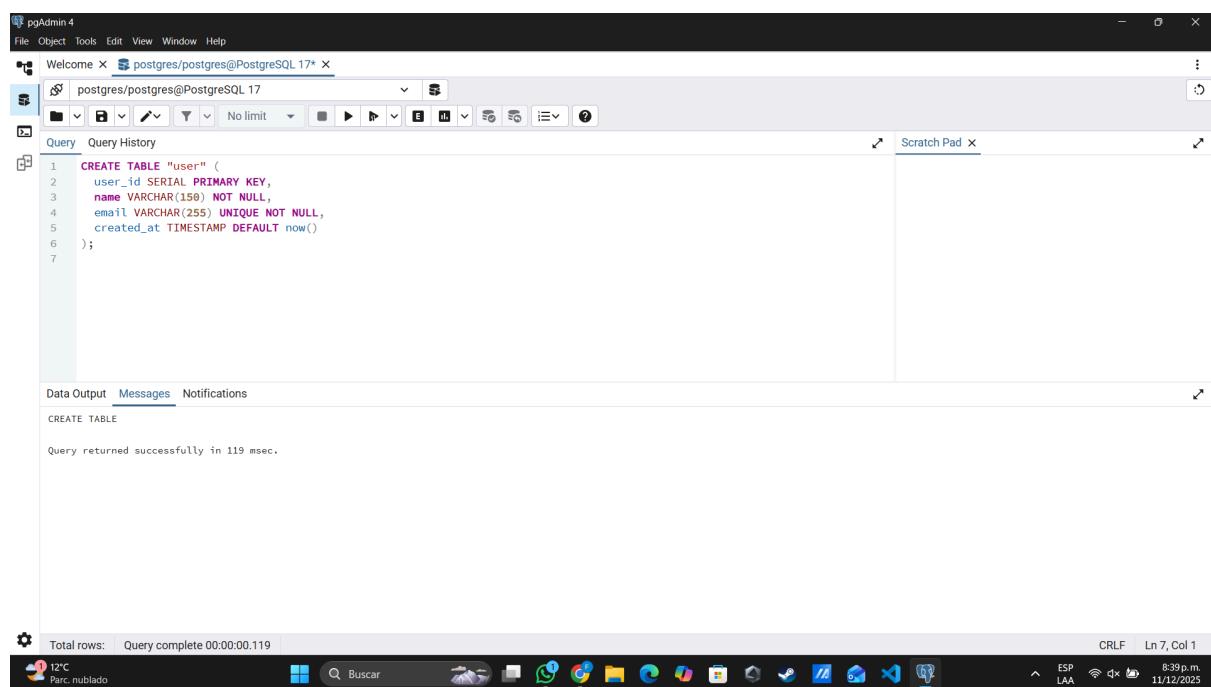
Data Output Messages Notifications

```
CREATE DATABASE
```

Query returned successfully in 611 msec.

Total rows: Query complete 00:00:00.611 CRLF Ln 2, Col 1

12°C Parc. nublado Buscar 8:37 p.m. 11/12/2025



pgAdmin 4

Welcome × postgres/postgres@PostgreSQL 17*

Query Query History

```
1 CREATE TABLE "user" (
2     user_id SERIAL PRIMARY KEY,
3     name VARCHAR(150) NOT NULL,
4     email VARCHAR(255) UNIQUE NOT NULL,
5     created_at TIMESTAMP DEFAULT now()
6 );
7
```

Data Output Messages Notifications

```
CREATE TABLE
```

Query returned successfully in 119 msec.

Total rows: Query complete 00:00:00.119 CRLF Ln 7, Col 1

12°C Parc. nublado Buscar 8:39 p.m. 11/12/2025

The screenshot shows the pgAdmin 4 interface with the following details:

- File Bar:** File, Object, Tools, Edit, View, Window, Help.
- Title Bar:** Welcome × postgres/postgres@PostgreSQL 17*
- Toolbar:** Includes icons for New Query, Open, Save, Copy, Paste, Run, Stop, Refresh, and Help.
- Query Pad:** Contains the SQL code for creating the booking table:

```
1 CREATE TABLE booking (
2     booking_id SERIAL PRIMARY KEY,
3     customer_id INT NOT NULL REFERENCES customer(customer_id) ON DELETE CASCADE,
4     branch_id INT NOT NULL REFERENCES branch(branch_id),
5     scheduled_at TIMESTAMP NOT NULL,
6     status VARCHAR(30) NOT NULL DEFAULT 'scheduled',
7     created_at TIMESTAMP DEFAULT now()
8 );
```
- Data Output Tab:** Shows the result of the query: "Query returned successfully in 62 msec."
- System Tray:** Shows battery level (12°C), network status (Parc. nublado), system icons, and a message: "Query returned successfully in 62 msec." with a timestamp: 8:46 p.m. 11/12/2025.

The screenshot shows the pgAdmin 4 interface with the following details:

- File Bar:** File, Object, Tools, Edit, View, Window, Help.
- Title Bar:** Welcome × postgres/postgres@PostgreSQL 17*
- Toolbar:** Includes icons for New Query, Open, Save, Copy, Paste, Run, Stop, Refresh, and Help.
- Query Pad:** Contains the SQL code for creating the booking_service table:

```
1 CREATE TABLE booking_service (
2     booking_id INT NOT NULL REFERENCES booking(booking_id) ON DELETE CASCADE,
3     service_id INT NOT NULL REFERENCES service(service_id) ON DELETE CASCADE,
4     PRIMARY KEY (booking_id, service_id)
5 );
6
```
- Data Output Tab:** Shows the result of the query: "Query returned successfully in 63 msec."
- System Tray:** Shows battery level (12°C), network status (Parc. nublado), system icons, and a message: "Query returned successfully in 63 msec." with a timestamp: 8:46 p.m. 11/12/2025.

The screenshot shows the pgAdmin 4 interface with the following details:

- File Bar:** File, Object, Tools, Edit, View, Window, Help.
- Title Bar:** Welcome × postgres/postgres@PostgreSQL 17*
- Toolbar:** Includes icons for New Query, Open, Save, Copy, Paste, Run, Stop, Refresh, and Help.
- Query Pad:** Contains the SQL code for creating the booking_service table:

```
1 CREATE TABLE booking_service (
2     booking_id INT NOT NULL REFERENCES booking(booking_id) ON DELETE CASCADE,
3     service_id INT NOT NULL REFERENCES service(service_id) ON DELETE CASCADE,
4     PRIMARY KEY (booking_id, service_id)
5 );
6
```
- Data Output Tab:** Shows the result of the query: "Query returned successfully in 63 msec."
- System Tray:** Shows battery level (12°C), network status (Parc. nublado), system icons, and a message: "Query returned successfully in 63 msec." with a timestamp: 8:46 p.m. 11/12/2025.

The screenshot shows the pgAdmin 4 interface with the following details:

- Title Bar:** pgAdmin 4
- Toolbar:** File, Object, Tools, Edit, View, Window, Help
- Connections:** Welcome (selected), postgres/postgres@PostgreSQL 17*
- Query Editor:** Contains the SQL code for creating the payment table:

```
1 CREATE TABLE payment (
2     payment_id SERIAL PRIMARY KEY,
3     booking_id INT NOT NULL REFERENCES booking(booking_id) ON DELETE CASCADE,
4     amount NUMERIC(18,2) NOT NULL,
5     method VARCHAR(50),
6     paid_at TIMESTAMP
7 );
```
- Data Output:** Shows the message "Query returned successfully in 59 msec."
- Scratch Pad:** Available for use.

The screenshot shows the pgAdmin 4 interface with the following details:

- Title Bar:** pgAdmin 4
- Toolbar:** File, Object, Tools, Edit, View, Window, Help
- Connections:** Welcome (selected), postgres/postgres@PostgreSQL 17*
- Query Editor:** Contains the SQL code for creating the customer table:

```
1 CREATE TABLE customer (
2     customer_id SERIAL PRIMARY KEY,
3     user_id INT NOT NULL REFERENCES "user"(user_id) ON DELETE CASCADE,
4     document_id VARCHAR(50),
5     phone VARCHAR(30)
6 );
```
- Data Output:** Shows the message "Query returned successfully in 66 msec."
- Scratch Pad:** Available for use.

The screenshot shows the pgAdmin 4 interface with the following details:

- Title Bar:** pgAdmin 4
- Toolbar:** File, Object, Tools, Edit, View, Window, Help
- Connections:** Welcome (selected), postgres/postgres@PostgreSQL 17*
- Query Editor:** Contains the SQL code for creating the customer table:

```
1 CREATE TABLE customer (
2     customer_id SERIAL PRIMARY KEY,
3     user_id INT NOT NULL REFERENCES "user"(user_id) ON DELETE CASCADE,
4     document_id VARCHAR(50),
5     phone VARCHAR(30)
6 );
```
- Data Output:** Shows the message "Query returned successfully in 66 msec."
- Scratch Pad:** Available for use.

The screenshot shows the pgAdmin 4 interface with a query editor window. The query is:

```
1 CREATE VIEW vw_booking_details AS
2 SELECT b.booking_id, u.name as customer_name, br.name as branch_name, b.scheduled_at, b.status
3 FROM booking b
4 JOIN customer c ON c.customer_id = b.customer_id
5 JOIN "user" u ON u.user_id = c.user_id
6 JOIN branch br ON br.branch_id = b.branch_id;
```

The status bar at the bottom indicates "Query returned successfully in 70 msec." and "CRLF Ln 6, Col 46".

The screenshot shows the pgAdmin 4 interface with a query editor window. The query is:

```
1 CREATE TABLE employee (
2     employee_id SERIAL PRIMARY KEY,
3     user_id INT NOT NULL REFERENCES "user"(user_id) ON DELETE CASCADE,
4     role VARCHAR(50)
5 );
```

The status bar at the bottom indicates "Query returned successfully in 70 msec." and "CRLF Ln 5, Col 3".

The image shows three separate sessions in pgAdmin 4, each displaying a successful SQL query execution. The top session creates a 'branch' table with columns: branch_id (SERIAL PRIMARY KEY), name (VARCHAR(150) NOT NULL), and address (TEXT). The middle session creates a 'service' table with columns: service_id (SERIAL PRIMARY KEY), name (VARCHAR(150) NOT NULL), duration_min (INT NOT NULL), and price (NUMERIC(10,2) NOT NULL). The bottom session creates a 'data' table with columns: id (SERIAL PRIMARY KEY), service_id (INT NOT NULL), quantity (INT NOT NULL), and unit (VARCHAR(50)). Each session includes a status bar at the bottom showing the total rows affected, the query completion time (e.g., 00:00:00.081), and the operating system's taskbar.

```
CREATE TABLE branch (
    branch_id SERIAL PRIMARY KEY,
    name VARCHAR(150) NOT NULL,
    address TEXT
);

CREATE TABLE service (
    service_id SERIAL PRIMARY KEY,
    name VARCHAR(150) NOT NULL,
    duration_min INT NOT NULL,
    price NUMERIC(10,2) NOT NULL
);

CREATE TABLE data (
    id SERIAL PRIMARY KEY,
    service_id INT NOT NULL,
    quantity INT NOT NULL,
    unit VARCHAR(50)
);
```

/sql/insert_data.sql

```
pgAdmin 4
File Object Tools Edit View Window Help
Welcome × postgres/postgres@PostgreSQL 17* ×
postgres/postgres@PostgreSQL 17
Query History
1 INSERT INTO branch (name, address) VALUES
2 ('Sucursal Centro','Calle 1 #10-20'),
3 ('Sucursal Norte','Av 5 #30-40');

Data Output Messages Notifications
INSERT 0 2
Query returned successfully in 80 msec.
```

```
Total rows: Query complete 00:00:00.080
12°C Parc. nublado Buscar CRLF Ln 3, Col 34
8:49 p.m. 11/12/2025

pgAdmin 4
File Object Tools Edit View Window Help
Welcome × postgres/postgres@PostgreSQL 17* ×
postgres/postgres@PostgreSQL 17
Query History
1 INSERT INTO service (name, duration_min, price) VALUES
2 ('Cambio de aceite', 30, 45.00),
3 ('Alineación', 60, 80.00),
4 ('Revisión general', 45, 60.00);

Data Output Messages Notifications
INSERT 0 3
Query returned successfully in 63 msec.
```

```
Total rows: Query complete 00:00:00.063
12°C Parc. nublado Buscar CRLF Ln 4, Col 33
8:49 p.m. 11/12/2025

pgAdmin 4
File Object Tools Edit View Window Help
Welcome × postgres/postgres@PostgreSQL 17* ×
postgres/postgres@PostgreSQL 17
Query History
```

The image displays three separate sessions of the pgAdmin 4 interface, each showing a successful SQL query execution. The top session shows an 'INSERT' into the 'booking' table with three rows. The middle session shows an 'INSERT' into the 'booking_service' table with two rows. The bottom session shows an 'INSERT' into the 'booking_service' table with three rows. Each session includes a message indicating success and the time taken.

```
pgAdmin 4
File Object Tools Edit View Window Help
Welcome × postgres/postgres@PostgreSQL 17* ×
postgres/postgres@PostgreSQL 17
Query History
Scratch Pad ×

1 INSERT INTO booking (customer_id, branch_id, scheduled_at, status) VALUES
(1, 1, now() + interval '2 day', 'scheduled'),
(1, 2, now() + interval '7 day', 'scheduled'),
(2, 1, now() + interval '1 day', 'cancelled');

Data Output Messages Notifications
INSERT 0 3
Query returned successfully in 73 msec.

Total rows: Query complete 00:00:00.073
12°C Parc. nublado Buscar ESP LAA 8:51 p.m. 11/12/2025
CRLF Ln 4, Col 47

pgAdmin 4
File Object Tools Edit View Window Help
Welcome × postgres/postgres@PostgreSQL 17* ×
postgres/postgres@PostgreSQL 17
Query History
Scratch Pad ×

1 INSERT INTO booking_service (booking_id, service_id) VALUES
(1,1),(1,3),(2,2);

Data Output Messages Notifications
INSERT 0 3
Query returned successfully in 65 msec.

Total rows: Query complete 00:00:00.065
12°C Parc. nublado Buscar ESP LAA 8:51 p.m. 11/12/2025
CRLF Ln 2, Col 19
```

```
pgAdmin 4
File Object Tools Edit View Window Help
Welcome × postgres/postgres@PostgreSQL 17* ×
postgres/postgres@PostgreSQL 17
Query History
1 INSERT INTO payment (booking_id, amount, method, paid_at) VALUES
2 (1, 105.00, 'card', now()),
3 (2, 80.00, 'cash', null);

Data Output Messages Notifications
INSERT 0 2
Query returned successfully in 66 msec.

Total rows: Query complete 00:00:00.066
12°C Parc. nublado Buscar CRLF Ln 3, Col 26
8:52 p.m. 11/12/2025
File Object Tools Edit View Window Help
Welcome × postgres/postgres@PostgreSQL 17* ×
postgres/postgres@PostgreSQL 17
Query History
1 INSERT INTO "user" (name, email) VALUES
2 ('Ana Perez', 'ana.perez@example.com'),
3 ('Carlos Lopez', 'carlos.lopez@example.com'),
4 ('Sofia Martinez', 'sofia.m@example.com'),
5 ('Diego Ruiz', 'diego.ruiz@example.com');

Data Output Messages Notifications
INSERT 0 4
Query returned successfully in 90 msec.

Total rows: Query complete 00:00:00.090
12°C Parc. nublado Buscar CRLF Ln 5, Col 41
8:49 p.m. 11/12/2025
```

The image shows three separate sessions in pgAdmin 4. Each session consists of a query editor window and a results pane.

- Session 1:** Inserts into the `payment` table. The query is:

```
1 INSERT INTO payment (booking_id, amount, method, paid_at) VALUES
2 (1, 105.00, 'card', now()),
3 (2, 80.00, 'cash', null);
```

It returns a success message: "Query returned successfully in 66 msec."
- Session 2:** Inserts into the `"user"` table. The query is:

```
1 INSERT INTO "user" (name, email) VALUES
2 ('Ana Perez', 'ana.perez@example.com'),
3 ('Carlos Lopez', 'carlos.lopez@example.com'),
4 ('Sofia Martinez', 'sofia.m@example.com'),
5 ('Diego Ruiz', 'diego.ruiz@example.com');
```

It returns a success message: "Query returned successfully in 90 msec."
- Session 3:** Inserts into another `"user"` table. The query is:

```
1 INSERT INTO "user" (name, email) VALUES
2 ('Ana Perez', 'ana.perez@example.com'),
3 ('Carlos Lopez', 'carlos.lopez@example.com'),
4 ('Sofia Martinez', 'sofia.m@example.com'),
5 ('Diego Ruiz', 'diego.ruiz@example.com');
```

It returns a success message: "Query returned successfully in 90 msec."

The status bar at the bottom of each session window shows the total rows inserted, the query completion time, and the current date and time (11/12/2025). The system tray icons indicate battery level (12°C), signal strength, and network status.

The image shows three separate sessions of the pgAdmin 4 interface, each displaying a successful SQL query execution. The top session inserts three rows into the 'customer' table. The middle session inserts three rows into the 'employee' table. The bottom session inserts three rows into the 'customer' table. Each session includes a terminal window showing the command entered, the number of rows inserted, and a success message. The pgAdmin interface features a toolbar, a query history panel, and a scratch pad.

```
1 INSERT INTO customer (user_id, document_id, phone) VALUES
2 (1, 'CC12345', '30801112222'),
3 (3, 'CC98765', '30803334444');
4
```

```
1 INSERT INTO employee (user_id, role) VALUES
2 (2, 'technician'),
3 (4, 'reception');
```

```
1 INSERT INTO customer (user_id, document_id, phone) VALUES
2 (1, 'CC12345', '30801112222'),
3 (3, 'CC98765', '30803334444');
```

/sql/queries_avanzadas.sql

```
pgAdmin 4
File Object Tools Edit View Window Help
Welcome × postgres/postgres@PostgreSQL 17* ×
postgres/postgres@PostgreSQL 17
Query History
Scratch Pad ×
Query
1 SELECT s.service_id, s.name, p.payment_id
2   FROM service s
3 FULL OUTER JOIN booking_service bs ON bs.service_id = s.service_id
4 FULL OUTER JOIN booking b ON b.booking_id = bs.booking_id
5 FULL OUTER JOIN payment p ON p.booking_id = b.booking_id;
```

Data Output Messages Notifications

	service_id	name	payment_id
1	3	Revisión general	1
2	1	Cambio de aceite	1
3	2	Alineación	2
4	[null]	[null]	[null]

Showing rows: 1 to 4 Page No: 1 of 1 < > << >> <<< >>>

Total rows: 4 Query complete 00:00:00.094 ✓ CRLF Ln 5, Col 58
12°C Parc. nublado Buscar 8:53 p.m. 11/12/2025

```
pgAdmin 4
File Object Tools Edit View Window Help
Welcome × postgres/postgres@PostgreSQL 17* ×
postgres/postgres@PostgreSQL 17
Query History
Scratch Pad ×
Query
1 SELECT br.branch_id, br.name, SUM(p.amount) AS total
2   FROM payment p
3   JOIN booking b ON b.booking_id = p.booking_id
4   JOIN branch br ON br.branch_id = b.branch_id
5   GROUP BY br.branch_id, br.name
6   HAVING SUM(p.amount) > 100;
```

Data Output Messages Notifications

	branch_id	name	total
1	1	Sucursal Centro	105.00

Showing rows: 1 to 1 Page No: 1 of 1 < > << >> <<< >>>

Total rows: 1 Query complete 00:00:00.410 ✓ CRLF Ln 6, Col 28
12°C Parc. nublado Buscar 8:53 p.m. 11/12/2025

pgAdmin 4

File Object Tools Edit View Window Help

Welcome × postgres/postgres@PostgreSQL 17*

postgres/postgres@PostgreSQL 17

Query History

Scratch Pad

```
1 SELECT c.customer_id, u.name
2 FROM customer c
3 JOIN "user" u ON u.user_id = c.user_id
4 WHERE (SELECT SUM(amount) FROM payment p JOIN booking b ON b.booking_id = p.booking_id WHERE b.customer_id = c.customer_id)
      (SELECT AVG(total) FROM (SELECT SUM(amount) as total FROM payment GROUP BY booking_id) t);
```

Data Output Messages Notifications

customer_id name

customer_id	name
1	Ana Perez

Showing rows: 1 to 1 Page No: 1 of 1 << >> >>>

Total rows: 1 Query complete 00:00:00.101 ✓ Successfully run. Total query runtime: 101 msec. 1 rows affected. CRLF Ln 5, Col 97

12°C Parc. nublado Buscar

8:53 p.m. 11/12/2025

pgAdmin 4

File Object Tools Edit View Window Help

Welcome × postgres/postgres@PostgreSQL 17*

postgres/postgres@PostgreSQL 17

No limit

Query History

Scratch Pad

```

1 SELECT b.booking_id, u.name AS customer_name, br.name AS branch_name
2 FROM booking b
3 JOIN customer c ON c.customer_id = b.customer_id
4 JOIN "user" u ON u.user_id = c.user_id
5 JOIN branch br ON br.branch_id = b.branch_id
6 WHERE b.status = 'scheduled';

```

Data Output Messages Notifications

Showing rows: 1 to 2 | Page No: 1 of 1 | < > << >> <<< >>>

booking_id	customer_name	branch_name
1	Ana Perez	Sucursal Norte
2	Ana Perez	Sucursal Centro

Total rows: 2 Query complete 00:00:00.102

12°C Parc. nublado Buscar ESP LAA 8:52 p.m. 11/12/2025

Successfully run CRLF Ln 6, Col 30

pgAdmin 4

File Object Tools Edit View Window Help

Welcome × postgres/postgres@PostgreSQL 17*

postgres/postgres@PostgreSQL 17

No limit

Query History

Scratch Pad

```

1 SELECT u.name, b.booking_id, b.scheduled_at
2 FROM "user" u
3 LEFT JOIN customer c ON c.user_id = u.user_id
4 LEFT JOIN booking b ON b.customer_id = c.customer_id
5 ORDER BY u.user_id;
6

```

Data Output Messages Notifications

Showing rows: 1 to 5 | Page No: 1 of 1 | < > << >> <<< >>>

name	booking_id	scheduled_at
Ana Perez	1	2025-12-13 20:51:32.196644
Ana Perez	2	2025-12-18 20:51:32.196644
Carlos Lopez	[null]	[null]
Sofia Martinez	3	2025-12-12 20:51:32.196644
Diego Ruiz	[null]	[null]

Total rows: 5 Query complete 00:00:00.090

12°C Parc. nublado Buscar ESP LAA 8:52 p.m. 11/12/2025

Successfully run. Total query runtime: 90 msec. 5 rows affected. CRLF Ln 6, Col 1

pgAdmin 4

Welcome × postgres/postgres@PostgreSQL 17*

postgres/postgres@PostgreSQL 17

No limit

Query History

```

1 WITH bookings_in_range AS (
2   SELECT branch_id, scheduled_at
3   WHERE scheduled_at BETWEEN now() AND now() + interval '10 day'
4 )
5   SELECT br.branch_id, br.name, COUNT(bir.scheduled_at) AS reserved_in_next_10_days
6   FROM branch br
7   LEFT JOIN bookings_in_range bir ON bir.branch_id = br.branch_id
8   GROUP BY br.branch_id, br.name;
9 
```

Data Output

branch_id	name	reserved_in_next_10_days
1	Sucursal Norte	1
2	Sucursal Centro	2

Showing rows: 1 to 2 | Page No: 1 of 1 | < > << >> <<< >>>

Total rows: 2 Query complete 00:00:00.144

Successfully run. Total query runtime: 141 msec. 2 rows affected.

CRLF Ln 9, Col 1

12°C Parc. nublado Buscar 8:53 p.m. 11/12/2025

pgAdmin 4

Welcome × postgres/postgres@PostgreSQL 17*

postgres/postgres@PostgreSQL 17

No limit

Query History

```

1 WITH payments_per_booking AS (
2   SELECT booking_id, SUM(amount) AS paid_total FROM payment GROUP BY booking_id
3 )
4   SELECT b.booking_id, u.name, COALESCE(ppb.paid_total,0) as paid_total
5   FROM booking b
6   JOIN customer c ON c.customer_id = b.customer_id
7   JOIN "user" u ON u.user_id = c.user_id
8   LEFT JOIN payments_per_booking ppb ON ppb.booking_id = b.booking_id
9   WHERE COALESCE(ppb.paid_total,0) < (
10    SELECT AVG(amount) FROM payment
11 ); 
```

Data Output

booking_id	name	paid_total
1	Ana Perez	80.00
2	Sofia Martinez	0

Showing rows: 1 to 2 | Page No: 1 of 1 | < > << >> <<< >>>

Total rows: 2 Query complete 00:00:00.091

Noticias para usted... Nuevo temblor en... CRLF Ln 11, Col 3

8:54 p.m. 11/12/2025

/mongodb/inserts.json

A screenshot of a Windows desktop environment. In the center is a code editor window titled "mongodb". The left sidebar shows "OPEN EDITORS" with "consultas_aggregation.md" and "inserts.json" listed under "MONGODB". The main editor area contains the following JSON data:

```
[{"_id": {"$oid": "64b1f1a1c2a3e1a1a1a1a1"}, "user_id": 1, "preferences": {"notifications": true, "language": "es", "favorite_services": ["Cambio de aceite", "Alineación"]}, "addresses": [{"label": "Casa", "address": "Calle 1 #10-20"}, {"label": "Trabajo", "address": "Calle 2 #20-30"}], "created_at": {"$date": "2025-12-01T10:00:00Z"}, {"_id": {"$oid": "64b1f1a1c2a3e1a1a1a1a2"}, "user_id": 3, "preferences": {"notifications": false, "language": "en"}, "notes": "Cliente VIP", "created_at": {"$date": "2025-11-28T12:00:00Z"}]}
```

The status bar at the bottom shows "Ln 16, Col 2" and "1045 p.m. 11/12/2025".

/mongodb/consultas_aggregation.md

A screenshot of a Windows desktop environment. In the center is a code editor window titled "mongodb". The left sidebar shows "OPEN EDITORS" with "consultas_aggregation.md" and "inserts.json" listed under "MONGODB". The main editor area contains the following MongoDB aggregation pipeline:

```
// pipeline: agrupar por user_id, contar reservas, sumar pagos
[{"$match": { "events.o": { $exists: true } } },
 {"$unwind": "$events" },
 {"$match": { "events.type": "payment" } },
 {"$group": { _id: "$user_id", total_paid: { $sum: "$events.amount" }, payments_count: { $sum: 1 } } },
 {"$sort": { total_paid: -1 } }

[{"$match": { user_id: 1 } },
 {"$project: { user_id: 1, preferences: 1, last_event: { $arrayElemAt: [ "$events", -1 ] }, services: "$events.service_name" } },
 {"$unwind": { path: "$services", preserveNullAndEmptyArrays: true } },
 {"$group: { _id: "$user_id", servicesList: { $push: "$services" }, last_event: { $first: "$last_event" } } }
```

The status bar at the bottom shows "Ln 17, Col 2" and "1046 p.m. 11/12/2025".

/integracion/docker-compose.yml

The screenshot shows a Visual Studio Code interface with the following details:

- File Explorer:** On the left, it lists files and folders: **EXPLORER**, **OPEN EDITORS**, **INTEGRACION**, **node_modules**, **demo_integration.js**, **docker-compose.yml**, **package-lock.json**, and **package.json**.
- Editor:** The main editor area displays the **docker-compose.yml** file content. The file defines three services: **postgres**, **mongodb**, and **redis**.

```
version: '3.8'
services:
  postgres:
    image: postgres:15
    environment:
      POSTGRES_DB: appdb
      POSTGRES_USER: appuser
      POSTGRES_PASSWORD: apppass
    ports:
      - "5432:5432"
    volumes:
      - pgdata:/var/lib/postgresql/data

  mongodb:
    image: mongo:6
    ports:
      - "27017:27017"
    volumes:
      - mongodata:/data/db

  redis:
    image: redis:7
    ports:
      - "6379:6379"
    command: redis-server --save 60 1
    volumes:
      - pgdata:
          - mongodata:
```
- Search Bar:** At the top center, it says **integación**.
- Bottom Status Bar:** Shows **Ln 7, Col 25**, **Spaces: 2**, **UTF-8**, **CRLF**, **Compose**, and other system icons.

/integracion/demo_integration.js

The screenshot shows a Node.js application running in a terminal window. The application connects to PostgreSQL, MongoDB, and Redis, and performs operations such as querying users from PostgreSQL, finding profiles from MongoDB, incrementing a booking counter in Redis, and pushing notifications to Redis.

```
// Demo Node.js que muestra la integración conceptual entre PostgreSQL, MongoDB y Redis
// Ejecutar node demo_integration.js (aseguran npm i pg mongodb ioredis)

const Client = require('pg');
const MongoClient = require('mongodb');
const Redis = require('ioredis');

async function main() {
    // Conexiones
    const pg = new Client({ host: 'localhost', port: 5432, user: 'usuario', password: '1234', database: 'postgres' });
    const mongo = new MongoClient('mongodb://localhost:27017');
    const redis = new Redis();

    await pg.connect();
    await mongo.connect();
    const db = mongo.db('appdb');

    // 1) Leer usuario y perfil extendido
    const res = await pg.query("SELECT u.user_id, u.name FROM public.\"user\" u LIMIT 1");
    console.log(`User From PG: ${res.rows[0]}`);

    const profile = await db.collection('profiles').findOne({ user_id: res.rows[0].user_id });
    console.log(`Profile from MongoDB: ${profile}`);

    // 2) Incrementar contador global en Redis
    const bookingsCount = await redis.incr('counter:bookings_total');
    console.log(`Bookings counter (Redis): ${bookingsCount}`);

    // 3) Push a queue para procesar notificación
    await redis.push('queue:notifications', `notify:user:${res.rows[0].user_id}`);
    console.log(`Pushed notification job to Redis list`);

    // Clean up
    await pg.end();
    await mongo.close();
    redis.disconnect();
}

main().catch(err => { console.error(err); process.exit(1) });


```

CAPTURA FINAL FUNCIONAMIENTO AL 100%

The screenshot shows a Windows desktop environment with a Visual Studio Code (VS Code) window open. The window title is "C:\ integracion".

File Explorer: Shows files in the "INTEGRACION" folder, including "demo_integration.js" (selected), "docker-compose.yml", "package-lock.json", and "package.json".

Code Editor: Displays the content of "demo_integration.js". The code uses Node.js, PostgreSQL, MongoDB, and Redis to perform user profile integration.

```
JS demo_integration.js | docker-compose.yml
JS demo_integration.js > main
6 const redis = require('ioredis');
7
8 async function main() {
9     // conexiones
10    const pg = new Client({ host: 'localhost', port: 5432, user: 'usuario', password: '1234', database: 'postgres' });
11    const mongo = new MongoClient('mongodb://localhost:27017');
12    const redis = new Redis();
13
14    await pg.connect();
15    await mongo.connect();
16
17    const db = mongo.db('appdb');
18
19    // 1) Leer usuario y perfil extendido
20    const res = await pg.query(`SELECT u.user_id, u.name FROM public."user" u LIMIT 1`);
21
22    console.log('User from Pg:', res.rows[0]);
23
24    const profile = await db.collection('profiles').findOne({ user_id: res.rows[0].user_id });
25    console.log('Profile from MongoDB:', profile);
26
27    // 2) Incrementar contador global en Redis
```

Terminal: Shows the command "node demo_integration.js" being run, and the output indicating a user was found from PostgreSQL and a profile was found from MongoDB.

```
PS C:\Users\villa\OneDrive\Desktop\integracion> node demo_integration.js
column: undefined,
datatype: undefined,
constraint: undefined,
file: 'parse_relation.c',
line: 1431,
routine: 'parserOpenTable'
}
PS C:\Users\villa\OneDrive\Desktop\integracion> User from Pg: { user_id: 1, name: 'Ana Perez' }
Profile from MongoDB: null
Bookings counter (Redis): 1
Pushed notification job to Redis list
% PS C:\Users\villa\OneDrive\Desktop\integracion>
```

System Tray: Shows standard Windows icons for network, battery, and system status.