

1. Creare connessione con pgAdmin

1. Apri **pgAdmin**
2. Nel pannello a sinistra, clicca col tasto destro su "**Servers**" > **Register** > **Server...**
3. Inserisci un nome (es. *Postgres Locale*)
4. Tab **Connection**:
 - Host: localhost
 - Port: 5432
 - Maintenance database: postgres
 - Username: postgres
 - Password: inserisci la password
5. Clicca **Save**

2. Creare una tabella con pgAdmin

1. Espandi il server > Databases > postgres > Schemas > public > Tables
2. Tasto destro su **Tables** > **Create** > **Table...**
3. Inserisci:
 - **Table Name**: usa lo snake_case, es. automobile
4. Tab **Columns**:
 - Aggiungi i campi:
 - id - Integer - Primary Key - Not null
 - modello - Text - Not null
 - costo - Numeric - Not null
 - numero_vendite - Integer
 - anno_immatricolazione - Date
5. Clicca **Save**

3. Modificare una tabella

1. Tasto destro sulla tabella > **Properties**
2. Modifica i campi o aggiungine altri dal tab **Columns**
3. **Save** per salvare

4. Visualizzare dati

1. Tasto destro sulla tabella > **View/Edit Data** > **All Rows**

5. Eseguire stringhe SQL

1. Tasto destro sul database > **Query Tool**
2. Scrivi la query SQL
3. Premi il tasto ► **Execute** (F5)

6. Operazioni SQL

SELECT solo alcuni campi

```
-- Estrae solo modello e costo da automobile  
SELECT modello, costo FROM automobile;
```

INSERT

```
-- Inserisce un'auto nella tabella automobile  
INSERT INTO automobile (modello, costo, numero_vendite, anno_immatricolazione)  
VALUES ('Fiat Panda', 12000, 50, '2023-01-10');
```

UPDATE

```
-- Aggiorna il costo dell'auto  
UPDATE automobile SET costo = 12500 WHERE modello = 'Fiat Panda';
```

DELETE

```
-- Elimina un'auto  
DELETE FROM automobile WHERE modello = 'Fiat Panda';
```

7. Usare script da menu in pgAdmin

1. Tasto destro sul database > **Query Tool**
2. Menu **File** > **Open...** > seleziona file .sql
3. Esegui con ► **Execute**

8. Commenti SQL

```
-- Questo è un commento
/* Questo è un commento su più righe */
```

9. Alias

```
-- Alias per tabella e campo
SELECT a.modello AS nome_modello FROM automobile AS a;
```

10. JOIN con USING

```
-- Creazione seconda tabella: motore
CREATE TABLE motore (
  id INTEGER PRIMARY KEY,
  modello TEXT,
  tipo TEXT,
  data_produzione DATE
);

-- Inserimento dati
INSERT INTO motore VALUES (1, 'Fiat Panda', 'benzina', '2022-10-01');

-- JOIN tra automobile e motore usando USING
SELECT a.modello, costo, tipo
FROM automobile a
JOIN motore m USING (modello);
```

11. Condizioni: NULL, NOT NULL, LIKE, =, BETWEEN, >, >=, <, <=

```
-- Auto con costo tra 10000 e 15000
SELECT modello FROM automobile WHERE costo BETWEEN 10000 AND 15000;

-- Auto con anno >= 2022
SELECT modello FROM automobile WHERE anno_immatricolazione >= '2022-01-01';

-- Auto con modello che inizia con 'Fiat'
SELECT modello FROM automobile WHERE modello LIKE 'Fiat%';
```

12. Funzioni sulle date

```
-- Estrai anno da una data
SELECT modello, EXTRACT(YEAR FROM anno_immatricolazione) AS anno FROM automobile;
```

13. Raggruppamento dati

SENZA GROUP BY

```
-- Somma totale auto vendute
SELECT SUM(numero_vendite) FROM automobile;
```

CON GROUP BY

```
-- Totale vendite per modello
SELECT modello, SUM(numero_vendite) AS tot_vendite
FROM automobile
GROUP BY modello;
```

HAVING

```
-- Trova anno con più di 3 auto immatricolate
SELECT EXTRACT(YEAR FROM anno_immatricolazione) AS anno, COUNT(*) AS numero_auto
FROM automobile
GROUP BY anno
HAVING COUNT(*) > 3;
```

14. Subquery in WHERE

```
-- Mostra auto con costo maggiore della media
SELECT * FROM automobile
WHERE costo > (SELECT AVG(costo) FROM automobile);
```

15. Commenti su query esempio

```
-- Estrai modello e numero vendite
SELECT modello, numero_vendite
FROM automobile
```

-- SELECT: indica i campi da estrarre
-- FROM: tabella da cui prendere i dati

```
ORDER BY numero_vendite DESC;
```

-- ORDER BY: ordina per numero vendite decrescen

-- Calcola media costi

```
SELECT AVG(costo) AS media_costo  
FROM automobile;
```

-- AVG: calcola media, AS: assegna nome al risul

-- Conta quante auto ci sono

```
SELECT COUNT(*) FROM automobile;
```

```
COUNT(*) FROM automobile;
```