# **PROMPT**: give me the detailed documentation of insert command with examples in postgresql**.**

# **PostgreSQL INSERT Command Documentation**

## **What is INSERT in PostgreSQL?**

The INSERT command is used to add **new rows of data** into a table.

## Basic Syntax:

INSERT INTO table\_name (column1, column2, ...)

VALUES (value1, value2, ...);

* table\_name: The name of the table where data will be inserted.
* (column1, column2, ...): The list of columns you are inserting data into.
* (value1, value2, ...): The corresponding values to be inserted.

## **Table Used For Insertion**

CREATE TABLE students (

student\_id SERIAL PRIMARY KEY,

name VARCHAR(100),

age INT,

email VARCHAR(100),

enrollment\_date DATE

);

## 

## 

## **Examples**

### **1. Insert a single row**

INSERT INTO students (name, age, email, enrollment\_date)

VALUES ('John Doe', 20, 'john@example.com', '2025-08-01');

### **2. Insert multiple rows at once**

INSERT INTO students (name, age, email, enrollment\_date)

VALUES

('Alice Smith', 22, 'alice@example.com', '2025-08-02'),

('Bob Johnson', 21, 'bob@example.com', '2025-08-03');

### **3. Insert data into all columns (in order)**

INSERT INTO students

VALUES (DEFAULT, 'Charlie Brown', 23, 'charlie@example.com', '2025-08-04');

Note: DEFAULT lets PostgreSQL auto-fill the SERIAL student\_id.

### **4. Insert using data from another table**

INSERT INTO students (name, age, email, enrollment\_date)

SELECT name, age, email, CURRENT\_DATE

FROM temp\_students;

### **5. Insert with RETURNING clause (get back inserted row)**

INSERT INTO students (name, age, email, enrollment\_date)

VALUES ('Daisy', 24, 'daisy@example.com', CURRENT\_DATE)

RETURNING student\_id, name;

### **6. Insert NULL values**

INSERT INTO students (name, age, email, enrollment\_date)

VALUES ('Ethan', NULL, NULL, CURRENT\_DATE);

## **Notes on Data Types and Insertion**

* Match the data type of each column while inserting.
* Use NULL for unknown/missing values.
* Use DEFAULT if a column has a default value (e.g., SERIAL or CURRENT\_DATE).

### 

### **7. Insert with Subquery Example**

INSERT INTO students (name, age, email, enrollment\_date)

SELECT name, age, email, CURRENT\_DATE

FROM applicants

WHERE status = 'approved';

## **Insert with Conflict Handling**

### **8. Use ON CONFLICT to handle duplicates**

If email has a unique constraint, and you try to insert a duplicate:

INSERT INTO students (name, age, email, enrollment\_date)

VALUES ('Frank', 25, 'john@example.com', CURRENT\_DATE)

ON CONFLICT (email) DO NOTHING;

You can also update existing rows on conflict:

INSERT INTO students (name, age, email, enrollment\_date)

VALUES ('Frank', 25, 'john@example.com', CURRENT\_DATE)

ON CONFLICT (email) DO UPDATE

SET name = EXCLUDED.name,

age = EXCLUDED.age;

## **Insert from CSV or External Source (psql)**

\copy students(name, age, email, enrollment\_date)

FROM '/path/to/data.csv' DELIMITER ',' CSV HEADER;

## **Insert Performance Tips**

* Use bulk insert (multiple rows) to speed up large data insertion.
* Disable indexes and constraints temporarily for huge imports (use carefully).
* Use COPY for importing large data efficiently.

## **Common Errors**

| **Error** | **Reason** |
| --- | --- |
| ERROR: null value in column ... | Column is NOT NULL, but you inserted NULL |
| ERROR: duplicate key value violates unique constraint | Inserting duplicate in UNIQUE or PRIMARY KEY column |
| ERROR: column "..." does not exist | Wrong column name |
| ERROR: syntax error at or near ... | Typing or quotation mistake |

## **Summary**

| **Feature** | **Syntax** |
| --- | --- |
| Insert one row | INSERT INTO table (...) VALUES (...); |
| Insert multiple rows | INSERT INTO table (...) VALUES (...), (...); |
| Insert all columns | INSERT INTO table VALUES (...); |
| Use DEFAULT values | DEFAULT |
| Handle conflict | ON CONFLICT ... DO NOTHING/UPDATE |
| Return inserted data | RETURNING column |
| Insert from another table | INSERT INTO ... SELECT ... |