**Promt : Give in detail documentation of insert command and syntax. Explain in detail with 2 examples as a beginner in postgreSQL.**

**INSERT Command**

The **INSERT** command is used to add new rows (or records) to a table. It's a key part of **Data Manipulation Language (DML)**, which is concerned with managing data within a database's structure.

**Syntax**

There are two main ways to use the INSERT command:

**Method 1: Inserting values into all columns**

If you are providing values for every column in the table, you don't need to specify the column names. The values must be in the same order as the columns were defined when the table was created.

SQL

INSERT INTO table\_name VALUES (value1, value2, value3, ...);

* **table\_name**: The name of the table you want to insert data into.
* **VALUES:** A keyword that signals the start of the data you're inserting.
* **value1, value2, ...:** The data for each column, separated by commas.

**Method 2: Inserting values into specific columns**

This method is more flexible and is recommended for clarity. It allows you to specify which columns you're providing values for. This is particularly useful when some columns have a DEFAULT value or are set to auto-increment.

**SQL**

INSERT INTO table\_name (column1, column2, column3, ...) VALUES (value1, value2, value3, ...);

* (column1, column2, ...): A comma-separated list of the columns you are providing values for.
* (value1, value2, ...): The corresponding values for each column listed, also comma-separated. The order of the values must match the order of the columns.

**Examples for Beginners**

Let's use a sample table named employees to demonstrate the INSERT command.

**SQL**

-- Assume this table already exists

CREATE TABLE employees (

employee\_id SERIAL PRIMARY KEY,

first\_name VARCHAR(50) NOT NULL,

last\_name VARCHAR(50) NOT NULL,

hire\_date DATE DEFAULT CURRENT\_DATE,

department VARCHAR(50)

);

**Example 1: Inserting a new employee with full details 🧑‍💼**

Here, we'll insert a new employee and provide values for all the columns we can. We'll use the column-specific method, which is best practice. Notice we don't provide a value for employee\_id because it's a SERIAL type and will be automatically generated.

SQL

INSERT INTO employees (first\_name, last\_name, department)

VALUES ('John', 'Doe', 'Sales');

* We explicitly list the columns first\_name, last\_name, and department.
* The VALUES clause provides the data for these columns in the same order: 'John', 'Doe', and 'Sales'.
* The employee\_id will be automatically assigned (e.g., 1, 2, 3, etc.).
* The hire\_date will automatically default to the current date because of the DEFAULT CURRENT\_DATE constraint.

**Example 2: Inserting multiple rows at once 📝**

You can insert multiple rows in a single INSERT statement by separating the sets of values with a comma. This is more efficient than running multiple individual INSERT statements.

**SQL**

INSERT INTO employees (first\_name, last\_name, department)

VALUES

('Jane', 'Smith', 'Marketing'),

('Peter', 'Jones', 'Engineering'),

('Mary', 'Brown', 'Sales');

* After the VALUES keyword, we provide multiple sets of parentheses, each containing the data for a new row.
* Each set of values is separated by a comma.
* This one command adds three new employees to the employees table.

**🎯 Important Notes:**

* The **order of columns** in the INSERT query must match the order of values.
* You can **skip columns** that have DEFAULT values or are auto-incremented (SERIAL), like a primary key.
* Strings or text values should be wrapped in **single quotes** ('John', 'Email@gmail.com')
* If the data types don’t match the column types, you’ll get an error.

**Common Mistakes to Avoid:**

1. **Forgetting Quotes** around text:
   * Wrong: 'name' = John
   * Right: name = 'John'
2. **Wrong data type**:
   * Inserting text into an integer column will throw an error.
3. **Duplicate UNIQUE values**:
   * If a column like email is UNIQUE, you cannot insert the same email twice.
4. **NOT NULL constraint violation**:
   * If a column is marked NOT NULL, and you try to skip it, you’ll get an error.