Chapter: SQL's Data Addition Tool – INSERT

Topic: Inserting Data into Tables in SQL

Section 1: Learn

1.1 What Is the INSERT Statement?

The INSERT statement in SQL is used to **add new rows of data** to a table. It is one of the most common and essential DML (Data Manipulation Language) commands.

1.2 Basic Syntax

Inserting into All Columns:

```
INSERT INTO table_name
VALUES (value1, value2, ...);
```

> The order of values must match the table's column order.

Inserting into Specific Columns:

```
INSERT INTO table_name (column1, column2)
VALUES (value1, value2);
```

> Recommended for clarity and flexibility.

1.3 Example: Insert a New Student

```
INSERT INTO students (id, name, age, class)
VALUES (101, 'Ravi Kumar', 16, '10A');
```

1.4 Inserting Multiple Rows (Bulk Insert)

You can insert multiple rows using a single INSERT statement.

```
INSERT INTO students (id, name, age, class)
VALUES
(102, 'Priya Sharma', 15, '9B'),
(103, 'Aman Joshi', 17, '11C');
```

> Useful for uploading large datasets efficiently.

1.5 Inserting Data Using SELECT

Copy data from one table to another using:

```
INSERT INTO alumni (id, name, grad_year)
SELECT id, name, 2024
FROM students
WHERE class = '12A';
```

> This approach is useful for backups, transformations, and archiving.

1.6 Handling Identity (Auto-Increment) Columns

If a table has an auto-increment column (e.g., id), you **should not insert values** into that column explicitly unless needed.

Example:

```
CREATE TABLE employees (
  id INT AUTO_INCREMENT PRIMARY KEY,
  name VARCHAR(100),
  department VARCHAR(50)
);

INSERT INTO employees (name, department)
VALUES ('Anjali Desai', 'HR');
```

> The ID will be generated automatically.

1.7 Inserting NULL Values

If some columns are optional, you can insert NULL to represent missing data:

```
INSERT INTO students (id, name, age, email)
VALUES (104, 'Neha', 15, NULL);
```

> Ensure the column allows NULL values.

1.8 Error Handling and Best Practices

- Always specify column names to avoid errors during schema changes.
- Use transactions for bulk inserts in critical operations.
- Validate data types and constraints (e.g., NOT NULL, UNIQUE).
- Use INSERT IGNORE or ON DUPLICATE KEY UPDATE (MySQL) to avoid duplicate errors.

Section 2: Practise

Exercise 1: Insert a New Customer

```
INSERT INTO customers (name, email, city)
VALUES ('Ajay Singh', 'ajay@gmail.com', 'Delhi');
```

Exercise 2: Insert Multiple Products

```
INSERT INTO products (product_name, price, stock)
VALUES
('Laptop', 55000, 10),
('Keyboard', 750, 50),
('Mouse', 400, 80);
```

Exercise 3: Insert Using SELECT

```
INSERT INTO old_orders (order_id, customer_id, order_date)
SELECT id, customer_id, created_at
FROM orders
WHERE created_at < '2023-01-01';
```

Exercise 4: Insert Into Table with Auto-Increment ID

```
INSERT INTO feedback (comment, rating)
VALUES ('Excellent service', 5);
```

Exercise 5: Insert with NULLs

```
INSERT INTO employees (name, department, email)
VALUES ('Karan Mehta', 'Finance', NULL);
```

Section 3: FAQ – Know More

Q1. What if I insert fewer values than columns?

If the missing columns are defined with **default values** or allow **NULL**, the insert will work. Otherwise, it throws an error.

Q2. Can I insert into a view?

Yes, but only if the view is **updatable** and maps directly to a single table.

Q3. How do I insert the result of a calculation?

You can use expressions inside the VALUES clause:

```
INSERT INTO salaries (emp_id, annual_salary)
VALUES (101, 50000 * 12);
```

Q4. How do I prevent duplicate inserts?

- Use PRIMARY KEY or UNIQUE constraints.
- In MySQL, use INSERT IGNORE or ON DUPLICATE KEY UPDATE.

Q5. Can I insert data from CSV or Excel?

Yes, using:

- LOAD DATA INFILE (MySQL)
- COPY (PostgreSQL)
- Import wizard or tools like DBeaver, phpMyAdmin

End of Notes for Chapter: SQL's Data Addition Tool – INSERT