Data Manipulation Language(DML)

- Data Manipulation Language statements are used for manipulating data in database
- DML commands are not auto-committed like DDL Statements

  Changes done by DML commands can be rolled back
- Under DML Commands we perform:
  - **INSERT Statement**
  - **UPDATE Statement**
  - **DELETE Statement**
  - **SELECT Statement**

Insert Statement

- INSERT Statement used to add records to existing table
- To insert data into table, SQL INSERT INTO command can be used
- To insert few values in table as per columns names we can use following
- **Syntax:**

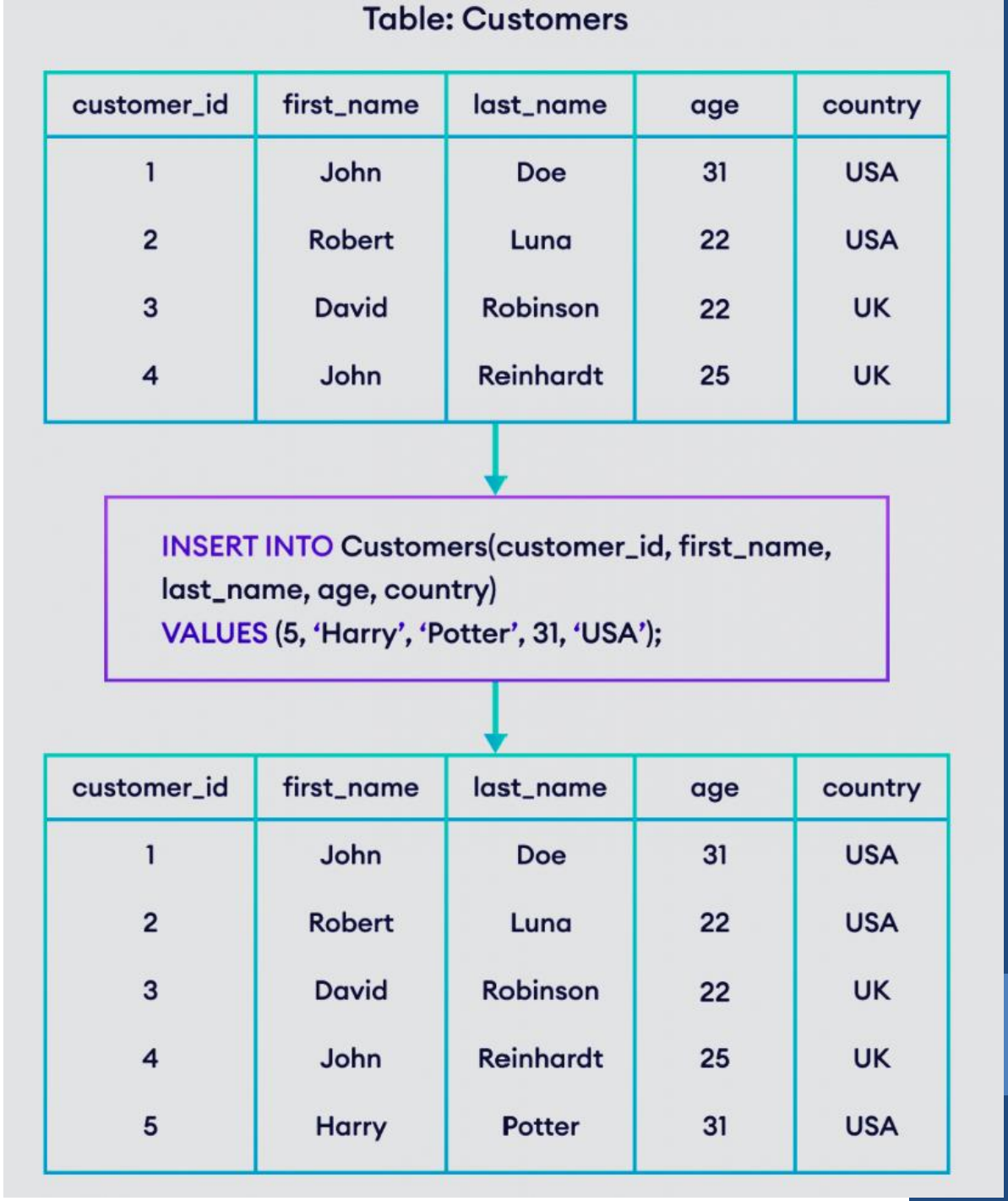
```
INSERT INTO table_name (column1, column2, column3, ...)
VALUES (value1, value2, value3, ...);
```

Insert Statement

- If all values for all the columns of the table are to be added then **no need to specify** the column names in SQL Query
- However, make sure the order of the values is in the same order as the columns in the table.
- Here, **syntax** would be as follows:

```
INSERT INTO table_name
VALUES (value1, value2, value3, ...);
```

Insert Statement



Insert Statement

Insert Row Providing Value Explicitly

- It's possible to provide default values to a column (for example, auto incrementing a column). In a database table, the **ID** field is usually unique auto incremented.
- In such cases, we can omit the value for that column during row insertion
- For example,

```sql
CREATE TABLE Customers(
    ID int AUTO_INCREMENT,
    FirstName varchar(25) NOT NULL,
    LastName varchar(25),
    Age int,
    Country varchar(25)
);
```

```sql
INSERT INTO Customers(first_name, last_name, age, country)
VALUES
('James', 'Bond', 48, 'USA');
```

Insert Statement

Insert Multiple Rows at Once in SQL

- It's also possible to insert multiple rows to a database table at once.
- For example,

```
INSERT INTO Customers(first_name, last_name, age, country)
VALUES
('Harry', 'Potter', 31, 'USA'),
('Chris', 'Hemsworth', 43, 'USA'),
('Tom', 'Holland', 26, 'UK');
```

Insert Statement

Insert rows Without Specifying Column Names

- It is also possible to insert values in a row without specifying column names.
- For example

```
INSERT INTO Customers
VALUES
(5, 'Chris', 'Evans', 42, 'USA');
```

Insert Statement

Not Including All Columns During Insertion

- If we skip column names during row insertion, the values of those columns will be NULL

```
INSERT INTO Customers(first_name, last_name, age)
VALUES
('Brad', 'Pitt', 58);
```

Update Statement

- Update statement is used to modify the existing data present in the table
- To update data in table, SQL UPDATE command can be used
- To update all rows in table we can use following
- **Syntax:**

UPDATE <Table_name>

SET column1=new_value

- For example,

```
UPDATE Customers
SET country = 'NP';
```

Update Statement

- The SQL UPDATE statement is used to edit existing rows in a database table. For example,

```
UPDATE Customers
SET first_name = 'Johnny'
WHERE customer_id = 1;
```

Update Multiple Values in a Row

- We can also update multiple values in a row at once. For example,

```
UPDATE Customers
SET first_name = 'Johnny', last_name = 'Depp'
WHERE customer_id = 1;
```

Update Statement

Update Multiple Rows

- The UPDATE statement can update multiple rows at once.
- For example,

```
UPDATE Customers
SET country = 'NP'
WHERE age = 22;
```

Delete Statement

- DELETE Statement is used to delete some or all records from existing table
- To delete data into a table, SQL DELETE command can be used

Delete all Rows in a Table

- The WHERE clause determines which rows to delete. However, we can delete all rows at once if we omit the WHERE clause.
- Syntax: DELETE FROM <TABLE_NAME>
- For example,

```
DELETE FROM Customers;
```

Delete Statement

- In SQL, we use the DELETE statement to delete specific row(s) from a database table.
- **Syntax**:

DELETE FROM <TABLE_NAME> WHERE<Condition>

- For example,

```
DELETE FROM Customers
WHERE customer_id = 5;
```

SELECT Statement

- Select statement is used to retrieve data from database.
- SELECT query can never make any change in the database.
- The data returned by the SELECT query is in the form of result sets
- **Syntax**:

```
SELECT column1, column2, columnN FROM table_name;
```

- The SQL SELECT statement is used to select (retrieve) data from a database table. For example,

```
SELECT first_name, last_name
FROM Customers;
```

SELECT Statement

- To select all columns from a database table, we use the * character. For example,

```
SELECT *
FROM Customers;
```

- A SELECT statement can have an optional WHERE clause. The WHERE clause allows us to fetch records from a database table that matches specified condition(s). For example,

```
SELECT *
FROM Customers
WHERE last_name = 'Doe';
```