

### **DML (Data Manipulation Language):**

- **SELECT:** एक या एक से अधिक table से data retrieve करने के लिए।
- **INSERT:** नया record enter करने के लिए।
- **UPDATE:** Record को modify करने के लिए।
- **DELETE:** रिकॉर्ड डिलीट करने के लिए।

## SQL-SELECT Statement

SQL में ज्यादातर SELECT Statement का इस्तेमाल किया जाता है |

SELECT Statement का इस्तेमाल database के table से data पाने (retrieve) के लिए किया जाता है |

जो returned हुआ data होता है वो table जैसा होता है और उस returned हुए data को result-set कहा जाता है |

### Syntax for SELECT Statement in SQL

#### For Single or Multiple Columns in SQL

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

#### For All Columns using \* in SQL

```
SELECT *  
  
FROM table_name;
```

### Sample Table

```
+-----+-----+-----+-----+-----+  
| id | Dealer   | Appliances           | Cost  | GST  |  
+-----+-----+-----+-----+-----+  
|  1 | Balaji   | T.V.                 | 19000 | 3420 |  
|  2 | Ravikant | Washing Machine     | 15000 | 4200 |  
|  3 | Electra  | Refrigerator        | 10000 | 2800 |  
|  4 | K.K.     | Cooler              | 18000 | 5040 |  
|  5 | Ganesh   | A.C.                | 35000 | 9800 |  
|  6 | Rajaram  | Iron                |  1500 |  420 |  
|  7 | Ravi     | Laptop              | 30000 | 8400 |  
+-----+-----+-----+-----+-----+
```

## Example For SELECT Statement with Single Column in SQL

Example पर Appliances इस table का सिर्फ Dealer ये column select किया गया है |

Source Code :

```
SELECT Dealer FROM Appliances;
```

Output :

```
+-----+
| Dealer |
+-----+
| Balaji |
| Ravikant |
| Electra |
| K.K. |
| Ganesh |
| Rajaram |
| Ravi |
+-----+
```

## Example For SELECT Statement with Multiple Columns in SQL

Example पर Appliances इस table के Dealer, Appliances और GST ये columns select किये गए हैं |

Source Code :

```
SELECT Dealer,Appliances,GST FROM Appliances;
```

Output :

```
+-----+-----+-----+
| Dealer | Appliances | GST |
+-----+-----+-----+
| Balaji | T.V. | 3420 |
| Ravikant | Washing Machine | 4200 |
```

Electra	Refrigerator	2800	
K.K.	Cooler	5040	
Ganesh	A.C.	9800	
Rajaram	Iron	420	
Ravi	Laptop	8400	
+-----+-----+-----+			

## Example For SELECT Statement with All (\*) Columns in SQL

Example पर Appliances इस table के सभी columns select किये गए है |

Source Code :

```
SELECT * FROM Appliances;
```

Output :

+---+-----+-----+-----+-----+					
id	Dealer	Appliances	Cost	GST	
+---+-----+-----+-----+-----+					
1	Balaji	T.V.	19000	3420	
2	Ravikant	Washing Machine	15000	4200	
3	Electra	Refrigerator	10000	2800	
4	K.K.	Cooler	18000	5040	
5	Ganesh	A.C.	35000	9800	
6	Rajaram	Iron	1500	420	
7	Ravi	Laptop	30000	8400	
+---+-----+-----+-----+-----+					

SELECT\_DISTINCT Statement का इस्तेमाल सिर्फ Unique values; return करने के लिए किया जाता है | कई बार Database के table के columns में duplicate values होती है | इन duplicate values को remove करने के लिए SELECT के साथ DISTINCT keyword का इस्तेमाल किया जाता है |

## Syntax for SELECT Statement in SQL

### For Single or Multiple Columns in SQL

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

### For Select all Columns using \* in SQL

```
SELECT DISTINCT *  
  
FROM table_name;
```

## Sample Table

id	Dealer	Appliances	Cost	GST
1	Balaji	T.V.	19000	3420
2	Ravikant	Washing Machine	15000	4200
3	Electra	Refrigerator	10000	2800
4	K.K.	Cooler	18000	5040
5	Ganesh	A.C.	35000	9800
6	Rajaram	Iron	1500	420
7	Ravi	Laptop	30000	8400
8	Ravikant	Washing Machine	15000	4200

## Example For SELECT\_DISTINCT Statement in SQL

Example पर Appliances table का 'Appliances' ये column select करके उसमे से unique values को return किया गया है |

Source Code :

```
SELECT DISTINCT Appliances FROM Appliances;
```

Output :

```
+-----+
| Appliances      |
+-----+
| T.V.            |
| Washing Machine |
| Refrigerator    |
| Cooler          |
| A.C.            |
| Iron            |
| Laptop          |
+-----+
```

## **SQL - WHERE Clause**

WHERE Clause का इस्तेमाल ज्यादातर SELECT, UPDATE और DELETE Statement के साथ किया जाता है |

WHERE Clause ये दिए गए condition के मुताबिक data को retrieve, delete या update किया जाता है |

WHERE Clause की condition अगर True होती है तो result-set return होता है |

### **Syntax for WHERE Clause in SQL**

**For Single or Multiple Columns in SQL**

```
SELECT column1, column2,..., column3

FROM table_name

WHERE condition;
```

**For All Columns in SQL**

```
SELECT *

FROM table_name
```

```
WHERE condition;
```

## Sample Table

emp_id	emp_name	emp_salary
1	Shweta Pandit	20000
3	Rupal Patil	30000
4	Shankar Mane	NULL
7	Raj Raman	25000
12	Rajesh Mujumdar	26000
15	Ramesh Mohane	NULL

## Example for WHERE Clause using SELECT Statement in SQL

Example पर सभी columns को select करके Where clause की मदद से 7 इस emp\_id से कम ids को return किया गया है |

Source Code :

```
SELECT * FROM Employee WHERE emp_id<7;
```

Output :

emp_id	emp_name	emp_salary
1	Shweta Pandit	20000
3	Rupal Patil	30000
4	Shankar Mane	NULL

```
+-----+-----+-----+
```

**Note :** WHERE Clause के condition पर <, >, <=, >=, =, !=, <>, BETWEEN, IN, LIKE इन operators का इस्तेमाल किया जा सकता है।

## **SQL - INSERT INTO Statement**

INSERT INTO Statement का इस्तेमाल Table पर नया record insert करने के लिए किया जाता है।

### **Syntax for INSERT INTO Statement in SQL**

#### **With Columns Name**

```
INSERT INTO table_name(column1,column2,...,columnN)

VALUES (value1, value2,...,valueN);
```

#### **Without Column Name**

यहाँ पर दिए गए table के columns की सभी values दी जाती है।

```
INSERT INTO table_name

VALUES (value1, value2,...,valueN);
```

### **Sample Table**

```
+-----+-----+-----+
| emp_id | emp_name      | emp_salary |
+-----+-----+-----+
|      1 | Shweta Pandit |      20000 |
|      3 | Rupal Patil   |      30000 |
|      4 | Shankar Mane  |      15000 |
|     12 | Rajesh Mujumdar |     26000 |
|     15 | Ramesh Mohane |      30000 |
+-----+-----+-----+
```



## Example for INSERT INTO Statement With Multiple Column Names in SQL

Example पर Employee इस table पर 'Shreya Bhalekar' ये emp\_name और '40000' ये emp\_salary insert की गयी है और पूरे result-set को return किया गया है |

Source Code :

```
INSERT INTO Employee(emp_name, emp_salary)

VALUES ('Shreya Bhalekar', '40000');

SELECT * FROM Employee;
```

Output :

```
+-----+-----+-----+
| emp_id | emp_name      | emp_salary |
+-----+-----+-----+
|      1 | Shweta Pandit |      20000 |
|      3 | Rupal Patil   |      30000 |
|      4 | Shankar Mane  |      15000 |
|     12 | Rajesh Mujumdar |     26000 |
|     15 | Ramesh Mohane |      30000 |
|    16 | Shreya Bhalekar |     40000 |
+-----+-----+-----+
```

## Example for INSERT INTO Statement With Single Column Names in SQL

Sample Table

```
+-----+-----+-----+
| emp_id | emp_name      | emp_salary |
+-----+-----+-----+
|      1 | Shweta Pandit |      20000 |
|      3 | Rupal Patil   |      30000 |
|      4 | Shankar Mane  |      15000 |
```

12	Rajesh Mujumdar	26000
15	Ramesh Mohane	30000
16	Shreya Bhalekar	40000
+-----+-----+-----+		

Example पर सिर्फ emp name 'Ravi Rajan' ये insert किया गया है | लेकिन emp salary ये column की value दी नहीं गयी है | Column create करते वक्त जब default value दी जाती है वो default value वहा पर दी जाती है | For Example, NULL, NOT NULL, AUTO INCREMENT

Source Code :

```
INSERT INTO Employee(emp_name)
VALUES ('Ravi Rajan');

SELECT * FROM Employee;
```

Output :

+-----+-----+-----+		
emp_id	emp_name	emp_salary
+-----+-----+-----+		
1	Shweta Pandit	20000
3	Rupal Patil	30000
4	Shankar Mane	15000
12	Rajesh Mujumdar	26000
15	Ramesh Mohane	30000
22	Shreya Bhalekar	40000
23	Ravi Rajan	NULL
+-----+-----+-----+		

## Example For INSERT INTO Statement Without Columns in SQL

Sample Table

emp_id	emp_name	emp_salary
1	Shweta Pandit	20000
3	Rupal Patil	30000
4	Shankar Mane	15000
12	Rajesh Mujumdar	26000
15	Ramesh Mohane	30000
22	Shreya Bhalekar	40000
23	Ravi Rajan	NULL

Example पर Employee इस table पर बिना columns दिए record को insert किया गया है | अगर columns दिए नहीं जाते है तो सभी columns की values दी जाती है |

INSERT INTO Employee

VALUES(20, 'Rahul Kapoor', '27000');

SELECT \* FROM Employee;

Output :

emp_id	emp_name	emp_salary
1	Shweta Pandit	20000
3	Rupal Patil	30000
4	Shankar Mane	15000

12	Rajesh Mujumdar	26000
15	Ramesh Mohane	30000
20	Rahul Kapoor	27000
22	Shreya Bhalekar	40000
23	Ravi Rajan	NULL
+-----+-----+-----+		

### Sample Tables

+-----+-----+-----+		
emp_id	emp_name	emp_salary
+-----+-----+-----+		
1	Shweta Pandit	30000
3	Rupal Patil	35000
4	Shankar Mane	15000
15	Ramesh Mohane	10000
20	Rahul Kapoor	27000
21	Shankar Mane	20000
22	Maria Shaikh	30000
23	Vikram Kank	40000
24	Vaibhav Vichare	10000
25	Shweta Pandit	25000
+-----+-----+-----+		

+----+-----+-----+-----+-----+

id	Dealer	Appliances	Cost	GST
1	Balaji	T.V.	19000	3420
2	Ravikant	Washing Machine	15000	4200
3	Electra	Refrigerator	10000	2800
4	K.K.	Cooler	18000	5040
5	Ganesh	A.C.	35000	9800
6	Rajaram	Iron	1500	420
7	Ravi	Laptop	30000	8400
8	Ravikant	Washing Machine	15000	4200

## SQL - UPDATE Statement

UPDATE Statement का इस्तेमाल मौजूदे records को update करने के लिए किया जाता है ।

UPDATE Statement द्वारा एक से ज्यादा भी records को update किया जाता है ।

### Syntax for UPDATE Statement in SQL

**Without WHERE Clause(For All Rows)**

```
UPDATE table_name

SET coloum1=value1,column2=value2,...,column3=value3;
```

**With WHERE Clause (For Specific Row)**

```
UPDATE table_name

SET coloum1=value1,column2=value2,...,column3=value3

WHERE condition;
```

### Sample Table

```
+-----+-----+-----+
```

emp_id	emp_name	emp_salary
1	Shweta Pandit	20000
3	Rupal Patil	30000
4	Shankar Mane	15000
12	Rajesh Mujumdar	26000
15	Ramesh Mohane	30000
20	Rahul Kapoor	27000

### Example for UPDATE Statement Without WHERE Clause in SQL

Example पर WHERE Clause के बिना UPDATE Statement को दिया गया है | अगर WHERE Clause की condition दी नहीं जाती है तो SET पर दिए column की values सभी row पर Update होती है |

Source Code :

```
UPDATE Employee

SET emp_salary=30000;

SELECT * FROM Employee;
```

Output :

emp_id	emp_name	emp_salary
1	Shweta Pandit	30000
3	Rupal Patil	30000
4	Shankar Mane	30000
12	Rajesh Mujumdar	30000
15	Ramesh Mohane	30000
20	Rahul Kapoor	30000

+-----+-----+-----+

## Example for UPDATE Statement With WHERE Clause in SQL

Sample Table

+-----+-----+-----+		
emp_id	emp_name	emp_salary
+-----+-----+-----+		
1	Shweta Pandit	20000
3	Rupal Patil	30000
4	Shankar Mane	15000
12	Rajesh Mujumdar	26000
15	Ramesh Mohane	30000
20	Rahul Kapoor	27000
+-----+-----+-----+		

Example पर Employee table के 1 इस emp id के emp salary इस column के value को update किया गया है ।

Source Code :

```
UPDATE Employee  
  
SET emp_salary=30000  
  
WHERE emp_id=1;  
  
SELECT * FROM Employee;
```

Output :

+-----+-----+-----+		
emp_id	emp_name	emp_salary
+-----+-----+-----+		
1	Shweta Pandit	30000

3	Rupal Patil	30000
4	Shankar Mane	15000
12	Rajesh Mujumdar	26000
15	Ramesh Mohane	30000
20	Rahul Kapoor	27000
+-----+-----+-----+		

## SQL - DELETE Statement

DELETE Statement से table से मौजूदे record या पूरे table के records को delete/remove किया जाता है ।

### Syntax for DELETE Statement in Python

#### For Specific Record

```
DELETE FROM table_name

WHERE condition;
```

#### For All Records

```
DELETE FROM table_name;
```

or

```
DELETE * FROM table_name;
```

### Sample Table

+-----+-----+-----+		
emp_id	emp_name	emp_salary
+-----+-----+-----+		
1	Shweta Pandit	30000
3	Rupal Patil	30000
4	Shankar Mane	15000
12	Raj Mehra	10000



	15	Ramesh Mohane		30000	
	20	Rahul Kapoor		27000	
+-----+-----+-----+					

## Example for DELETE Statement with WHERE Clause in SQL

Example में Employee इस table में जहा पर emp\_name 'Raj Mehra' है वो record table से delete किया गया है |

Source Code :

```
DELETE FROM Employee

WHERE emp_name='Raj Mehra';

SELECT * FROM Employee;
```

Output :

+-----+-----+-----+					
	emp_id		emp_name		emp_salary
+-----+-----+-----+					
	1		Shweta Pandit		30000
	3		Rupal Patil		30000
	4		Shankar Mane		15000
	15		Ramesh Mohane		30000
	20		Rahul Kapoor		27000
+-----+-----+-----+					

## DELETE All Records From Table 'Employee'

Source Code :

```
DELETE FROM Employee;
```

```
SELECT * FROM Employee;
```

Output :

```
+-----+-----+-----+  
| emp_id | emp_name | emp_salary |  
+-----+-----+-----+  
  
+-----+-----+-----+
```