



# **SNS COLLEGE OF ENGINEERING**

Kurumbapalayam (Po), Coimbatore – 641 107

**An Autonomous Institution**

Accredited by NBA – AICTE and Accredited by NAAC – UGC with ‘A’ Grade  
Approved by AICTE, New Delhi & Affiliated to Anna University, Chennai

**DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING(IoT and  
Cybersecurity Including BCT)**

**COURSE NAME : 19SB504 DATABASE MANAGEMENT SYSTEMS**

**III YEAR / V SEMESTER**

**Unit II- SQL**

**Topic :Set Operations, Sub-queries, Correlated Sub-Queries**

# Set Operation

The SQL Set operation is used to combine the two or more SQL SELECT statements.

## Types of Set Operation

1. Union
2. UnionAll
3. Intersect
4. Minus





## 1. Union

- The SQL Union operation is used to combine the result of two or more SQL SELECT queries.
- In the union operation, all the number of datatype and columns must be same in both the tables on which UNION operation is being applied.
- The union operation eliminates the duplicate rows from its result set.

### Syntax

```
1.SELECT column_name FROM table1
```

```
UNION
```

```
SELECT column_name FROM table2;
```

### Example:

```
SELECT * FROM First
```

```
UNION
```

```
SELECT * FROM Second;
```



## The First table

ID	NAME
1	Jack
2	Harry
3	Jackson

## The Second table

ID	NAME
3	Jackson
4	Stephan
5	David



The resultset table will look like:

ID	NAME
1	Jack
2	Harry
3	Jackson
4	Stephan
5	David



## 2. Union All

Union All operation is equal to the Union operation. It returns the set without removing duplication and sorting the data.

### **Syntax:**

```
SELECT column_name FROM table1
```

```
UNION ALL
```

```
SELECT column_name FROM table2;
```

**Example:** Using the above First and Second table.

### **Union All query will be like:**

```
SELECT * FROM First
```

```
UNION ALL
```

```
SELECT * FROM Second;
```



The result set table will look like:

ID	NAME
1	Jack
2	Harry
3	Jackson
3	Jackson
4	Stephan
5	David



### 3. Intersect

- It is used to combine two SELECT statements. The Intersect operation returns the **common rows from both the SELECT statements**.
- In the Intersect operation, the number of datatype and columns must be the same.
- It has no duplicates and it arranges the data in ascending order by default.

#### Syntax

```
SELECT column_name FROM table1
```

```
INTERSECT
```

```
SELECT column_name FROM table2;
```

#### Example:

**Using the above First and Second table.**

Intersect query will be:

```
SELECT * FROM First
```

```
INTERSECT
```

```
SELECT * FROM Second;
```





The resultset table will look like:

ID	NAME
3	Jackson



## 4. Minus

- It combines the result of **two SELECT statements**. Minus operator is used to display the **rows which are present in the first query but absent in the second query**.
- It has no duplicates and data arranged in ascending order by default.

### Syntax:

```
1.SELECT column_name FROM table1
```

```
MINUS
```

```
SELECT column_name FROM table2;
```

### Example

```
SELECT * FROM First
```

```
MINUS
```

```
SELECT * FROM Second;
```



The result set table will look like:

ID	NAME
1	Jack
2	Harry



## Sub-Queries



A Subquery is a query within another SQL query and embedded within the WHERE clause.

### Important Rule:

- ✓ A subquery can be placed in a number of SQL clauses like WHERE clause, FROM clause, HAVING clause.
- ✓ You can use **Subquery with SELECT, UPDATE, INSERT, DELETE** statements along with the operators like =, <, >, >=, <=, IN, BETWEEN, etc.
- ✓ A subquery is a query within another query. The **outer query is known as the main query**, and the inner query is known as a subquery.
- ✓ Subqueries are on the **right side of the comparison operator**.
- ✓ A subquery is enclosed in parentheses.



## 1. Subqueries with the Select Statement

SQL subqueries are most frequently used with the Select statement.

### Syntax

```
SELECT column_name  
FROM table_name  
WHERE column_name expression operator  
(SELECT column_name from table_name WHERE ...);
```



## Example

Consider the EMPLOYEE table have the following records:

ID	NAME	AGE	ADDRESS	SALARY
1	John	20	US	2000.00
2	Stephan	26	Dubai	1500.00
3	David	27	Bangkok	2000.00
4	Alina	29	UK	6500.00
5	Kathrin	34	Bangalore	8500.00
6	Harry	42	China	4500.00
7	Jackson	25	Mizoram	10000.00



The subquery with a SELECT statement will be:

```
SELECT * FROM EMPLOYEE WHERE ID IN (SELECT ID  
FROM EMPLOYEE WHERE SALARY > 4500);
```

This would produce the following result:

ID	NAME	AGE	ADDRESS	SALARY
4	Alina	29	UK	6500.00
5	Kathrin	34	Bangalore	8500.00
7	Jackson	25	Mizoram	10000.00



## 2. Subqueries with the INSERT Statement

- SQL subquery can also be used with the Insert statement. In the insert statement, data returned from the subquery is used to insert into another table.
- In the subquery, the selected data can be modified with any of the character, date functions.

### Syntax:

```
1.INSERT INTO table_name (column1, column2, column3....)  
SELECT * FROM table_name WHERE VALUE OPERATOR
```

### Example

Consider a table EMPLOYEE\_BKP with similar as EMPLOYEE.

Now use the following syntax to copy the complete EMPLOYEE table into the EMPLOYEE\_BKP table.

```
1.INSERT INTO EMPLOYEE_BKP SELECT * FROM EMPLOYEE  
WHERE ID IN (SELECT ID FROM EMPLOYEE);
```





### 3. Subqueries with the UPDATE Statement

The subquery of SQL can be used in conjunction with the Update statement. When a subquery is used with the Update statement, then either single or multiple columns in a table can be updated.

#### Syntax

1.UPDATE table SET column\_name = new\_value WHERE VALUE OPERATOR (SELECT COLUMN\_NAME FROM TABLE\_NAME WHERE condition);

#### Example

Let's assume we have an EMPLOYEE\_BKP table available which is backup of EMPLOYEE table. The given example updates the SALARY by .25 times in the EMPLOYEE table for all employee whose AGE is greater than or equal to 29.

1.UPDATE EMPLOYEE SET SALARY = SALARY \* 0.25 WHERE AGE IN (SELECT AGE FROM CUSTOMERS\_BKP WHERE AGE >= 29);



This would impact three rows, and finally, the EMPLOYEE table would have the following records.

ID	NAME	AGE	ADDRESS	SALARY
1	John	20	US	2000.00
2	Stephan	26	Dubai	1500.00
3	David	27	Bangkok	2000.00
4	Alina	29	UK	1625.00
5	Kathrin	34	Bangalore	2125.00
6	Harry	42	China	1125.00
7	Jackson	25	Mizoram	10000.00



## 4. Subqueries with the DELETE Statement

The subquery of SQL can be used in conjunction with the Delete statement just like any other statements mentioned above.

### Syntax

- 1.DELETE FROM TABLE\_NAME
- 2.WHERE VALUE OPERATOR
3. (SELECT COLUMN\_NAME
4. FROM TABLE\_NAME
5. WHERE condition);

### Example

```
DELETE FROM EMPLOYEE
WHERE AGE IN (SELECT AGE FROM EMPLOYEE_BKP
WHERE AGE >= 29 );
```



This would impact three rows, and finally, the EMPLOYEE table would have the following records.



ID	NAME	AGE	ADDRESS	SALARY
1	John	20	US	2000.00
2	Stephan	26	Dubai	1500.00
3	David	27	Bangkok	2000.00
7	Jackson	25	Mizoram	10000.00



# Correlated Sub Queries



Correlated Sub Queries are also similar to sub queries but here the **outer query is executed first and inner query is executed for each records of outer query.**

That is inner query is executed as many times as the outer query results.

## Example

```
SELECT *  
FROM STUDENT  
WHERE STD_ID IN (SELECT STD_ID FROM MARKS m WHERE s.STD_ID =  
m.STD_ID);
```

Here we can see that outer query column and inner query column are joined to get the result. This query fetches all the records from STUDENT table and joins with the STD\_ID in MARKS table. It returns the records only if there is a matching STD\_ID in MARKS.



Any Query????

Thank you.....