



DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING-IOT Including CS&BCT
UNIT-II

SET Operations in SQL

SQL supports few Set operations which can be performed on the table data. These are used to get meaningful results from data stored in the table, under different special conditions.

In this tutorial, we will cover 4 different types of SET operations, along with example:

- ✓ UNION
- ✓ UNION ALL
- ✓ INTERSECT
- ✓ MINUS

UNION Operation

UNION is used to combine the results of two or more SELECT statements. However it will eliminate duplicate rows from its resultset. In case of union, number of columns and datatype must be same in both the tables, on which UNION operation is being applied.

Example of UNION

The First table,

ID	Name
1	abhi
2	adam

The Second table,

ID	Name
2	adam
3	Chester

Union SQL query will be,

```
SELECT * FROM First
```

```
UNION
```

```
SELECT * FROM Second;
```

The resultset table will look like,

ID	NAME
1	abhi
2	adam
3	Chester



DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING-IOT Including CS&BCT
UNIT-II

UNION ALL

This operation is similar to Union. But it also shows the duplicate rows.

Example of Union All

The First table,

ID	NAME
1	abhi
2	adam

The Second table,

ID	NAME
2	adam
3	Chester

Union All query will be like,

```
SELECT * FROM First
```

```
UNION ALL
```

```
SELECT * FROM Second;
```

The resultset table will look like,

ID	NAME
1	abhi
2	adam
2	adam
3	Chester

INTERSECT

Intersect operation is used to combine two SELECT statements, but it only returns the records which are common from both SELECT statements. In case of Intersect the number of columns and datatype must be same.

Example of Intersect

The First table,

ID	NAME
----	------



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 Including CS&BCT
UNIT-II

1 abhi

2 adam

The Second table,

ID NAME

2 adam

3 Chester

Intersect query will be,

SELECT * FROM First

INTERSECT

SELECT * FROM Second;

The resultset table will look like

ID NAME

2 adam

MINUS

The Minus operation combines results of two SELECT statements and return only those in the final result, which belongs to the first set of the result.

Example of Minus

The First table,

ID NAME

1 abhi

2 adam

The Second table,

ID NAME

2 adam

3 Chester

Minus query will be,

SELECT * FROM First

MINUS



**DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING-IOT Including CS&BCT
UNIT-II**

SELECT * FROM Second;

The resultset table will look like,

ID	NAME
1	Abhi

SUBQUERY

In SQL a Subquery can be simply defined as a query within another query. In other words we can say that a Subquery is a query that is embedded in WHERE clause of another SQL query. Important rules for Subqueries:

You can place the Subquery in a number of SQL clauses: WHERE clause, HAVING clause, FROM clause. Subqueries can be used with SELECT, UPDATE, INSERT, DELETE statements along with expression operator. It could be equality operator or comparison operator such as =, >, =, <= and Like operator.

A subquery is a query within another query. The outer query is called as main query and inner query is called as subquery.

The subquery generally executes first when the subquery doesn't have any co-relation with the main query, when there is a co-relation the parser takes the decision on the fly on which query to execute on precedence and uses the output of the subquery accordingly.

- ✓ Subquery must be enclosed in parentheses.
- ✓ Subqueries are on the right side of the comparison operator.

ORDER BY command cannot be used in a Subquery. GROUPBY command can be used to perform same function as ORDER BY command.

Use single-row operators with singlerow Subqueries. Use multiple-row operators with multiple-row Subqueries.

Syntax: There is not any general syntax for Subqueries. However, Subqueries are seen to be used most frequently with SELECT statement as shown below:

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

Sample Table:

DATABASE

NAME	ROLL_NO	LOCATION	PHONE_NUMBER
Ram	101	Chennai	9988775566
Raj	102	Coimbatore	8877665544



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 Including CS&BCT
UNIT-II

Sasi	103	Madurai	7766553344
Ravi	104	Salem	8989898989
Sumathi	105	Kanchipuram	8989856868

STUDENT

NAME	ROLL_NO	SECTION
Ravi	104	A
Sumathi	105	B
Raj	102	A

Sample Queries

To display NAME, LOCATION, PHONE_NUMBER of the students from DATABASE table whose section is A

Select NAME, LOCATION, PHONE_NUMBER from DATABASE WHERE ROLL_NO IN(SELECT ROLL_NO from STUDENT where SECTION='A');

Explanation : First subquery executes “ SELECT ROLL_NO from STUDENT where SECTION='A' ” returns ROLL_NO from STUDENT table whose SECTION is ‘A’.Then outer-query executes it and return the NAME, LOCATION, PHONE_NUMBER from the DATABASE table of the student whose ROLL_NO is returned from inner subquery.

Output:

NAME	ROLL_NO	LOCATION	PHONE_NUMBER
Ravi	104	Salem	8989898989
Raj	102	Coimbatore	8877665544

Insert Query Example:

Table1: Student1

NAME	ROLL_NO	LOCATION	PHONE_NUMBER
Ram	101	chennai	9988773344
Raju	102	coimbatore	9090909090
Ravi	103	salem	8989898989

Table2: Student2

NAME	ROLL_NO	LOCATION	PHONE_NUMBER
------	---------	----------	--------------



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 Including CS&BCT
UNIT-II

Raj	111	chennai	8787878787
Sai	112	mumbai	6565656565
Sri	113	coimbatore	7878787878

To insert Student2 into Student1 table:

```
INSERT INTO Student1 SELECT * FROM Student2;
```

Output:

NAME	ROLL_NO	LOCATION	PHONE_NUMBER
Ram	101	chennai	9988773344
Raju	102	coimbatore	9090909090
Ravi	103	salem	8989898989
Raj	111	chennai	8787878787
Sai	112	mumbai	6565656565
Sri	113	coimbatore	7878787878

To delete students from Student2 table whose rollno is same as that in Student1 table and having location as chennai

```
DELETE FROM Student2
```

```
WHERE ROLL_NO IN ( SELECT ROLL_NO FROM Student1 WHERE LOCATION = 'chennai');
```

Output:

1 row delete successfully.

Display Student2 table:

NAME	ROLL_NO	LOCATION	PHONE_NUMBER
Sai	112	mumbai	6565656565
Sri	113	coimbatore	7878787878

To update name of the students to sns in Student2 table whose location is same as Raju,Ravi in Student1 table

```
UPDATE Student2 SET NAME='sns' WHERE LOCATION IN ( SELECT LOCATION FROM Student1 WHERE NAME IN ('Raju','Ravi'));
```

Output:



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 Including CS&BCT
UNIT-II

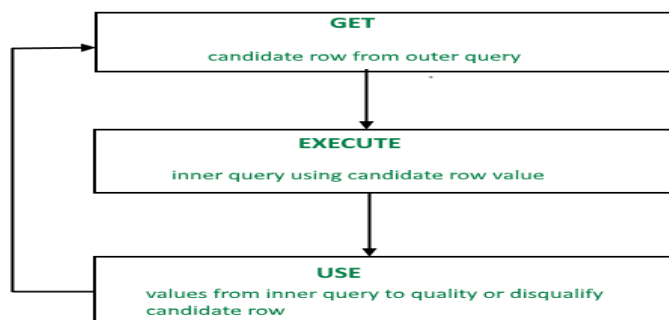
1 row updated successfully.

Display Student2 table:

NAMEROLL_NO	LOCATION	PHONE_NUMBER
Sai 112	mumbai	6565656565
sns 113	coimbatore	7878787878

CORRELATED SUBQUERIES

Correlated subqueries are used for row-by-row processing. Each subquery is executed once for every row of the outer query.



A correlated subquery is evaluated once for each row processed by the parent statement. The parent statement can be a SELECT, UPDATE, or DELETE statement.

```
SELECT column1, column2, ... FROM table1 outer WHERE column1 operator (SELECT column1, column2 FROM table2 WHERE expr1 = outer.expr2);
```

A correlated subquery is one way of reading every row in a table and comparing values in each row against related data. It is used whenever a subquery must return a different result or set of results for each candidate row considered by the main query. In other words, you can use a correlated subquery to answer a multipart question whose answer depends on the value in each row processed by the parent statement.

Nested Subqueries Versus Correlated Subqueries :

With a normal nested subquery, the inner SELECT query runs first and executes once, returning values to be used by the main query. A correlated subquery, however, executes once



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 Including CS&BCT
UNIT-II**

for each candidate row considered by the outer query. In other words, the inner query is driven by the outer query.

NOTE : You can also use the ANY and ALL operator in a correlated subquery.

EXAMPLE of Correlated Subqueries : Find all the employees who earn more than the average salary in their department.

```
SELECT last_name, salary, department_id FROM employees outer WHERE salary >
(SELECT AVG(salary) FROM employee WHERE department_id =outer.department_id);
```

Other use of correlation are in UPDATE and DELETE

CORRELATED UPDATE :

```
UPDATE table1 alias1
```

```
SET column = (SELECT expression FROM table2 alias2 WHERE alias1.column =
alias2.column);
```

Use a correlated subquery to update rows in one table based on rows from another table.

CORRELATED DELETE :

```
DELETE FROM table1 alias1 WHERE column1 operator (SELECT expression FROM table2
alias2 WHERE alias1.column = alias2.column);
```

Use a correlated subquery to delete rows in one table based on the rows from another table.