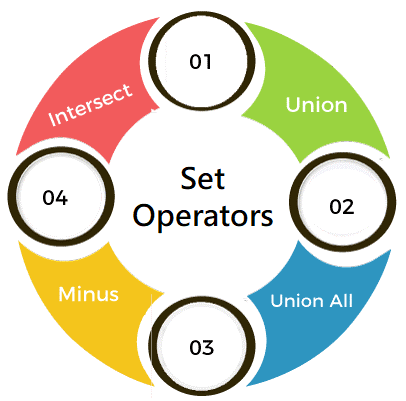
SET Operators in SQL

SET operators are special type of operators which are used to *combine the result of two queries.*

Operators covered under SET operators are:

1. **UNION**
2. **UNION ALL**
3. **INTERSECT**
4. **MINUS**



There are certain rules which must be followed to perform operations using SET operators in SQL. Rules are as follows:

1. **The number and order of columns must be the same.**
2. **Data types must be compatible.**

Let us see each of the SET operators in more detail with the help of examples.

Backward Skip 10sPlay VideoForward Skip 10s

All the examples will be written using the MySQL database.

Consider we have the following tables with the given data.

**Table 1: t\_employees**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **ID** | **Name** | **Department** | **Salary** | **Year\_of\_Experience** |
| 1 | Aakash Singh | Development | 72000 | 2 |
| 2 | Abhishek Pawar | Production | 45000 | 1 |
| 3 | Pranav Deshmukh | HR | 59900 | 3 |
| 4 | Shubham Mahale | Accounts | 57000 | 2 |
| 5 | Sunil Kulkarni | Development | 87000 | 3 |
| 6 | Bhushan Wagh | R&D | 75000 | 2 |
| 7 | Paras Jaiswal | Marketing | 32000 | 1 |

**Table 2: t2\_employees**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **ID** | **Name** | **Department** | **Salary** | **Year\_of\_Experience** |
| 1 | Prashant Wagh | R&D | 49000 | 1 |
| 2 | Abhishek Pawar | Production | 45000 | 1 |
| 3 | Gautam Jain | Development | 56000 | 4 |
| 4 | Shubham Mahale | Accounts | 57000 | 2 |
| 5 | Rahul Thakur | Production | 76000 | 4 |
| 6 | Bhushan Wagh | R&D | 75000 | 2 |
| 7 | Anand Singh | Marketing | 28000 | 1 |

**Table 3: t\_students**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **ID** | **Name** | **Hometown** | **Percentage** | **Favourite\_Subject** |
| 1 | Soniya Jain | Udaipur | 89 | Physics |
| 2 | Harshada Sharma | Kanpur | 92 | Chemistry |
| 3 | Anuja Rajput | Jaipur | 78 | History |
| 4 | Pranali Singh | Nashik | 88 | Geography |
| 5 | Renuka Deshmukh | Panipat | 90 | Biology |
| 6 | Swati Kumari | Faridabad | 93 | English |
| 7 | Prachi Jaiswal | Gurugram | 96 | Hindi |

**Table 4: t2\_students**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **ID** | **Name** | **Hometown** | **Percentage** | **Favourite\_Subject** |
| 1 | Soniya Jain | Udaipur | 89 | Physics |
| 2 | Ishwari Dixit | Delhi | 86 | Hindi |
| 3 | Anuja Rajput | Jaipur | 78 | History |
| 4 | Pakhi Arora | Surat | 70 | Sanskrit |
| 5 | Renuka Deshmukh | Panipat | 90 | Biology |
| 6 | Jayshree Patel | Pune | 91 | Maths |
| 7 | Prachi Jaiswal | Gurugram | 96 | Hindi |

1. UNION:

* UNION will be used to combine the result of two select statements.
* Duplicate rows will be eliminated from the results obtained after performing the UNION operation.

**Example 1:**

Write a query to perform union between the table t\_employees and the table t2\_employees.

**Query:**

1. mysql> **SELECT** \***FROM** t\_employees **UNION** **SELECT** \***FROM** t2\_employees;

Here, in a single query, we have written two SELECT queries. The first SELECT query will fetch the records from the t\_employees table and perform a UNION operation with the records fetched by the second SELECT query from the t2\_employees table.

You will get the following output:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **ID** | **Name** | **Department** | **Salary** | **Year\_of\_Experience** |
| 1 | Aakash Singh | Development | 72000 | 2 |
| 2 | Abhishek Pawar | Production | 45000 | 1 |
| 3 | Pranav Deshmukh | HR | 59900 | 3 |
| 4 | Shubham Mahale | Accounts | 57000 | 2 |
| 5 | Sunil Kulkarni | Development | 87000 | 3 |
| 6 | Bhushan Wagh | R&D | 75000 | 2 |
| 7 | Paras Jaiswal | Marketing | 32000 | 1 |
| 1 | Prashant Wagh | R&D | 49000 | 1 |
| 3 | Gautam Jain | Development | 56000 | 4 |
| 5 | Rahul Thakur | Production | 76000 | 4 |
| 7 | Anand Singh | Marketing | 28000 | 1 |

Since we have performed union operation between both the tables, so only the records from the first and second table are displayed except for the duplicate records.

**Example 2:**

Write a query to perform union between the table t\_students and the table t2\_students.

**Query:**

1. mysql> **SELECT** \***FROM** t\_students **UNION** **SELECT** \***FROM** t2\_students;

Here, in a single query, we have written two SELECT queries. The first SELECT query will fetch the records from the t\_students table and perform a UNION operation with the records fetched by the second SELECT query from the t2\_students table.

You will get the following output:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **ID** | **Name** | **Department** | **Salary** | **Year\_of\_Experience** |
| 1 | Soniya Jain | Udaipur | 89 | Physics |
| 2 | Harshada Sharma | Kanpur | 92 | Chemistry |
| 3 | Anuja Rajput | Jaipur | 78 | History |
| 4 | Pranali Singh | Nashik | 88 | Geography |
| 5 | Renuka Deshmukh | Panipat | 90 | Biology |
| 6 | Swati Kumari | Faridabad | 93 | English |
| 7 | Prachi Jaiswal | Gurugram | 96 | Hindi |
| 2 | Ishwari Dixit | Delhi | 86 | Hindi |
| 4 | Pakhi Arora | Surat | 70 | Sanskrit |
| 6 | Jayshree Patel | Pune | 91 | Maths |

Since we have performed union operation between both the tables, so only the records from the first and second table are displayed except for the duplicate records.

2. UNION ALL

* This operator combines all the records from both the queries.
* Duplicate rows will be not be eliminated from the results obtained after performing the UNION ALL operation.

**Example 1:**

Write a query to perform union all operation between the table t\_employees and the table t2\_employees.

**Query:**

1. mysql> **SELECT** \***FROM** t\_employees **UNION** ALL **SELECT** \***FROM** t2\_employees;

Here, in a single query, we have written two SELECT queries. The first SELECT query will fetch the records from the t\_employees table and perform UNION ALL operation with the records fetched by the second SELECT query from the t2\_employees table.

You will get the following output:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **ID** | **Name** | **Department** | **Salary** | **Year\_of\_Experience** |
| 1 | Aakash Singh | Development | 72000 | 2 |
| 2 | Abhishek Pawar | Production | 45000 | 1 |
| 3 | Pranav Deshmukh | HR | 59900 | 3 |
| 4 | Shubham Mahale | Accounts | 57000 | 2 |
| 5 | Sunil Kulkarni | Development | 87000 | 3 |
| 6 | Bhushan Wagh | R&D | 75000 | 2 |
| 7 | Paras Jaiswal | Marketing | 32000 | 1 |
| 1 | Prashant Wagh | R&D | 49000 | 1 |
| 2 | Abhishek Pawar | Production | 45000 | 1 |
| 3 | Gautam Jain | Development | 56000 | 4 |
| 4 | Shubham Mahale | Accounts | 57000 | 2 |
| 5 | Rahul Thakur | Production | 76000 | 4 |
| 6 | Bhushan Wagh | R&D | 75000 | 2 |
| 7 | Anand Singh | Marketing | 28000 | 1 |

Since we have performed union all operation between both the tables, so all the records from the first and second table are displayed, including the duplicate records.

**Example 2:**

Write a query to perform union all operation between the table t\_students and the table t2\_students.

**Query:**

1. mysql> **SELECT** \***FROM** t\_students **UNION** ALL **SELECT** \***FROM** t2\_students;

Here, in a single query, we have written two SELECT queries. The first SELECT query will fetch the records from the t\_students table and perform UNION ALL operation with the records fetched by the second SELECT query from the t2\_students table.

You will get the following output:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **ID** | **Name** | **Hometown** | **Percentage** | **Favourite\_Subject** |
| 1 | Soniya Jain | Udaipur | 89 | Physics |
| 2 | Harshada Sharma | Kanpur | 92 | Chemistry |
| 3 | Anuja Rajput | Jaipur | 78 | History |
| 4 | Pranali Singh | Nashik | 88 | Geography |
| 5 | Renuka Deshmukh | Panipat | 90 | Biology |
| 6 | Swati Kumari | Faridabad | 93 | English |
| 7 | Prachi Jaiswal | Gurugram | 96 | Hindi |
| 1 | Soniya Jain | Udaipur | 89 | Physics |
| 2 | Ishwari Dixit | Delhi | 86 | Hindi |
| 3 | Anuja Rajput | Jaipur | 78 | History |
| 4 | Pakhi Arora | Surat | 70 | Sanskrit |
| 5 | Renuka Deshmukh | Panipat | 90 | Biology |
| 6 | Jayshree Patel | Pune | 91 | Maths |
| 7 | Prachi Jaiswal | Gurugram | 96 | Hindi |

Since we have performed union all operation between both the tables, so all the records from the first and second table are displayed, including the duplicate records.

3. INTERSECT:

* It is used to combine two SELECT statements, but it only returns the records which are common from both SELECT statements.

**Example 1:**

Write a query to perform intersect operation between the table t\_employees and the table t2\_employees.

**Query:**

1. mysql> **SELECT** \***FROM** t\_employees **INTERSECT** **SELECT** \***FROM** t2\_employees;

Here, in a single query, we have written two SELECT queries. The first SELECT query will fetch the records from the t\_employees table and perform INTERSECT operation with the records fetched by the second SELECT query from the t2\_employees table.

You will get the following output:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **ID** | **Name** | **Hometown** | **Percentage** | **Favourite\_Subject** |
| 2 | Abhishek Pawar | Production | 45000 | 1 |
| 4 | Shubham Mahale | Accounts | 57000 | 2 |
| 6 | Bhushan Wagh | R&D | 75000 | 2 |

Since we have performed intersect operation between both the tables, so only the common records from both the tables are displayed.

**Example 2:**

Write a query to perform intersect operation between the table t\_students and the table t2\_students.

**Query:**

1. mysql> **SELECT** \***FROM** t\_students **INTERSECT** **SELECT** \***FROM** t2\_students;

Here, in a single query, we have written two SELECT queries. The first SELECT query will fetch the records from the t\_students table and perform a UNION operation with the records fetched by the second SELECT query from the t2\_students table.

You will get the following output:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **ID** | **Name** | **Hometown** | **Percentage** | **Favourite\_Subject** |
| 1 | Soniya Jain | Udaipur | 89 | Physics |
| 3 | Anuja Rajput | Jaipur | 78 | History |
| 5 | Renuka Deshmukh | Panipat | 90 | Biology |
| 7 | Prachi Jaiswal | Gurugram | 96 | Hindi |

Since we have performed intersect operation between both the tables, so only the common records from both the tables are displayed.

1. **MINUS**

* It displays the rows which are present in the first query but absent in the second query with no duplicates.

**Example 1:**

Write a query to perform a minus operation between the table t\_employees and the table t2\_employees.

**Query:**

1. mysql> **SELECT** \***FROM** t\_employees MINUS **SELECT** \***FROM** t2\_employees;

Here, in a single query, we have written two SELECT queries. The first SELECT query will fetch the records from the t\_employees table and perform MINUS operation with the records fetched by the second SELECT query from the t2\_employees table.

You will get the following output:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **ID** | **Name** | **Department** | **Salary** | **Year\_of\_Experience** |
| 1 | Aakash Singh | Development | 72000 | 2 |
| 3 | Pranav Deshmukh | HR | 59900 | 3 |
| 5 | Sunil Kulkarni | Development | 87000 | 3 |
| 7 | Paras Jaiswal | Marketing | 32000 | 1 |

Since we have performed Minus operation between both the tables, so only the unmatched records from both the tables are displayed.

**Example 2:**

Write a query to perform a minus operation between the table t\_students and the table t2\_students.

**Query:**

1. mysql> **SELECT** \***FROM** t\_students MINUS **SELECT** \***FROM** t2\_students;

Here, in a single query, we have written two SELECT queries. The first SELECT query will fetch the records from the t\_employees table and perform a UNION operation with the records fetched by the second SELECT query from the t2\_employees table.

You will get the following output:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **ID** | **Name** | **Hometown** | **Percentage** | **Favourite\_Subject** |
| 2 | Harshada Sharma | Kanpur | 92 | Chemistry |
| 4 | Pranali Singh | Nashik | 88 | Geography |
| 6 | Swati Kumari | Faridabad | 93 | English |

Since we have performed a minus operation between both the tables, so only the Unmatched records from both the tables are displayed.