eBOX
BY AMPHISOFT

Set Operators

In this session, you will learn:

- How to use UNION Operator
- How to use UNION ALL Operator
- How to use INTERSECT Operator
- How to use MINUS Operator

UNION Operator

- allows you to combine two or more result sets of queries into a single result set.

Syntax

```
SELECT column_name(s) FROM table1
UNION
SELECT column_name(s) FROM table2;
```

Basic Rules

✓ First, the number and the orders of columns that appear in all SELECT statements must be the same.

✓ Second, the data types of columns must be the same or convertible.

UNION Operator- Example

First

ID	Name
1	Abhi
2	Anu

Second

ID	Name
2	Anu
3	Meena

Example

```
SELECT * from First
UNION
SELECT * from Second
```

→

ID	Name
1	Abhi
2	Anu
3	Meena

UNION ALL Operator

- The UNION operator selects only distinct values by default. To allow duplicate values, use UNION ALL.

Syntax

```
SELECT column_name(s) FROM table1
UNION ALL
SELECT column_name(s) FROM table2;
```

UNION ALL Operator- Example

First

ID	Name
1	Abhi
2	Anu

Second

ID	Name
2	Anu
3	Meena

Example

```
SELECT * from First
UNION ALL
SELECT * from Second
```

→

ID	Name
1	Abhi
2	Anu
2	Anu
3	Meena

INTERSECT Operator

- The INTERSECT operator is a set operator that returns only distinct rows of two queries or more queries.

Syntax

```
SELECT column_name(s) FROM table1
INTERSECT
SELECT column_name(s) FROM table2;
```

INTERSECT Operator- Example

First

ID	Name
1	Abhi
2	Anu

Second

ID	Name
2	Anu
3	Meena

Example

```
SELECT * from First
INTERSECT
SELECT * from Second
```

→

ID	Name
2	Anu

MINUS Operator

- MINUS compares results of two queries and returns distinct rows from the first query that aren’t output by the second query.

Syntax

```
SELECT column_name(s) FROM table1
MINUS
SELECT column_name(s) FROM table2;
```

MINUS Operator- Example

First

ID	Name
1	Abhi
2	Anu

Second

ID	Name
2	Anu
3	Meena

Example

```
SELECT * from First
MINUS
SELECT * from Second
```

→

ID	Name
1	Abhi

THANKS

eBOX
BY AMPHISOFT