Government College of Engineering, Jalgaon
(An Autonomous Institute of Government of Maharashtra)

Name : Janhavi Ramesh Wankhede	**Semester :** V	**PRN :** 2141027
Class : T. Y. B.Tech Computer	**Academic Year :** 2023-24	**Subject :** CO307U

Course Teacher : Mr. Vinit Kakde

Date of Performance : **Date of Completion :**

Practical no. 4

Aim : Design SQL queries that involve set operation and set comparison in the database.

Requirements:

1. Computer System with Open Source Operating System.
2. Mysql

Theory:

Set Operations:

The SQL operations union, intersect, and except operate on relations and correspond to the mathematical set-theory operations ∪, ∩, and −.

The Union Operation

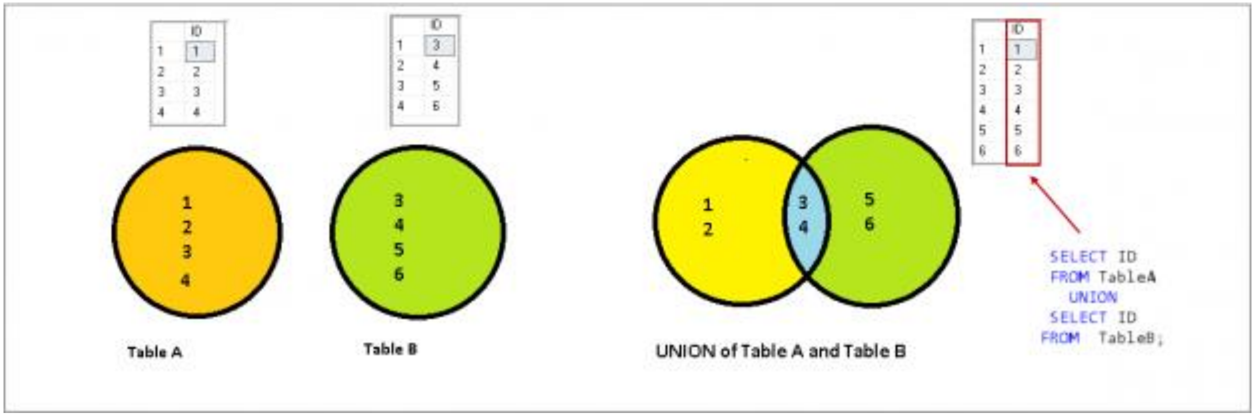
To find the set of all employees who belong to either “IT” dept. or “Comp” dept, or both,
(select * from employee where dept_name=‟IT‟) union(select *from employee where dept_name = ‟Comp‟);

The union operation automatically eliminates duplicates, unlike the select clause.

On the similar concepts, intersect and minus/except operations works by retaining their individual properties.

The UNION operator is used to combine the result-set of two or more SELECT statements.

- Every SELECT statement within UNION must have the same number of columns
- The columns must also have similar data types
- The columns in every SELECT statement must also be in the same order.



	ID
1	1
2	2
3	3
4	4

	ID
1	3
2	4
3	5
4	6

	ID
1	1
2	2
3	3
4	4
5	5
6	6

```
SELECT ID
FROM TableA
  UNION
SELECT ID
FROM  TableB;
```

UNION Syntax:

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

UNION ALL Syntax

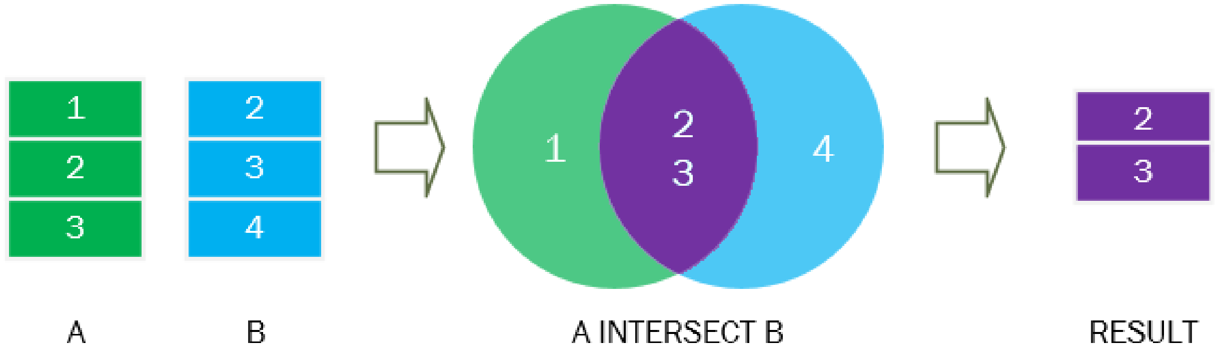
The UNION operator selects only distinct values by default. To allow duplicate values, use UNION ALL:

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

The Intersect Operation

The INTERSECT operator in SQL is used to retrieve the records that are identical/common between the result sets of two or more tables.

The INTERSECT clause in SQL is used to combine two SELECT statements but the dataset returned by the INTERSECT statement will be the intersection of the data sets of the two SELECT statements. In simple words, the INTERSECT statement will return only those rows which will be common to both of the SELECT statements.



INTERSECT Syntax

SELECT *column1, column2,..., columnN*
FROM *table1, table2,..., tableN*
INTERSECT
SELECT *column1, column2,..., columnN*
FROM *table1, table2,..., tableN*

Conclusion:
Design SQL queries that involve set operation and set comparison in the database.

Questions:

1) What is the purpose of the UNION operator in MySQL?
The **UNION** operator in MySQL is used to combine the result sets of two or more **SELECT** statements. It returns a combined set of unique rows from all the SELECT statements

2) How does the UNION ALL operator differ from the UNION operator in MySQL?
While **UNION** removes duplicate rows from the combined result set, **UNION ALL** includes all rows, including duplicates. **UNION ALL** is more efficient than **UNION**, but it doesn't eliminate duplicate rows.

3) Can you simulate the INTERSECT operation in MySQL?

```
SELECT column1, column2, ...
FROM table1
INNER JOIN table2 ON table1.common_column = table2.common_column;
```

4) How do you find records in one table that are not present in another table in MySQL?

You can use the **LEFT JOIN** and **IS NULL** pattern or the **NOT EXISTS** clause. For example:

```
-- Using LEFT JOIN and IS NULL
SELECT table1.*
FROM table1
LEFT JOIN table2 ON table1.common_column = table2.common_column
WHERE table2.common_column IS NULL;

-- Using NOT EXISTS
SELECT column1, column2, ...
FROM table1
WHERE NOT EXISTS (SELECT 1 FROM table2 WHERE table1.common_column =
table2.common_column);
```

5) How can you combine the results of two tables and include all rows, even if they don't have matches in the other table?

MySQL does not have a direct **FULL OUTER JOIN** or **EXCEPT** operator. However, you can achieve similar results using a combination of **LEFT JOIN**, **RIGHT JOIN**, and **UNION**. For example:

```
SELECT column1, column2, ...
FROM table1
LEFT JOIN table2 ON table1.common_column = table2.common_column

UNION

SELECT column1, column2, ...
FROM table1
RIGHT JOIN table2 ON table1.common_column = table2.common_column
WHERE table1.common_column IS NULL;
```

Course Teacher

Mr. Vinit Kakde