Lab Manual for Database Management System

Lab No. 5

Set Operations in SQL

LAB 5: SET OPERATIONS IN SQL

1. INTRODUCTION:

SQL set operations provide a robust framework for data manipulation, enabling efficient data retrieval, comparison, and integration. These operations play a crucial role in addressing complex analytical tasks and enhancing the capabilities of relational database management systems.

2. OBJECTIVE:

After completing this lab the student should be able to:

a. *Master SQL set operations (UNION, INTERSECT, EXCEPT) for versatile data manipulation.*
b. *Efficiently integrate data from multiple sources, handling duplicates with set operations.*

3. THEORY

SQL set operations are a fundamental aspect of relational databases, offering powerful mechanisms to manipulate and analyze data. These operations allow you to combine, compare, and extract information from different sets of data within the database.

The four most common set operations are:

i. **UNION**

ii. **UNION ALL**

iii. **INTERSECT**

iv. **EXCEPT (or MINUS)**

i. UNION:

One of the most common set operations is the UNION operation, which combines the results of two or more SELECT statements and returns only the unique rows. 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.

Syntax → ***SELECT column1 FROM Tabel1***
UNION SELECT column1 FROM table2;

Example

The following query aims to retrieve a unique list of cities found in either the "Employees" or "Customers" table from the Northwind database, merging and eliminating duplicates using the UNION operator.

```
SELECT city FROM Employees
UNION
SELECT city FROM Customers;
```

ii. UNION ALL:

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

Syntax →

```
SELECT column1 FROM Tabel1
UNION ALL  SELECT column1 FROM table2;
```

Example

The following query retrieves all city names from both the "Employees" and "Customers" tables in the Northwind database, including duplicates, and combines them using the `UNION ALL` operator.

```
SELECT city FROM Employees
UNION ALL
SELECT city FROM Customers;
```

iii. INTERSECT:

Another useful set operation is the INTERSECT operation.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 column1 FROM Tabel1
INTERSECT  SELECT column1 FROM table2;
```

Example

The following query finds the common city names between the "Employees" and "Customers" tables in the Northwind database using the `INTERSECT` operator.

```
SELECT city FROM Employees
INTERSECT
SELECT city FROM Customers;
```

iv. EXCEPT:

Finally, the EXCEPT operation returns only the rows that are unique to the first SELECT statement.

Syntax →

```
SELECT column1 FROM Tabel1
EXCEPT  SELECT column1 FROM table2;
```

Example

The following query retrieves the unique city names from the "Employees" table in the Northwind database that do not appear in the "Customers" table, using the `EXCEPT` operator.

```
SELECT city FROM Employees
EXCEPT
SELECT city FROM Customers;
```

4. ACTIVITY TIME BOXING

Activity Name	Activity Time	Total Time
Instruments Allocation + Setting up Lab	10 mints	10 mints
Walk through Theory & Tasks (Lecture)	60 mints	60 mints
Implementation & Practice time	90 mints	80 mints
Evaluation Time	20 mints	20 mints
	Total Duration	180 mints

5. EXERCISE:

PERFORMANCE TASKS:

1. ***All Example Tasks.***

LAB FILE TASKS:

1. *Write query that retrieve those cities of Customers where any supplier belongs to.*
2. *Write query that retrieve those cities of Customers where no single supplier belongs to.*
3. *Write query that retrieve the regions of Employees and Suppliers.*
4. *List distinct cities from of customers and suppliers, country must be Germany.*
5. *Lists all customers and suppliers (ContactName, City, and Country).*