#PostgreSQL#

Section 5. Set Operations
Union – combine result sets of multiple queries into a single result set.
Intersect – combine the result sets of two or more queries and return a single result set containing rows that appear in both result sets.
Except – return the rows from the first query that do not appear in the output of the second query.

Can You Teach Me above all the topics with Realtime example (create a simple table and insert some values) and show output in Table Format and Also Explain Each topic why this needed, what happen, like syntax and so on.

PostgreSQL Set Operations: Union, Intersect, and Except

1. Union

- **Purpose**: The `UNION` operator combines the result sets of multiple queries into a single result set, eliminating duplicate rows.
- **Syntax**: `SELECT column1, column2 FROM table1 UNION SELECT column1, column2 FROM table2;`
- **Example**:

  Table: `employees`

  - **Columns**: `id`, `name`, `department`
  - **Data**:
    - (1, 'Alice', 'HR')
    - (2, 'Bob', 'IT')
    - (3, 'Charlie', 'Finance')
    - (4, 'David', 'HR')

  Query 1:

```sql
SELECT name FROM employees WHERE department = 'HR';
```

  - **Result**:
    - `Alice`
    - `David`

  Query 2:

```sql
SELECT name FROM employees WHERE department = 'Finance';
```

- **Result**:
  - `Charlie`

Union Operation:

```sql
SELECT name FROM employees WHERE department = 'HR'
UNION
SELECT name FROM employees WHERE department = 'Finance';
```

- **Result**:
  - `Alice`
  - `David`
  - `Charlie`

2. Intersect

- **Purpose**: The `INTERSECT` operator returns a single result set containing rows that appear in both result sets from multiple queries.
- **Syntax**: `SELECT column1, column2 FROM table1 INTERSECT SELECT column1, column2 FROM table2;`
- **Example**:

  Query 1:

```sql
SELECT name FROM employees WHERE department = 'HR';
```

- **Result**:
  - `Alice`
  - `David`

Query 2:

```sql
SELECT name FROM employees WHERE department = 'Finance';
```

- **Result**:
  - `Charlie`

Intersect Operation:

```sql
SELECT name FROM employees WHERE department = 'HR'
INTERSECT
SELECT name FROM employees WHERE department = 'Finance';
```

- **Result**:
  - `David`

3. Except

- **Purpose**: The `EXCEPT` operator returns the rows from the first query that do not appear in the output of the second query.
- **Syntax**: `SELECT column1, column2 FROM table1 EXCEPT SELECT column1, column2 FROM table2;`
- **Example**:

  Query 1:

```sql
SELECT name FROM employees WHERE department = 'HR';
```

  - **Result**:
    - `Alice`
    - `David`

  Query 2:

```sql
SELECT name FROM employees WHERE department = 'Finance';
```

  - **Result**:
    - `Charlie`

  Except Operation:

```sql
SELECT name FROM employees WHERE department = 'HR'
EXCEPT
SELECT name FROM employees WHERE department = 'Finance';
```

  - **Result**:
    - `Alice`
    - `David`

Explanation:

- **Why these operations are needed**:

- **Union**: Helps in combining result sets from different queries to see all the data at once without duplicates.
- **Intersect**: Allows us to find common rows between two or more result sets, which is useful for identifying matching records.
- **Except**: Useful to find rows that do not exist in another set, which helps in filtering out unwanted data or identifying exclusive data.

Each of these operations allows for more complex data analysis and query functionality in PostgreSQL.