| Experiment No.5 |
|---|
| Perform simple queries, string manipulation operations and aggregate functions. |
| Date of Performance: |
| Date of Submission: |

**Aim :-** Write simple query to manipulate string operations and perform aggregate functions like (MIN, MAX, SUM, AVERAGE, COUNT).

**Objective :-** To apply aggregate functions and string manipulation functions to perform simple queries in the database system

**Theory:**

Simple Queries in SQL:

In SQL, a simple query is a request for data from a database table or tables. It allows users to retrieve specific information by specifying the columns they want to retrieve and any conditions for filtering rows based on certain criteria. Simple queries are the backbone of interacting with databases, enabling users to extract the data they need for analysis, reporting, or further processing.

String Manipulation Operations:

String manipulation operations in SQL involve modifying or transforming string values stored in database columns. These operations are crucial for tasks such as formatting data, combining strings, converting case, or extracting substrings. By using string functions and operators, users can manipulate text data to suit their requirements, whether it's for display purposes or for further analysis.

Aggregate Functions:

Aggregate functions in SQL are used to perform calculations on sets of values and return a single result. These functions allow users to summarize data across multiple rows, providing insights into the overall characteristics of the dataset. Common aggregate functions include calculating counts, sums, averages, minimums, and maximums of numerical values. They are essential tools for data analysis, enabling users to derive meaningful insights from large datasets.

Benefits of Understanding These Concepts:

- Data Retrieval: Simple queries allow users to fetch specific data from databases, facilitating data retrieval for various purposes.

- Data Transformation: String manipulation operations enable users to format and transform text data according to their needs, improving data consistency and readability.

- Data Analysis: Aggregate functions help users summarize and analyze large datasets, providing valuable insights into trends, patterns, and statistical measures.

- Data Reporting: By combining simple queries, string manipulation operations, and aggregate functions, users can generate reports and visualizations that communicate key findings effectively.

**Implementation:**

Database:

| employee_id | employee_name | department | salary |
|---|---|---|---|
| 1 | John Doe | Finance | 50000.00 |
| 2 | Jane Smith | Finance | 55000.00 |
| 3 | Michael Johnson | HR | 48000.00 |
| 4 | Emily Brown | HR | 52000.00 |
| 5 | David Lee | IT | 60000.00 |
| 6 | Sarah Clark | IT | 65000.00 |
| 7 | Jennifer White | Marketing | 52000.00 |
| 8 | Kevin Taylor | Marketing | 58000.00 |

1)Concatenate two strings:

```sql
select concat(customer_name, ' lives in ' ,city) as info
from customer;
```

| info |
|---|
| Saurabh lives in Vasai |
| Harsh lives in Mumbai |

2)Get the length of employee names:

```sql
select customer_name, length(customer_name) as name_length
from customer;
```

| customer_name | name_length |
|---|---|
| Saurabh | 7 |
| Harsh | 5 |

3)Find the department with the longest name:

```sql
select customer_name
from customer
group by customer_name
order by length(customer_name) desc
limit 1;
```

```
customer_name
```

Saurabh

4)Calculate the total salary expenditure for all employees:

```
select sum(pincode) as total_pincode
from customer;
```

```
total_pincode
```
802407

5)Find the highest and lowest salary in the company:

```
select max(customer_id) as highest_id,
min(customer_id) as lowest_id
from customer;
```
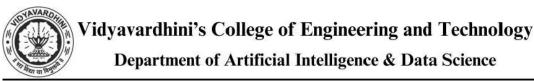
| highest_id | lowest_id |
|---|---|
| 1234567 | 98765 |

6)Calculate the average salary of employees in each department:

```
select customer_id, avg(pincode) as avg_pincode
from customer
group by customer_id;
```

| customer_id | avg_pincode |
|---|---|
| 98765 | 401205 |
| 1234567 | 401202 |

**Conclusion:**

The syntax and brief explanations for five common aggregate functions in SQL:

1. SUM():

Syntax: 'SUM(column_name)'

Explanation: Calculates the sum of all values in the specified column. It is commonly used to find the total of numeric values in a column.

2. AVG():

Syntax: 'AVG(column_name)'

Explanation: Calculates the average (mean) of all values in the specified column. It is useful for finding the average value of numeric data.

3. MAX():

Syntax: 'MAX(column_name)'

Explanation: Returns the maximum value from the specified column. It is used to find the highest value in a set of data.

4. MIN():

Syntax: 'MIN(column_name)'

Explanation: Returns the minimum value from the specified column. It is used to find the lowest value in a set of data.

5. COUNT():

Syntax: 'COUNT(column_name)' or 'COUNT(*)'

Explanation: Returns the number of rows that match the specified condition. It can count the number of non-null values in a column when a column name is provided, or it can count all rows when `COUNT(*)` is used.

These aggregate functions are commonly used in SQL queries to perform calculations and summarizations on data within a table.