|  |  |
| --- | --- |
| **Course Code: CSL402** | **Course Name: Database Management System Lab** |
| **Class: SE-CO** | **Batch: 2020-24** |
| **Roll no: 20CO24** | **Name: Khan Arshad Abdulla** |

**Experiment: 05**

**Aim: Perform simple queries, string manipulation operations and aggregate functions.**

**Case Study Title – Currency Converter**

**Output -**

**Execute the following commands**

================================

1. **WHERE C1 AND C2**

Each selected row must satisfy both the conditions C1 and C2.

Syntax: select \* from table where column1 = ‘value’ AND column2 = ‘value’;

1. **WHERE C2 OR C2**

Each selected row must satisfy either the conditions C1 or the condition C2 or both the conditions C1 and C2.

Syntax: select \* from table where column1 = ‘value’ OR column2 = ‘value’;

AND and OR operators may be combined. For example, in a condition like A AND B OR C. In such compound conditions, the AND operation is done first.

1. **WHERE NOT C1 AND C2**

Each selected row must satisfy both the conditions C2 but not satisfy the condition C1.

Syntax: select \* from table where NOT column1 = ‘value’ and column2 = ‘value’;

1. **WHERE A IN**

Each selected row must have the value of A in the list that follows IN. NOT IN may be used to select all that are not in the list that follows IN.

Syntax: select \* from table where column1 IN (‘value1’,’value2’,’value3’);

1. **WHERE A OPERATOR ANY**

Each selected row must satisfy both the condition for each of the list that follows ANY. Any of the following operators may be used in such a clause, greater than (>), less than (<), less than equal to (<=) and greater than equal to (>=).

Syntax: select \* from table where column1 (>,<,>=,<=) = ‘value’;

1. **WHERE A LIKE X**

Each selected row must have a value of A that satisfies the string-matching condition specified in X. The expression that follow LIKE must be a character string and enclosed in apostrophes (‘).

Syntax: select \* from table where column1 LIKE ‘value1’ ;

LIKE is a powerful string-matching operator that uses two special characters (called as wildcards). These characters are underscore (\_) and percent (%). Underscore represents any single character while percent represents any sequence of n characters.

For eg. ‘Del%’, ‘Gup\_\_’, ‘%ing%’, and ‘%ame\_’

1. **WHERE A BETWEEN X AND Y**

Each selected row must have a values of A between X and Y including values X and Y. NOT BETWEEN may also be used to find rows that are outside the range (X,Y).

Syntax: select \* from table where column1 BETWEEN ‘value1’ AND ‘value2’;

1. **WHERE A IS NULL**

Each selected row must satisfy the condition that A is NULL.

Syntax: select \* from table where column1 IS NULL;

**9. AGGREGATE FUNCTIONS**

**COUNT –** Counts the occurrences of the rows.

**MIN –** Find the smallest value

**MAX** – Find the largest value

**SUM –** Computes the sum

**AVG –** Compute the average of unique values.

**Output:**

**Attach the output of the above commands in txt format.**

**Conclusion:**

|  |
| --- |
| In this experiment we have successfully implemented string manipulation operations and aggregate functions .SQL aggregate functions return a single value, calculated from values in a column. |