Bangladesh University of Business and Technology
Department of Computer Science and Engineering

CSE 232: Database Systems Lab

Lab 04 Tasks - LIKE Operator, Wildcards, Aggregate Functions, GROUP BY, HAVING

LIKE Operator

In simple terms, the LIKE operator is used to match substrings in a query. It is used in a WHERE clause to search for a specified pattern in a column. There are two wildcards often used in conjunction with the LIKE operator:

- % - The percent sign represents zero, one, or multiple characters.
- _ - The underscore represents a single character.

```
SELECT column1, column2, ...
FROM table_name
WHERE columnN LIKE pattern;
```

Examples showing the use of LIKE operator with % and _:

LIKE Operator	Description
WHERE CustomerName LIKE 'a%'	Finds any values that start with "a"
WHERE CustomerName LIKE '%a'	Finds any values that end with "a"
WHERE CustomerName LIKE '%or%'	Finds any values that have "or" in any position
WHERE CustomerName LIKE '_r%'	Finds any values that have "r" in the second position
WHERE CustomerName LIKE 'a_%'	Finds any values that start with "a" and are at least 2 characters in length
WHERE CustomerName LIKE 'a__%'	Finds any values that start with "a" and are at least 3 characters in length
WHERE ContactName LIKE 'a%o'	Finds any values that start with "a" and ends with "o"

Wildcards

A wildcard character is used to substitute one or more characters in a string. Wildcard characters are used with the LIKE operator. The LIKE operator is used in a WHERE clause to search for a specified pattern in a column.
Some of the wildcards recognized by SQL are as follows:

LIKE Operator	Description
WHERE CustomerName LIKE 'a%'	Finds any values that starts with "a"
WHERE CustomerName LIKE '%a'	Finds any values that ends with "a"
WHERE CustomerName LIKE '%or%'	Finds any values that have "or" in any position
WHERE CustomerName LIKE '_r%'	Finds any values that have "r" in the second position
WHERE CustomerName LIKE 'a_%_%'	Finds any values that starts with "a" and are at least 3 characters in length
WHERE ContactName LIKE 'a%o'	Finds any values that starts with "a" and ends with "o"

Aggregate Functions

Aggregate functions are functions that take a collection (a set or multiset) of values as input and return a single value. SQL offers five built-in aggregate functions:

- Average: avg
- Minimum: min
- Maximum: max
- Total: sum
- Count: count

AVG Function

```
SELECT AVG(column_name)
FROM table_name
WHERE condition;

--Example--
SELECT AVG(Price)
FROM Products;
```

MIN Function

```
SELECT MIN(column_name)
FROM table_name
WHERE condition;

--Example--
SELECT MIN(Price) AS LowestPrice
FROM Products;
```

MAX Function

```
SELECT MAX(column_name)
FROM table_name
WHERE condition;

--Example--
SELECT MAX(Price) AS HighestPrice
FROM Products;
```

SUM Function

```
SELECT SUM(column_name)
FROM table_name
WHERE condition;

--Example--
SELECT SUM(Quantity)
FROM OrderDetails;
```

COUNT Function

```
SELECT COUNT(column_name)
FROM table_name
WHERE condition;

--Example--
SELECT COUNT(ProductID)
FROM Products;
```

GROUP BY

There are circumstances where we would like to apply the aggregate function not only to a single set of tuples, but also to a group of sets of tuples; we specify this wish in SQL using the GROUP BY clause.
The attribute or attributes given in the group by clause are used to form groups.

```
SELECT DEPT_NAME, AVG(SALARY) AS AVG_SALARY
FROM INSTRUCTOR
GROUP BY DEPT_NAME;
```

Note: Attributes that appear in the SELECT statement but not in the aggregate function must appear in the GROUP BY clause, otherwise the query will be considered erroneous. For example, the following query is incorrect as ID appears in the SELECT statement, but not in the GROUP BY clause:

```
SELECT DEPT_NAME, ID, AVG(SALARY) AS AVG_SALARY
FROM INSTRUCTOR
GROUP BY DEPT_NAME;
```

HAVING

The HAVING clause is used to specify conditions on groups rather than on tuples. For example, we might be interested in only those departments where the average salary of the instructors is more than $42,000. We cannot use the WHERE clause in this case because WHERE is used to specify conditions on a single set of tuples, whereas in this case, we are interested to find a group of departments where the average salary is more than $42,000.

```
SELECT DEPT_NAME, AVG(SALARY) AS AVG_SALARY
FROM INSTRUCTOR
GROUP BY DEPT_NAME
HAVING AVG(SALARY)>42000;
```

Note: As was the case for the SELECT statement, any attribute that is present in the HAVING clause without being aggregated must appear in the GROUP BY clause, otherwise the query is treated as erroneous.

LIMIT Clause

The LIMIT clause is used to specify the number of records to return. For example, we can find the

```
SELECT DISTINCT salary FROM citizen
ORDER BY salary DESC LIMIT 1,1;
```

Note: The DISTINCT keyword returns distinct salary values. If after sorting, the 1st and 2nd rows both have salary=100000, then if we do not use the DISTINCT keyword, this will return 100000. If we use the DISTINCT Keyword, it will look for the next highest salary which is less than 100000. LIMIT n,m means skip nth row and then return m row(s). So, LIMIT 1,1 skips 1st row and then returns the next 1 row.

POINTS TO REMEMBER

The sequence of operations whenever multiple of this clauses are involved is:
FROM →WHERE →GROUP BY →AGGREGATE FUNCTION →HAVING →ORDER BY

1. As was the case for queries without aggregation, the from clause is first evaluated to get a relation.
2. If a where clause is present, the predicate in the where clause is applied on the result relation of the from clause.
3. Tuples satisfying the where predicate are then placed into groups by the group by clause if it is present. If the group by clause is absent, the entire set of tuples satisfying the where predicate is treated as being in one group.
4. The having clause, if it is present, is applied to each group; the groups that do not satisfy the having clause predicate are removed.
5. The select clause uses the remaining groups to generate tuples of the result of the query, applying the aggregate functions to get a single result tuple for each group.

Tasks

Run the codes in Citizen_Lab4.txt

1. What is the maximum salary?
2. Show Name, Salary, age of citizens whose name excludes 'EE'
3. Show Name, occupation of citizens whose name includes 'AA' and 'LL'
4. Which citizens have 'y' as the third character in their name?
5. Which citizens names start with 'A' and have at least 4 characters?
6. What is the total income of all the teachers of the table?
7. What is the maximum salary of citizens whose name starts with 'A'?
8. Find 5th highest salary.
9. Of all the citizens, how many are doctors?
10. Among all the male citizens, how many are musicians?
11. Which person has the maximum income?
12. How many citizens are in each occupation?
13. Show the maximum salary of each occupation.
14. What is the number of cities with 'aka' in their name?
15. Which male person has the maximum income?
16. (Bonus) Citizens of which occupations have average salary greater than the average salary of all the citizens?