Information and Communication Engineering

Noakhali-3814

Lab Report

Course Title: Database Management System Lab

Course Code: ICE 2208

Report No.: 02

Submission Date: 11-10-2023

Remarks

Submitted To:	**Submitted By:**
Md. Sabbir Ejaz	Name: Habiba Akter
Lecturer	Roll No: BKH2111039F
Department of ICE	Session: 2020-2021
Noakhali Science and Technology University	Year: 2 Term: II

Theory: SQL is a standard language for sorting, manipulating and retrieving data in databases. In this lab, we will work as fundamental database operation including aggregate function (Numeric function - Max, Min, Avg, power, sqrt, etc. and string fuction - upper, lower), Dual, Aliasing, Like(-, %), In and Non In operator.

Aggregate Function Creation: An aggregate function performs a calculation on a set of values, and returns a single value.

Numeric Function:

Max(): The Max() function returns the largest value of the selected column.

Syntax: select Max(col-name) from table-name;

Min(): The Min() function returns the ~~Largest~~ smallest value of the selected column.

Syntax: select Min(col-name) from table-name;

Avg(): The Avg() function returns the average value of the numerical column.

Syntax: select Avg(col-name) from table-name;

POWER(): The POWER () functions returns the value of a number raised to the power of another number.

Syntax: select power(col-name, a) from table-name;

sqrt(): The sqrt() function returns the square root of a number.

Syntax: select sqrt(col-name) from table-name;

String Function:

Upper(): The upper() function converts the text to upper-case.

Syntax: select upper(col-name) from table-name;

Lower(): The Lower() function convert the text to Lower-case.

Syntax: select lower(col-name) from table-name;

Dual Table Creation: There may be a situation where we want to query something that is not from a table. It is a table that is automatically created by Oracle Database along with the data dictionary.

Syntax: select * from Dual;

Aliases Creation: SQL aliases are used to give a table, or a column in a table, a temporary name. Aliases are often used to make column names more readable. An alias only exists for the duration of that query. An alias is created with the 'As' keyword.

Syntax: select col-name As alias-name from table-name;

Like operator creation: The Like operator is used in a where clause to search for a specified pattern in a column.

%: The percent sign % represents zero, one or multiple characters.

_: The underscore sign _ represents one, single character.

Syntax: select col-name from table-name where pattern;

Starts with: To return records that starts with a specific letter or phrase, add the % at the end of the letter or phrase.

Ends with: To return records that ends with a specific letter or phrase, add the % at the beginning of the letter or phrase.

Contains: To return records that contains a specific letter or phrase, add the % both before and after the letter or phrase.

In Operator Creation: The In operator allows we to specify multiple values in a where clause.

The In operator is a shorthand for multiple or conditions.

Syntax: select col-name from table-name where col-name In (value1, value2, ...);

Not In Operator Creation: By using the Not keyword in front of the In operator, we return all records that are Not any of the values in the list.

Syntax: select col-name from table-name where col-name Not In (value1, value2, ...);

Problem Name: Create Aggregate (Numeric) Function

Solution: The create Aggregate (Numeric) Fuction is used to Max(), Min(), Avg(), Power(), sqrt() function.

Sample Input:

Max():

```
select Max(marks) from STUDENT-INFO;
```

Min():

```
select Min(marks) from STUDENT-INFO;
```

Avg():

```
select Avg(marks) from STUDENT-INFO;
```

Power():

```
select power(marks, 2) from STUDENT-INFO;
```

sqrt():

```
select sqrt(marks) from STUDENT_INFO;
```

Sample Output:

BEFORE:

Result Grid | Filter Rows:

	name	roll	session	course	marks
▸	Shifa	39	2020-21	DC	76
	Jemi	40	2020-21	ITF	80
	Jarir	1	2020-21	DC	90
	Jarir	2	2020-21	IPE	86
	Jarir	3	2020-21	DBMS	85

AFTER (Numeric function):

Result Grid | Filter Rows:

	Max(marks)
▸	90

Result Grid | Filter Rows:

	Min(marks)
▸	76

Result Grid | Filter Rows:

	Avg(marks)
▸	83.4

Result Grid | Filter Rows:

	power(marks,2)
▸	5776
	6400
	8100
	7396
	7225

Result Grid | Filter Rows:

	sqrt(marks)
▸	8.717797887081348
	8.94427190999916
	9.486832980505138
	9.273618495495704
	9.219544457292887

Problem Name: Create Aggregate (String) Function

Solution: The create Aggregate (String) Function is used to Upper(), Lower() function.

Sample Input:

Lower():

```
select lower(name) from STUDENT_INFO;
```

Upper():

```
select upper(name) from STUDENT_INFO;
```

Sample Output:

BEFORE:

Result Grid | Filter Rows:

	name	roll	session	course	marks
▸	Shifa	39	2020-21	DC	76
	Jemi	40	2020-21	ITF	80
	Jarir	1	2020-21	DC	90
	Jarir	2	2020-21	IPE	86
	Jarir	3	2020-21	DBMS	85

AFTER (String function):

Result Grid | Filter Rows:

	upper(name)
▸	SHIFA
	JEMI
	JARIR
	JARIR
	JARIR

Result Grid | Filter Rows:

	lower(name)
▸	shifa
	jemi
	jarir
	jarir
	jarir

Problem Name: Dual table creation

Solution: The Dual table creation is used to 'Dual' keyword.

Sample Input:

```
select 2*2 from dual;
select 2*2, 5+8 from dual;
```

Sample Output:

BEFORE:

Result Grid | Filter Rows:

name	roll	session	course	marks
Shifa	39	2020-21	DC	76
Jemi	40	2020-21	ITF	80
Jarir	1	2020-21	DC	90
Jarir	2	2020-21	IPE	86
Jarir	3	2020-21	DBMS	85

AFTER (dual):

Result Grid | Filter Rows:

2*2
4

Result Grid | Filter Rows:

2*2	5+8
4	13

Problem Name: Aliases creation

Solution: Aliases creation is used to as keyword.

Sample Input:

```
select 2*2 as result from dual;
select 2*2 as result, 2+1 as Credit from dual;
```

Sample Output:

BEFORE:

Result Grid | Filter Rows:

name	roll	session	course	marks
Shifa	39	2020-21	DC	76
Jemi	40	2020-21	ITF	80
Jarir	1	2020-21	DC	90
Jarir	2	2020-21	IPE	86
Jarir	3	2020-21	DBMS	85

AFTER (Aliasing):

Result Grid | Filter Rows:

result
4

Result Grid | Filter Rows:

result	Credit
4	3

Problem Name: Like operator creation

Solution: Like operator creation is used to like operator starts with, Ends with and both before and after the letter or phrase.

Sample Input:

```
Select * from STUDENT-INFO where name like '%a';
Select * from STUDENT-INFO where name like 'J%';
Select * from STUDENT-INFO where name like '%e%';
Select * from STUDENT-INFO where name like '_a%';
Select * from STUDENT-INFO where name like '__i%';
Select * from STUDENT-INFO where name like '_e%';
```

Sample Output:

BEFORE:

Result Grid | Filter Rows:

name	roll	session	course	marks
Shifa	39	2020-21	DC	76
Jemi	40	2020-21	ITF	80
Jarir	1	2020-21	DC	90
Jarir	2	2020-21	IPE	86
Jarir	3	2020-21	DBMS	85

AFTER(Like):

Result Grid | Filter Rows:

name	roll	session	course	marks
Shifa	39	2020-21	DC	76

Result Grid | Filter Rows:

name	roll	session	course	marks
Jemi	40	2020-21	ITF	80
Jarir	1	2020-21	DC	90
Jarir	2	2020-21	IPE	86
Jarir	3	2020-21	DBMS	85

Result Grid | Filter Rows:

name	roll	session	course	marks
Jemi	40	2020-21	ITF	80

Result Grid | Filter Rows:

name	roll	session	course	marks
Jarir	1	2020-21	DC	90
Jarir	2	2020-21	IPE	86
Jarir	3	2020-21	DBMS	85

Result Grid | Filter Rows:

name	roll	session	course	marks
Shifa	39	2020-21	DC	76

Result Grid | Filter Rows:

name	roll	session	course	marks
Jemi	40	2020-21	ITF	80

Problem Name: In Operator Creation

Solution: In operator creation is used to where clause.

Sample Input:

Select * from STUDENT-INFO where name in('jarir');

select * from STUDENT-INFO where ~~name~~ roll in (39, 40);

Sample Output:

BEFORE:

Result Grid | Filter Rows:

name	roll	session	course	marks
Shifa	39	2020-21	DC	76
Jemi	40	2020-21	ITF	80
Jarir	1	2020-21	DC	90
Jarir	2	2020-21	IPE	86
Jarir	3	2020-21	DBMS	85

AFTER(In):

Result Grid | Filter Rows:

name	roll	session	course	marks
Jarir	1	2020-21	DC	90
Jarir	2	2020-21	IPE	86
Jarir	3	2020-21	DBMS	85

Result Grid | Filter Rows:

name	roll	session	course	marks
Shifa	39	2020-21	DC	76
Jemi	40	2020-21	ITF	80

Problem Name: Not In Operator Creation

Solution: Not In operator creation is used to where clause.

Sample Input:

select * from STUDENT_INFO where name not in('jarir');

select * from STUDENT_INFO where roll not in(39);

Sample Output:

BEFORE:

Result Grid | Filter Rows:

name	roll	session	course	marks
Shifa	39	2020-21	DC	76
Jemi	40	2020-21	ITF	80
Jarir	1	2020-21	DC	90
Jarir	2	2020-21	IPE	86
Jarir	3	2020-21	DBMS	85

AFTER (Not In):

Result Grid | Filter Rows:

name	roll	session	course	marks
Shifa	39	2020-21	DC	76
Jemi	40	2020-21	ITF	80

Result Grid | Filter Rows:

name	roll	session	course	marks
Jemi	40	2020-21	ITF	80
Jarir	1	2020-21	DC	90
Jarir	2	2020-21	IPE	86
Jarir	3	2020-21	DBMS	85

Discussion: This lab report provide a concise overview of fundamental database operation. These operation are crucial for efficiency managing dat with database for various application. Understanding these concepts is essential for maintaining data integrity and optimizing database performance.