LAB # 3

COMPARISON & LOGICAL OPERATORS

OBJECTIVE

Retrieve data from your database using SELECT Statement

THEORY

The WHERE Clause

The WHERE clause allows you to place conditions on what is returned to you. What you have seen thus far is unrestricted information, in the sense that every row in the table specified has been included in your results.

1. Using the Comparison Operators

Type in the following query in the Editor Pane using a comparison operator in the condition of WHERE clause.

```
a.  select * from employee where sal>18000

b.  select * from employee where eno=525

c.  select * from employee where ename='Jamil'
```

2. The BETWEEN Operator

You can retrieve rows in a range of values by using this operator. The range of values should contain a lower and an upper range.

```
select * from employee where sal between 10000 and 15000
```

Salary of employees ranging from 10000 to 15000 is shown in the results pane.

3. The IN Operator

To check for values in a specific list or column, we use the IN operator.

```
select * from employee where ename in ('Mukaram')
```

You should try the following query and observe the result:

```
select * from employee where emp_name in ('Mukaram', 'Jamil')

select * from employee where emp_commission is null

select * from employee where emp_head in(525,125,231)
```

4. Using the LIKE operator

When not knowing the exact value to search for, then you should use the LIKE operator to match a character pattern. The character pattern matching operation is termed as wildcard search. Following is a list of possible wildcard searches:

WILDCARD	DESCRIPTION
_	Represents single character.
%	Represents a string of any length.
[]	Represents a single character within the range enclosed in brackets.
[^]	Represents any single character not within the range enclosed in the brackets.

The following query list the name(s) of those employees in the employee table whose names start with the letter 'B'.

```
select ename from employee where ename like 'b%'
```

Run these queries using the wildcards:

```
select * from employee where ename like '%al%'
select * from employee where ename like '_a%'
select * from employee where ename like 'a%'
select * from employee where ename like 'ar[s]%'
select * from employee where ename like 'a_e_l'
select * from employee where ename like 'a[e-z]%'
select * from employee where ename like 'a[^r-z]%'
```

The LIKE operator can be used as a shortcut for some BETWEEN comparisons. Following is an example displaying the names and hire dates of employees January 1997 to December 1997.

```
select * from employee where hiredate like '%97%'
```

5. Using the IS NULL operator

The IS NULL operator looks for values that are null. **A null value means that the value is unavailable, unassigned, unknown, or inapplicable.** That is why we do not use (=) sign when testing with the IS NULL operator, because a null value cannot be equal or unequal to any value.

For example if you want to know the detail of those employees whose head is null, then use the following SQL statement:

```
select * from employee where  head is null
```

Test this yourself :

```
select ename,head,sal,comm from employee where  comm is null
```

Logical operators

A logical operator combines the result of two component conditions to produce a single result based on them or to invert the result of a single condition. The logical operators are **AND, OR,** and **NOT**. AND and OR are used to connect search conditions in WHERE clauses. NOT negates the search condition. The following table shows a summary of these operators.

OPERATOR	MEANING
AND	Returns TRUE if *both or all* conditions are true
OR	Returns TRUE if any of the component conditions is true
NOT	Returns TRUE if the following condition is false

6. Using the NOT operator

The query showing the details of those employees whose employee number is not greater than or equal to 200.

```
select * from employee where not eno>=200
```

In the following example the **AND** operator joins two conditions (**not ename='Karim'**) and (**not ename='Jamil'**)and returns TRUE only when both conditions are true. The query showing the result of employees number, their names, and salary whereas whose names are not Karim and Jamil.

Run these queries yourself:

```
select ename, job, sal from employee
where job not in ('Manager','clerk','assistant')

select ename, job, sal from employee where ename not like 'j%'
```

7. Using the AND operator

Type the following query. It will show a department information such that department name is sales and located in Lahore.

```
select * from department where dname='sales' and dept_location='lahore'
```

Run this query yourself:

```
select * from employee where(comm IS NULL and dno=10)
```

8. Using the OR operator

OR operator connects two conditions, but returns TRUE for any of the condition being true.

Type the following query. It shows information from department table where department number is either 10 or 20.

```
select * from department where dno=10 or dno=20
```

RULES OF PRECEDENCE

We can use a mix of logical operators within a single SELECT statement.
Following are the rules:

ORDER EVALUATED	OPERATOR
1	All comparison operators
2	NOT
3	AND
4	OR

You can override rules of precedence by using parentheses.
Consider the following example:

```
select * from employee
```

```
where eno>=500 or  eno<=150 and ename like 'j%'
```

In this statement first the last two conditions are tested for the AND operator as shown in the shaded region above then its result is compared to the OR operator i.e. "*select those employees whose names start with the letter 'J' and whose employee number is smaller than or equal to 150 or if the employee number is greater than or equal to 500*".

Test the query yourself:

Using Parentheses to force priority

Now consider the same example with a change by using parentheses.

```
select * from employee
where (eno>=500 or  eno<=150) and ename like 'j%'
```

Now the precedence will shift from the AND operator to the parentheses (shaded region).

Second line will be evaluated first, and third line will be evaluated second i.e. "*select employees whose employee number is greater than or equal 500 or less than or equal to 150 and if the employee name starts with the letter 'J'.*