

Date : 19.8.25

## EXERCISE-5

### Restricting and Sorting data

After the completion of this exercise, the students will be able to do the following:

- Limit the rows retrieved by the queries

- Sort the rows retrieved by the queries

#### Limits the Rows selected

- Using WHERE clause
- Alias cannot be used in WHERE clause

#### Syntax

SELECT----- FROM----- WHERE condition;

Example: SELECT employee\_id, last\_name, job\_id, department\_id FROM employees WHERE department\_id=90;

#### Character strings and Dates

Character strings and date values are enclosed in single quotation marks. Character values are case sensitive and date values are format sensitive.

Example: SELECT employee\_id, last\_name, job\_id, department\_id FROM employees WHERE last\_name='WHALEN';

Comparison Conditions All relational operators can be used. (=, >, >=, <, <=, <>, !=)

#### Example:

SELECT last\_name, salary FROM employees WHERE salary<=3000;

#### Other comparison conditions

Operator	Meaning
BETWEEN ...AND ...	Between two values
IN	Match any of a list of values
LIKE	Match a character pattern
IS NULL	Is a null values

#### Example:1

SELECT last\_name, salary  
FROM employees  
WHERE salary BETWEEN 2500 AND 3500;

### Example:2

```
SELECT employee_id, last_name, salary, manager_id
FROM employees
WHERE manager_id IN (101, 100, 201);
```

### Example:3

- Use the LIKE condition to perform wildcard searches of valid string values. •
- Two symbols can be used to construct the search string
- % denotes zero or more characters
- \_ denotes one character

```
SELECT first_name, salary
FROM employees
WHERE first_name LIKE '%s';
```

### Example:4

```
SELECT last_name, salary
FROM employees
WHERE last_name LIKE '_o%';
```

### Example:5

**ESCAPE option**-To have an exact match for the actual % and \_ characters  
To search for the string that contain 'SA\_'

```
SELECT employee_id, first_name, salary, job_id
FROM employees
WHERE job_id LIKE
'%sa\_%' ESCAPE '\';
```

### Test for NULL

- Using IS NULL operator

**Example:** SELECT employee\_id, last\_name, salary, manager\_id
FROM employees
WHERE manager\_id IS NULL;

**Logical Conditions** All logical operators can be used.( AND, OR, NOT)

**Example:1** SELECT employee\_id, last\_name, salary, job\_id
FROM employees
WHERE salary >= 10000 AND job\_id LIKE '%MAN%';

**Example:2** SELECT employee\_id, last\_name, salary, job\_id
FROM employees
WHERE salary >= 10000 OR job\_id LIKE '%MAN%';

**Example:3** SELECT employee\_id, last\_name, salary, job\_id
FROM employees
WHERE job\_id NOT IN ('it\_prog', 'st\_clerk', 'sa\_rep');

### Example:1

```
SELECT employee_id, last_name, salary, job_id
FROM employees
WHERE job_id = 'sa_rep' OR
job_id = 'ad_pres' AND salary > 15000;
```

### Example:2

```
SELECT employee_id, last_name, salary, job_id
FROM employees
WHERE (job_id = 'sa_rep' OR
job_id = 'ad_pres') AND salary > 15000;
```

### Sorting the rows

Using ORDER BY Clause

**ASC**-Ascending Order,Default

**DESC**-Descending order

**Example:1**

```
SELECT last_name, salary , job_id,department_id,hire_date FROM employees ORDER BY hire_date;
```

**Example:2**

```
SELECT last_name, salary , job_id,department_id,hire_date FROM employees ORDER BY hire_date DESC;
```

**Example:3**

**Sorting by column alias**

```
SELECT last_name, salary*12 annsal , job_id,department_id,hire_date FROM employees ORDER BY annsal;
```

**Example:4**

**Sorting by Multiple columns**

```
SELECT last_name, salary , job_id,department_id,hire_date FROM employees ORDER BY department_id, salary DESC;
```

**Rules of Precedence**

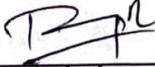
Order Evaluated	Operator
1	Arithmetic
2	Concatenation
3	Comparison
4	IS [NOT] NULL, LIKE, [NOT] IN
5	[NOT] BETWEEN
6	Logical NOT
7	Logical AND
8	Logical OR

Find the Solution for the following:

1. Create a query to display the last name and salary of employees earning more than 12000.  
Ans: `SELECT last_name, salary FROM employees WHERE salary > 12000;`
2. Create a query to display the employee last name and department number for employee number 176.  
Ans: `SELECT last_name, department_id FROM employees WHERE employee_id = 176;`
3. Create a query to display the last name and salary of employees whose salary is not in the range of 5000 and 12000. (hints: not between )  
Ans: `SELECT last_name, salary FROM employees WHERE salary NOT BETWEEN 5000 AND 12000;`
4. Display the employee last name, job ID, and start date of employees hired between February 20,1998 and May 1,1998.order the query in ascending order by start date.(hints: between)  
Ans: `SELECT last_name, job_id, hire_date FROM employees WHERE hire_date BETWEEN '1998-02-20' AND '1998-05-01' ORDER BY hire_date ASC;`
5. Display the last name and department number of all employees in departments 20 and 50 in alphabetical order by name.(hints: in, orderby)  
Ans: `SELECT last_name, department_id FROM employees WHERE department_id IN (20, 50) ORDER BY last_name ASC;`
6. Display the last name and salary of all employees who earn between 5000 and 12000 and are in departments 20 and 50 in alphabetical order by name. Label the columns EMPLOYEE, MONTHLY SALARY respectively.(hints: between, in)  
Ans: `SELECT last_name AS EMPLOYEE, salary AS 'MONTHLY SALARY' FROM employees WHERE salary BETWEEN 5000 AND 12000 AND department_id IN (20, 50) ORDER BY last_name ASC;`
7. Display the last name and hire date of every employee who was hired in 1994.(hints: like)  
Ans: `SELECT last_name, hire_date FROM employees WHERE hire_date LIKE '1994%';`
8. Display the last name and job title of all employees who do not have a manager.(hints: is null)  
Ans: `SELECT last_name, job_id FROM employees WHERE manager_id IS NULL;`
9. Display the last name, salary, and commission for all employees who earn commissions. Sort data in descending order of salary and commissions.(hints: is not null,orderby)  
Ans: `SELECT last_name, salary, commission_pct FROM employees WHERE commission_pct IS NOT NULL ORDER BY salary DESC, commission_pct DESC;`
10. Display the last name of all employees where the third letter of the name is a.(hints:like)  
Ans: `SELECT last_name FROM employees WHERE last_name LIKE '__a%';`
11. Display the last name of all employees who have an a and an e in their last name.(hints: like)  
Ans: `SELECT last_name FROM employees WHERE last_name LIKE '%a%' AND last_name LIKE '%e%';`
12. Display the last name and job and salary for all employees whose job is sales representative or stock clerk and whose salary is not equal to 2500 ,3500 or 7000.(hints:in,not in)  
Ans: `SELECT last_name, job_id, salary FROM employees WHERE job_id IN ('SA_REP', 'ST_CLERK') AND salary NOT IN (2500, 3500, 7000);`

13. Display the last name, salary, and commission for all employees whose commission amount is 20%.(hints:use predicate logic)

Ans: **SELECT last\_name, salary, commission\_pct FROM employees WHERE commission\_pct = 0.20;**

Evaluation Procedure	Marks awarded
Query(5)	5
Execution (5)	5
Viva(5)	5
Total (15)	15
Faculty Signature	 29/25

## Pracice Questions

### Sorting Rows

1. In the example below, assign the employee\_id column the alias of "Number." Complete the SQL statement to order the result set by the column alias.

SELECT employee\_id, first\_name, last\_name FROM employees;

Ans: SELECT employee\_id AS Number, first\_name, last\_name FROM employees ORDER BY Number;

2. Create a query that will return all the DJs on Demand CD titles ordered by year with titles in alphabetical order by year.

Ans: SELECT cd\_title, release\_year FROM djs\_on\_demand\_cds ORDER BY release\_year, cd\_title;

3. Order the DJs on Demand songs by descending title. Use the alias "Our Collection" for the song title.

Ans: SELECT song\_title AS "Our Collection" FROM djs\_on\_demand\_songs ORDER BY song\_title DESC;

4. Write a SQL statement using the ORDER BY clause that could retrieve the information needed.

Ans: SELECT first\_name, last\_name FROM employees ORDER BY last\_name;

Evaluation Procedure	Marks Awarded
Practice Evaluation (5)	5
Viva (5)	5
Total (10)	10
Faculty Signature	 2/9/25