12/8/25

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
-

Limiting the Rows selected

- Using WHERE clause
- Alias cannot used in WHERE clause

Syntax

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

Example:

SELECT employee_id,last_name, job_id, deparment_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, deparment_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');
```

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

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;
```

Find the Solution for the following:

1. Create a query to display the last name and salary of employees earning more than 12000.

```
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.

```
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)

```
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)

```
SELECT last_name, job_id - hire_date. from employees
where hire_date between "1998-02-20" and
"1998-05-01" ORDER BY hire_date;
```

5 Display the last name and department number of all employees in departments 20 and 50 in alphabetical order by name.(hints: in, orderby)

SELECT last_name and department_id from employees where departmentid between 20 and 50 order by first_name;

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)

SELECT last_name as EMPLOMEE, Salary as "MONTHY SALARY" from employees where Salary between 500 and 12000 and department_id between 20 and 50 order by first_name;

7. Display the last name and hire date of every employee who was hired in 1994.(hints: like)

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)

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 nul,orderby)

SELECT last_name, Salary, Commission_pct from employees where commission_pct is NOT NULL order by Salary and commission_pct desc;

10. Display the last name of all employees where the third letter of the name is *a*.(hints:like)

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)

```
SELECT last_name from employees where last_name
like "%a%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)

```
SELECT last_name ,job_id , Salary from employees,
where job_id in ("Sales representatives", "Stock 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)

```
SELECT last_name , Salary , commission_pct from
employees where commission_pct = Salary * 0.2;
```

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

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;

SELECT employee_id AS "Number", first_name, last_name FROM employee. 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.

SELECT title, year FROM d_cds ORDER BY YEAR, title;

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

SELECT song_title AS "our Collection" FROM d_Songs ORDER BY song_title DESC;

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

SELECT id, first_name, last_name, Salary FROM d_staffs ORDER BY Salary DESC, last_name;