

1. Write a query to find employees whose name contains the letter 'A' anywhere in the name, and order the result by their hire date.

```
- Select empname, hiredate  
  From employee  
  Where empname LIKE '%A%'  
  Order by hiredate;
```

2. Write a query to find employees whose name starts with 'M' and whose job is 'SALESMAN'.

```
- Select empname, job  
  From employee  
  Where empname LIKE 'M%' AND job = salesman;
```

3. Write a query to find employees whose name ends with 'R', whose job is either 'SALESMAN' or 'MANAGER', and display their salaries.

```
- Select empname, job, sal  
  From employee  
  Where empname LIKE '%R' AND JOB IN  
    ('SALESMAN', 'MANAGER');
```

4. Write a query to find employees whose name has exactly 6 letters, and display their names, jobs, and salaries.

```
- Select empname, job, sal  
  From employee  
  Where empname LIKE '_____';
```

5. Write a query to find employees whose name contains the letter 'S' in the second position and display their names, jobs, and hire dates.

```
- Select empname, job, hiredate  
  From employee  
  Where empname LIKE '_S%' ;
```

6. Write a query to find employees ordered by their hire date in ascending order, but if two employees have the same hire date, order by their salary in descending order.

```
- Select *  
  From employee  
  Order by hiredate asc, sal desc;
```

7. Write a query to find employees ordered by their commission in descending order, and then by their job.

```
- Select *  
  From employee  
  Order by comm desc, job;
```

8. Write a query to find employees in department 30, ordered by their job title in ascending order and then by salary in descending order.

```
- Select *  
  From employee  
  Where deptno = 30  
  Order by job asc, sal desc;
```

9. Write a query to find employees whose salary is between 1000 and 3000, ordered by their hire date and then by their job in descending order.

```
select * from employee  
Where sal between 1000 and 3000  
Order by hiredate asc, job desc;
```

10. Write a query to find employees in departments 10 and 20, ordered by their hire date, and if the hire date is the same, order by their name in ascending order.

```
Select * from employee  
Where deptno in (10,20)  
Order by hiredate asc, empname asc;
```

11. Write a query to display the top 5 employees with the highest commission who work as a SALESMAN.

```
- Select empname, comm  
From employee  
Where job = 'salesman'  
Order by comm desc  
Limit 5;
```

12. Write a query to display the top 3 employees with the earliest hire dates in department 30.

```
- Select empname, hiredate  
From employee  
Where deptno=30  
Order by hiredate asc  
Limit 3;
```

13. Write a query to find the top 5 employees who have the lowest salary and display their names, jobs, and salaries.

```
- Select empname, job, sal  
From employee  
Order by sal asc  
Limit 5;
```

14. Write a query to display the first 3 employees in terms of salary from department 20, ordered by salary in ascending order.

```
Select empname, sal  
From employee  
Where deptno=20  
Order by sal asc  
Limit 3;
```

15. Write a query to display the bottom 3 employees (by salary) in departments 10 and 20, ordered by salary in descending order.

```
Select empname, sal  
From employee  
Where deptno in (10,20)  
Order by sal desc  
Limit 3;
```

● HAVING CLAUSE:

It is used to filter records based on aggregate functions(count(), sum(),.....)

Purpose:

- It filters the grouped data based on a condition.
- While where clause filters the rows before grouping.
- Having clause filters the group after aggregation
- It is used in conjunction with group by clause to filter grouped data
- Having clause is used after the group by clause
- We can use aggregate functions in having clause

Ex. Get the departments where total salary is more than 9000.

```
Select deptno, sum(sal) AS total_salary  
From employee  
Group by deptno  
Having sum(sal) > 9000;
```

Explanation:

- The data is grouped by department
- The sum(sal) for each department is calculated
- Only groups where total salary is more than 9000 will be in resultset.

Ex. To get departments with more than 5 employees.

```
Select deptno, count(*) AS employee_count  
From employee  
Group by deptno  
Having count(*) > 5;
```

Ex. To get the departments where total salary is more than 9200 and the maximum salary is less than 10000.

```
Select deptno, sum(sal) as Total_salary, max(sal) as max_salary  
From employee  
Group by deptno  
Having sum(sal) > 9200 AND max(sal) < 10000;
```

(Recheck)

D5 assignment

1. Write a query to display all distinct job titles from the emp table.

```
Select distinct job  
From employee;
```

2. Write a query to display distinct commission (COMM) values that employees earn.

```
Select distinct comm  
From employee;
```

3. Write a query to find how many distinct managers (MGR) are present in the emp table.

```
Select count(distinct mgr) AS distinct_manager  
From employee;
```

4. Write a query to display distinct combinations of salary (SAL) and commission (COMM).

```
Select distinct sal,comm  
From employee;
```

5. Write a query to display distinct department number (DEPTNO) and job title (JOB) combinations.

```
Select distinct deptno, job  
From employee;
```

6. Write a query to display the total and average salary (SAL) for each job title (JOB).

```
Select job, sum(sal) as total_salary, avg(sal) as  
average_salary  
From employee  
Group by job;
```

7. Write a query to find the highest salary (SAL) among employees who have non-null commission (COMM).

```
Select max(sal) as highest_sal
```

```
From employee
```

```
Where comm is not null;
```

8. Write a query to count the number of employees earning more than the average salary.

```
Select count(*) as employee_count
```

```
From employee
```

```
Where sal > (Select avg(sal) from employee);
```

```
To calculate the average salary of employees:
```

```
Select avg(sal) from employee;
```

9. Write a query to calculate the total salary for employees under manager (MGR = 7839).

```
Select sum(sal) as total_salary
```

```
From employee
```

```
Where mgr=7839;
```

10. Write a query to find the earliest hire date (HIREDATE) for each department.

```
Select deptno, min(hiredate) as earliest_hiredates
```

```
From employee
```

```
Group by deptno;
```

11. Write a query to count the number of employees for each department (DEPTNO) and job (JOB).

```
Select deptno, job, count(*) as employee_count
```

```
From employee
```

```
Group by deptno, job;
```

12. Write a query to calculate the total salary of employees working under each manager (MGR).

```
Select mgr, sum(sal) as total_salary  
From employee  
Group by mgr;
```

13. Write a query to display the average salary by job for departments that have more than 3 employees.

```
Select deptno, job, avg(sal) as average_salary  
From employee  
Group by deptno, job  
Having count(*) >3;
```

14. Write a query to find the departments where the maximum salary is greater than 2000.

```
Select deptno  
From employee  
Group by deptno  
Having max(sal) > 3000;
```

15. Write a query to find the total salary and employee count by manager (MGR).

```
Select mgr, sum(sal) as total_salary, count(*) as  
employee_count  
From employee  
Group by mgr;
```


Subquery

1. Single row subquery: returns a single value
2. Multi-row subquery : returns multiple rows
3. Correlated queries: depends on outer query for execution

Ex. Find the name of employee who earns the highest salary

```
Select empname
```

```
From employee
```

```
Where sal = (select max(sal) from employee);
```

Ex. Find all employees who work in the same department as employees earning more than 3500.

```
SELECT *
```

```
FROM EMPLOYEE
```

```
WHERE DEPTNO IN (
```

```
    SELECT DEPTNO
```

```
    FROM EMPLOYEE
```

```
    WHERE SAL > 3500
```

```
);
```

Correlated Query:

the inner query depends on value from outer query

Ex. Find employees who earn more than the average salary of their department.

```
SELECT E1.EMPNO, E1.EMPNAME, E1.SAL, E1.DEPTNO
FROM EMPLOYEE E1
WHERE E1.SAL > (
    SELECT AVG(E2.SAL)
    FROM EMPLOYEE E2
    WHERE E1.DEPTNO = E2.DEPTNO
);
```

EXPLANATION:

1. Outer Query: Retrieves employees whose salaries are higher than the department's average salary
2. Inner Query: Calculates the average salary for the department of the current employee(e1.deptno)

3. Correlation(**E1.DEPTNO = E2.DEPTNO**): Links the inner query to the current row of the outer query .

TASK: Find employees with the same job as their manager.

DATE FUNCTIONS:

1. Curdate() :

To fetch the current date (YYYY-MM-DD)

Ex. Select curdate() ;

O/P:

| |
|------------|
| +-----+ |
| curdate() |
| +-----+ |
| 2025-01-23 |
| +-----+ |

2. NOW ()

- To fetch the current date and time

- (YYYY-MM-DD HH:MM:SS)

- Ex. Select now() ;

| |
|-------------------------|
| - +-----+ |
| - now() |
| - +-----+ |
| - 2025-01-23 11:31:06 |
| - +-----+ |

3. DATE_FORMAT()

To display dates in user-friendly format
With the help of placeholders

Common Placeholders:

%d :Day of month(2 digits)

%D :Day of the month with suffix(1st,2nd)

%m :Month in numeric format

%M :Full Month name(March, ...)

%y : Year in 2-digit format(25 for 2025)

%Y : Year in 4 digit format(2025)

Ex. To display the curr date in custom format

```
Select DATE_FORMAT(CURdate(), ' %M %d %Y' );
```

```
+-----+
| DATE_FORMAT(CURdate(), ' %M %d %Y' ) |
+-----+
| January 23 2025                        |
+-----+
```

```
Select DATE_FORMAT(CURdate(), '%M %D %Y');
```

```
+-----+
| DATE_FORMAT(CURdate(), '%M %D %Y') |
+-----+
| January 23rd 2025                  |
+-----+
```

Ex. Display the hiredate in custom format;

```
Select empname, DATE_FORMAT(hiredate, '%M %D %Y')
AS NEW_DATE
```

```
From employee;
```

4. DATEDIFF()

Calculates the number of days between two dates

```
DATEDIFF(from_date, to_date);
```

Ex. To find days between two dates

```
Select datediff( '2022-11-23' , ' 2000-12-01' )
AS days;
```

Ex. To find the experience in years from employee table.

```
Select empno, empname,  
floor(datediff(curdate(), hiredate)/365) AS  
Experience  
From employee;
```

5. DATE_ADD()

Adds an interval (like days, weeks, months or years) to a given date.

SYNTAX:

```
DATE_ADD(date, INTERVAL value);
```

Ex. To add 1 week to current date.

```
Select Date_Add(curdate(), INTERVAL 1 week) As  
Future_date;
```

```
Select Date_Add(curdate(), INTERVAL 1 year) As  
Future_date;
```

```
Select Date_Add(curdate(), INTERVAL 4 day) As  
Future_date;
```

Ex. To find employees who joined in feb.

Month() -> function to extract month of a date

Select empname,hiredate

From employee

Where month(hiredate) = 2 ;

TASK:

Table: Medicine

Columns: MID, Mname, price, exp_date

Requirement: Find medicines expiring in 3 months.

Data Manipulation Language (DML)

Insert: adds new rows to table

Update: modifies the existing data in table

Delete: removes rows from table

Update statement

Syntax:

Update table_name

Set column1 = value1, column2 = value2,

Where condition

Ex. To increase the salary of employees in department 30 by 15%.

UPDATE employee

Set sal=sal*1.15

Where deptno=30;

Ex. To change the job of SMITH(7369) from clerk to senior clerk.

Update employee

Set job = 'SENIOR CLERK'

Where empno = 7369;

DELETE Statement:

SYNTAX:

Delete from table_name

Where condition;

Ex.

To remove the employee whose empno = 7900.

DELETE FROM EMPLOYEE

Where empno = 7900;

Ex. Delete all employees whose salary is less than 1000.

Delete from employee

Where sal < 1000;

