1. Retrieve the name and age of employees who **joined in the year 2023**.

------- Select ename, age from Employee where year(hire\_date)=2023;

2. Display the difference between **maximum salary and minimum salary** from the employee table. Use suitable column alias.

------- Select max(salary) – min(salary) as “Sal Diff” from Employee;

1. Create a **procedure** which accepts values of **Department table** as parameters and does the following:
   1. Inserts the values to the Department table.
   2. Displays the contents of the Department table

----------- Delimiter $$

Create procedure test\_procedure (dno int, dname varchar(10), location varchar(10))

Begin

Insert into Department values (dno, dname, location);

Select \* from Department;

End$$

1. Retrieve the **highest salary** in each department.

----- Select dnum, max(salary) from Employee group by dnum;

5.Retrieve the names of employees whose name starts with **letter ‘t’ and letter ‘m’.**

-------- Select ename from Employee where ename like ‘t%’ or ename like ‘m%’;

6.Display the name and salary of employees who **work in ‘Mysuru’.**

------- Select ename, salary from Employee where dnum in (select dno from Department where location= “Mysuru”;

7.Retrieve the **lowest salary in each department.**

------- Select dnum, min(salary) from Employee group by dnum;

8.Create a **trigger** which inserts values into another table whenever an employee is moved from one department to another department. The values to be recorded are **“eno, old\_dnum, new\_dnum”.**

------ Delimiter $$

Create trigger test\_trigger

after update of dnum on Employee

for each row

if (new.dnum!=old.dnum) then

insert into test\_dept values (new.eno,new.dnum, new.dnum);

$$

9.Retrieve the names of employees who are **30 years old and joined in the year 2020.**

---- Select ename from Employee where age=30 and year(hire\_date)=2020;

10.Display the names of employees who earn **between 50000 and 100000.**

----- Select ename from Employee where salary between 50000 and 100000;

11.Display the **department number and number of employees in each department.**

------- Select dnum, count(eno) from Employee group by dnum;

12.Create a **view** which displays the ***‘Equi’*** join results of Employee and Department table. Note: Results must contain meaningful data.

------------- Create view test\_view as

Select \* from Employee, Department where dnum=dno;

13.Retrieve the name and age of employees who joined **between ‘01-08-2008’ and ’31-07-2012’.**

-------------- Select ename, age from Employee where hire\_date between ‘01-08-2008’ and ’31-07-2012’;

14.Display the **cross product of Employee and Department table.**

------------- Select \* from Employee, Department;

15.Write the statement used to **check whether a query used an index or not.**

------------ Explain select eno, ename from Employee;

16.Add **a new column ‘temp’** in the Employee table. The position of the newly added column **mustbe theafter ‘salary’** column. Remove the newly added column.

--------- Alter table Employee add column temp int after salary;

17.Delete the employees who are **older than 60 years.**

------- Alter table Employee drop column temp;

Delete from Employee where age > 60;

18.Display the names of employees whose name **does not end with letter ‘a’.**

------ Select ename from Employee where ename not like ‘%a’;

19.Display **the top 5 salaries** from the employee table.

------ Select salary from Employee order by salary desc limit 5;

20.Create a **view** which displays the ***‘Natural’ join*** results of Employee and Department table. Note: Results must contain meaningful data.

------ Create view test\_view as

Select \* from Employee natural join Department;

ename, dname from Employee, Department where dnum=dno;

21.Retrieve the **employee name and department name** of employees who joined in the year 2023.

--------- Select ename, dname from Employee, Department where dnum=dno and year(hire\_date)=2023;

22.Display the difference between **maximum salary and minimum salary** from each department. Use suitable column alias.

----------- Select dnum, max(salary) – min(salary) as “Sal Diff” from Employee group by dnum;

24.Create a **procedure** which accepts values of **Employee** table as parameters and does the following:

1. Inserts the values to the Employee table.

Displays the contents of the Employee table.

------------ Delimiter $$

Create procedure test\_procedure (eno int, ename varchar(10), salary int, age int, hire\_date date, dnum int)

Begin

Insert into Employee values (eno, ename, salary, age, hire\_date, dnum);

Select \* from Employee;

End$$

25.Retrieve the names of employees whose name **does not start with letter ‘a’or letter ‘b’.**

------ Select ename from Employee where ename not like ‘a%’ or ename not like ‘b%’;

26.Display the **name and hire date of employees who work in ‘Bengaluru’.**

------ Select ename, hire\_date from Employee where dnum in (select dno from Department where location = “Bengaluru”;

27.Retrieve the **average salary in each department.**

------- Select dnum, avg(salary) from Employee group by dnum;

28.Create a **trigger** which inserts valuesinto another table whenever an employeenumberis changed. The values to be recorded are **“old\_eno, new\_eno”.**

-------- Delimiter $$

Create trigger test\_trigger

after update of ename on Employee

for each row

if (new.eno!=old.eno) then

insert into test\_emp values (old.eno, new.eno);

$$

29.Retrieve the names of employees who **earn more than 100000 and are older than 30 years.**

----- Select ename from Employee where age>30 and salary>100000;

30.Display the names of employees who joined the company **between ‘10-10-2021’ and   
‘10-10-2022’.**

------- Select ename from Employee where hire\_date between ’10-10-2021’ and ’10-10-2022’;

31.Display the **department number and number of employees in each department.**

Select dnum, count(eno) from Employee group by dnum;

32.Create **a view** which displays the ***‘Natural’ join*** results of Employee and Department table.

Create view test\_view as

Select \* from Employee natural join Department;

33.Retrieve the name and salary of employees who joined **between** ‘01-08-2008’ and ’31-07-2012’ and work **for ‘Sales’ department.**

Select ename, salary from Employee, Department where hire\_date between ‘01-08-2008’ and ’31-07-2012’ and dnum=dno and dname= ‘Sales’;

34.Display the **cartesian product** of Employee and Department table.

Select \* from Employee, Department;

35.**Increase the salary** of all employees who work in ‘Sales’ department.

Update Employee set salary=salary\*1.1 where dnum in (select dno from Department where dname= ‘Sales’;

36.Write SQL statement to **modify an existing column’s data type in Employee table. Display the table structure before and after modification.**

Describe Employee;

Alter table Employee modify column salary deminal (5,2);

Describe Employee;

37**.Delete** the employees who are older than **65 yearsand who are working in ‘Mysuru’**.

Delete from Employee where age > 65 and dnum in (select dno from Department where location = ‘Mysuru’);

38.Display the names of employees whose name **does not start with letter ‘a’.**

Select ename from Employee where ename not like ‘a%’;

39.Display **the least 5 salaries** from the employee table.

Select salary from Employee order by salary limit 5;

40.Create a **view** which displays the ***‘Equi’*** join results of Employee and Department table. Note: View must not contain ‘Location’ attribute and must contain all other attributes.

Create view test\_view as

Select eno, ename, salary, age, hire\_date, dnum, dno, dname from Employee, Deprtament where dno=dnum;