1) Consider the relation employee (emp\_id,e\_name,salary ,Date of Joining,Dapt\_no,Designation) perform basic SQL operations.

1. Create table employee.

Create table employee(emp\_id int , e\_name varchar(20),salary int, Date\_of\_joining date ,Dept\_no int ,Designation varchar(20), Department varchar(20));

1. Insert 10 records in table.
2. Create a view emp\_vl of table employee which has emp\_id , name and dept-attributes.

Create view emp\_v1 as select emp\_id, e\_name, Dept\_no from employee;

1. Create view of table.

Create view emp\_v2 as select \* from employee;

1. Update dept of any employee in view. Check whether it gets updated or not.

Update employee set Dept\_no=2 where emp\_id=1;

1. Create emp\_id as primary key and show indices on table employee.

Alter table employee add primary key (emp\_id);

1. Show indices on table

Show index from employee;.

1. Create user defined index on any column.

Create index in1 on employee (Dept\_no , Designation);

2) Consider the relation employee (emp\_id,e\_name,salary ,Date of Joining,Dapt\_no,Designation) perform basic SQL operations.

1. Display employees whose name contains letter ‘e’.

select \* from employee where e\_name like '%e%';

1. Display different types of designation

select distinct(Designation) from employee;

1. Display name and salary of employee whose location is Mumbai

select e\_name,salary from employee where city='Mumbai';

1. Display name and department of employee working in Manager or Marketing department

select e\_name,Dept\_no from employee where Designation in ('Manager','Finance Manager');

1. Display the department name whose employees are more than one

select Department,count(\*) as no\_emp from employee group by Department Having count(\*)>1;

1. Rename employee table as emp1

alter table employee rename to emp1;

1. Add a new column city in the employee table.

Alter table employee add column city varchar(20);

3)Consider the relation employee(emp\_id,e\_name,salary ,Date of Joining,Dapt\_no,Designation) perform basic SQL operations.

1. Find department in which maximum employees work.

select Department,count(\*) as no\_emp from emp1 group by Department order by no\_emp desc limit 1;

1. Display name, designation and department no of employees whose name starts with either ‘A’ or ‘P’.

SELECT e\_name, Designation, Dept\_no

-> FROM emp1

-> WHERE e\_name LIKE 'A%' OR e\_name LIKE 'P%';

1. Display max. salary from department 2 and min. salary from department 4.

select a.salary as dep\_2\_sal , b.salary as dep\_4\_sal from emp1 a,emp1 b where a.salary=(select max(salary) from emp1 where Dept\_no=2) and b.salary=(select min(salary) from emp1 where Dept\_no=4);

1. Display employee data where salary is less than average salary from department 3.

SELECT \*

-> FROM emp1

-> WHERE salary < (SELECT AVG(salary) FROM emp1 WHERE Dept\_no = 3);

1. Display employees who were hired earliest or latest.

SELECT \*

-> FROM emp1

-> WHERE Date\_of\_joining = (SELECT MIN(Date\_of\_joining) FROM emp1) OR Date\_of\_joining = (SELECT MAX(Date\_of\_joining) FROM emp1);

1. Display name and department no of employees who are manager, market analysts. Use prediactes

SELECT e\_name, Dept\_no

-> FROM emp1

-> WHERE Designation IN ('Manager', 'Finance Analyst');

1. List employees hired in August.

SELECT \*

-> FROM emp1

-> WHERE EXTRACT(MONTH FROM Date\_of\_joining) = 8;

1. List employees who are hired after 31/12/2006.

SELECT \*

-> FROM emp1

-> WHERE Date\_of\_joining > '2006-12-31';

1. Find average annual salary per department.

SELECT Dept\_no, AVG(salary) AS avg\_salary

-> FROM emp1

-> GROUP BY Dept\_no;

4)Consider two tables Customer(c\_id, c\_name , email , city , pincode)Order(order\_id , date , amount , cust\_id.

1. Create both the tables with primary key and foreign key constraints.

CREATE TABLE Customer ( c\_id INT PRIMARY KEY, c\_name VARCHAR(255), email VARCHAR(255), city VARCHAR(255), pincode VARCHAR(10));

CREATE TABLE Order ( order\_id INT PRIMARY KEY, date DATE, amount DECIMAL(10, 2), cust\_id INT, FOREIGN KEY (cust\_id) REFERENCES Customer(c\_id) );

1. insert 10 records each.
2. Find all orders placed by customers with cust\_id 2

SELECT \* FROM Order WHERE cust\_id = 2;

1. Find list of customers who placed their order and details of order

SELECT c.c\_name, o.order\_id, o.date, o.amount

FROM Customer c

Inner join Order o ON c.c\_id = o.cust\_id;

1. List of customers who haven’t placed order

SELECT c.c\_name

FROM Customer c

LEFT JOIN Order o ON c.c\_id = o.cust\_id

WHERE o.cust\_id IS NULL;

1. List all orders and append to customer table
2. Display all records
3. Display customer that are from same city

SELECT c1.c\_name, c2.c\_name

FROM Customer c1

JOIN Customer c2 ON c1.city = c2.city AND c1.c\_id <> c2.c\_id;

5) Consider tables Borrower (RollNo, Name, DateofIssue, NameofBook, Status) and

Fine (Roll\_no,Date,Amt). Status is either Issued or Returned.

1. Create both the tables with primary key.

CREATE TABLE Borrower (

RollNo INT PRIMARY KEY,

Name VARCHAR(255),

DateofIssue DATE,

NameofBook VARCHAR(255),

Status VARCHAR(10)

);

CREATE TABLE Fine (

Roll\_no INT PRIMARY KEY,

Date DATE,

Amt DECIMAL(10, 2)

);

2. Insert 10 records each.

3. Find count of books with Issued status.

SELECT COUNT(\*) AS IssuedBookCount

FROM Borrower WHERE Status = 'Issued';

4. Display all records.

5. Display RollNo whose date of issue is same.

8) For the given relation schema: employee(employee-name, street, city)

works (employee-name, company-name, salary)

company (company-name, city)

manages (employee-name, manager-name)

Give an expression in SQL for each of the following queries:

a) Find the names, street address, and cities of residence for all employees who work for same company and earn more than $10,000.

SELECT e.employee\_name, e.street, e.city

FROM employee e, works w

WHERE e.employee\_name = w.employee\_name

AND w.salary > 10000

AND (w.company\_name, w.salary) IN (

SELECT w1.company\_name, MAX(w1.salary)

FROM works w1

GROUP BY w1.company\_name

HAVING COUNT(\*) > 1

);

b) Find the names of all employees in the database who live in the same cities as the companies for which they work.

select e.employee-namefrom employee e, works w, company cwhere e.employee-name = w.employee-name and e.city = c.city and w.company-name = c.company-name;

c) Find the names of all employees who earn more than the average salary of all employees of their company. Assume that all people work for at most one company.

select employee-namefrom works t where salary >(select avg(salary) from works s where t.company-name = s.company-name);

9) For the given relation schema: employee(employee-name, street, city)

works (employee-name, company-name, salary)

company (company-name, city)

manages (employee-name, manager-name)

Give an expression in SQL for each of the following queries:

1. Find the name of the company that has the smallest payroll.

select company-namefrom worksgroup by company-namehaving sum(salary) <= all (select sum(salary) from works group by company-name)

1. Find the names of all employees in the database who live in the same cities and on the same streets as do their managers.

select p.employee-namefrom employee p, employee r, manages mwhere p.employee-name = m.employee-name and m.manager-name =r.employee-name and p.street = r.street and p.city = r.city;