**Task-1: create a table with name products and entries are**

ProductID, Product name, supplier ID, categoryID, Quantity per unit, unit price, units in stock, units on order, reorder level, discontinued.

create table products(ProductID int not null,

Product\_name varchar(100) primary key,

supplier\_ID int not null,

category\_ID int not null,

quantity\_per\_unit int ,

unit\_price int,

units\_in\_stock int,

units\_on\_order int,

reorder\_level int);

Indexes: primary key product name, foreign key is products categories & products suppliers.

Queries:

1. Write a mysql query to get Product name and quantity/unit.

Select product\_name,quantity\_per\_unit from products;

1. Write a MySQL query to get current Product list (Product ID and name).

Select product\_ID,Product\_name from products where qualification=’continued’;

1. Write a MySQL query to get discontinued Product list (Product ID and name).

Select product\_ID,Product\_name from products where qualification=’discontinued’;

1. Write a MySQL query to get most expense and least expensive Product list (name and unit price).

Select product\_name,unit\_price from products order by unit\_price desc limit 5;

Select product\_name,unit\_price from products order by unit\_price asc limit 5;

1. Write a MySQL query to get Product list (id, name, unit price) where current products cost less than 20 rupees.

Select product\_id,product\_name,unit\_price from products where qualification=’continued’ and unit\_price<20;

**Task-2:**

1. Create a table name departments with primary key column(department\_id).
2. Columns in the table should be department\_id, department\_name & location\_id.

Hint:

CREATE TABLE departments

( department\_id INTEGER PRIMARY KEY

, department\_name VARCHAR(30)

, location\_id INTEGER

) ;

1. Create another table with name employees with a foreign key.

Hint:

CREATE TABLE employees

( employee\_id INTEGER

, first\_name VARCHAR(20)

, last\_name VARCHAR(25)

, email VARCHAR(25)

, phone\_number VARCHAR(20)

, hire\_date DATE

, job\_id VARCHAR(10)

, salary INTEGER

, commission\_pct INTEGER

, manager\_id INTEGER

, department\_id INTEGER

, constraint pk\_emp primary key (employee\_id)

, constraint fk\_deptno foreign key (department\_id) references departments(department\_id)

) ;

1. **Insert 16 Records into departments Table.**

insert into department(department\_name, location\_id)values('HR',101),

('Finance',102),('IT',103),('Marketing',104),('Sales',105),('Customer Support',106),

('DevOps Engineer',107),('Data Analyst',108),('Training',109),('Testing',110),

('Research & Devlopment',111),('Admin',112),('Quality Assurance',113),('Security',114),

('Developer',115),('Logistics',116);

1. **Insert 20 Records into employees Table.**

insert into employees(employee\_id,first\_name,last\_name,email,phone\_number,hire\_date,

job\_id,salary,commission\_pct,manager\_id,department\_id)values

(101,'Nikith','Ram','nikithram2@gmail.com','5627398126', '2024-06-27','SAL\_DEP',50000,2,201,5),

(102,'Sinivasa','Rao','srinjvasrao3@gmail.com','6953269838', '2025-01-20','IT\_DEP',25000,1,202,3),

(103,'Hema','Nisitha','hemanisitha9@gmail.com','7029318739', '2024-07-22','HR\_DEP',60000,2,203,1),

(104,'Ajay','Kumar','ajaykumar2@gmail.com','7349302753', '2021-07-19','LOG\_DEP',100000,5,204,16),

(105,'Manvitha','Sri','manvithasri7@gmail.com','7643687927', '2022-09-05','DEVOPS\_DEP',80000,4,205,7),

(106,'Rachin','Ravindra','rachinravindra8@gmail.com','7654912819', '2020-11-30','TESTING\_DEP',150000,10,206,10),

(107,'Rama','Krishna','ramakrishna23@gmail.com','7705681672', '2017-08-25','TRAINING\_DEP',250000,15,207,9),

(108,'Anushka','Sharma','anushkasharma93@gmail.com','7798342164', '2019-12-12','R&D\_DEP',125000,12,208,11),

(109,'Hapsa','Kausar','hapsakausar56@gmail.com','7842945167', '2023-01-09','MAR\_DEP',75000,4,209,4),

(110,'Venu','Madhav','venumadhav64@gmail.com','7953416729', '2016-05-18','CUST\_DEP',100000,8,210,6),

(111,'Lahari','Priya','laharipriya39@gmail.com','8609234019', '2021-02-14','DATA\_ANALYST\_DEP',150000,11,211,8),

(112,'Reyansh','Sai','reyanshsai02@gmail.com','8623884923', '2020-04-27','ADMIN\_DEP',90000,7,212,12),

(113,'Chitra','Lahari','chitralahari31@gmail.com','9623985377', '2023-12-14','IT\_DEP',60000,3,213,3),

(114,'Rama','Devi','ramadevi94@gmail.com','8623985279', '2023-08-14','Q&A\_DEP',50000,1,214,13),

(115,'Vinay','Varma','vinayvarma83@gmail.com','9183294576', '2022-12-11','SECU\_DEP',20000,0,215,14),

(116,'Bhavya','Sri','bhavyasri20@gmail.com','9234835924', '2020-10-05','CUST\_DEP',90000,7,216,6),

(117,'Jai','Karthikeya','jaikarthikeya11@gmail.com','9550516723', '2019-11-23','DEV\_DEP',150000,9,217,15),

(118,'Reetu','Varma','reetuvarma07@gmail.com','9653945928', '2023-07-17','HR\_DEP',55000,3,218,1),

(119,'Uday','Kishore','udaykishore28@gmail.com','9823761954', '2022-06-06','R&D\_DEP',250000,15,219,11),

(120,'Sai','Charan','saicharan062@gmail.com','9959867217', '2023-10-19','ADMIN\_DEP',65000,2,220,12);

Queries:

1. **Select employees first name, last name, job\_id and salary whose first name starts with alphabet S.**

select first\_name,last\_name,job\_id,salary from employees where first\_name like 'S%';

1. **Write a query to select employee with the highest salary**.

select first\_name,last\_name,salary from employees order by salary desc limit 1;

1. **Select employee with the second highest salary**

select first\_name,last\_name,salary from employees order by salary desc limit 1 offset 1 ;

1. **Fetch employees with 2nd or 3rd highest salary**.

select first\_name,last\_name,salary from employees order by salary desc limit 1 offset 2 ;

1. **Write a query to select employees and their corresponding managers and their salaries.**

Now, this is a classic example of **SELF JOIN** in SQL exercises. Also, use the **CONCAT** function to concatenate the first name and last name of each employee and manager.

select concat(first\_name,' ',last\_name) as emp\_name, manager\_id,salary from employees;

1. **Write a query to show count of employees under each manager in descending order.**

select manager\_id,count(\*) as employee\_count from employees group by manager\_id order by employee\_count asc;

1. **Find the count of employees in each department**.

select department\_name,count(\*) as employee\_count from employees

join department on employees.department\_id=department.department\_id

group by department\_name order by employee\_count desc;

1. **Get the count of employees hired year wise.**

select year(hire\_date) as hire\_year,count(\*) as employee\_count from employees

join department on employees.department\_id=department.department\_id

group by hire\_year order by employee\_count desc;

1. **Find the salary range of employees**.
2. **Write a query to divide people into three groups based on their salaries.**

select first\_name,last\_name,salary,

case when salary>170000 then 'High'

when salary>125000 then 'Medium'

else 'Low'

end as salary\_group

from employees;

1. **Select the employees whose first\_name contains “an”**.

select \* from employees where first\_name like 'an%';

1. **Select employee first name and the corresponding phone number in the format (\_ \_ \_)-(\_ \_ \_)-(\_ \_ \_ \_).**
2. **Find the employees who joined in August, 1994.**

select \* from employees where month(hire\_date)=8 and year(hire\_date)=1994;

1. **Write an SQL query to display employees who earn more than the average salary in that company**.

select \* from employees where salary>125000 and salary<170000;

1. **Find the maximum salary from each department.**

SELECT department\_id, MAX(salary) AS max\_salary FROM employees GROUP BY department\_id;

1. **Write a SQL query to display the 5 least earning employees**.

select \* from employees order by salary asc limit 5;

1. **Find the employees hired in the 80s.**

select \* from employees where year(hire\_date) between 1980 and 1989;

1. **Display the employees first name and the name in reverse order**.

select first\_name,reverse(first\_name) as reverse\_name from employees;

1. **Find the employees who joined the company after 15th of the month.**

select \* from employees where day(hire\_date)>15;

1. **Display the managers and the reporting employees who work in different departments**.