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.

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.
2. Write a MySQL query to get current Product list (Product ID and name).
3. Write a MySQL query to get discontinued Product list (Product ID and name).
4. Write a MySQL query to get most expense and least expensive Product list (name and unit price).
5. Write a MySQL query to get Product list (id, name, unit price) where current products cost less than 20 rupees.

Answer:-

```
create table suppliers (

   supplierid int primary key,

   suppliername varchar(50)

);

create table categories (

   categoryid int primary key,

   categoryname varchar(50)

);

create table products (

   productid int primary key,

   productname varchar(50) unique,

   supplierid int,

   categoryid int,

   quantityperunit varchar(50),
    unitprice decimal,
    unitsinstock int,
    unitsonorder int,
    reorderlevel int,
    discontinued boolean,
    foreign key (supplierid) references suppliers(supplierid),
    foreign key (categoryid) references categories(categoryid)
);
insert into suppliers (supplierid, suppliername) values
(1, 'food shop'),
(2, 'fruit vendor'),
(3, 'leela botique');
insert into categories (categoryid, categoryname) values
(1, 'food'),
(2, 'fruits'),
(3, 'clothing');
insert into products (productid,productname, supplierid, categoryid, quantityperunit, unitprice, unitsinstock, unitsonorder, reorderlevel, discontinued) values
(001,'bread', 1, 1, '1 loaf', 25.00, 20, 0, 5, false),
(002,'apples', 2, 2, '1 kg', 80.00, 50, 0, 10, true),
(003,'tops', 3, 3, '1 piece', 200.00, 10, 2, 3, false),
(004,'bun', 1, 1, '1 piece', 10.00, 30, 0, 10, true),
(005,'bananas', 2, 2, '1 kg', 40.00, 40, 0, 15, false),
(006,'shirt', 3, 3, '1 piece', 300.00, 25, 5, 8, false);
```

```
select * from products;

select productname, quantityperunit from products;

select productid, productname from products where discontinued = false;

select productid, productname from products where discontinued = true;

select productname, unitprice from products where unitprice = (select max(unitprice) from products)

union

select productname, unitprice from products where unitprice = (select min(unitprice) from products);

select productid, productname, unitprice from products where unitprice < 20 and discontinued = false;
```

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
      ) ;
   ```

3. 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)
  ) ;
```

4. **Insert 16 Records into departments Table.**
5. **Insert 20 Records into employees Table.**

   Queries:

   1. **Select employees first name, last name, job_id and salary whose first name starts with alphabet S.**
   2. **Write a query to select employee with the highest salary**.
   3. **Select employee with the second highest salary**
   4. **Fetch employees with 2nd or 3rd highest salary**.
   5. **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.
   6. **Write a query to show count of employees under each manager in descending order.**
   7. **Find the count of employees in each department**.
   8. **Get the count of employees hired year wise.**
   9. **Find the salary range of employees**.
   10. **Write a query to divide people into three groups based on their salaries.**
   11. **Select the employees whose first_name contains "an"**.
   12. **Select employee first name and the corresponding phone number in the format (_ _ _)-(_ _ _)-(_ _ _ _).**
   13. **Find the employees who joined in August, 1994.**
   14. **Write an SQL query to display employees who earn more than the average salary in that company**.
   15. **Find the maximum salary from each department.**
   16. **Write a SQL query to display the 5 least earning employees**.
   17. **Find the employees hired in the 80s.**
   18. **Display the employees first name and the name in reverse order**.
   19. **Find the employees who joined the company after 15th of the month.**

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

Answer:

```
create table departments (
   department_id int primary key,
   department_name varchar(40),
   location_id int
);
create table employees (
   employee_id int primary key,
   first_name varchar(30),
   last_name varchar(35),
   email varchar(35),
   phone_number varchar(10),
   hire_date date,
   job_id varchar(10),
   salary int,
   commission_pct int,
   manager_id int,
   department_id int,
   constraint fk_deptno foreign key (department_id) references departments(department_id)
);
insert into departments(department_id, department_name, location_id) values
(001, 'sales', 500001), (002, 'marketing', 500002), (003, 'hr', 500003), (004, 'finance', 500004),
(005, 'it', 500005), (006, 'production', 500006), (007, 'Acountant', 500007), (008, 'customer service', 500008),

(009, 'research', 500009), (010, 'administration', 500010), (011, 'retail', 500011), (012, 'training', 500012),

(013, 'maintenance', 500013), (014, 'security', 500014), (015, 'quality control', 500015), (016, 'policy', 500016);

select * from departments;


insert into employees (employee_id, first_name, last_name, email, phone_number, hire_date, job_id, salary, commission_pct, manager_id, department_id) values

(460, 'leela', 'vathi', 'leela@gmail.com', '9876543210', '2025-01-27', 'dev', 50000, 10, 103, 001),

(461, 'anil', 'reddy', 'anil@gmail.com', '9876543211', '2025-02-15', 'fin', 45000, 5, 103, 002),

(462, 'ramesh', 'rao', 'ramesh@gmail.com', '9876543212', '2024-03-20', 'hr1', 60000, null, null, 003),

(463, 'janaki', 'devi', 'janaki@gmail.com', '9876543213', '2024-04-25', 'fin', 55000, 8, 103, 004),

(464, 'venkat', 'naidu', 'venkat@gmail.com', '9876543214', '2024-05-30', 'it1', 70000, null, 103, 005),

(465, 'lakshmi', 'priya', 'lakshmi@gmail.com', '9876543215', '2024-06-05', 'prod1', 48000, 3, 103, 006),

(466, 'narayana', 'murthy', 'narayana@gmail.com', '9876543216', '2023-07-10', 'log1', 42000, null, 103, 007),

(467, 'saritha', 'devi', 'saritha@gmail.com', '9876543217', '2025-08-15', 'cust1', 52000, 6, 103, 008),

(468, 'gopal', 'krishna', 'gopal@gmail.com', '9876543218', '2024-09-20', 'res1', 65000, null, 103, 009),

(469, 'rani', 'kumari', 'rani@gmail.com', '9876543219', '2023-10-25', 'adm1', 58000, 7, 103, 010),

(470, 'mohan', 'rao', 'mohan@gmail.com', '9876543220', '2020-11-30', 'ret1', 49000, null, 103, 011),
(471, 'suma', 'devi', 'suma@gmail.com', '9876543221', '2021-12-05', 'train1', 43000, 4, 103,
012),

(472, 'kiran', 'kumar', 'kiran@gmauil.com', '9876543222', '2022-01-10', 'main1', 53000, null,
103, 013),

(473, 'geetha', 'rao', 'geetha@gmail.com', '9876543223', '2019-02-15', 'sec1', 68000, 9, 103,
014),

(474, 'rajesh', 'naidu', 'rajesh@gmail.com', '9876543224', '2023-03-20', 'qc1', 62000, null, 103,
015),

(475, 'sravani', 'priya', 'sravani@gmail.com', '9876543225', '2020-04-25', 'legal1', 56000, 2, 103,
016),

(476, 'santhosh', 'kumar', 'santhosh@gmail.com', '987654226', '2021-08-01', 'sales2', 51000,
11,103, 001),

(477, 'anusha', 'reddy', 'anusha@gmail.com', '9876543227', '2022-09-02', 'mktg2', 46000, 1,
103, 002),

(478, 'madhu', 'rao', 'madhu@gmail.com', '9876543228', '2023-10-03', 'hr2', 61000, null, 103,
003),

(479, 'swathi', 'devi', 'swathi@gmail.com', '9876543229', '2019-05-04', 'fin2', 57000,
12,103,004);

select * from employees;

select first_name, last_name, job_id, salary from employees where first_name like 's%';

select * from employees order by salary desc limit 1;

select * from employees order by salary desc limit 1 offset 1;

select * from employees order by salary desc limit 2 offset 1;

select

  concat(e.first_name, ' ', e.last_name) as employee_name,

  concat(m.first_name, ' ', m.last_name) as manager_name,

  e.salary as employee_salary,
    m.salary as manager_salary
from employees e
left join employees m on e.manager_id = m.employee_id;
select
    concat(m.first_name, ' ', m.last_name) as manager_name,
    count(e.employee_id) as employee_count
from employees e
left join employees m on e.manager_id = m.employee_id
group by m.employee_id
order by employee_count desc;
select
    d.department_name,
    count(e.employee_id) as employee_count
from employees e
join departments d on e.department_id = d.department_id
group by d.department_id;
select
    year(hire_date) as hire_year,
    count(employee_id) as employee_count
from employees
group by year(hire_date);
select
    min(salary) as min_salary,
    max(salary) as max_salary
```

```
from employees;
select
    first_name,
    salary,
    case
        when salary < (select min(salary) + (max(salary)-min(salary))/3 from employees) then 'low'
        when salary < (select min(salary) + 2*(max(salary)-min(salary))/3 from employees) then 'medium'
        else 'high'
    end as salary_group
from employees;
select * from employees where first_name like '%an%';
select
    first_name,
    concat('(', substr(phone_number, 1, 3), ')-', substr(phone_number, 5, 3), '-', substr(phone_number, 9, 4)) as formatted_phone_number
from employees;
select * from employees where month(hire_date) = 8 and year(hire_date) = 1994;
select * from employees where salary > (select avg(salary) from employees);
select
    d.department_name,
    max(e.salary) as max_salary
from employees e
join departments d on e.department_id = d.department_id
group by d.department_id;
```

```sql
select * from employees order by salary asc limit 5;

select * from employees where year(hire_date) between 1980 and 1989;

select first_name, reverse(first_name) as reversed_name from employees;


select * from employees where day(hire_date) > 15;

select
   concat(m.first_name, ' ', m.last_name) as manager_name,
   concat(e.first_name, ' ', e.last_name) as employee_name
from employees e
join employees m on e.manager_id = m.employee_id
where e.department_id <> m.department_id;
```