**Task-2: 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 list (id, name, unit price) where products cost between $15 and 25 rupees.
2. Write a MySQL query to get Product list (name, unit price) of above average price.
3. Write a MySQL query to get Product list (name, unit price) of ten most expensive products.
4. Write a MySQL query to count current and discontinued products.
5. Write a MySQL query to get Product list (name, units on order , units in stock) of stock is less than the quantity on order.

CREATE TABLE products (

ProductID INT PRIMARY KEY AUTO\_INCREMENT,

Product\_name VARCHAR(20),

supplierID INT,

categoryID INT,

QuantityPerUnit int,

UnitPrice DECIMAL(10, 2),

UnitsInStock INT,

UnitsOnOrder INT,

ReorderLevel INT,

Discontinued BOOLEAN,

foreign key (supplierID) references suppliers(supplierID),

foreign key (categoryID) references categories(categoryID)

);

create table suppliers(

supplierID INT primary key auto\_increment,

supplier\_name varchar(20)

);

create table categories(

categoryID INT PRIMARY key auto\_increment,

category\_name varchar(20)

);

insert into suppliers(supplier\_name)

values('flipkart'),

('Amazon'),

('D-mart');

insert into categories(category\_name)

values('electronics'),

('food'),

('grocesories');

INSERT INTO products (Product\_name, supplierID, categoryID, QuantityPerUnit, UnitPrice, UnitsInStock, UnitsOnOrder, ReorderLevel, Discontinued)

VALUES

('mobile', 1, 1, 10, 20, 10, 5, 10, false),

('laptop', 2, 1, 1, 17, 50, 2, 4, false),

('maggie', 3, 2, 50, 21, 100, 5, 1, true),

('happy-happy\_biscuits', 2, 2, 10, 12, 200, 20, 20, false),

('fruits', 2, 3, 10, 22, 500, 20, 12, false),

('vegetables', 3, 3, 20, 25, 800, 20, 20, false),

('tab', 1, 1, 10, 10, 20, 10, 50, true),

('minese', 1, 2, 4, 26, 20, 1, 0, true),

('head-phones', 2, 1, 100, 50, 500, 20, 4, false),

('biryani', 1, 2, 1, 20, 1, 20, 40, false),

('cable', 2, 1, 400, 50, 400, 50, 40, false),

('fast\_food', 2, 2, 1, 10, 1, 20, 10, false);

**Write a MySQL query to get Product list (id, name, unit price) where products cost between $15 and 25 rupees.**

select ProductID, product\_name, unitprice

from products

where unitprice between 15 and 25;

**Write a MySQL query to get Product list (name, unit price) of ten most expensive products.**

select product\_name, unitprice from products

order by unitprice DESC limit 10;

**Write a MySQL query to get Product list (name, units on order , units in stock) of stock is less than the quantity on order.**

select product\_name, UnitsOnOrder, UnitsInStock

from products

where UnitsInStock < UnitsOnOrder;

**Write a MySQL query to get Product list (name, unit price) of above average price.**

select product\_name, UnitPrice

from products

where UnitPrice > (SELECT AVG(UnitPrice) FROM products);

**Write a MySQL query to count current and discontinued products.**

SELECT

(SELECT COUNT(\*) FROM products WHERE discontinued = TRUE) AS discontinued\_count,

(SELECT COUNT(\*) FROM products WHERE discontinued = FALSE) AS current\_count;

**Task -2**

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

select product\_name, quantityperunit from products;

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

select product\_name, unitprice from products where Discontinued = false;

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

select productId, product\_name from products where Discontinued = true;

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

select productID, product\_name, quantityperunit from products where unitprice < 20;

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

(select product\_name, Quantityperunit, 'most\_expensive' as unit\_price from products

order by quantityperunit desc limit 1)

union

(select product\_name,quantityperunit, 'least\_expensive' as unit\_price from products

order by quantityperunit limit 1);