```sql
-- Retail Mart Management Project--

-- Task 1 : Write a query to create a database named SQL basics
CREATE DATABASE SQL_basics;


-- Task 2 : Write a query to select SQL basics
USE SQL_basics;

/* Task 3 : Write a query to create a product table with the fields product code,
product name, price, stock, and category, a customer table with the
fields customer ID, customer name, customer location, and customer phone number, and a
sales table with the fields date, order number,
product code, product name, quantity, and price*/
 -- PRODUCT_TABLE
CREATE TABLE PRODUCT_TABLE
(
p_code VARCHAR (50) NOT NULL,
p_name VARCHAR (50) NOT NULL,
price INT NOT NULL,
stock VARCHAR (100) NOT NULL,
category VARCHAR (100) NOT NULL,
PRIMARY KEY(p_code)
);

 -- CUSTOMER_TABLE
 CREATE TABLE CUSTOMER_TABLE
(
c_id VARCHAR (50) NOT NULL,
c_name VARCHAR (50) NOT NULL,
c_location VARCHAR (50) NOT NULL,
c_phoneno INT NOT NULL,
PRIMARY KEY(c_id)
);

 -- SALES_TABLE
CREATE TABLE SALES_TABLE
(
order_date DATE NOT NULL,
order_no  VARCHAR (50) NOT NULL,
c_id VARCHAR (50) NOT NULL,
c_name VARCHAR (50) NOT NULL,
s_code VARCHAR (50) NOT NULL,
p_name VARCHAR (50) NOT NULL,
qty INT NOT NULL,
price INT NOT NULL,
PRIMARY KEY(order_date)
);


-- Task 4 : Write a query to insert values into the customer, product, and sales table.
 -- PRODUCT_TABLE

INSERT INTO
PRODUCT_TABLE ( p_code, p_name, price, stock, category )
VALUES
("1", "tulip", "198", "5", "perfume"),
("2", "cornoto", "50", "21", "icecream"),
("3", "Pen", "10", "52", "Stationary"),
("4", "Lays", "10", "20", "snacks"),
("5", "mayanoise", "90", "10", "dip"),
("6", "jam", "105", "10", "spread"),
("7", "shampoo", "5", "90", "hair product"),
("8", "axe", "210", "4", "perfume"),
("9", "park avenue", "901", "2", "perfume"),
("10", "wattagirl", "201", "3", "perfume"),
("11", "pencil", "4", "10", "Stationary"),
("12", "sharpener", "5", "90", "Stationary"),
("13", "sketch pen", "30", "10", "Stationary"),
```

```
("14", "tape", "15", "30", "Stationary"),
("15", "paint", "60", "12", "Stationary"),
("16", "chocolate", "25", "50", "snacks"),
("17", "biscuts", "60", "26", "snacks"),
("18", "mango", "100", "21", "fruits"),
("19", "apple", "120", "9", "fruits"),
("20", "kiwi", "140", "4", "fruits"),
("21", "carrot", "35", "12", "vegetable"),
("22", "onion", "22", "38", "vegetable"),
("23", "tomato", "21", "15", "vegetable"),
("24", "serum", "90", "4", "hair product"),
("25", "conditioner", "200", "5", "hair product"),
("26", "oil bottle", "40", "2", "kitchen utensil");


 -- CUSTOMER_TABLE
INSERT INTO
CUSTOMER_TABLE( c_id, c_name, c_location, c_phoneno )
VALUES
("1111" , "Nisha" , "kerala" , "8392320"),
("1212" , "Oliver" , "kerala" , "4353891"),
("1216" , "Nila" , "delhi" , "3323242"),
("1246" , "Vignesh" , "chennai" , "1111212"),
("1313" , "shiny" , "Maharastra" , "5454543"),
("1910" , "Mohan" , "mumbai" , "9023941"),
("2123" , "Biyush" , "Bombay" , "1253358"),
("3452" , "Alexander" , "West Bengal" , "1212134"),
("3921" , "Mukesh" , "Manipur" , "4232321"),
("5334" , "Christy" , "pakistan" , "2311111"),
("9021" , "Rithika" , "Kashmir" , "1121344"),
("9212" , "Jessica" , "banglore" , "1233435"),
("9875" , "Stephen" , "chennai" , "1212133");


-- SALES_TABLE
INSERT INTO
SALES_TABLE( order_date, order_no, c_id, c_name, s_code, p_name, qty, price)
VALUES
("2016-07-24", "HM06" ,"9212" ,"Jessica" ,"11" ,"pencil" ,"3", "30"),
("2016-10-19", "HM09" ,"3921" ,"Mukesh" ,"17" ,"biscuits" ,"10", "600"),
("2016-10-30", "HM10" ,"9875" ,"Stephen" ,"2" ,"cornoto" ,"10", "500"),
("2018-12-04", "HM03" ,"1212" ,"Oliver" ,"20" ,"kiwi" ,"3", "420"),
("2018-02-05", "HM05" ,"1910" ,"Mohan" ,"20" ,"kiwi" ,"2", "280"),
("2018-09-20", "HM08" ,"5334" ,"Chirsty" ,"16" ,"chocolate" ,"2", "50"),
("2019-11-01", "HM07" ,"1246" ,"Vignesh" ,"19" ,"apple" ,"5", "600"),
("2019-03-15", "HM01" ,"1910" ,"Mohan" ,"5" ,"mayanoise" ,"4", "360"),
("2021-10-02", "HM04" ,"1111" ,"Nisha" ,"25" ,"conditioner" ,"5", "1000"),
("2021-12-02", "HM02" ,"2123" ,"Biyush" ,"3" ,"Pen" ,"2", "20");

-- Task 5 : Write a query to add new columns, such as serial number and categories, to
the sales table.

ALTER TABLE SALES_TABLE
ADD SERIAL_NUMBER VARCHAR(50) NOT NULL,
ADD CATEGORIES VARCHAR(100) NOT NULL;


-- Task 6: Write a query to change the stock field type to varchar in the product table.

ALTER TABLE PRODUCT_TABLE
MODIFY STOCK VARCHAR(50);


-- Task 7: Write a query to change the table name from customer to customer details.

ALTER TABLE CUSTOMER_TABLE
RENAME TO CUSTOMER_DETAILS;

-- Task 8 : Write a query to drop the sl. no. and categories columns from the sales
```

```
table.

ALTER TABLE SALES_TABLE
DROP COLUMN SERIAL_NUMBER;

ALTER TABLE SALES_TABLE
DROP COLUMN CATEGORIES;

-- Task 9 : Write a query to display the order ID, customer ID, order date, price, and quantity columns of the sales table.

SELECT order_no, order_date, price, qty
FROM SALES_TABLE;


-- Task 10 : Write a query to display the details where the category is stationary from the product table

SELECT * FROM PRODUCT_TABLE
WHERE CATEGORY = "STATIONARY";

-- Task 11 :  Write a query to display the unique category from the product table

SELECT  DISTINCT CATEGORY
FROM PRODUCT_TABLE
ORDER BY CATEGORY;


-- Task 12 :  Write a query to display the details of the sales from the sales table where quantity is greater than 2 and the price is less than 500.

SELECT * FROM PRODUCT_TABLE
WHERE stock > 2
AND price < 500;


-- Task 13 :  Write a query to display every customer whose name ends with an 'a'

SELECT * FROM CUSTOMER_DETAILS
WHERE c_name LIKE '%a';


-- Task 14 :  Write a query to display the product details in descending order of price.

SELECT * FROM PRODUCT_TABLE
ORDER BY price DESC;


-- Task 15 :  Write a query to display the product code and category from categories that have two or more products.

SELECT p_code,category FROM PRODUCT_TABLE GROUP BY category HAVING
COUNT(category)>=2;


-- Task 16 :  Write a query to combine the sales and product tables based on the order number and customer's name including duplicated rows.

SELECT order_no,c_name FROM SALES_TABLE LEFT JOIN PRODUCT_TABLE ON
SALES_TABLE.s_code = PRODUCT_TABLE.p_code
UNION ALL
SELECT order_no,c_name FROM SALES_TABLE RIGHT JOIN PRODUCT_TABLE ON
SALES_TABLE.s_code = PRODUCT_TABLE.p_code;
```