Retail Mart Management

SQL project1 - PC BA DEC 2022 Cohort 1

3/22/2023

Manohari Wijesooriya

Contents

1. Introduction 2
2. Objective 2
3. Sequence of Tasks 2

1. Introduction

A data analyst of a retail shop, Happy Mart, wants to store the product details, the customer details, and the order details to provide unparalleled insights about customer behavior and product stock details daily.

2. Objective

The design of the database helps to easily evaluate and identify the performance of the shop to increase the daily sales.

3. Sequence of Tasks

3.1. Write a query to create a database named SQL basics.

3.2. Write a query to select the database SQL basics.

SQL code
CREATE DATABASE SQL_basics; USE SQL_basics;

3.3. Write a query to create a product table with fields as product code, product name, price, stock and category, customer table with the fields as customer id, customer name, customer location, and customer phone number and, sales table with the fields as date, order number, product code, product name, quantity, and price.

SQL code
CREATE TABLE product(product_code int, product_name varchar(15), price float, stock int, category varchar(15));
CREATE TABLE customer(customer_id int, customer_name varchar(15), customer_location varchar(20), customer_phone_number int);

```
CREATE TABLE sales(
 date date,
 order_number VARCHAR(4),
 product_code int,
 product_name varchar(15),
 quantity int,
 price float,
 customer_id
);
```

3.4. Write a query to insert values into the tables.

SQL code

```
INSERT INTO
SQL_basics.product(product_code, product_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", "biscuits", "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");
```

```
INSERT INTO
SQL_basics.sales(date, order_number, product_code, product_name, quantity, price, customer_id)
VALUES
(STR_TO_DATE("24-07-2016","%d-%m-%Y"),"HM06","11", "pencil", "3", "30" ,"9212"),
(STR_TO_DATE("19-10-2016","%d-%m-%Y"),"HM09","17", "biscuits", "10", "600" ,"3921"),
(STR_TO_DATE("30-10-2016","%d-%m-%Y"),"HM10","2", "cornoto", "10", "500","9875"),
(STR_TO_DATE("12-04-2018","%d-%m-%Y"),"HM03","20", "kiwi", "3", "420","1212"),
(STR_TO_DATE("02-05-2018","%d-%m-%Y"),"HM05","20", "kiwi", "2", "280","1910"),
(STR_TO_DATE("20-09-2018","%d-%m-%Y"),"HM08","16", "chocolate", "2", "50","5334"),
```

```
(STR_TO_DATE("11-01-2019","%d-%m-%Y"),"HM07","19", "apple", "5", "600","1246"),
(STR_TO_DATE("15-03-2019","%d-%m-%Y"),"HM01","5", "mayanoise", "4", "360","1910"),
(STR_TO_DATE("10-02-2021","%d-%m-%Y"),"HM04","25", "conditioner", "5", "1000","1111"),
(STR_TO_DATE("12-02-2021","%d-%m-%Y"),"HM02","3", "Pen", "2", "20","2123");
```

```
INSERT INTO
SQL_basics.customer(customer_id, customer_name, customer_location, customer_phone_number )
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");
```

3.5. Write a query to add two new columns such as S_no and categories to the sales table.

SQL code

```
ALTER TABLE SQL_basics.sales
ADD s_no int,
ADD categories varchar(15);
```

3.6. Write a query to change the column type of stock in the product table to varchar.

SQL code

```
ALTER TABLE product
  MODIFY stock varchar(4);
```

3.7. Write a query to change the table name from customer-to-customer details.

SQL code

```
ALTER TABLE customer
RENAME TO customer_details;
```

3.8. Write a query to drop the columns S_no and categories from the sales table.

SQL code

```
ALTER TABLE sales
DROP COLUMN s_no;

ALTER TABLE sales
DROP COLUMN categories;
```

3.9. Write a query to display order id, customer id, order date, price, and quantity from the sales table.

SQL code

```
SELECT order_number as order_id, customer_id, date as order_date, price,  quantity
from sales;
```

Results

order_id	customer_	order_date	price	quantity
HM06	9212	2016-07-24	30	3
HM09	3921	2016-10-19	600	10
HM10	9875	2016-10-30	500	10
HM03	1212	2018-04-12	420	3
HM05	1910	2018-05-02	280	2
HM08	5334	2018-09-20	50	2
HM07	1246	2019-01-11	600	5
HM01	1910	2019-03-15	360	4
HM04	1111	2021-02-10	1000	5
HM02	2123	2021-02-12	20	2

3.10. Write a query to display all the details in the product table if the category is stationary.

SQL code

```
SELECT * from product WHERE category = "stationary";
```

Results

product_code	product_n	price	stock	category
3	Pen	10	52	Stationary
11	pencil	4	10	Stationary
12	sharpener	5	90	Stationary
13	sketch per	30	10	Stationary
14	tape	15	30	Stationary
15	paint	60	12	Stationary

3.11. Write a query to display a unique category from the product table.

SQL code

```
SELECT DISTINCT
        category from product;
```

Results

category
perfume
icecream
Stationary
snacks
dip
spread
hair product
fruits
vegetable
kitchen utensil

3.12. Write a query to display the sales details if quantity is greater than 2 and price is lesser than 500 from the sales table.

SQL code

```
SELECT *
        from sales
        where quantity > 2 and price < 500;
```

Results

date	order_nur	product_c	product_n	quantity	price	customer_id
2016-07-24	HM06	11	pencil	3	30	9212
2018-04-12	HM03	20	kiwi	3	420	1212
2019-03-15	HM01	5	mayanois	4	360	1910

3.13. Write a query to display the customer's name if the name ends with a.

SQL code

```
SELECT customer_name
        FROM customer_details
        WHERE customer_name like "%a";
```

Results

customer_name
Nisha
Nila
Rithika
Jessica

3.14. Write a query to display the product details in descending order of the price.

SQL code

```
SELECT * FROM product ORDER BY price DESC;
```

3.15. Write a query to display the product code and category from similar categories that are greater than or equal to 2.

SQL code

```
SELECT category, count(product_code) as number_of_products
FROM product GROUP BY category
HAVING number_of_products >= 2;
```

Results

category	number_of_products
perfume	4
Stationary	6
snacks	3
hair product	3
fruits	3
vegetable	3

3.16. Write a query to display the order number and the customer name to combine the results of the order and the customer tables including duplicate rows.

SQL code

```
SELECT order_number, customer_name
from sales left join customer_details
ON sales.customer_id = customer_details.customer_id
ORDER BY customer_name;
```

Results

order_number	customer_name
HM02	Biyush
HM08	Christy
HM06	Jessica
HM05	Mohan
HM01	Mohan
HM09	Mukesh
HM04	Nisha
HM03	Oliver
HM10	Stephen
HM07	Vignesh