

Name : Sarthak Shandilya

Coding Challenge

Paper Solved :- Coding Challenge 6

SQL Tables:

1. customers table:

- customer_id (Primary Key)
- name
- email
- password

2. products table:

- product_id (Primary Key)
- name
- price
- description
- stockQuantity

3. cart table:

- cart_id (Primary Key)
- customer_id (Foreign Key)
- product_id (Foreign Key)
- quantity

4. orders table:

- order_id (Primary Key)
- customer_id (Foreign Key)
- order_date
- total_price
- shipping_address

5. order_items table (to store order details):

- order_item_id (Primary Key)
- order_id (Foreign Key)
- product_id (Foreign Key)
- quantity

Created Tables:

```
mysql> create table customers(  
  -> customer_id int primary key,  
  -> name varchar(30),  
  -> email varchar(40),  
  -> password varchar(20));  
Query OK, 0 rows affected (0.09 sec)  
  
mysql> create table products(  
  -> product_id int primary key,  
  -> name varchar(20),  
  -> price decimal,  
  -> description text,  
  -> stockQuantity int);  
Query OK, 0 rows affected (0.05 sec)  
  
mysql> create table cart(  
  -> cart_id int primary key,  
  -> customer_id int,  
  -> product_id int,  
  -> quantity int,  
  -> foreign key(customer_id) references customers(customer_id) on delete  
cascade on update cascade,  
  -> foreign key(product_id) references products(product_id) on delete cascade on update cascade);  
Query OK, 0 rows affected (0.07 sec)
```

```
mysql> create table orders(  
  -> order_id int primary key,  
  -> customer_id int,  
  -> order_date date,  
  -> total_price decimal,  
  -> shipping_address text,  
  -> foreign key(customer_id) references customers(customer_id) on delete  
cascade on update cascade);  
Query OK, 0 rows affected (0.07 sec)  
  
mysql> create table order_items(  
  -> order_item_id int primary key,  
  -> order_id int,  
  -> product_id int,  
  -> quantity int,  
  -> foreign key(order_id) references orders(order_id) on delete cascade on update cascade,  
  -> foreign key(product_id) references products(product_id) on delete cascade on update cascade);  
Query OK, 0 rows affected (0.07 sec)
```

Records Inserted:

Products :-

```
mysql> insert into products values
-> (1, "Laptop",800.00 ,"High-performance laptop" ,10);
Query OK, 1 row affected (0.02 sec)

mysql> insert into products values
-> (2, "Smartphone" ,600.00,"Latest smartphone",15),
-> (3, "Tablet",300.00, "Portable tablet",20),
-> (4, "Headphones",150.00 ,"Noise-canceling" ,30),
-> (5, "TV", 900.00,"4K Smart TV" ,5),
-> (6, "Coffee Maker",50.00 ,"Automatic coffee maker" ,25),
-> (7, "Refrigerator",700.00 ,"Energy-efficient" ,10),
-> (8, "Microwave",80.00 ,"Oven Countertop microwave" ,15),
-> (9, "Blender" ,70.00,"High-speed blender" ,20),
-> (10, "Vacuum Cleaner",120.00 ,"Bagless vacuum cleaner" ,10);
Query OK, 9 rows affected (0.02 sec)
Records: 9 Duplicates: 0 Warnings: 0
```

Customers:-

```
mysql> insert into customers values
-> (1, "John Doe" ,"johndoe@example.com" ,"123 Main St, City"),
-> (2, "Jane Smith" ,"janesmith@example.com" ,"456 Elm St, Town"),
-> (3, "Robert Johnson" ,"robert@example.com" ,"789 Oak St, Village"),
-> (4, "Sarah Brown" ,"sarah@example.com" ,"101 Pine St, Suburb"),
-> (5, "David Lee" ,"david@example.com" ,"234 Cedar St, District"),
-> (6, "Laura Hall" ,"laura@example.com" ,"567 Birch St, County"),
-> (7, "Michael Davis" ,"michael@example.com" ,"890 Maple St, State"),
-> (8, "Emma Wilson" ,"emma@example.com" ,"321 Redwood St, Country"),
-> (9, "William Taylor" ,"william@example.com" ,"432 Spruce St, Province"),
-> (10, "Olivia Adams" ,"olivia@example.com" ,"765 Fir St, Territory");
Query OK, 10 rows affected (0.01 sec)
Records: 10 Duplicates: 0 Warnings: 0
```

Orders:-

```
mysql> insert into orders values
-> (1, 1, '2023-01-05' ,1200.00,"123 Main St, City"),
-> (2, 2, '2023-02-10' ,900.00,"456 Elm St, Town"),
-> (3, 3, '2023-03-15' ,300.00,"789 Oak St, Village");
Query OK, 3 rows affected (0.02 sec)
Records: 3 Duplicates: 0 Warnings: 0
```

```
mysql> insert into orders values
-> (4,4,'2023-04-20',150.00,"101 Pine St, Suburb"),
-> (5,5,'2023-05-25',1800.00,"234 Cedar St, District"),
-> (6,6,'2023-06-30',400.00,"567 Birch St, County"),
-> (7,7,'2023-07-05',700.00,"890 Maple St, State"),
-> (8,8,'2023-08-10',160.00,"321 Redwood St, Country"),
-> (9,9,'2023-09-15',140.00,"432 Spruce St, Province"),
-> (10,10,'2023-10-20',1400.00,"765 Fir St, Territory");
Query OK, 7 rows affected (0.02 sec)
Records: 7 Duplicates: 0 Warnings: 0
```

Order_items:-

```
mysql> insert into order_items values
-> (1,1,1,2,1600.00),
-> (2,1,3,1,300.00),
-> (3,2,2,3,1800.00),
-> (4,3,5,2,1800.00),
-> (5,4,4,4,600.00),
-> (6,4,6,1,50.00),
-> (7,5,1,1,800.00),
-> (8,5,2,2,1200.00),
-> (9,6,10,2,240.00),
-> (10,6,9,3,210.00);
Query OK, 10 rows affected (0.02 sec)
Records: 10 Duplicates: 0 Warnings: 0
```

Cart:-

```
mysql> insert into cart values
-> (1,1,1,2),
-> (2,1,3,1),
-> (3,2,2,3),
-> (4,3,4,4),
-> (5,3,5,2),
-> (6,4,6,1),
-> (7,5,1,1),
-> (8,6,10,2),
-> (9,6,9,3),
-> (10,7,7,2);
Query OK, 10 rows affected (0.02 sec)
Records: 10 Duplicates: 0 Warnings: 0
```

Descriptions of my tables:-

Customers:

```
mysql> desc customers  
-> ;
```

Field	Type	Null	Key	Default	Extra
customer_id	int	NO	PRI	NULL	
name	varchar(30)	YES		NULL	
email	varchar(40)	YES		NULL	
address	text	YES		NULL	

4 rows in set (0.00 sec)

Products:

NOTE: I've later on added a category section in product table as it was required in a question.

I used the query as below:

```
ALTER TABLE products ADD COLUMN category varchar(20);
```

```
mysql> desc products;
```

Field	Type	Null	Key	Default	Extra
product_id	int	NO	PRI	NULL	
name	varchar(20)	YES		NULL	
price	decimal(10,2)	YES		NULL	
description	text	YES		NULL	
stockQuantity	int	YES		NULL	
category	varchar(20)	YES		NULL	

6 rows in set (0.00 sec)

Orders:

```
mysql> desc orders;
```

Field	Type	Null	Key	Default	Extra
order_id	int	NO	PRI	NULL	
customer_id	int	YES	MUL	NULL	
order_date	date	YES		NULL	
total_price	decimal(10,2)	YES		NULL	
shipping_address	text	YES		NULL	

5 rows in set (0.00 sec)

Order_items:

```
mysql> desc order_items;
```

Field	Type	Null	Key	Default	Extra
order_item_id	int	NO	PRI	NULL	
order_id	int	YES	MUL	NULL	
product_id	int	YES	MUL	NULL	
quantity	int	YES		NULL	
total_amount	decimal(10,2)	YES		NULL	

5 rows in set (0.00 sec)

Cart:

```
mysql> desc cart;
```

Field	Type	Null	Key	Default	Extra
cart_id	int	NO	PRI	NULL	
customer_id	int	YES	MUL	NULL	
product_id	int	YES	MUL	NULL	
quantity	int	YES		NULL	

4 rows in set (0.00 sec)

Questions and Solutions:-

1. Update refrigerator product price to 800.

```
mysql> update products set price=800.00 where name="Refrigerator";
Query OK, 1 row affected (0.01 sec)
Rows matched: 1  Changed: 1  Warnings: 0

mysql> select * from products;
```

product_id	name	price	description	stockQuantity
1	Laptop	800.00	High-performance laptop	10
2	Smartphone	600.00	Latest smartphone	15
3	Tablet	300.00	Portable tablet	20
4	Headphones	150.00	Noise-canceling	30
5	TV	900.00	4K Smart TV	5
6	Coffee Maker	50.00	Automatic coffee maker	
25				
7	Refrigerator	800.00	Energy-efficient	10
8	Microwave	80.00	Oven Countertop microwave	15
9	Blender	70.00	High-speed blender	20
10	Vacuum Cleaner	120.00	Bagless vacuum cleaner	10

```
10 rows in set (0.01 sec)
```

2. Remove all cart items for a specific customer.

```
mysql> delimiter @@
mysql> create procedure deleteFromCart(in var int)
-> begin
-> delete from cart where customer_id=var;
-> end@@
Query OK, 0 rows affected (0.03 sec)

mysql> delimiter ;
mysql> set @cust_id=6;
Query OK, 0 rows affected (0.01 sec)

mysql> call deleteFromCart(@cust_id);
Query OK, 2 rows affected (0.02 sec)
```

Cart item for customer with customer_id = 6 deleted

```
mysql> select * from cart;
```

cart_id	customer_id	product_id	quantity
1	1	1	2
2	1	3	1
3	2	2	3
4	3	4	4
5	3	5	2
6	4	6	1
7	5	1	1
10	7	7	2

```
8 rows in set (0.00 sec)
```

3. Retrieve Products Priced Below \$100.

```
mysql> select * from products where price<100;
```

product_id	name	price	description	stockQuantity
6	Coffee Maker	50.00	Automatic coffee maker	25
8	Microwave	80.00	Oven Countertop microwave	15
9	Blender	70.00	High-speed blender	20

```
3 rows in set (0.01 sec)
```

4. Find Products with Stock Quantity Greater Than 5.

```
mysql> select * from products where stockQuantity > 5;
```

product_id	name	price	description	stockQuantity
1	Laptop	800.00	High-performance laptop	10
2	Smartphone	600.00	Latest smartphone	15
3	Tablet	300.00	Portable tablet	20
4	Headphones	150.00	Noise-canceling	30
6	Coffee Maker	50.00	Automatic coffee maker	25
7	Refrigerator	800.00	Energy-efficient	10
8	Microwave	80.00	Oven Countertop microwave	15
9	Blender	70.00	High-speed blender	20
10	Vacuum Cleaner	120.00	Bagless vacuum cleaner	10

```
9 rows in set (0.00 sec)
```


5. Retrieve Orders with Total Amount Between \$500 and \$1000.

```
mysql> select * from orders where total_price between 500 and 1000;
+-----+-----+-----+-----+-----+
| order_id | customer_id | order_date | total_price | shipping_address |
+-----+-----+-----+-----+-----+
| 2 | 2 | 2023-02-10 | 900.00 | 456 Elm St, Town |
| 7 | 7 | 2023-07-05 | 700.00 | 890 Maple St, State |
+-----+-----+-----+-----+-----+
2 rows in set (0.01 sec)
```

6. Find Products which name end with letter 'r'.

```
mysql> select * from products where name like '%r';
+-----+-----+-----+-----+-----+
| product_id | name | price | description | stockQuantity |
+-----+-----+-----+-----+-----+
| 6 | Coffee Maker | 50.00 | Automatic coffee maker | 25 |
| 7 | Refrigerator | 800.00 | Energy-efficient | 10 |
| 9 | Blender | 70.00 | High-speed blender | 20 |
| 10 | Vacuum Cleaner | 120.00 | Bagless vacuum cleaner | 10 |
+-----+-----+-----+-----+-----+
4 rows in set (0.00 sec)
```

7. Retrieve Cart Items for Customer 5.

```
mysql> select p.product_id,p.name,p.price,p.description,c.cart_id,c.customer_id,c.quantity from products p
-> join cart c on p.product_id=c.product_id
-> where c.customer_id=5;
+-----+-----+-----+-----+-----+-----+
| product_id | name | price | description | cart_id | customer_id | quantity |
+-----+-----+-----+-----+-----+-----+
| 1 | Laptop | 800.00 | High-performance laptop | 7 | 5 | 1 |
+-----+-----+-----+-----+-----+-----+
1 row in set (0.00 sec)
```

8. Find Customers Who Placed Orders in 2023.

```
mysql> select c.customer_id,c.name,o.order_id,o.order_date
-> from customers c
-> join orders o on c.customer_id=o.order_id
-> where year(o.order_date)=2023;
```

customer_id	name	order_id	order_date
1	John Doe	1	2023-01-05
2	Jane Smith	2	2023-02-10
3	Robert Johnson	3	2023-03-15
4	Sarah Brown	4	2023-04-20
5	David Lee	5	2023-05-25
6	Laura Hall	6	2023-06-30
7	Michael Davis	7	2023-07-05
8	Emma Wilson	8	2023-08-10
9	William Taylor	9	2023-09-15
10	Olivia Adams	10	2023-10-20

10 rows in set (0.00 sec)

9. Determine the Minimum Stock Quantity for Each Product Category.

```
mysql> select category,min(stockquantity) from products
-> group by category;
```

category	min(stockquantity)
Computers	10
Mobile Phone	15
Tablets	20
Audio	30
Television	5
Kitchen Appliances	10
Home cleaning	10

7 rows in set (0.01 sec)

10. Calculate the Total Amount Spent by Each Customer.

```
mysql> select c.customer_id,c.name,sum(o.total_price) as TotalAmount
-> from customers c
-> join orders o on c.customer_id=o.customer_id
-> group by c.customer_id,c.name;
```

customer_id	name	TotalAmount
1	John Doe	1200.00
2	Jane Smith	900.00
3	Robert Johnson	300.00
4	Sarah Brown	150.00
5	David Lee	1800.00
6	Laura Hall	400.00
7	Michael Davis	700.00
8	Emma Wilson	160.00
9	William Taylor	140.00
10	Olivia Adams	1400.00

10 rows in set (0.00 sec)

11. Find the Average Order Amount for Each Customer.

```
mysql> select c.customer_id,c.name,avg(o.total_price) as AverageAmount
-> from customers c
-> join orders o on c.customer_id=o.customer_id
-> group by c.customer_id,c.name;
```

customer_id	name	AverageAmount
1	John Doe	1200.000000
2	Jane Smith	900.000000
3	Robert Johnson	300.000000
4	Sarah Brown	150.000000
5	David Lee	1800.000000
6	Laura Hall	400.000000
7	Michael Davis	700.000000
8	Emma Wilson	160.000000
9	William Taylor	140.000000
10	Olivia Adams	1400.000000

10 rows in set (0.01 sec)

12. Count the Number of Orders Placed by Each Customer.

```
mysql> select c.customer_id,c.name,count(o.order_id) as NumberOfOrders
-> from customers c
-> join orders o on o.customer_id=c.customer_id
-> group by c.customer_id,c.name;
```

customer_id	name	NumberOfOrders
1	John Doe	1
2	Jane Smith	1
3	Robert Johnson	1
4	Sarah Brown	1
5	David Lee	1
6	Laura Hall	1
7	Michael Davis	1
8	Emma Wilson	1
9	William Taylor	1
10	Olivia Adams	1

10 rows in set (0.01 sec)

13. Find the Maximum Order Amount for Each Customer.

```
mysql> select c.customer_id,c.name,max(o.total_price) as MaximumOrderAmount
-> from customers c
-> join orders o on c.customer_id=o.order_id
-> group by c.customer_id,c.name;
```

customer_id	name	MaximumOrderAmount
1	John Doe	1200.00
2	Jane Smith	900.00
3	Robert Johnson	300.00
4	Sarah Brown	150.00
5	David Lee	1800.00
6	Laura Hall	400.00
7	Michael Davis	700.00
8	Emma Wilson	160.00
9	William Taylor	140.00
10	Olivia Adams	1400.00

10 rows in set (0.00 sec)

14. Get Customers Who Placed Orders Totaling Over \$1000.

```
mysql> select c.customer_id,c.name,sum(o.total_price) as TotalAmount
-> from customers c
-> join orders o on c.customer_id=o.customer_id
-> group by c.customer_id,c.name
-> having TotalAmount > 1000;
```

customer_id	name	TotalAmount
1	John Doe	1200.00
5	David Lee	1800.00
10	Olivia Adams	1400.00

3 rows in set (0.00 sec)

15. Subquery to Find Products Not in the Cart.

```
mysql> select product_id,name from products
-> where product_id not in(select product_id from cart);
```

product_id	name
8	Microwave
9	Blender
10	Vacuum Cleaner

3 rows in set (0.01 sec)

16. Subquery to Find Customers Who Haven't Placed Orders.

```
mysql> select * from customers
      -> where customer_id not in(select customer_id from orders);
+-----+-----+-----+-----+
| customer_id | name      | email           | address           |
+-----+-----+-----+-----+
|          5 | David Lee | david@example.com | 234 Cedar St, District |
+-----+-----+-----+-----+
1 row in set (0.00 sec)
```

17. Subquery to Calculate the Percentage of Total Revenue for a Product.

```
mysql> select p.product_id,p.name,
      -> (select sum(total_amount) from order_items) as totalRevenue,
      -> ((select sum(oi.total_amount) from order_items oi where oi.product_id=p.product_id)/(select sum(total_amount) from order_items))*100 as PercentageOfRevenue
      -> from products p;
+-----+-----+-----+-----+
| product_id | name      | totalRevenue | PercentageOfRevenue |
+-----+-----+-----+-----+
|          1 | Laptop    | 6600.00      | 24.242424           |
|          2 | Smartphone | 6600.00      | 27.272727           |
|          3 | Tablet    | 6600.00      | 4.545455            |
|          4 | Headphones | 6600.00      | 9.090909            |
|          5 | TV        | 6600.00      | 27.272727           |
|          6 | Coffee Maker | 6600.00      | 0.757576            |
|          7 | Refrigerator | 6600.00      | NULL                |
|          8 | Microwave | 6600.00      | NULL                |
|          9 | Blender   | 6600.00      | 3.181818            |
|         10 | Vacuum Cleaner | 6600.00      | 3.636364            |
+-----+-----+-----+-----+
10 rows in set (0.00 sec)
```

18. Subquery to Find Products with Low Stock.

Note:- I've considered a product's stock to be low if it's available stock quantity is less than total average stock quantity

```
mysql> select * from products where product_id in(
-> select product_id from products where stockQuantity < any(
-> select avg(stockQuantity) from products));
```

product_id	name	price	description	stockQuantity	category
1	Laptop	800.00	High-performance laptop	10	Computers
2	Smartphone	600.00	Latest smartphone	15	Mobile Phone
5	TV	900.00	4K Smart TV	5	Television
7	Refrigerator	800.00	Energy-efficient	10	Kitchen Appliances
8	Microwave	80.00	Oven Countertop microwave	15	Kitchen Appliances
10	Vacuum Cleaner	120.00	Bagless vacuum cleaner	10	Home cleaning

6 rows in set (0.00 sec)

19. Subquery to Find Customers Who Placed High-Value Orders.

```
mysql> select * from customers
-> where customer_id in(select customer_id from orders where total_price > any(
-> select avg(total_price) from orders));
```

customer_id	name	email	address
1	John Doe	johndoe@example.com	123 Main St, City
2	Jane Smith	janesmith@example.com	456 Elm St, Town
7	Michael Davis	michael@example.com	890 Maple St, State
10	Olivia Adams	olivia@example.com	765 Fir St, Territory

4 rows in set (0.01 sec)