CODING CHALLENGE

E COMMERCE

NAME: S. NIKHIL SAI

Coding Challenge: E – Commerce

Table Creation:

/* Creating Database*/

use Ecommerce

/*Customers

- Customerid (primary key)
- Name
- Email
- Password */

```
create table customers(
customerid int IDENTITY PRIMARY KEY,
name varchar(20),
email varchar(300),
password varchar(300)
);
```

/* products

- Productid (primary key)
- Name
- Description
- Price
- stockQuantity */

```
create table products(
productid int PRIMARY KEY,
name varchar(30),
description varchar(200),
price decimal(10,2),
stockQuantity int
);
```

/* cart

- cartid (primary key)
- customerid (foreign key)
- productid (foreign key)
- Quantity */

```
create table cart(
cartid int PRIMARY KEY,
customerid INT,
productid INT,
Quantity INT,
FOREIGN KEY(customerid) REFERENCES customers(customerid) ON DELETE CASCADE,
FOREIGN KEY(productid) REFERENCES products(productid) ON DELETE CASCADE);
```

/* orders

- Ordered (primary key)
- Customerid (foreign key)
- Orderdate
- Totalamount*/

create table orders(

orderid int PRIMARY KEY,

customerid INT,

orderdate date,

totalamount decimal(10,2),

FOREIGN KEY(customerid) REFERENCES customers(customerid) ON DELETE CASCADE

);

/* orderitems

- Orderitemid (primary key)
- Orderid (foreign key)
- Productid (foreign key)
- Quantity
- Itemamount */

create table orderitems(

orderitemid int PRIMARY KEY,

orderid INT,

productid INT,

Quantity INT,

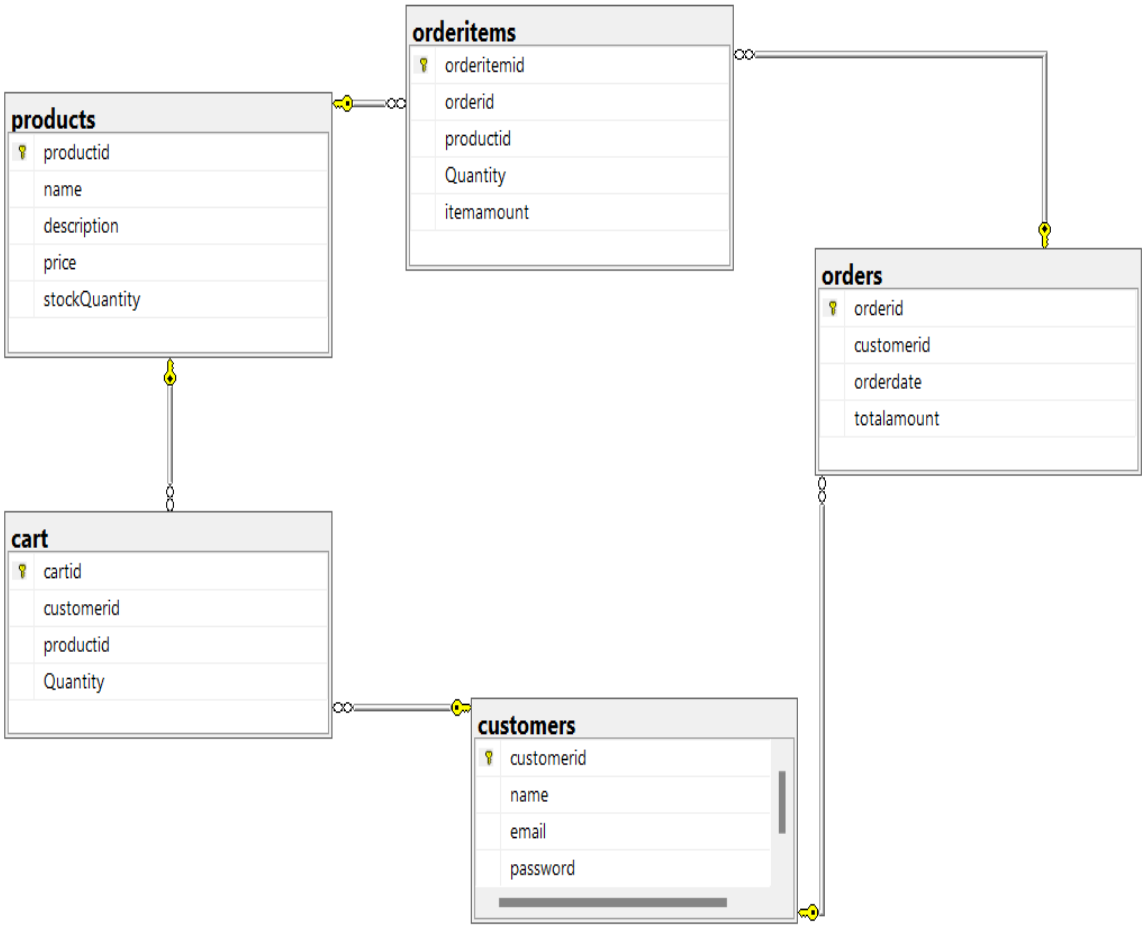
itemamount decimal(10,2),

FOREIGN KEY(orderid) REFERENCES orders(orderid) ON DELETE CASCADE,

FOREIGN KEY(productid) REFERENCES products(productid) ON DELETE CASCADE

);

ER Diagram:



Inserting Values:

```
INSERT INTO customers VALUES
('Rahul Sharma', 'rahul.sharma@example.com', 'rahul1234'),
('Priya Kapoor', 'priya.kapoor@example.com', 'priya_secure'),
('Anjali Mehta', 'anjali.mehta@example.com', 'mehta2024'),
('Vikram Singh', 'vikram.singh@example.com', 'vikram!789'),
('Sneha Patel', 'sneha.patel@example.com', 'patel#567'),
('Rohit Verma', 'rohit.verma@example.com', 'rohit_pass'),
('Kavita Nair', 'kavita.nair@example.com', 'kavita_123'),
('Arjun Rao', 'arjun.rao@example.com', 'arjun987'),
('Meera Iyer', 'meera.iyer@example.com', 'meera$pass'),
('Sanjay Gupta', 'sanjay.gupta@example.com', 'gupta@2024');
 select * from customers

INSERT INTO products VALUES
(1, 'Laptop', 'High-performance laptop', 800.00, 10),
(2, 'Smartphone', 'Latest smartphone', 600.00, 15),
(3, 'Tablet', 'Portable tablet', 300.00, 20),
(4, 'Headphones', 'Noise-canceling', 150.00, 30),
(5, 'TV', '4K Smart TV', 900.00, 5),
(6, 'Coffee Maker', 'Automatic coffee maker', 50.00, 25),
(7, 'Refrigerator', 'Energy-efficient', 700.00, 10),
(8, 'Microwave Oven', 'Countertop microwave', 80.00, 15),
```

(9, 'Blender', 'High-speed blender', 70.00, 20),

(10, 'Vacuum Cleaner', 'Bagless vacuum cleaner', 120.00, 10);

select * from products

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);

select * from cart

INSERT INTO orders VALUES

(1, 1, '2023-01-05', 1200.00),

(2, 2, '2023-02-10', 900.00),

(3, 3, '2023-03-15', 300.00),

(4, 4, '2023-04-20', 150.00),

(5, 5, '2023-05-25', 1800.00),

(6, 6, '2023-06-30', 400.00),

(7, 7, '2023-07-05', 700.00),

(8, 8, '2023-08-10', 160.00),

(9, 9, '2023-09-15', 140.00),

(10, 10, '2023-10-20', 1400.00);

select * from orders

INSERT INTO orderitems 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),

(8,5,2,2,200.00),

(9,6,10,2,240.00),

(10,6,9,3,210.00);

select * from orderitems;

TASKS:

1. Update refrigerator product price to 800.

Ans.

```
update products
set price=800
where name like 'Refrigerator';
select * from products
```

esults | Messages

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

2. Remove all cart items for a specific customer.

Ans.

```
delete  from cart
where customerid=3
select * from cart
```

esults | Messages

cartid	customerid	productid	Quantity
1	1	1	2
2	1	3	1
3	2	2	3
6	4	6	1
7	5	1	1
8	6	10	2
9	6	9	3
10	7	7	2

3. Retrieve Products Priced Below $100.

Ans.

```
select name, price from products
where price<100;
```

sults | Messages

name	price
Coffee Maker	50.00
Microwave Oven	80.00
Blender	70.00

4. Find Products with Stock Quantity Greater Than 5.

Ans.

```
select name, stockquantity from products
where stockquantity > 5;
```

Results | Messages

name	stockquantity
Laptop	10
Smartphone	15
Tablet	20
Headphones	30
Coffee Maker	25
Refrigerator	10
Microwave Oven	15
Blender	20
Vacuum Cleaner	10

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

Ans.

```
select  * from orders
where totalamount>500 and totalamount<1000;
```

Results | Messages

orderid	customerid	orderdate	totalamount
2	2	2023-02-10	900.00
7	7	2023-07-05	700.00

6. Find Products which name end with letter ‘r’.

Ans.

```
select * from  products
where name like '%r'
```

Results | Messages

productid	name	description	price	stockQuantity	category
6	Coffee Maker	Automatic coffee maker	50.00	25	Electronics
7	Refrigerator	Energy-efficient	800.00	10	Accessories
9	Blender	High-speed blender	70.00	20	Accessories
10	Vacuum Cleaner	Bagless vacuum cleaner	120.00	10	Accessories

7. Retrieve Cart Items for Customer 5.

Ans.

```
select * from cart
where customerid=5
```

Results | Messages

cartid	customerid	productid	Quantity
7	5	1	1

8. Find Customers Who Placed Orders in 2023.

Ans.

```
select c.name, o.orderdate
from customers c
join orders o on c.customerid=o.customerid
where orderdate like '2023%'
```

esults | Messages

name	orderdate
Rahul Sharma	2023-01-05
Priya Kapoor	2023-02-10
Anjali Mehta	2023-03-15
Vikram Singh	2023-04-20
Sneha Patel	2023-05-25
Rohit Verma	2023-06-30
Kavita Nair	2023-07-05
Arjun Rao	2023-08-10
Meera Iyer	2023-09-15
Sanjay Gupta	2023-10-20

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

Ans.

```
alter table products
add category varchar(70);
update products
set category = 'Electronics'
where productid IN (1, 2, 3, 5, 6, 8);
update products
set category = 'Accessories'
where productid IN (4,7,9,10);


select category, min(stockQuantity) as min_stock_quantity
from products
group by category;
```

esults | Messages

category	min_stock_quantity
Accessories	10
Electronics	5

10. Calculate the Total Amount Spent by Each Customer.

Ans.

```
select customers.name,
sum(orders.totalamount) as TotalAmount
from customers
JOIN orders
on customers.customerid = orders.customerid
group by customers.name;
```

esults | Messages

name	TotalAmount
Anjali Mehta	300.00
Arjun Rao	160.00
Kavita Nair	700.00
Meera Iyer	140.00
Priya Kapoor	900.00
Rahul Sharma	1200.00
Rohit Verma	400.00
Sanjay Gupta	1400.00
Sneha Patel	1800.00
Vikram Singh	150.00

11. Find the Average Order Amount for Each Customer.

Ans .

```
select
c.customerid,
c.name,
avg(o.totalamount) as Avgamount
from customers c
join orders o on c.customerid=o.customerid
group by c.customerid, c.name
order by Avgamount desc
```

Results | Messages

customerid	name	Avgamount
5	Sneha Patel	1800.000000
10	Sanjay Gupta	1400.000000
1	Rahul Sharma	1200.000000
2	Priya Kapoor	900.000000
7	Kavita Nair	700.000000
6	Rohit Verma	400.000000
3	Anjali Mehta	300.000000
8	Arjun Rao	160.000000
4	Vikram Singh	150.000000
9	Meera Iyer	140.000000

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

Ans.

```
select customers.name,
count(orders.orderid) as No_of_Orders
from customers
JOIN orders
on customers.customerid = orders.customerid
group by customers.name;
```

Results | Messages

name	No_of_Orders
Anjali Mehta	1
Arjun Rao	1
Kavita Nair	1
Meera Iyer	1
Priya Kapoor	1
Rahul Sharma	1
Rohit Verma	1
Sanjay Gupta	1
Sneha Patel	1
Vikram Singh	1

13. Find the Maximum Order Amount for Each Customer.

Ans.

```
SELECT
c.customerid,
c.name,
max(o.totalamount) as maxamount
from customers c
join orders o on c.customerid=o.customerid
group by c.customerid, c.name;
```

Results | Messages

customerid	name	maxamount
1	Rahul Sharma	1200.00
2	Priya Kapoor	900.00
3	Anjali Mehta	300.00
4	Vikram Singh	150.00
5	Sneha Patel	1800.00
6	Rohit Verma	400.00
7	Kavita Nair	700.00
8	Arjun Rao	160.00
9	Meera Iyer	140.00
10	Sanjay Gupta	1400.00

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

Ans.

```
select
c.customerid,
c.name,
sum(o.totalamount) as total_spent
from customers c
JOIN orders o on c.customerid = o.customerid
group by c.customerid, c.name
having sum(o.totalamount) > 1000;
```

Results Messages

customerid	name	total_spent
1	Rahul Sharma	1200.00
5	Sneha Patel	1800.00
10	Sanjay Gupta	1400.00

15. Subquery to Find Products Not in the Cart.

Ans.

```
select
p.productid,
p.name
from products p
where p.productid not in (select productid from cart);
```

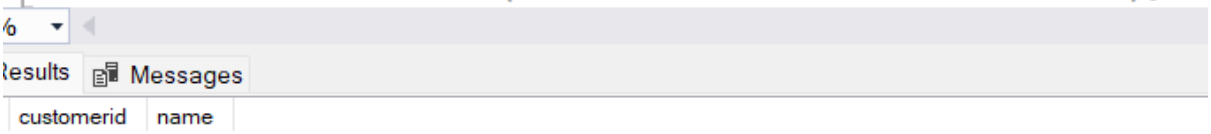
Results Messages

productid	name
4	Headphones
5	TV
8	Microwave Oven

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

Ans.

```
select
customerid,
name
from  customers
where customerid not in (select customerid from customers);
```

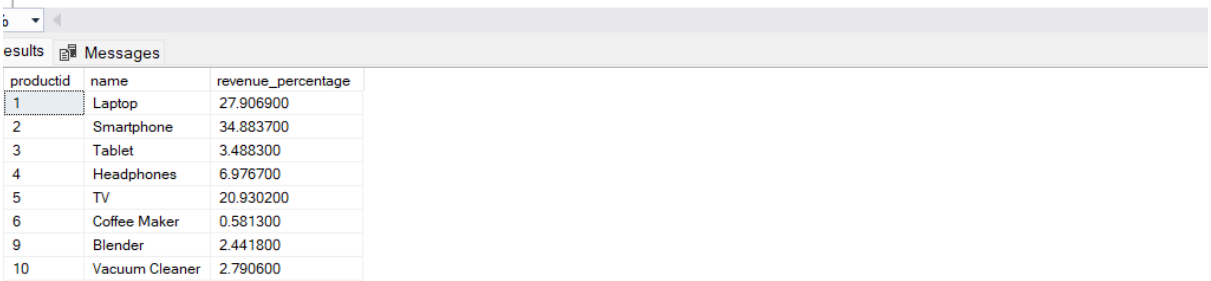


customerid	name

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

Ans.

```
select p.productid, p.name, (sum(oi.Quantity * p.price) / (select sum(oi.Quantity * p.price)
       from orderitems oi
       JOIN products p on oi.productid = p.productid)) * 100 as revenue_percentage
from orderitems oi
join products p on oi.productid = p.productid
group by p.productid, p.name;
```

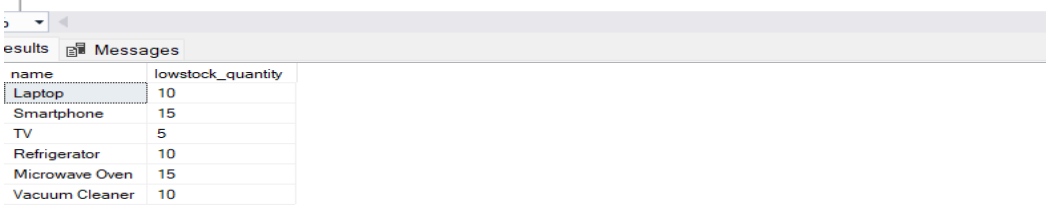


productid	name	revenue_percentage
1	Laptop	27.906900
2	Smartphone	34.883700
3	Tablet	3.488300
4	Headphones	6.976700
5	TV	20.930200
6	Coffee Maker	0.581300
9	Blender	2.441800
10	Vacuum Cleaner	2.790600

18. Subquery to Find Products with Low Stock.

Ans.

```
select
name,
stockquantity as lowstock_quantity
from products
where stockquantity < (select avg(stockquantity) from products);
```



name	lowstock_quantity
Laptop	10
Smartphone	15
TV	5
Refrigerator	10
Microwave Oven	15
Vacuum Cleaner	10

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

Ans.

```
select
name, customerid from customers
where customerid in
(select customerid from orders where totalamount > (select avg(totalamount) from orders));
```

esults | Messages

name	customerid
Rahul Sharma	1
Priya Kapoor	2
Sneha Patel	5
Sanjay Gupta	10