

```
create database Ecommerce;  
use Ecommerce;
```

```
create table customer  
(customer_id int primary key,  
first_name varchar(10),  
last_name varchar(10),  
email varchar(50),  
address varchar(50),  
phone_number int);
```

```
select * from customer
```

```
create table category  
(category_id int primary key,  
category_name varchar(50));
```

```
select * from category
```

```
create table products  
(product_id int primary key,  
product_name varchar(50),  
product_desc varchar(50),  
price int,  
categoryid int ,constraint fk foreign key(categoryid) references category  
    (category_id));  
select * from products
```



```
create table orders  
(order_id int primary key,  
customerid int ,constraint f foreign key(customerid) references customer  
    (customer_id),  
order_date date,  
total_amount int,  
order_status varchar(15));
```



```
select * from orders
```

```
create table payments  
(payment_id int primary key,  
orderid int ,constraint k foreign key(orderid) references orders(order_id),  
payment_date date,  
payment_amount int,  
payment_status varchar(50));
```

```
select * from payments
```

```
create table shipping  
(shipping_id int primary key,
```

```
ordid int,constraint fkr foreign key(ordid) references orders(order_id),
shipping_add varchar(50),
shipping_date date,
shipping_status varchar(50));
```

```
select * from shipping
```

```
create table order_item_id(
order_id int,constraint kr foreign key(order_id) references orders(order_id),
productid int ,constraint a foreign key(productid) references products(product_id),
quantity int,
item_total_amount int);
```

```
select * from order_item_id;
```

```
INSERT INTO customer
(customer_id ,first_name, last_name ,email , address, phone_number ) values
(001, 'gaurav', 'prasad', 'gaurav123@gmail.com', 'patna', 744343443),
(002, 'vivek', 'kumar', 'vivek123@gmail.com', 'kolkata', 947233388),
(003, 'aman', 'gupta', 'aman123@gmail.com', 'goa', 983443534),
(004, 'ankur', 'adarsh', 'ankur123@gmail.com', 'jharkhand', 765656664),
(005, 'riya', 'sen', 'riya123@gmail.com', 'jaipur', 788786879);
```

```
select * from customer;
```

```
INSERT INTO category (category_id ,category_name ) values
(11, 'fashion'),
(12, 'moblies'),
(13, 'grocery'),
(14, 'sports'),
(15, 'home');
```

```
select * from category
```

```
INSERT INTO products
(product_id ,product_name,product_desc ,price ,categoryid ) values
(21, 'shirts', 'formal-shirt', 699, 11),
(22, 'oneplus9', 'mobile', 38000, 12),
(23, 'tea', 'grocery', 500, 13),
(24, 'skates', 'sports', 7999, 14),
(25, 'tool kits', 'home', 899, 15);
```

```
select * from products
```

```
INSERT INTO orders
(order_id ,customerid ,order_date ,total_amount,order_status) values
(31,001, '2023-02-07', 999, 'pending'),
(32,002, '2022-02-06', 8090, 'shipped'),
(33,002, '2021-10-05', 1300, 'delivered'),
```

```
(34,003,'2020-05-09',4000,'pending'),
(35,004,'2019-09-03',8000,'delivered');
```

```
select * from orders
```

```
INSERT INTO payments
```

```
(payment_id ,orderid ,payment_date ,payment_amount ,payment_status ) values
(41,31,'2022-02-09',8009, 'pending'),
(42,32,'2023-05-06',9000, 'paid'),
(43,33,'2021-03-05',6000, 'refunded'),
(44,34,'2020-02-06',3400, 'paid'),
(45,35,'2019-01-04',5400, 'refunded');
```

```
select * from payments
```

```
INSERT INTO order_item_id( order_id,productid ,quantity,item_total_amount) values
```

```
(31,21,2,1398 ),(32,22,3,114000),(33,23,1,500),(34,24,4,31996),
(35,25,3,9434);
select * from order_item_id;
```

```
INSERT INTO shipping
```

```
(shipping_id ,ordid ,shipping_add ,shipping_date ,shipping_status ) values
(51,31,'goa','2021-04-20','not shipped'),
(52,32,'jaipur','2022-05-30',' shipped'),
(53,33,'jaipur','2022-05-30','not shipped'),
(54,34,'jharkhand','2020-03-23','shipped'),
(55,35,'kolkata','2019-02-22','delivered');
```

```
select * from shipping
```

```
--Q3
```

```
--curd operations (create,update,read,delete)
```

```
--here we adding new products
```

```
select * from products
```

```
INSERT INTO products
```

```
(product_id ,product_name,product_desc ,price ,categoryid ) values
(26, 'furnisher' , 'home', 80000,15);
```

```
-- here we updating product_name
```

```
update products SET product_name= 'furnis' WHERE product_id =26;
```

```
select * from products
```

```
-- here read or viewing customer information
```

```
select * from customer;
```

```
--here we delete product_desc with help of product_id
```

```
delete from products where product_desc = 'home' and product_id = 26;
```

```
select * from products
```

```
--here we deleting order_item_id
```

```
--delete from order_item_id;
```

--Q4 advanced queries

-- here we finding products with the highest sales

```
select pro.product_id, pro.product_name, sum(oi.quantity) as total_sales
from Products pro
join order_item_id oi on pro.product_id = oi.productid
group by pro.product_id, pro.product_name
order by total_sales
```

-- here calculating the total revenue for a specific time period

```
select sum(ord.total_amount) as total_revenue
from orders ord
where ord.order_date between '2019-01-01' and '2023-05-30';
```

--here identifying the most active customers.

```
select cus.customer_id, cus.first_name, cus.last_name, count(ord.order_id) AS
    total_orders
from customer cus
join orders ord on cus.customer_id = ord.customerid
group by cus.customer_id, cus.first_name, cus.last_name
order by total_orders
```

➤

--Q5 Business needed Views:

-- Monthly sales View with revenue, number of orders, and top-selling products.

create view MonthlySales as

```
select
sum(ord.total_amount) as revenue,
count(ord.order_id) as num_orders,
p.product_name as top_selling_product
from orders ord
join order_item_id oi on ord.order_id = oi.order_id
join products p on oi.productid = p.product_id
group by p.product_name
order by p.product_name
```

-- Customer loyalty view, showing the number of repeat customers and their purchase history.

➤

create view CustomersLoyalty AS

```
select
customerid,
count( distinct order_id) as number_of_orders,
sum(total_amount) AS historypurchase
from orders
group by customerid
having count(distinct order_id) > 1;
```

```
select *from CustomersLoyalty
```

```
-- Shipping performance view, analysing average delivery times and tracking delayed orders. ➤
```

```
create view ShippingPerformances AS
```

```
select
shipping_status,

avg(shipping_id) as average_shipping,
count(ordid) as number_of_delayed_orders
from Shipping
group by shipping_status;
```

```
select *from ShippingPerformances
```

```
--Q6 . Data Analysis:
--analyzing
```

```
--here we identifying the most popular product name
```

```
select
product_name,
sum(total_amount) as revenue,
count(*) as num_orders
from orders
join order_item_id on orders.order_id = order_item_id.order_id
join products on order_item_id.productid = products.product_id
join category on products.categoryid = category.category_id
group by
product_name
order by
revenue DESC;
```

```
-- regions with the highest sales in customer.
```

```
select
customer_id,
sum(total_amount) as revenue,
```

```
count(*) as num_orders
from orders
join customer on orders.customerid = customer.customer_id
group by
    customer_id
order by
    revenue ;
-- regions with the highest sales in product.
select
product_name,
sum(total_amount) AS revenue
from orders
join order_item_id on orders.order_id = order_item_id.order_id
join products on order_item_id.productid = products.product_id
group by product_name
order by revenue desc;

--Q7. Security:
--security check done

--Q8 Transactions and Rollbacks:

BEGIN TRANSACTION;

INSERT INTO orders (customerid, total_amount) VALUES (005, 100);

-- Insert a new order item.
INSERT INTO order_item_id (order_id, productid, quantity) VALUES (37, 26, 5);
select * from order_item_id
-- Check if the order was created successfully.
SELECT order_id from orders where order_id = 36;

rollback transaction ;

COMMIT TRANSACTION;

--Q9 Optimization:

--use in predicate when querying an index column

--original Query

select*from products
where product_id = 2
OR product_id = 4;

--Improved query:
```

```
select*from products
where product_id IN (4, 7);
```

--instead of * use column names in a select statement

--original Query :

```
select * from products;
```

--Improved Query :

```
select product_id from products;
```

--Q10. Data Backup and Recovery:

--backup done and recovery