ASSIGNMENT 6

mysql> use ecommerce\_system\_011; Database changed

# Creating tables :

mysql> create table customers\_011(

-> customer\_id int primary key,

-> name varchar(50) ,

-> email varchar(50) unique ,

-> city varchar(50)

-> );

mysql>

mysql> create table orders\_011(

-> order\_id int primary key,

-> customer\_id int ,

-> order\_date date ,

-> total\_amount decimal(10,2) ,

-> foreign key (customer\_id) references customers\_011(customer\_id)

-> );

mysql>

mysql> create table products\_011 (

-> product\_id int primary key,

-> product\_name varchar(100) ,

-> category varchar(50) ,

-> price decimal(10,2)

-> );

mysql>

mysql> create table order\_items\_011 (

-> order\_item\_id int primary key,

-> order\_id int ,

-> product\_id int ,

-> quantity int ,

-> subtotal decimal(10,2) ,

-> foreign key (order\_id) references orders\_011(order\_id) ,

-> foreign key (product\_id) references products\_011(product\_id)

-> );

mysql> mysql>

mysql> create table sellers\_011(

-> seller\_id int primary key,

-> seller\_name varchar(100) ,

-> city varchar(50)

-> );

mysql>

mysql> create table product\_sellers\_011 (

-> seller\_id int ,

-> product\_id int ,

-> stock int ,

-> primary key (seller\_id, product\_id),

-> foreign key (seller\_id) references sellers\_011(seller\_id) on delete cascade,

-> foreign key (product\_id) references products\_011(product\_id) on delete cascade

-> );

# Inserting data in tables :

mysql> select \* from customers\_011;

+ + + + +

| customer\_id | name | email | city |

+ + + + +

| 1 | prasad bange | [prasad@gmail.com](mailto:prasad@gmail.com) | jalna |

| 2 | aarush borkar | [aarush@gmail.com](mailto:aarush@gmail.com) | jalna |

| 3 | Devraj Bafna | [devrajbafna@gmail.com](mailto:devrajbafna@gmail.com) | pune |

| 4 | shreya bhagat | [shreya@gmail.com](mailto:shreya@gmail.com) | pune |

| 5 | mihir patil | [mihir@gmail.com](mailto:mihir@gmail.com) | anded |

| 6 | yuvraj shembale | [yuvraj@gmail.com](mailto:yuvraj@gmail.com) | anded |

| 7 | aniket damedhar | [aniket@gmail.com](mailto:aniket@gmail.com) | mumbai |

| 8 | dhruv pujari | [dhruv@gmail.com](mailto:dhruv@gmail.com) | pune |

| 9 | anuj ahire | [anuj@gmail.com](mailto:anuj@gmail.com) | jalna |

| 10 | prathamesh bhangari | [prathamesh@gmail.com](mailto:prathamesh@gmail.com) | Kolhapur |

| 11 | Vishwajit Bhamre | [vishwajitbhamre@gmail.com](mailto:vishwajitbhamre@gmail.com) | pune |

+ + + + +

11 rows in set (0.00 sec)

mysql> select \* from order\_items\_011;

+ + + + + +

| order\_item\_id | order\_id | product\_id | quantity | subtotal |

+ + + + + +

| 1 | 1 | 1 | 1 | 800.00 |

| 2 | 1 | 3 | 2 | 200.00 |

| 3 | 2 | 2 | 1 | 600.00 |

| 4 | 2 | 5 | 3 | 150.00 |

| 5 | 3 | 4 | 1 | 200.00 |

| 6 | 3 | 6 | 2 | 60.00 |

| 7 | 4 | 7 | 1 | 150.00 |

| 8 | 5 | 8 | 1 | 250.00 |

| 9 | 6 | 9 | 2 | 160.00 |

| 10 | 7 | 10 | 1 | 120.00 |

| 11 | 1 | 5 | 1 | 200.00 |

| 12 | 1 | 6 | 1 | 100.00 |

| 13 | 1 | 7 | 1 | 50.00 |

| 14 | 2 | 8 | 1 | 60.00 |

| 15 | 2 | 9 | 1 | 500.00 |

| 16 | 2 | 10 | 1 | 50.00 |

+ + + + + +

16 rows in set (0.00 sec)

mysql> select \* from orders\_011;

+ + + + +

| order\_id | customer\_id | order\_date | total\_amount |

+ + + + +

| 1 | 1 | 2025-03-01 | 1000.00 |

| 2 | 2 | 2025-03-02 | 750.00 |

| 3 | 3 | 2025-03-03 | 300.00 |

| 4 | 4 | 2025-03-04 | 450.00 |

| 5 | 5 | 2025-03-05 | 600.00 |

| 6 | 6 | 2025-03-06 | 1200.00 |

| 7 | 7 | 2025-03-07 | 900.00 |

| 8 | 8 | 2025-03-08 | 500.00 |

| 9 | 9 | 2025-03-09 | 700.00 |

| 10 | 10 | 2025-03-10 | 350.00 |

| 11 | 1 | 2025-03-11 | 550.00 |

| 12 | 2 | 2025-03-12 | 800.00 |

| 13 | 3 | 2025-03-13 | 950.00 |

| 14 | 4 | 2025-03-14 | 400.00 |

+ + + + +

14 rows in set (0.00 sec)

mysql> select \* from product\_sellers\_011;

+ + + +

| seller\_id | product\_id | stock |

+ + + +

| 1 | 1 | 50 |

| 1 | 2 | 30 |

| 2 | 3 | 40 |

| 2 | 4 | 20 |

| 3 | 5 | 25 |

| 3 | 6 | 60 |

| 4 | 7 | 15 |

| 4 | 8 | 35 |

| 5 | 9 | 10 |

| 5 | 10 | 45 |

| 6 | 1 | 20 |

| 7 | 2 | 25 |

| 8 | 3 | 30 |

| 9 | 4 | 15 |

| 10 | 5 | 40 |

+ + + +

15 rows in set (0.00 sec)

mysql> select \* from products\_011;

+ + + + +

| product\_id | product\_name | category | price |

+ + + + +

| 1 | Smartphone | Electronics | 699.99 |

| 2 | Laptop | Electronics | 999.99 |

| 3 | T-Shirt | Clothing | 19.99 |

| 4 | Jeans | Clothing | 49.99 |

| 5 | Microwave Oven | Home Appliances | 199.99 |

| 6 | Fiction Book | Books | 14.99 |

| 7 | Basketball | Sports | 29.99 |

| 8 | flerfume | Beauty fi Health | 39.99 |

| 9 | Dining Table | Furniture | 499.99 |

| 10 | Toy Car | Toys | 24.99 |

| 11 | Smartwatch | Electronics | 299.99 |

| 12 | Sofa Set | Furniture | 999.99 |

| 13 | Gaming Console | Electronics | 499.99 |

+ + + + +

13 rows in set (0.00 sec) mysql> select \* from sellers\_011;

+ + + +

| seller\_id | seller\_name | city |

+ + + +

| 1 | Vijay Sales | flune |

| 2 | Croma | flune |

| 3 | Ajio | flune |

| 4 | Reliance Digital | flune |

| 5 | flantaloon | flune |

| 6 | Shoppers Stop | flune |

| 7 | Decathlon | flune |

| 8 | D-Mart | flune |

| 9 | Westside | flune |

| 10 | Big Bazaar | flune |

| 11 | Metro Mart | flune |

| 12 | Trendz Fashion | flune |

| 13 | Style Hub | flune |

+ + + +

13 rows in set (0.00 sec)

**SQL Queries:**

## Retrieve all customers along with their corresponding orders (including customers who

**haven't ordered).**

-> mysql> select c.customer\_id, c.name, o.order\_id, o.order\_date, o.total\_amount

-> from customers\_011 c

-> left join orders o on c.customer\_id = o.customer\_id;

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| + | + | + | + | + | + |
| | customer\_id | | name | | order\_id | | order\_date | | total\_amount | | |
| + | + | + | + | + | + |
| | 1 | | prasad bange | | 1 | | 2025-03-01 | | 1000.00 | | |
| | 1 | | prasad bange | | 11 | | 2025-03-11 | | 550.00 | | |
| | 2 | | aarush borkar | | 2 | | 2025-03-02 | | 750.00 | | |
| | 2 | | aarush borkar | | 12 | | 2025-03-12 | | 800.00 | | |
| | 3 | | Devraj Bafna | | 3 | | 2025-03-03 | | 300.00 | | |
| | 3 | | Devraj Bafna | | 13 | | 2025-03-13 | | 950.00 | | |
| | 4 | | shreya bhagat | | 4 | | 2025-03-04 | | 450.00 | | |
| | 4 | | shreya bhagat | | 14 | | 2025-03-14 | | 400.00 | | |
| | 5 | | mihir patil | | 5 | | 2025-03-05 | | 600.00 | | |
| | 6 | | yuvraj shembale | | 6 | | 2025-03-06 | | 1200.00 | | |
| | 7 | | aniket damedhar | | 7 | | 2025-03-07 | | 900.00 | | |

| 8 | dhruv pujari | 8 | 2025-03-08 | 500.00 |

| 9 | anuj ahire | 9 | 2025-03-09 | 700.00 |

| 10 | prathamesh bhangari | 10 | 2025-03-10 | 350.00 |

| 11 | Vishwajit Bhamre | ULL | ULL | ULL |

+ + + + + +

15 rows in set (0.01 sec)

## List all orders along with the product names and their quantities.

-> mysql> select oi.order\_id, p.product\_name, oi.quantity

-> from order\_items\_011 oi

-> join products p on oi.product\_id = p.product\_id;

+ + + +

| order\_id | product\_name | quantity |

+ + + +

| 1 | Smartphone | 1 |

| 1 | T-Shirt | 2 |

| 2 | Laptop | 1 |

| 2 | Microwave Oven | 3 |

| 3 | Jeans | 1 |

| 3 | Fiction Book | 2 |

| 4 | Basketball | 1 |

| 5 | flerfume | 1 |

| 6 | Dining Table | 2 |

| 7 | Toy Car | 1 |

| 1 | Microwave Oven | 1 |

| 1 | Fiction Book | 1 |

| 1 | Basketball | 1 |

| 2 | flerfume | 1 |

| 2 | Dining Table | 1 |

| 2 | Toy Car | 1 |

+ + + +

16 rows in set (0.00 sec)

## Find the total number of orders placed by each customer.

-> mysql> select c.customer\_id, c.name, count(o.order\_id) as total\_orders

-> from customers\_011 c

-> left join orders o on c.customer\_id = o.customer\_id

-> group by c.customer\_id, c.name;

+ + + +

| customer\_id | name | total\_orders |

+ + + +

| 1 | prasad bange | 2 |

| 2 | aarush borkar | 2 |

| 3 | Devraj Bafna | 2 |

| 4 | shreya bhagat | 2 |

| 5 | mihir patil | 1 |

| 6 | yuvraj shembale | 1 |

| 7 | aniket damedhar | 1 |

| 8 | dhruv pujari | 1 |

| 9 | anuj ahire | 1 |

| 10 | prathamesh bhangari | 1 |

| 11 | Vishwajit Bhamre | 0 |

+ + + +

11 rows in set (0.01 sec)

## Find the total number of products available in each category.

-> mysql> select category, count(product\_id) as total\_products

-> from products\_011

-> group by category;

+ + +

| category | total\_products |

+ + +

| Electronics | 4 |

| Clothing | 2 |

| Home Appliances | 1 |

| Books | 1 |

| Sports | 1 |

| Beauty fi Health | 1 |

| Furniture | 2 |

| Toys | 1 |

+ + +

## Retrieve the order details, including customer name and total amount, for all orders

placed in the last 30 days.

-> mysql> select o.order\_id, c.name, o.total\_amount

-> from orders\_011 o

-> join customers\_011 c on o.customer\_id = c.customer\_id

-> where order\_date >= curdate() - interval 30 day;

+ + + +

| order\_id | name | total\_amount |

+ + + +

| 8 | dhruv pujari | 500.00 |

| 9 | anuj ahire | 700.00 |

| 10 | prathamesh bhangari | 350.00 |

| 11 | prasad bange | 550.00 |

| 12 | aarush borkar | 800.00 |

| 13 | Devraj Bafna | 950.00 |

| 14 | shreya bhagat | 400.00 |

+ + + +

7 rows in set (0.01 sec)

## Retrieve the names of sellers who sell a specific product (e.g., "Laptop").

-> mysql> select s.seller\_name

-> from sellers\_011 s

-> join product\_sellers\_011 ps on s.seller\_id = ps.seller\_id

-> join products\_011 p on ps.product\_id = p.product\_id

-> where p.product\_name = 'Laptop';

+ +

| seller\_name |

+ +

| Vijay Sales |

| Decathlon |

+ +

2 rows in set (0.01 sec)

## Show all customers who have never placed an order.

-> mysql> select \* from customers\_011

-> where customer\_id not in (select customer\_id from orders);

+ + + + +

| customer\_id | name | email | city |

+ + + + +

| 11 | Vishwajit Bhamre | [vishwajitbhamre@gmail.com](mailto:vishwajitbhamre@gmail.com) | pune |

+ + + + +

1 row in set (0.01 sec)

## Retrieve details of orders where the total amount is greater than the average order total.(

**solve suing subquery)**

-> mysql> select \* from orders\_011

-> where total\_amount > (select avg(total\_amount) from orders\_011);

+ + + + +

| order\_id | customer\_id | order\_date | total\_amount |

+ + + + +

| 1 | 1 | 2025-03-01 | 1000.00 |

| 2 | 2 | 2025-03-02 | 750.00 |

| 6 | 6 | 2025-03-06 | 1200.00 |

| 7 | 7 | 2025-03-07 | 900.00 |

| 9 | 9 | 2025-03-09 | 700.00 |

| 12 | 2 | 2025-03-12 | 800.00 |

| 13 | 3 | 2025-03-13 | 950.00 |

+ + + + +

7 rows in set (0.01 sec)

## Find customers who have placed at least two orders.

-> mysql> select c.customer\_id, c.name, count(o.order\_id) as total\_orders

-> from customers\_011 c

-> join orders\_011 o on c.customer\_id = o.customer\_id

-> group by c.customer\_id, c.name

-> having total\_orders >= 2;

+ + + +

| customer\_id | name | total\_orders |

+ + + +

| 1 | prasad bange | 2 |

| 2 | aarush borkar | 2 |

| 3 | Devraj Bafna | 2 |

| 4 | shreya bhagat | 2 |

+ + + +

4 rows in set (0.01 sec)

## Find the top 3 most ordered products based on quantity sold.

**->** mysql> select p.product\_name, sum(oi.quantity) as total\_quantity

-> from order\_items\_011 oi

-> join products\_011 p on oi.product\_id = p.product\_id

-> group by p.product\_name

-> order by total\_quantity desc

-> limit 3;

+ + +

| product\_name | total\_quantity |

+ + +

| Microwave Oven | 4 |

| Dining Table | 3 |

| Fiction Book | 3 |

+ + +

3 rows in set (0.01 sec)

## Display the product names that are sold by multiple sellers.

-> mysql> select p.product\_name

-> from product\_sellers\_011 ps

-> join products p on ps.product\_id = p.product\_id

-> group by p.product\_name

-> having count(ps.seller\_id) > 1;

|  |  |  |
| --- | --- | --- |
| + |  | + |
| | | product\_name | | |
| + |  | + |
| | | Smartphone | | |
| | | Laptop | | |
| | | T-Shirt | | |
| | | Jeans | | |
| | | Microwave Oven | | |

## Retrieve sellers who have never sold any product.

**->** mysql> select \* from sellers\_011

-> where seller\_id not in (select distinct seller\_id from product\_sellers\_011);

+ + + +

| seller\_id | seller\_name | city |

+ + + +

| 11 | Metro Mart | flune |

| 12 | Trendz Fashion | flune |

| 13 | Style Hub | flune |

+ + + +

3 rows in set (0.01 sec)

## Find all products that have never been ordered.

-> mysql> select \* from products\_011

-> where product\_id not in (select distinct product\_id from order\_items\_011);

+ + + + +

| product\_id | product\_name | category | price |

+ + + + +

| 11 | Smartwatch | Electronics | 299.99 |

| 12 | Sofa Set | Furniture | 999.99 |

| 13 | Gaming Console | Electronics | 499.99 |

+ + + + +

3 rows in set (0.01 sec)

## Retrieve the names of customers who have placed the highest number of orders.

-> mysql> select c.name

-> from customers\_011 c

-> join orders\_011 o on c.customer\_id = o.customer\_id

-> group by c.customer\_id, c.name

-> order by count(o.order\_id) desc

-> limit 1;

+ +

| name |

+ +

| prasad bange |

## Find all customers who have ordered more than 5 different products.

-> mysql> select c.customer\_id, c.name

-> from customers\_011 c

-> join orders\_011 o on c.customer\_id = o.customer\_id

-> join order\_items\_011 oi on o.order\_id = oi.order\_id

-> group by c.customer\_id, c.name

-> having count(distinct oi.product\_id) > 5; Empty set (0.00 sec)

## Find products that are sold by at least two different sellers but have never been ordered.

**->** mysql> select p.product\_name

-> from product\_sellers\_011 ps

-> join products\_011 p on ps.product\_id = p.product\_id

-> where not exists (

-> select 1

-> from order\_items\_011 oi

-> where oi.product\_id = p.product\_id

-> )

-> group by p.product\_name

-> having count(distinct ps.seller\_id) > 1; Empty set (0.00 sec)

## Find the customer who has spent the most money overall.

**->** mysql> select c.customer\_id, c.name, sum(o.total\_amount) as total\_spent

-> from customers\_011 c

-> join orders\_011 o on c.customer\_id = o.customer\_id

-> group by c.customer\_id, c.name

-> order by total\_spent desc

-> limit 1;

+ + + +

| customer\_id | name | total\_spent |

+ + + +

| 1 | prasad bange | 1550.00 |

+ + + +

1 row in set (0.00 sec)

## Find all customers who have either placed an order or live in the same city as a seller.

**->** mysql> select c.customer\_id, c.name, c.email, c.city, sum(o.total\_amount) as total\_spent

-> from customers\_011 c

-> join orders\_011 o on c.customer\_id = o.customer\_id

-> group by c.customer\_id, c.name, c.email, c.city

-> order by total\_spent desc

-> limit 1;

+ + + + + +

| customer\_id | name | email | city | total\_spent |

+ + + + + +

| 1 | prasad bange | [prasad@gmail.com](mailto:prasad@gmail.com) | jalna | 1550.00 |

+ + + + + +

1 row in set (0.00 sec)

## Retrieve all products that are either in stock with at least one seller or have been orderedat least once.

**->** mysql> select distinct p.product\_id, p.product\_name

-> from products\_011 p

-> join product\_sellers\_011 ps on p.product\_id = ps.product\_id

->

-> union

->

-> select distinct p.product\_id, p.product\_name

-> from products\_011 p

-> join order\_items\_011 oi on p.product\_id = oi.product\_id;

+ + +

| product\_id | product\_name |

+ + +

| 1 | Smartphone |

| 2 | Laptop |

| 3 | T-Shirt |

| 4 | Jeans |

| 5 | Microwave Oven |

| 6 | Fiction Book |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| | | 7 | | | Basketball | | |
| | | 8 | | | flerfume | | |
| | | 9 | | | Dining Table | | |
| | | 10 | | | Toy Car | | |
| + |  | + |  | + |
| 10 | rows in set | | (0.00 sec) | |

## Retrieve products that have been both ordered and are currently in stock.

**->** mysql> select distinct p.product\_id, p.product\_name

-> from products\_011 p

-> join order\_items\_011 oi on p.product\_id = oi.product\_id

-> join product\_sellers\_011 ps on p.product\_id = ps.product\_id;

+ + +

| product\_id | product\_name |

+ + +

| 1 | Smartphone |

| 2 | Laptop |

| 3 | T-Shirt |

| 4 | Jeans |

| 5 | Microwave Oven |

| 6 | Fiction Book |

| 7 | Basketball |

| 8 | flerfume |

| 9 | Dining Table |

| 10 | Toy Car |

+ + +

10 rows in set (0.00 sec)

## Find customers who have both placed an order and live in a city where a seller is located.

**->** mysql> select distinct c.customer\_id, c.name, c.city

-> from customers\_011 c

-> join orders\_011 o on c.customer\_id = o.customer\_id

-> join sellers\_011 s on c.city = s.city;

+ + + +

| customer\_id | name | city |

+ + + +

| 8 | dhruv pujari | pune |

| 4 | shreya bhagat | pune |

| 3 | Devraj Bafna | pune |

+ + + +

3 rows in set (0.00 sec)

## Retrieve all customers who have placed at least one order in each year available in the database.

**->** mysql> select c.customer\_id, c.name

-> from customers\_011 c

-> join orders\_011 o on c.customer\_id = o.customer\_id

-> group by c.customer\_id, c.name

-> having count(distinct year(o.order\_date)) = (

-> select count(distinct year(order\_date)) from orders\_011

-> );

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| + |  | + |  |  | + |
| | | customer\_id | | | name |  | | |
| + |  | + |  |  | + |
| | | 1 | | | prasad | bange | | |
| | | 2 | | | aarush | borkar | | |
| | | 3 | | | Devraj | Bafna | | |
| | | 4 | | | shreya | bhagat | | |
| | | 5 | | | mihir patil | | | |
| | | 6 | | | yuvraj shembale | | | |
| | | 7 | | | aniket damedhar | | | |
| | | 8 | | | dhruv pujari | | | |
| | | 9 | | | anuj ahire | | | |
| | | 10 | | | prathamesh bhangari | | | |
| + |  | + |  | | + |

10 rows in set (0.00 sec)