Assignment 4: SQL Computations in an E-Commerce Application

**mysql> show databases ;**

**+ +**

**| Database |**

**+ +**

**| banking\_system\_011 |**

**| classicmodels |**

**| information\_schema |**

**| mysql |**

**| pccoe |**

**| performance\_schema |**

**| practice |**

**| record |**

**| sql\_intro |**

**| sql\_joins |**

**| student |**

**| studentmanagement |**

**| sub\_queries |**

**| subqueries |**

**| sys |**

**| triggers |**

**| vehical |**

**+ +**

**17 rows in set (0.01 sec)**

mysql> create database ecommerce\_011; Query OK, 1 row affected (0.02 sec)

1. **Customers : Stores customer details. mysql> create table customers (**

-> customer\_id int primary key,

-> name varchar(50),

-> email varchar(100),

-> city varchar(50),

-> age int

**-> );**

Query OK, 0 rows affected (0.06 sec) mysql> select \* from customers ; Empty set (0.00 sec)

mysql> describe customers ;

**+ + + + + + +**

| Field | Type | ull | Key | Default | Extra |

**+ + + + + + +**

| customer\_id | int | O | flRI | ULL | |

| name | varchar(50) | YES | | ULL | |

| email | varchar(100) | YES | | ULL | |

| city | varchar(50) | YES | | ULL | |

| age | int | YES | | ULL | |

**+ + + + + + +**

5 rows in set (0.00 sec)

1. **Orders :Stores order details placed by customers. mysql> create table orders (**

-> order\_id int primary key ,customer\_id int ,order\_date date,

-> total\_amount decimal(10,2),status varchar(20),

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

**-> );**

Query OK, 0 rows affected (0.07 sec)

mysql> describe orders ;

**+ + + + + + +**

| Field | Type | ull | Key | Default | Extra |

**+ + + + + + +**

| order\_id | int | O | flRI | ULL | |

| customer\_id | int | YES | MUL | ULL | |

| order\_date | date | YES | | ULL | |

| total\_amount | decimal(10,2) | YES | | ULL | |

| status | varchar(20) | YES | | ULL | |

**+ + + + + + +**

5 rows in set (0.00 sec)

1. **flroducts : Stores product details. mysql> create table products (**

-> product\_id int primary key, name varchar(50),category varchar(50),

-> price decimal(10,2), stock int); Query OK, 0 rows affected (0.04 sec)

mysql> describe products ;

**+ + + + + + +**

| Field | Type | ull | Key | Default | Extra |

**+ + + + + + +**

| product\_id | int | O | flRI | ULL | |

| name | varchar(50) | YES | | ULL | |

| category | varchar(50) | YES | | ULL | |

| price | decimal(10,2) | YES | | ULL | |

| stock | int | YES | | ULL | |

**+ + + + + + +**

5 rows in set (0.00 sec)

3. Order\_Items : Stores products included in an order

mysql> create table order\_items (

-> order\_item\_id int primary key, order\_id int, product\_id int, quantity int,

-> subtotal decimal(10,2), foreign key (order\_id) references orders(order\_id),

-> foreign key (product\_id) references products(product\_id)); Query OK, 0 rows affected (0.07 sec)

mysql> describe order\_items ;

**+ + + + + + +**

| Field | Type | ull | Key | Default | Extra |

**+ + + + + + +**

| order\_item\_id | int | O | flRI | ULL | |

| order\_id | int | YES | MUL | ULL | |

| product\_id | int | YES | MUL | ULL | |

| quantity | int | YES | | ULL | |

| subtotal | decimal(10,2) | YES | | ULL | |

**+ + + + + + +**

5 rows in set (0.00 sec)

mysql> select \* from customers ;

**+ + + + +**

**+**

| customer\_id | name | email | city | age |

**+ + + + +**

**+**

| 1 | Alice Johnson | [alice@gmail.com](mailto:alice@gmail.com) | ew York |

28 |

| 2 | Bob Smith | [bob@gmail.com](mailto:bob@gmail.com) | Los Angeles |

35 |

| 3 | Charlie Davis | [charlie@gmail.com](mailto:charlie@gmail.com) | Chicago |

40 |

| 4 | Diana Miller | [diana@gmail.com](mailto:diana@gmail.com) | Houston |

25 |

| 5 | Ethan Wilson | [ethan@gmail.com](mailto:ethan@gmail.com) | Miami |

32 |

| 6 | Fayla Brown | [fayla@gmail.com](mailto:fayla@gmail.com) | San Francisco |

29 |

| 7 | George King | [george@gmail.com](mailto:george@gmail.com) | Dallas |

38 |

| 8 | Hannah White | [hannah@gmail.com](mailto:hannah@gmail.com) | flhoenix |

22 |

| 9 | Isaac Clark | [isaac@gmail.com](mailto:isaac@gmail.com) | flhiladelphia |

30 |

| 10 | Julia Evans | [julia@gmail.com](mailto:julia@gmail.com) | Seattle |

33 |

| 11 | Kevin Lewis | [kevin@gmail.com](mailto:kevin@gmail.com) | Denver |

45 |

| 12 | Lena Hall | [lena@gmail.com](mailto:lena@gmail.com) | Austin |

27 |

| 13 | Mike Scott | [mike@gmail.com](mailto:mike@gmail.com) | Boston |

36 |

| 14 | ina Young | [nina@gmail.com](mailto:nina@gmail.com) | Atlanta |

31 |

| 15 | Oscar Martinez | [oscar@gmail.com](mailto:oscar@gmail.com) | ew York |

40 |

**+ + + + +**

**+**

15 rows in set (0.00 sec)

mysql> insert into orders

-> values (101, 1, '2024-02-01', 150.75, 'shipped'),(102, 2,

'2024-02-02', 89.99, 'processing'),

-> (103, 3, '2024-02-03', 230.50, 'delivered'),(104, 4, '2024-02-

04', 120.00, 'cancelled'),

-> (105, 5, '2024-02-05', 310.25, 'shipped'),(106, 6, '2024-02-

06', 75.00, 'processing'),

-> (107, 7, '2024-02-07', 99.99, 'delivered'),(108, 1, '2024-02-

08', 200.00, 'shipped'),

-> (109, 2, '2024-02-09', 50.00, 'processing'),(110, 3, '2024-02-

10', 180.75, 'delivered'),

-> (111, 4, '2024-02-11', 160.50, 'cancelled'),(112, 5, '2024-02-

12', 400.00, 'shipped'),

-> (113, 6, '2024-02-13', 130.25, 'processing'),(114, 7, '2024-02-

14', 110.00, 'delivered'),

-> (115, 1, '2024-02-15', 190.00, 'shipped');

Query OK, 15 rows affected (0.02 sec) Records: 15 Duplicates: 0 Warnings: 0

mysql> select \* from orders ;

**+ + + + + +**

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

**+ + + + + +**

| 101 | 1 | 2024-02-01 | 150.75 | shipped |

| 102 | 2 | 2024-02-02 | 89.99 | processing |

| 103 | 3 | 2024-02-03 | 230.50 | delivered |

| 104 | 4 | 2024-02-04 | 120.00 | cancelled |

| 105 | 5 | 2024-02-05 | 310.25 | shipped |

| 106 | 6 | 2024-02-06 | 75.00 | processing |

| 107 | 7 | 2024-02-07 | 99.99 | delivered |

| 108 | 1 | 2024-02-08 | 200.00 | shipped |

| 109 | 2 | 2024-02-09 | 50.00 | processing |

| 110 | 3 | 2024-02-10 | 180.75 | delivered |

| 111 | 4 | 2024-02-11 | 160.50 | cancelled |

| 112 | 5 | 2024-02-12 | 400.00 | shipped |

| 113 | 6 | 2024-02-13 | 130.25 | processing |

| 114 | 7 | 2024-02-14 | 110.00 | delivered |

| 115 | 1 | 2024-02-15 | 190.00 | shipped |

**+ + + + + +**

15 rows in set (0.00 sec)

mysql> insert into products

-> values (1, 'Laptop', 'Electronics', 799.99, 15),(2,

'Smartphone', 'Electronics', 499.99, 30),

-> (3, 'Headphones', 'Accessories', 99.99, 50),(4, 'Desk Chair',

'Furniture', 120.00, 20),

-> (5, 'Coffee Maker', 'Appliances', 45.50, 25),(6, 'Backpack',

'Bags', 35.00, 40),

-> (7, 'Running Shoes', 'Footwear', 85.00, 35),(8, 'Smartwatch',

'Electronics', 199.99, 40),

-> (9, 'Gaming Mouse', 'Accessories', 59.99, 60),(10, 'Office

Desk', 'Furniture', 200.00, 15),

-> (11, 'Blender', 'Appliances', 60.00, 30),(12, 'Tote Bag',

'Bags', 25.00, 50),

-> (13, 'Sneakers', 'Footwear', 70.00, 45),(14, 'Tablet',

'Electronics', 249.99, 20),

-> (15, 'Yoga Mat', 'Accessories', 30.00, 80); Query OK, 15 rows affected (0.02 sec)

Records: 15 Duplicates: 0 Warnings: 0

mysql> select \* from products ;

**+ + + + + +**

| product\_id | name | category | price | stock |

**+ + + + + +**

| 1 | Laptop | Electronics | 799.99 | 15 |

| 2 | Smartphone | Electronics | 499.99 | 30 |

| 3 | Headphones | Accessories | 99.99 | 50 |

| 4 | Desk Chair | Furniture | 120.00 | 20 |

| 5 | Coffee Maker | Appliances | 45.50 | 25 |

| 6 | Backpack | Bags | 35.00 | 40 |

| 7 | Running Shoes | Footwear | 85.00 | 35 |

| 8 | Smartwatch | Electronics | 199.99 | 40 |

| 9 | Gaming Mouse | Accessories | 59.99 | 60 |

| 10 | Office Desk | Furniture | 200.00 | 15 |

| 11 | Blender | Appliances | 60.00 | 30 |

| 12 | Tote Bag | Bags | 25.00 | 50 |

| 13 | Sneakers | Footwear | 70.00 | 45 |

| 14 | Tablet | Electronics | 249.99 | 20 |

| 15 | Yoga Mat | Accessories | 30.00 | 80 |

**+ + + + + +**

15 rows in set (0.00 sec)

mysql> insert into order\_items

-> values (1, 101, 1, 1, 799.99),(2, 101, 3, 2, 199.98),

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **->** | **(3,** | **102,** | **2,** | **1,** | **499.99),** | **(4,** | **103,** | **4,** | **1,** | **120.00),** |
| **->** | **(5,** | **103,** | **6,** | **2,** | **70.00),** | **(6,** | **104,** | **5,** | **1,** | **45.50),** |
| **->** | **(7,** | **105,** | **7,** | **1,** | **85.00),** | **(8,** | **106,** | **1,** | **2,** | **1599.98),** |
| **->** | **(9,** | **107,** | **2,** | **3,** | **1499.97),** | **(10,** | **108,** | **4,** | **1,** | **120.00),** |
| **->** | **(11,** | **109,** | **5,** | **2,** | **91.00),** | **(12,** | **110,** | **6,** | **1,** | **35.00),** |
| **->** | **(13,** | **111,** | **7,** | **2,** | **170.00),** | **(14,** | **112,** | **3,** | **1,** | **99.99),** |
| **->** | **(15,** | **113,** | **1,** | **1,** | **799.99);** |  |  |  |  |  |

Query OK, 15 rows affected (0.02 sec) Records: 15 Duplicates: 0 Warnings: 0

mysql> select \* from order\_items ;

**+ + + + + +**

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

**+ + + + + +**

| 1 | 101 | 1 | 1 | 799.99 |

| 2 | 101 | 3 | 2 | 199.98 |

| 3 | 102 | 2 | 1 | 499.99 |

| 4 | 103 | 4 | 1 | 120.00 |

| 5 | 103 | 6 | 2 | 70.00 |

| 6 | 104 | 5 | 1 | 45.50 |

| 7 | 105 | 7 | 1 | 85.00 |

| 8 | 106 | 1 | 2 | 1599.98 |

| 9 | 107 | 2 | 3 | 1499.97 |

| 10 | 108 | 4 | 1 | 120.00 |

| 11 | 109 | 5 | 2 | 91.00 |

| 12 | 110 | 6 | 1 | 35.00 |

| 13 | 111 | 7 | 2 | 170.00 |

| 14 | 112 | 3 | 1 | 99.99 |

| 15 | 113 | 1 | 1 | 799.99 |

**+ + + + + +**

15 rows in set (0.00 sec)

1. **Find the total amount spent by customer ID 3 (sum of all their orders).**

mysql> select sum(total\_amount) as total\_spent

-> from orders

-> where customer\_id = 3;

**+ +**

| total\_spent |

**+ +**

| 411.25 |

**+ +**

1 row in set (0.00 sec)

1. **Find the total revenue generated from all orders mysql> select sum(total\_amount) as total\_revenue**

-> from orders;

**+ +**

| total\_revenue |

**+ +**

| 2497.98 |

**+ +**

1 row in set (0.00 sec)

1. **Calculate the stock value (price × stock) for each product. mysql> select product\_id, name, price \* stock as stock\_value**

-> from products;

**+ + + +**

| product\_id | name | stock\_value |

**+ + + +**

| 1 | Laptop | 11999.85 |

| 2 | Smartphone | 14999.70 |

| 3 | Headphones | 4999.50 |

| 4 | Desk Chair | 2400.00 |

| 5 | Coffee Maker | 1137.50 |

| 6 | Backpack | 1400.00 |

| 7 | Running Shoes | 2975.00 |

| 8 | Smartwatch | 7999.60 |

| 9 | Gaming Mouse | 3599.40 |

| 10 | Office Desk | 3000.00 |

| 11 | Blender | 1800.00 |

| 12 | Tote Bag | 1250.00 |

| 13 | Sneakers | 3150.00 |

| 14 | Tablet | 4999.80 |

| 15 | Yoga Mat | 2400.00 |

**+ + + +**

15 rows in set (0.00 sec)

1. **Retrieve customers who are above 30 years old and live in "ew York".**

mysql> select \* from customers

-> where age > 30 and city = 'ew York';

**+ + + + +**

**+**

| customer\_id | name | email | city | age

**|**

**+ + + + +**

**+**

| 15 | Oscar Martinez | [oscar@gmail.com](mailto:oscar@gmail.com) | ew York | 40

**|**

**+ + + + +**

**+**

1 row in set (0.00 sec)

1. **Find customers who either live in "Los Angeles" or have placed an order worth more than $500.**

mysql> select distinct c.\* from customers c

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

-> where c.city = 'Los Angeles' or o.total\_amount > 500;

**+ + + + + +**

| customer\_id | name | email | city | age |

**+ + + + + +**

| 2 | Bob Smith | [bob@gmail.com](mailto:bob@gmail.com) | Los Angeles | 35 |

**+ + + + + +**

1 row in set (0.00 sec)

1. **List products that are in stock but cost less than $50. mysql> select \* from products**

-> where stock > 0 and price < 50;

**+ + + + + +**

| product\_id | name | category | price | stock |

**+ + + + + +**

| 5 | Coffee Maker | Appliances | 45.50 | 25 |

| 6 | Backpack | Bags | 35.00 | 40 |

| 12 | Tote Bag | Bags | 25.00 | 50 |

| 15 | Yoga Mat | Accessories | 30.00 | 80 |

**+ + + + + +**

4 rows in set (0.00 sec)

1. **Find flroducts That Have Less Than 5 Items in Stock mysql> select \* from products**

-> where stock < 5; Empty set (0.00 sec)

1. **Find orders that are not completed (status is not 'Completed'). mysql> select \* from orders**

-> where status <> 'Completed';

**+ + + + + +**

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

**+ + + + + +**

| 101 | 1 | 2024-02-01 | 150.75 | shipped |

| 102 | 2 | 2024-02-02 | 89.99 | processing |

| 103 | 3 | 2024-02-03 | 230.50 | delivered |

| 104 | 4 | 2024-02-04 | 120.00 | cancelled |

| 105 | 5 | 2024-02-05 | 310.25 | shipped |

| 106 | 6 | 2024-02-06 | 75.00 | processing |

| 107 | 7 | 2024-02-07 | 99.99 | delivered |

| 108 | 1 | 2024-02-08 | 200.00 | shipped |

| 109 | 2 | 2024-02-09 | 50.00 | processing |

| 110 | 3 | 2024-02-10 | 180.75 | delivered |

| 111 | 4 | 2024-02-11 | 160.50 | cancelled |

| 112 | 5 | 2024-02-12 | 400.00 | shipped |

| 113 | 6 | 2024-02-13 | 130.25 | processing |

| 114 | 7 | 2024-02-14 | 110.00 | delivered |

| 115 | 1 | 2024-02-15 | 190.00 | shipped |

**+ + + + + +**

15 rows in set (0.00 sec)

1. **Find products that are out of stock or belong to the "Electronics" category**

mysql> select \* from products

-> where stock = 0 or category = 'Electronics';

**+ + + + + +**

| product\_id | name | category | price | stock |

**+ + + + + +**

| 1 | Laptop | Electronics | 799.99 | 15 |

| 2 | Smartphone | Electronics | 499.99 | 30 |

| 8 | Smartwatch | Electronics | 199.99 | 40 |

| 14 | Tablet | Electronics | 249.99 | 20 |

**+ + + + + +**

4 rows in set (0.00 sec)

1. **List all email addresses that belong to Gmail. mysql> select email from customers**

-> where email like '%@gmail.com';

**+ +**

| email |

**+ +**

| [alice@gmail.com](mailto:alice@gmail.com) |

| [bob@gmail.com](mailto:bob@gmail.com) |

| [charlie@gmail.com](mailto:charlie@gmail.com) |

| [diana@gmail.com](mailto:diana@gmail.com) |

| [ethan@gmail.com](mailto:ethan@gmail.com) |

| [fayla@gmail.com](mailto:fayla@gmail.com) |

| [george@gmail.com](mailto:george@gmail.com) |

| [hannah@gmail.com](mailto:hannah@gmail.com) |

| [isaac@gmail.com](mailto:isaac@gmail.com) |

| [julia@gmail.com](mailto:julia@gmail.com) |

| [kevin@gmail.com](mailto:kevin@gmail.com) |

| [lena@gmail.com](mailto:lena@gmail.com) |

| [mike@gmail.com](mailto:mike@gmail.com) |

| [nina@gmail.com](mailto:nina@gmail.com) |

| [oscar@gmail.com](mailto:oscar@gmail.com) |

**+ +**

15 rows in set (0.00 sec)

1. **Find product names containing the word "flhone".**

mysql> select \* from products

-> where name like '%flhone%';

**+ + + + + +**

| product\_id | name | category | price | stock |

**+ + + + + +**

| 2 | Smartphone | Electronics | 499.99 | 30 |

| 3 | Headphones | Accessories | 99.99 | 50 |

**+ + + + + +**

2 rows in set (0.00 sec)

1. **Find customers whose names have exactly 5 characters. mysql> select \* from customers**

-> where length(name) = 5; Empty set (0.00 sec)

1. **Find flroducts That Have "Laptop" or "Tablet" in Their ame**

mysql> select \* from products

-> where name like '%Laptop%' or name like '%Tablet%';

**+ + + + + +**

| product\_id | name | category | price | stock |

**+ + + + + +**

| 1 | Laptop | Electronics | 799.99 | 15 |

| 14 | Tablet | Electronics | 249.99 | 20 |

**+ + + + + +**

2 rows in set (0.00 sec)

1. **Retrieve orders placed in the year 2024. mysql> select \* from orders**

-> where year(order\_date) = 2024;

**+ + + + + +**

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

**+ + + + + +**

| 101 | 1 | 2024-02-01 | 150.75 | shipped |

| 102 | 2 | 2024-02-02 | 89.99 | processing |

| 103 | 3 | 2024-02-03 | 230.50 | delivered |

| 104 | 4 | 2024-02-04 | 120.00 | cancelled |

| 105 | 5 | 2024-02-05 | 310.25 | shipped |

| 106 | 6 | 2024-02-06 | 75.00 | processing |

| 107 | 7 | 2024-02-07 | 99.99 | delivered |

| 108 | 1 | 2024-02-08 | 200.00 | shipped |

| 109 | 2 | 2024-02-09 | 50.00 | processing |

| 110 | 3 | 2024-02-10 | 180.75 | delivered |

| 111 | 4 | 2024-02-11 | 160.50 | cancelled |

| 112 | 5 | 2024-02-12 | 400.00 | shipped |

| 113 | 6 | 2024-02-13 | 130.25 | processing |

| 114 | 7 | 2024-02-14 | 110.00 | delivered |

| 115 | 1 | 2024-02-15 | 190.00 | shipped |

**+ + + + + +**

15 rows in set (0.01 sec)

1. **Find customers who live in "ew York", "Los Angeles", or "Chicago".**

mysql> select \* from customers

-> where city in ('ew York', 'Los Angeles', 'Chicago');

**+ + + + + +**

| customer\_id | name | email | city | age |

**+ + + + + +**

| 1 | Alice Johnson | [alice@gmail.com](mailto:alice@gmail.com) | ew York | 28 |

| 2 | Bob Smith | [bob@gmail.com](mailto:bob@gmail.com) | Los Angeles | 35 |

| 3 | Charlie Davis | [charlie@gmail.com](mailto:charlie@gmail.com) | Chicago | 40 |

| 15 | Oscar Martinez | [oscar@gmail.com](mailto:oscar@gmail.com) | ew York | 40 |

**+ + + + + +**

4 rows in set (0.00 sec)

1. **List products that do not belong to the 'Electronics' or 'Clothing' categories.**

mysql> select \* from products

-> where category not in ('Electronics', 'Clothing');

**+ + + + + +**

| product\_id | name | category | price | stock |

**+ + + + + +**

| 3 | Headphones | Accessories | 99.99 | 50 |

| 4 | Desk Chair | Furniture | 120.00 | 20 |

| 5 | Coffee Maker | Appliances | 45.50 | 25 |

| 6 | Backpack | Bags | 35.00 | 40 |

| 7 | Running Shoes | Footwear | 85.00 | 35 |

| 9 | Gaming Mouse | Accessories | 59.99 | 60 |

| 10 | Office Desk | Furniture | 200.00 | 15 |

| 11 | Blender | Appliances | 60.00 | 30 |

| 12 | Tote Bag | Bags | 25.00 | 50 |

| 13 | Sneakers | Footwear | 70.00 | 45 |

| 15 | Yoga Mat | Accessories | 30.00 | 80 |

**+ + + + + +**

11 ows in set (0.00 sec)

17 Find orders placed by customers with IDs 1, 3, or 5.

mysql> select \* from orders

-> where customer\_id in (1, 3, 5);

**+ + + + + +**

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

**+ + + + + +**

| 101 | 1 | 2024-02-01 | 150.75 | shipped |

| 108 | 1 | 2024-02-08 | 200.00 | shipped |

| 115 | 1 | 2024-02-15 | 190.00 | shipped |

| 103 | 3 | 2024-02-03 | 230.50 | delivered |

| 110 | 3 | 2024-02-10 | 180.75 | delivered |

| 105 | 5 | 2024-02-05 | 310.25 | shipped |

| 112 | 5 | 2024-02-12 | 400.00 | shipped |

**+ + + + + +**

7 rows in set (0.01 sec)

18 Find customers who have not placed any orders mysql> select \* from customers

-> where customer\_id not in (select distinct customer\_id from

orders);

**+ + + + +-**

**+**

| customer\_id | name | email | city | age |

**+ + + + +-**

**+**

| 8 | Hannah White | [hannah@gmail.com](mailto:hannah@gmail.com) | flhoenix |

22 |

| 9 | Isaac Clark | [isaac@gmail.com](mailto:isaac@gmail.com) | flhiladelphia |

30 |

| 10 | Julia Evans | [julia@gmail.com](mailto:julia@gmail.com) | Seattle |

33 |

| 11 | Kevin Lewis | [kevin@gmail.com](mailto:kevin@gmail.com) | Denver |

45 |

| 12 | Lena Hall | [lena@gmail.com](mailto:lena@gmail.com) | Austin |

27 |

| 13 | Mike Scott | [mike@gmail.com](mailto:mike@gmail.com) | Boston |

36 |

| 14 | ina Young | [nina@gmail.com](mailto:nina@gmail.com) | Atlanta |

31 |

| 15 | Oscar Martinez | [oscar@gmail.com](mailto:oscar@gmail.com) | ew York |

40 |

**+ + + + +-**

**+**

8 rows in set (0.00 sec)

19 Show All Orders fllaced in the Last 30 Days mysql> select \* from orders

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

Empty set (0.01 sec)

20. Find Orders Where the Total Amount is an Exact Multiple of 100

mysql> select \* from orders

-> where total\_amount % 100 = 0;

**+ + + + + +**

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

**+ + + + + +**

| 108 | 1 | 2024-02-08 | 200.00 | shipped |

| 112 | 5 | 2024-02-12 | 400.00 | shipped |

**+ + + + + +**

2 rows in set (0.00 sec)