



AJEENKYA
D Y PATIL UNIVERSITY
THE INNOVATION UNIVERSITY

School of
Engineering

Name: Apoorv Thorat

Program: B. Tech ITDS

Semester: 4

Section: B

URN: 2022-B-18072004

Subject: Data Analytics using SQL

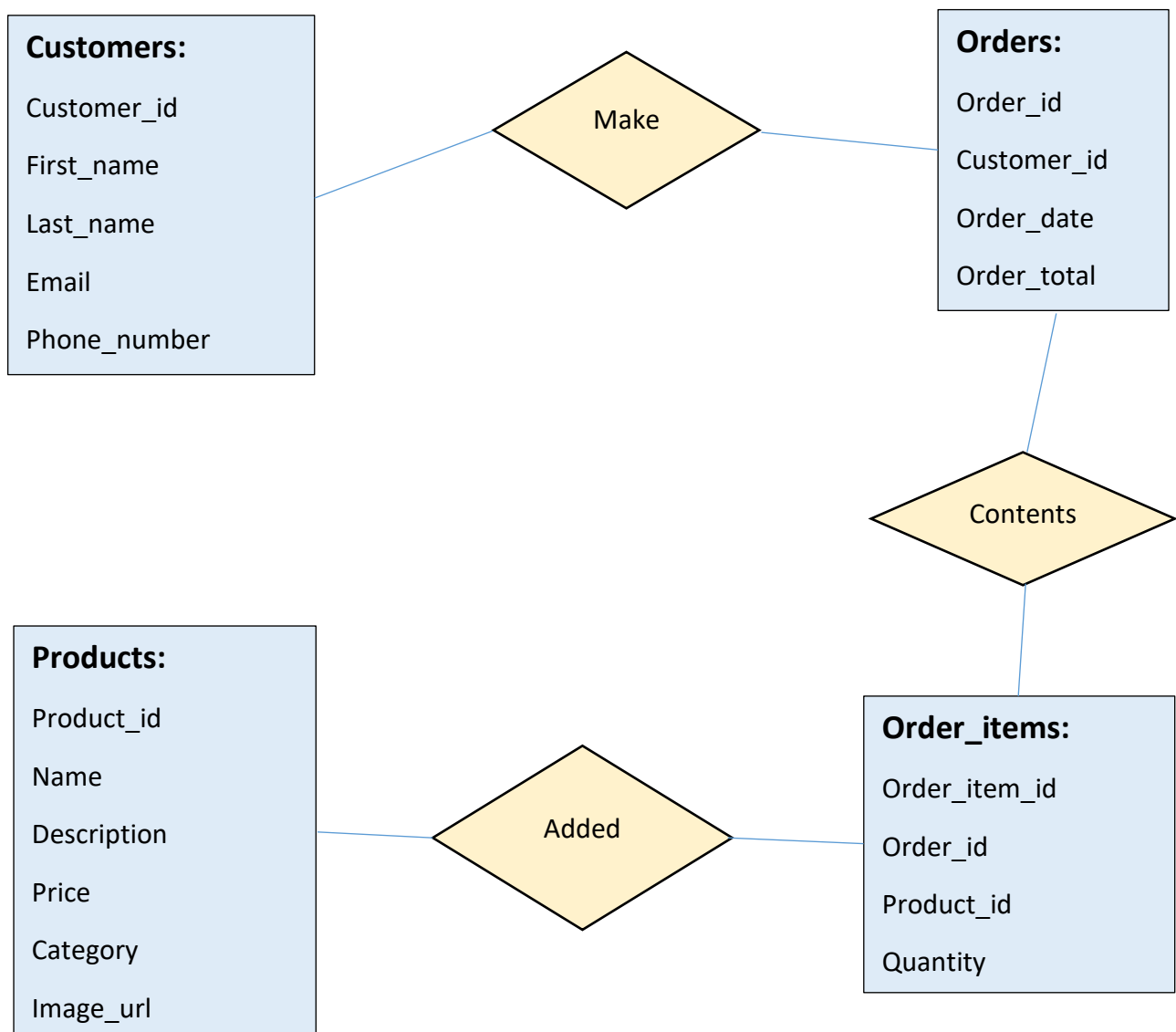
Professor: Dipmala Kamdi

Course Code: DS206E

Assignment 1

Problem Statement: - Creating a MySQL project for an Online Shopping System (Myntraa). Myntraa- like application involves designing a database to manage information related to clothes, customers, payments, orders, and other relevant entities.

ER-Diagram:



Database Schema:-

1. Customers table:

Customer_id, First_name, Last_name, Email, Phone_number

	customer_id	first_name	last_name	email	phone_number
▶	1	John	Doe	john.doe@example.com	+1234567890
	2	Jane	Smith	jane.smith@example.com	+9876543210
	3	Alice	Johnson	alice.johnson@example.com	+5551234567
	4	Michael	Brown	michael.brown@example.com	+8887654321
	5	David	Miller	david.miller@example.com	+4443210987
	6	Sarah	Garcia	sarah.garcia@example.com	+3332109876
	7	Emily	Clark	emily.clark@example.com	+2221098765
	8	Matthew	Lewis	matthew.lewis@example.com	+1110987654
	9	Jennifer	Robinson	jennifer.robinson@example.com	+9990876543
	10	Daniel	Williams	daniel.williams@example.com	+8880765432

2. Orders table:

Order_id, customer_id, order_date, order_total

	order_id	customer_id	order_date	order_total
▶	1	1	2024-04-10	150.00
	2	2	2024-04-11	225.50
	3	3	2024-04-12	80.75
	4	4	2024-04-08	312.99
	5	5	2024-04-09	107.20
	6	6	2024-04-10	450.00
	7	7	2024-04-11	188.40
	8	8	2024-04-08	271.15
	9	9	2024-04-09	623.50
	10	10	2024-04-10	99.99

3. Products table:

Product_id, name, description, price, category, image_url

	product_id	name	description	price	category	image_url
▶	1	Blue Denim Jeans	Classic straight-leg denim jeans	49.99	Clothing	https://example.com/denimjeans.jpg
	2	Cotton T-Shirt	Comfortable and stylish t-shirt	19.99	Clothing	https://example.com/tshirt.jpg
	3	Running Shoes	Lightweight shoes for running and exercise	79.99	Footwear	https://example.com/runningshoes.jpg
	4	Leather Wallet	Sleek and functional leather wallet	34.99	Accessories	https://example.com/wallet.jpg
	5	Wireless Headphones	High-quality sound for music and calls	99.99	Electronics	https://example.com/headphones.jpg
	6	Laptop Backpack	Durable backpack for carrying your laptop	54.99	Bags	https://example.com/backpack.jpg
	7	Wristwatch	Stylish and functional wristwatch	89.99	Accessories	https://example.com/watch.jpg
	8	Sunglasses	Protect your eyes from the sun in style	24.99	Accessories	https://example.com/sunglasses.jpg
	9	Dress	Elegant dress for any occasion	74.99	Clothing	https://example.com/dress.jpg
	10	Sports Bra	Supportive bra for workouts	29.99	Activewear	https://example.com/sportsbra.jpg

4. Order items table:

order_item_id, order_id, product_id, quantity

	order_item_id	order_id	product_id	quantity
▶	1	1	2	1
	2	2	4	2
	3	3	1	3
	4	4	7	1
	5	5	8	2
	6	6	3	1
	7	7	5	4
	8	8	6	2
	9	9	9	3
	10	10	10	1

Assignment 2

Questions:-

Let's consider only one table for generating 10 questions: -

Question 1: Retrieve first name of all the customers in the database.

Question: How to add column name 'age' in a table.

Question 3: Remove the column bag in the table customers.

Question 4: How to use WHERE clause in a table?

Question 5: Use AND clause in the following database.

Question 6: Use LIKE operator in the following database.

Question 7: Use NOT operator in the following database.

Question 8: Use OR operator in the following database.

Question 9: Use BETWEEN operators in the following database.

Question 10: Use IN operator in the following database.

Let's consider only two tables for generating 10 questions: -

Question 1: How many total orders were placed in the last week?

Question 2: What are the top 3 most expensive products in the Myntraa catalogue?

Question 3: For a specific customer (ID 10), what are their past 5 orders?

Question 4: What is the average order value (total spent per order) in the last month?

Question 5: Which product categories have seen the most sales in total quantity?

Question 6: Find all customers who have placed an order in the last month but haven't placed any orders before that.

Question 7: List the first name, last name, and total number of orders placed for each customer.

Question 8: Which customer has placed the most orders?

Question 9: Find all the customers who have not placed any orders in the last month.

Question 10: Find all orders placed before a specific date.

Let's consider only three tables for generating 10 questions: -

Question 1: What is the average order value for orders placed on weekends?

Question 2: Find all orders with a total amount exceeding Rs. 2000 that were placed by customers whose first name starts with the letter "A".

Question 3: Find all orders placed between two specific dates.

Question 4: List the 10 most recent orders placed.

Question 5: Find all products with a specific product name.

Question 6: List the 5 most expensive products in the catalogue.

Question 7: What is the total revenue generated from orders placed on a specific day of the week?

Question 8: List the 10 most recent orders placed.

Question 9: Find all customers with a specific email address.

Question 10: What percentage of customers who have placed an order in the last 3 months haven't placed any orders before that period?

Assignment 3

Question 1: Retrieve first name of all the customers in the database.

Ans. To retrieve all the first name in database, you can use SELECT query on the 'customers' table.

```
1 • select first_name
2   from customers;
```

Output:

Result Grid	
	first_name
▶	John
	Jane
	Alice
	Michael
	David
	Sarah
	Emily
	Matthew
	Jennifer
	Daniel

Question 2: How to add column name 'age' in a table.

Ans. To add column in a table we have to ALTER command.

```
1 • alter table customers
2   add age int;
```

Output:

Result Grid		Filter Rows:		Edit:		Export/Import:	
customer_id	first_name	last_name	email	phone_number	age		

Question 3: Remove the column bag in the table customers.

Ans. To remove a column bag in the table customers we need to use ALTER command.

```
1 • alter table customers
2 drop age;
```

Output:

	customer_id	first_name	last_name	email	phone_number
▶	1	John	Doe	john.doe@example.com	+1234567890
	2	Jane	Smith	jane.smith@example.com	+9876543210
	3	Alice	Johnson	alice.johnson@example.com	+5551234567
	4	Michael	Brown	michael.brown@example.com	+8887654321
	5	David	Miller	david.miller@example.com	+4443210987
	6	Sarah	Garcia	sarah.garcia@example.com	+3332109876
	7	Emily	Clark	emily.clark@example.com	+2221098765
	8	Matthew	Lewis	matthew.lewis@example.com	+1110987654
	9	Jennifer	Robinson	jennifer.robinson@example.com	+9990876543
	10	Daniel	Williams	daniel.williams@example.com	+8880765432

Question 4: How to use WHERE clause in a table?

Ans.

```
• SELECT * FROM orders
WHERE order_date >= '2024-04-01';
```

Output:

	order_id	customer_id	order_date	order_total
▶	1	1	2024-04-10	150.00
	2	2	2024-04-11	225.50
	3	3	2024-04-12	80.75
	4	4	2024-04-08	312.99
	5	5	2024-04-09	107.20
	6	6	2024-04-10	450.00
	7	7	2024-04-11	188.40
	8	8	2024-04-08	271.15
	9	9	2024-04-09	623.50
	10	10	2024-04-10	99.99

Question 5: Use AND clause in the following database.

Ans.

```
1  SELECT * FROM customers
2  WHERE first_name = 'Alice' AND last_name = 'Johnson';
3  |
```

Output:

	customer_id	first_name	last_name	email	phone_number
▶	3	Alice	Johnson	alice.johnson@example.com	+5551234567

Question 6: Use LIKE operator in the following database.

Ans.

```
• SELECT * FROM customers
  WHERE email NOT LIKE '%@gmail.com';
```

Output:

	customer_id	first_name	last_name	email	phone_number
▶	1	John	Doe	john.doe@example.com	+1234567890
	2	Jane	Smith	jane.smith@example.com	+9876543210
	3	Alice	Johnson	alice.johnson@example.com	+5551234567
	4	Michael	Brown	michael.brown@example.com	+8887654321
	5	David	Miller	david.miller@example.com	+4443210987
	6	Sarah	Garcia	sarah.garcia@example.com	+3332109876
	7	Emily	Clark	emily.clark@example.com	+2221098765
	8	Matthew	Lewis	matthew.lewis@example.com	+1110987654
	9	Jennifer	Robinson	jennifer.robinson@example.com	+9990876543
	10	Daniel	Williams	daniel.williams@example.com	+8880765432

Question 7: Use NOT operator in the following database.

Ans.

```
1 • SELECT * FROM customers
2 WHERE phone_number NOT LIKE '888%';
3
```

Output:

	customer_id	first_name	last_name	email	phone_number
▶	1	John	Doe	john.doe@example.com	+1234567890
	2	Jane	Smith	jane.smith@example.com	+9876543210
	3	Alice	Johnson	alice.johnson@example.com	+5551234567
	4	Michael	Brown	michael.brown@example.com	+8887654321
	5	David	Miller	david.miller@example.com	+4443210987
	6	Sarah	Garcia	sarah.garcia@example.com	+3332109876
	7	Emily	Clark	emily.clark@example.com	+2221098765
	8	Matthew	Lewis	matthew.lewis@example.com	+1110987654
	9	Jennifer	Robinson	jennifer.robinson@example.com	+9990876543
	10	Daniel	Williams	daniel.williams@example.com	+8880765432

Question 8: Use OR operator in the following database.

Ans.

```
1 • SELECT * FROM orders
2 WHERE order_date < '2024-04-08' OR order_date > '2024-04-10';
3
```

Output:

	order_id	customer_id	order_date	order_total
▶	2	2	2024-04-11	225.50
	3	3	2024-04-12	80.75
	7	7	2024-04-11	188.40

Question 9: Use BETWEEN operators in the following database.

Ans.

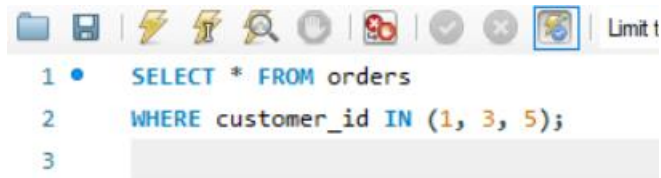
```
1 • SELECT * FROM orders
2 WHERE order_date BETWEEN '2024-04-08' AND '2024-04-11';
3
```

Output:

	order_id	customer_id	order_date	order_total
▶	1	1	2024-04-10	150.00
	2	2	2024-04-11	225.50
	4	4	2024-04-08	312.99
	5	5	2024-04-09	107.20
	6	6	2024-04-10	450.00
	7	7	2024-04-11	188.40
	8	8	2024-04-08	271.15
	9	9	2024-04-09	623.50
	10	10	2024-04-10	99.99

Question 10: Use IN operator in the following database.

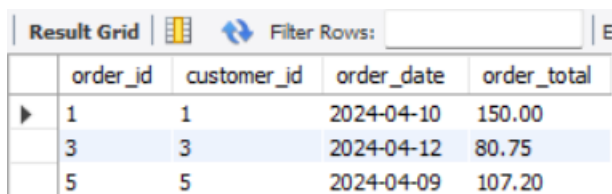
Ans.



The screenshot shows a SQL query editor with a toolbar at the top containing icons for file operations, execution, search, and window management. The query text is as follows:

```
1 • SELECT * FROM orders
2 WHERE customer_id IN (1, 3, 5);
3
```

Output:



The screenshot shows a 'Result Grid' with a toolbar at the top. The grid displays the results of the SQL query, with columns 'order_id', 'customer_id', 'order_date', and 'order_total'. The data is as follows:

	order_id	customer_id	order_date	order_total
▶	1	1	2024-04-10	150.00
	3	3	2024-04-12	80.75
	5	5	2024-04-09	107.20

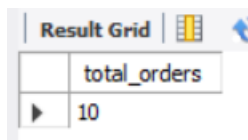
Assignment 4

Question 1: How many total orders were placed in the last week?

Ans.

```
1 • SELECT COUNT(*) AS total_orders
2   FROM orders
3   WHERE order_date >= DATE_SUB(CURDATE(), INTERVAL 7 DAY);
4
```

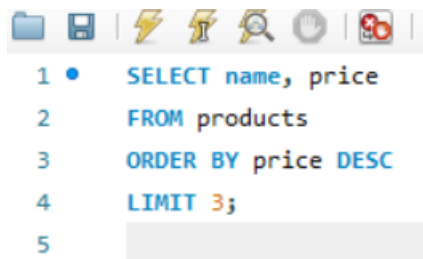
Output:



Result Grid	
	total_orders
▶	10

Question 2: What are the top 3 most expensive products in the Myntraa catalog?

Ans.



```
1 • SELECT name, price
2   FROM products
3   ORDER BY price DESC
4   LIMIT 3;
5
```

Output:

Result Grid			Filter Rows:
	name	price	
▶	Wireless Headphones	99.99	
	Wristwatch	89.99	
	Running Shoes	79.99	

Question 3: For a specific customer (ID 10), what are their past 5 orders?

Ans.

```
1 • SELECT * FROM orders
2 WHERE customer_id = 10
3 ORDER BY order_date DESC
4 LIMIT 5;
5
```

Output:

Result Grid					Filter Rows:	Edit
	order_id	customer_id	order_date	order_total		
▶	10	10	2024-04-10	99.99		

Question 4: What is the average order value (total spent per order) in the last month?

Ans.

```
1 • SELECT AVG(order_total) AS avg_order_value
2 FROM orders
3 WHERE order_date >= DATE_SUB(CURDATE(), INTERVAL 1 MONTH)
4 AND order_date < CURDATE();
5
```

Output:

Result Grid	Filter
avg_order_value	
250.948000	

Question 5: Which product categories have seen the most sales in total quantity?

Ans.

```
1 • SELECT p.category, SUM(oi.quantity) AS total_sales
2 FROM order_items oi
3 INNER JOIN products p ON oi.product_id = p.product_id
4 GROUP BY p.category
5 ORDER BY total_sales DESC
6 LIMIT 5;
7
```

Output:

Result Grid	Filter Rows:
category	total_sales
Clothing	7
Accessories	5
Electronics	4
Bags	2
Footwear	1

Question 6: Find all customers who have placed an order in the last month but haven't placed any orders before that.

Ans.

```
2 WITH NewCustomers AS (  
3     SELECT customer_id  
4     FROM orders  
5     WHERE order_date >= DATE_SUB(CURDATE(), INTERVAL 1 MONTH)  
6     GROUP BY customer_id  
7     HAVING COUNT(*) = 1  
8 )  
9 -- Step 2: Find these new customers in all orders  
10 SELECT * FROM customers c  
11 WHERE c.customer_id IN (  
12     SELECT customer_id FROM NewCustomers  
13 )  
14 AND c.customer_id NOT IN (  
15     SELECT customer_id FROM orders  
16     WHERE order_date < DATE_SUB(CURDATE(), INTERVAL 1 MONTH)  
17 )
```

Output:

Result Grid					
Filter Rows:		Export:		Wrap Cell Content: IA	
	customer_id	first_name	last_name	email	phone_number
▶	1	John	Doe	john.doe@example.com	+1234567890
	2	Jane	Smith	jane.smith@example.com	+9876543210
	3	Alice	Johnson	alice.johnson@example.com	+5551234567
	4	Michael	Brown	michael.brown@example.com	+8887654321
	5	David	Miller	david.miller@example.com	+4443210987
	6	Sarah	Garcia	sarah.garcia@example.com	+3332109876
	7	Emily	Clark	emily.clark@example.com	+2221098765
	8	Matthew	Lewis	matthew.lewis@example.com	+1110987654
	9	Jennifer	Robinson	jennifer.robinson@example.com	+9990876543
	10	Daniel	Williams	daniel.williams@example.com	+8880765432

Question 7: List the first name, last name, and total number of orders placed for each customer.

Ans.

```
1 • SELECT c.first_name, c.last_name, COUNT(o.order_id) AS total_orders
2 FROM customers c
3 INNER JOIN orders o ON c.customer_id = o.customer_id
4 GROUP BY c.customer_id, c.first_name, c.last_name;
```

Output:

Result Grid	Filter Rows:	Export
first_name	last_name	total_orders
John	Doe	1
Jane	Smith	1
Alice	Johnson	1
Michael	Brown	1
David	Miller	1
Sarah	Garcia	1
Emily	Clark	1
Matthew	Lewis	1
Jennifer	Robinson	1
Daniel	Williams	1

Question 8: Which customer has placed the most orders?

Ans.

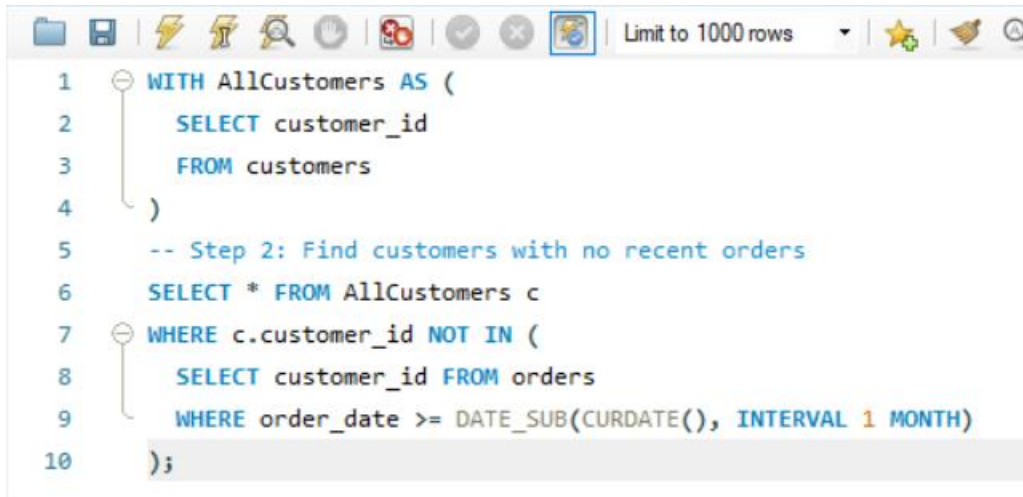
```
1 • SELECT c.first_name, c.last_name, COUNT(o.order_id) AS total_orders
2 FROM customers c
3 INNER JOIN orders o ON c.customer_id = o.customer_id
4 GROUP BY c.customer_id, c.first_name, c.last_name
5 ORDER BY total_orders DESC
6 LIMIT 1;
```

Output:

Result Grid	Filter Rows:	Export:
first_name	last_name	total_orders
John	Doe	1

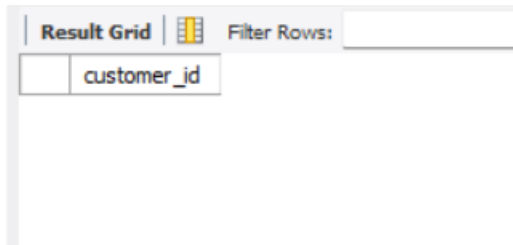
Question 9: Find all the customers who have not placed any orders in the last month.

Ans.



```
1 WITH AllCustomers AS (  
2     SELECT customer_id  
3     FROM customers  
4 )  
5 -- Step 2: Find customers with no recent orders  
6 SELECT * FROM AllCustomers c  
7 WHERE c.customer_id NOT IN (  
8     SELECT customer_id FROM orders  
9     WHERE order_date >= DATE_SUB(CURDATE(), INTERVAL 1 MONTH)  
10 );
```

Output:



customer_id

Question 10: Find all orders placed before a specific date.

Ans.



```
1 • SELECT * FROM orders WHERE order_date < '2024-04-20';
```

Output:

Result Grid		Filter Rows:		Edit:	
	order_id	customer_id	order_date	order_total	
▶	1	1	2024-04-10	150.00	
	2	2	2024-04-11	225.50	
	3	3	2024-04-12	80.75	
	4	4	2024-04-08	312.99	
	5	5	2024-04-09	107.20	
	6	6	2024-04-10	450.00	
	7	7	2024-04-11	188.40	
	8	8	2024-04-08	271.15	
	9	9	2024-04-09	623.50	
	10	10	2024-04-10	99.99	

Assignment 5

Question 1: What is the average order value for orders placed on weekends?

Ans.

```
1 • SELECT AVG(order_total) AS avg_weekend_order
2 FROM Orders
3 WHERE DAYNAME(order_date) IN ('Saturday', 'Sunday');
```

Output:

Result Grid		Filter Rows
	avg_weekend_order	
	NULL	

Question 2: Find all orders with a total amount exceeding Rs. 2000 that were placed by customers whose first name starts with the letter “A”.

Ans.

```
1 • SELECT o.order_id, c.first_name, c.last_name, o.order_total
2 FROM Orders o
3 INNER JOIN Customers c ON o.customer_id = c.customer_id
4 WHERE o.order_total > 100
5 AND c.first_name LIKE 'A%';
```

Output:

Result Grid		Filter Rows:	Exp
	order_id	first_name	last_name
	order_id	first_name	last_name

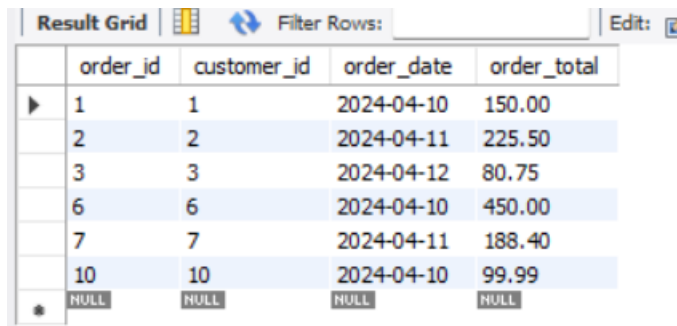
Question 3: Find all orders placed between two specific dates.

Ans.



```
1 • SELECT * FROM Orders
2 WHERE order_date >= '2024-04-10' AND order_date <= '2024-04-15';
```

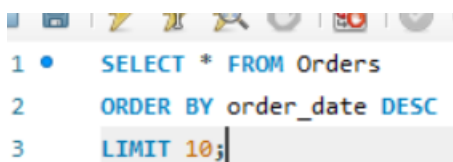
Output:



	order_id	customer_id	order_date	order_total
▶	1	1	2024-04-10	150.00
	2	2	2024-04-11	225.50
	3	3	2024-04-12	80.75
	6	6	2024-04-10	450.00
	7	7	2024-04-11	188.40
	10	10	2024-04-10	99.99
*	NULL	NULL	NULL	NULL

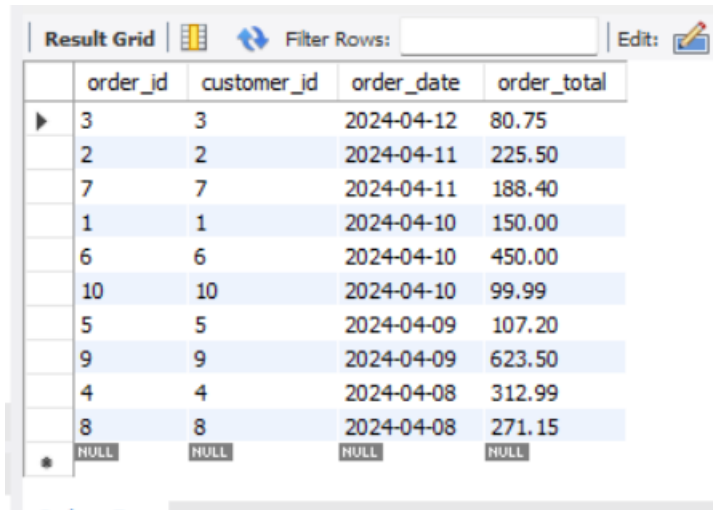
Question 4: List the 10 most recent orders placed.

Ans.



```
1 • SELECT * FROM Orders
2 ORDER BY order_date DESC
3 LIMIT 10;
```

Output:



The screenshot shows a database interface with a 'Result Grid' tab. The grid contains 11 rows of data. The first 10 rows represent individual orders, and the 11th row is a summary row with all NULL values. The columns are 'order_id', 'customer_id', 'order_date', and 'order_total'.

	order_id	customer_id	order_date	order_total
▶	3	3	2024-04-12	80.75
	2	2	2024-04-11	225.50
	7	7	2024-04-11	188.40
	1	1	2024-04-10	150.00
	6	6	2024-04-10	450.00
	10	10	2024-04-10	99.99
	5	5	2024-04-09	107.20
	9	9	2024-04-09	623.50
	4	4	2024-04-08	312.99
	8	8	2024-04-08	271.15
*	NULL	NULL	NULL	NULL

Question 5: Find all products with a specific product name.

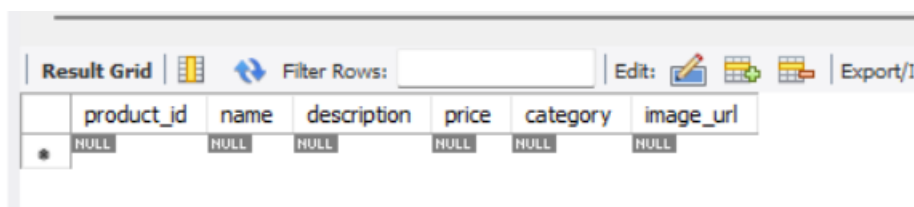
Ans.



The screenshot shows a SQL query editor with a toolbar at the top. The query is as follows:

```
1 • SELECT * FROM Products
2 WHERE name = 'T-Shirt';
```

Output:

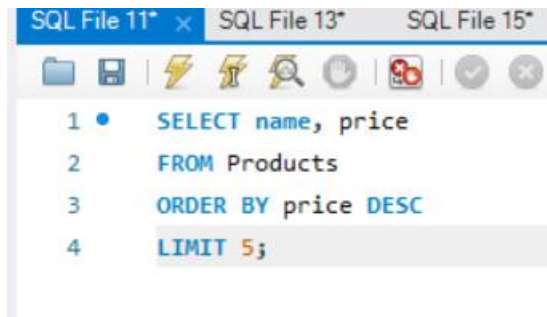


The screenshot shows a database interface with a 'Result Grid' tab. The grid contains 1 row of data, which is a summary row with all NULL values. The columns are 'product_id', 'name', 'description', 'price', 'category', and 'image_url'.

	product_id	name	description	price	category	image_url
*	NULL	NULL	NULL	NULL	NULL	NULL

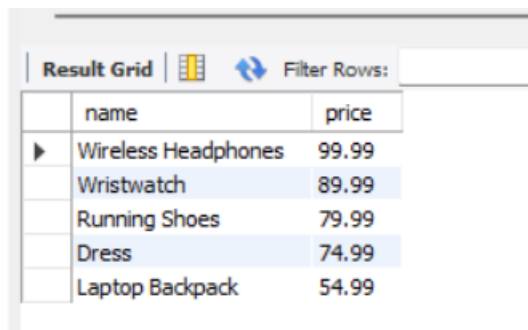
Question 6: List the 5 most expensive products in the catalogue.

Ans.



```
1 • SELECT name, price
2 FROM Products
3 ORDER BY price DESC
4 LIMIT 5;
```

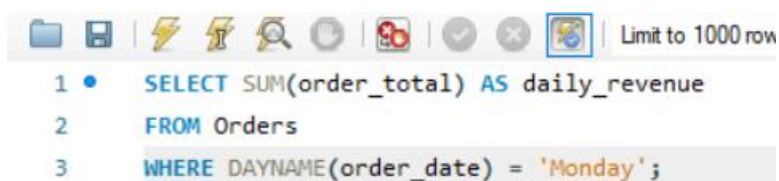
Output:



	name	price
▶	Wireless Headphones	99.99
	Wristwatch	89.99
	Running Shoes	79.99
	Dress	74.99
	Laptop Backpack	54.99

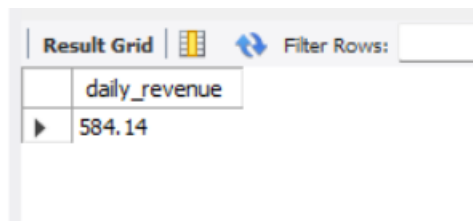
Question 7: What is the total revenue generated from orders placed on a specific day of the week?

Ans.



```
1 • SELECT SUM(order_total) AS daily_revenue
2 FROM Orders
3 WHERE DAYNAME(order_date) = 'Monday';
```

Output:

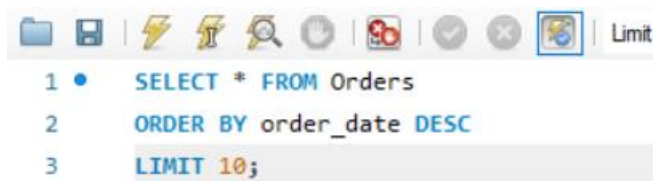


The screenshot shows a database interface with a 'Result Grid' tab. Below the tab, there is a table with one column labeled 'daily_revenue' and one row containing the value '584.14'.

daily_revenue
584.14

Question 8: List the 10 most recent orders placed.

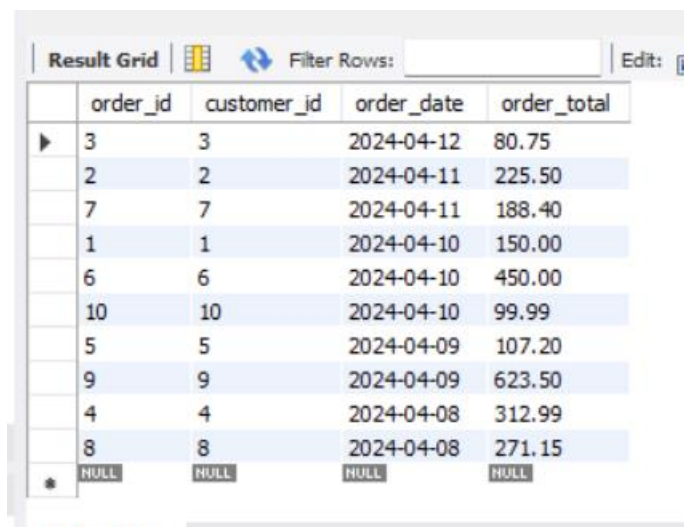
Ans.



The screenshot shows a SQL query editor with a toolbar at the top. The query text is as follows:

```
1 • SELECT * FROM Orders
2 ORDER BY order_date DESC
3 LIMIT 10;
```

Output:

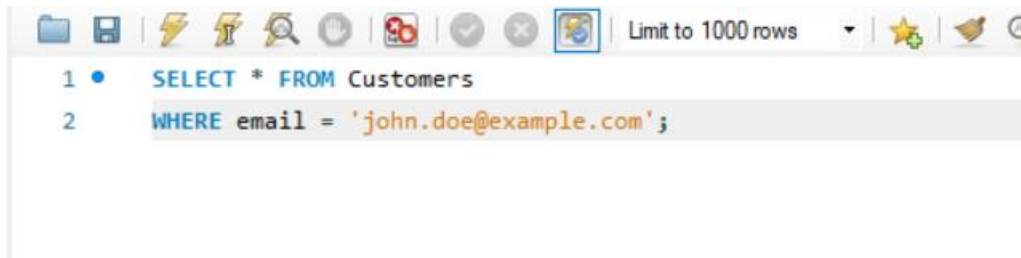


The screenshot shows a database interface with a 'Result Grid' tab. Below the tab, there is a table with 10 rows of data. The columns are 'order_id', 'customer_id', 'order_date', and 'order_total'. The data is as follows:

order_id	customer_id	order_date	order_total
3	3	2024-04-12	80.75
2	2	2024-04-11	225.50
7	7	2024-04-11	188.40
1	1	2024-04-10	150.00
6	6	2024-04-10	450.00
10	10	2024-04-10	99.99
5	5	2024-04-09	107.20
9	9	2024-04-09	623.50
4	4	2024-04-08	312.99
8	8	2024-04-08	271.15

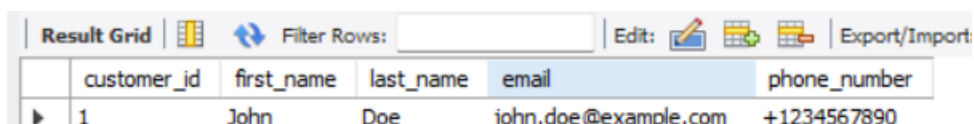
Question 9: Find all customers with a specific email address.

Ans.



```
1 • SELECT * FROM Customers
2 WHERE email = 'john.doe@example.com';
```

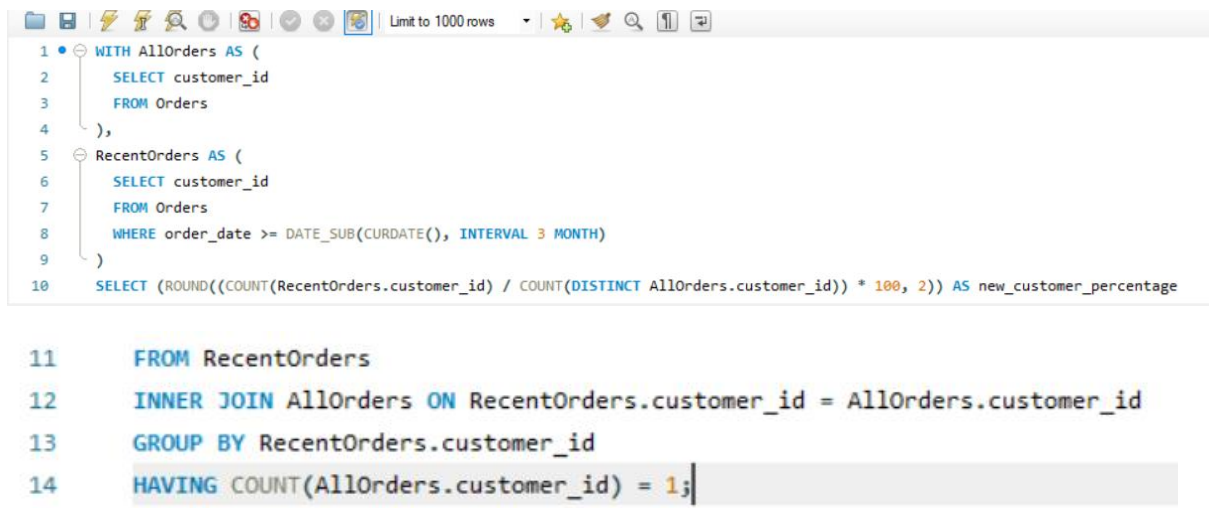
Output:



	customer_id	first_name	last_name	email	phone_number
▶	1	John	Doe	john.doe@example.com	+1234567890

Question 10: What percentage of customers who have placed an order in the last 3 months haven't placed any orders before that period?

Ans.



```
1 • WITH AllOrders AS (
2     SELECT customer_id
3     FROM Orders
4 ),
5 RecentOrders AS (
6     SELECT customer_id
7     FROM Orders
8     WHERE order_date >= DATE_SUB(CURDATE(), INTERVAL 3 MONTH)
9 )
10 SELECT (ROUND((COUNT(RecentOrders.customer_id) / COUNT(DISTINCT AllOrders.customer_id)) * 100, 2)) AS new_customer_percentage
11 FROM RecentOrders
12 INNER JOIN AllOrders ON RecentOrders.customer_id = AllOrders.customer_id
13 GROUP BY RecentOrders.customer_id
14 HAVING COUNT(AllOrders.customer_id) = 1;
```

Output:

[illegible]