LUMEN

Report of Nandyala Venkata Surendra

Practice Assignment - SQL A dvance

Attempt 1 | Submitted on Feb 21 2025 08:17:38

Programming Lab Exercise Result

Powered by UNEXT LABS

Practice Assignment - SQL Advance

Status: **graded** Attempts: **1/3 taken** Validity: **Feb 18 2025 - Feb 28 2025** Time Limit: **120 mins** Marks: **100**

Languages: **dbms , dbms , dbms , dbms , dbms** Archetypes: **mysql_large_dataset , mysql_large_dataset , mysql_large_dataset , mysql_large_dataset , oracle_large_dataset**

Overall Performance

118 mins 19 sec

Time Taken to Complete

Start: Feb 21 2025 08:15:57

End: Feb 21 2025 08:17:38

5/5

Projects Attempted

You have attempted 5 projects

100.00/ 100

Exercise Marks

You have received 100.00%

✓ **Passed**

Exercise Status

Percentage for passing - 60%

Project Performance

1. Masking Customer Email for Security

Level: **Medium** **DBMS > MYSQL_LARGE_DATASET**

20.0/20 marks

2. Number of Orders Placed by Each Customer

Level: **Difficult** **DBMS > MYSQL_LARGE_DATASET**

20.0/20 marks

3. Product Return Analysis

Level: **Difficult** **DBMS > MYSQL_LARGE_DATASET**

20.0/20 marks

4. Most Frequently Used Payment Method

Level: **Easy** **DBMS > MYSQL_LARGE_DATASET**

20.0/20 marks

5. CreditCard With Balance

Level: **Medium** **DBMS > ORACLE_LARGE_DATASET**

20.0/20 marks

Project 1: Masking Customer Email for Security

Marks

Compilation:	0%	0
User Black Box:	0%	0
Evaluator Black Box:	0%	0
User White Box:	0%	0
Evaluator White Box:	0%	0

Total Marks Received 20.00

Plagiarism- Not Available

Code Quality- Not Available

Final Marks Received **20.00**

Test Cases

User Black Box

Evaluator Black Box

User White Box

Evaluator White Box

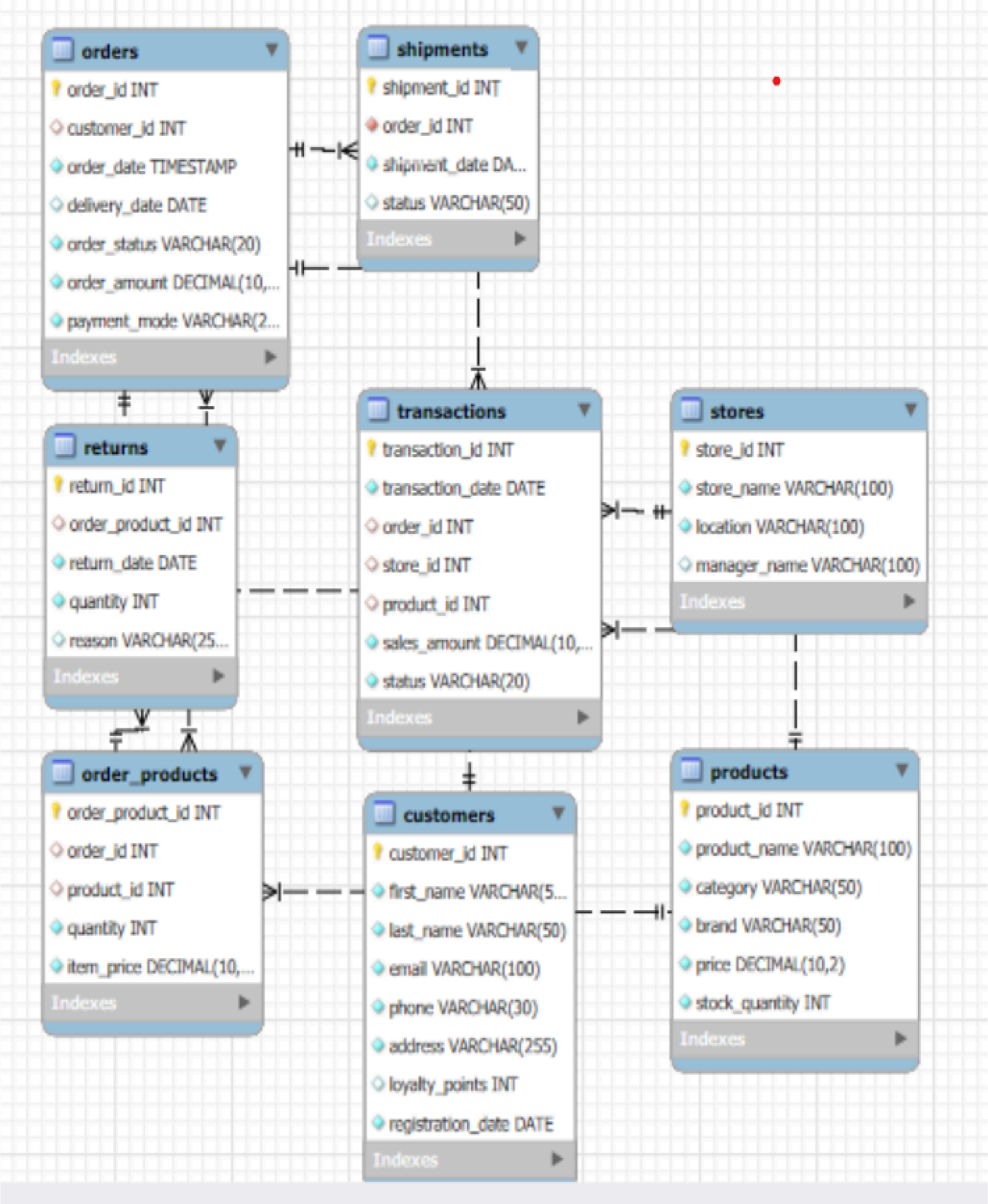
Problem Statement

For data privacy, the Retail company wants to **mask** part of customers’ email addresses. The format should be:

`j*****@gmail.com` (Mask everything except the first letter and domain).

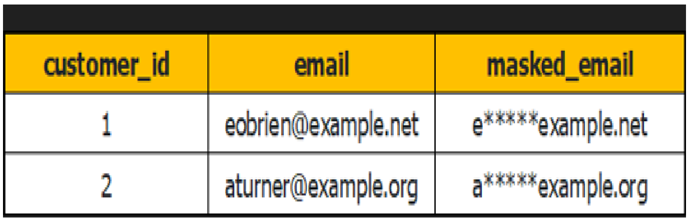
Write a query to retreive customerid,email and their masked email.Restrict the query to fetch 100 rows only

Instructions



Example

Note : Sample output is provided.Provide starting letter of table name in uppercase in your query(Eg.,Customers)



customer_id	email	masked_email
1	eobrien@example.net	e*****example.net
2	aturner@example.org	a*****example.org

Submitted Code

```
	select customer_id,email,CONCAT(
SUBSTR(email, 1, 1),
CONCAT(
   RPAD('', INSTR(email, '@') - 2, '*'),
   CONCAT(
      '@',
            SUBSTR(email, INSTR(email, '@') + 1)
        )
    )
) AS masked_email from Customers

limit 100
```

Project 2: Number of Orders Placed by Each Customer

Marks

Compilation:	0%	0
User Black Box:	0%	0
Evaluator Black Box:	0%	0
User White Box:	0%	0
Evaluator White Box:	0%	0

Total Marks Received 20.00

Plagiarism- Not Available

Code Quality- Not Available

Final Marks Received **20.00**

Test Cases

User Black Box

Evaluator Black Box

User White Box

Evaluator White Box

Problem Statement

The marketing team of the retail store wants to identify **customer engagement trends** by analyzing how frequently customers place orders. This data will be used to **segment customers** for personalized marketing campaigns.

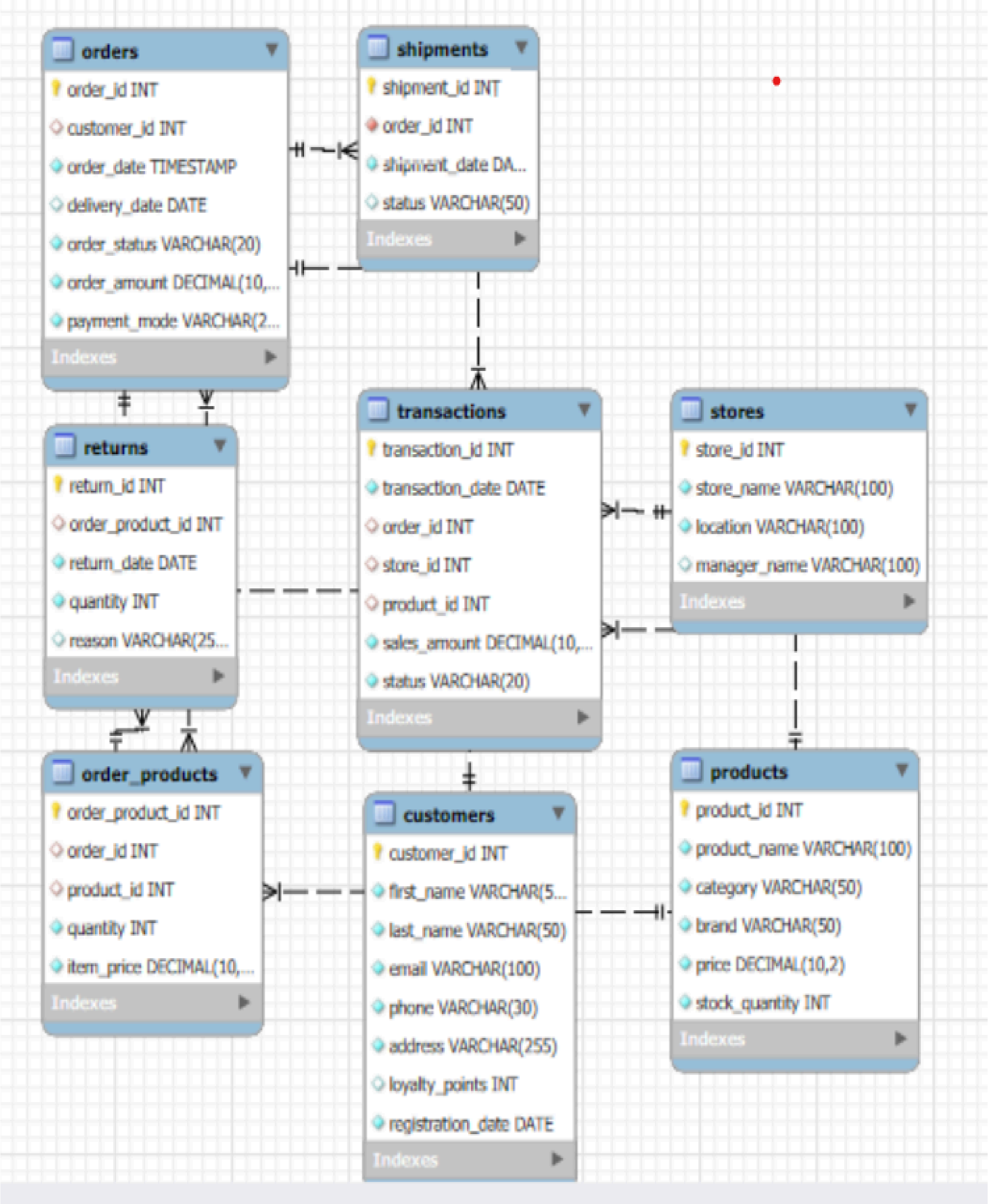
Customers who place **frequent orders** may receive **loyalty rewards** or **exclusive discounts**.

Customers who have placed only **one or two orders** might receive promotional emails encouraging repeat purchases.

Problem Statement:

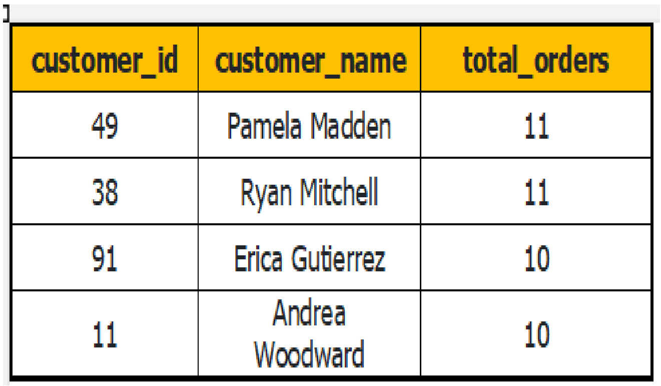
Write a SQL query to **fetch the customer ID, full name, and total number of orders placed** by each customer. The result should include **all customers**, even those who have never placed an order.Restrict query to fetch 100 rows only.

Instructions



Example

Note : Sample outut is provided. Provide starting letter of table name in uppercase in your query(Eg.,Customers)



customer_id	customer_name	total_orders
49	Pamela Madden	11
38	Ryan Mitchell	11
91	Erica Gutierrez	10
11	Andrea Woodward	10

Submitted Code

```
        select c.customer_id,concat(c.first_name,concat(' ',c.last_name)) as customer_name,count(order_id) as total_orders from Customers c join Orders o on c.customer_id=o.customer_id
group by customer_name order by count(order_id) desc;
```

Project 3: Product Return Analysis

Marks		
Compilation:	0%	0
User Black Box:	0%	0
Evaluator Black Box:	0%	0
User White Box:	0%	0
Evaluator White Box:	0%	0

Total Marks Received 20.00

Plagiarism- Not Available

Code Quality- Not Available

Final Marks Received **20.00**

Test Cases
User Black Box
Evaluator Black Box
User White Box
Evaluator White Box

Problem Statement

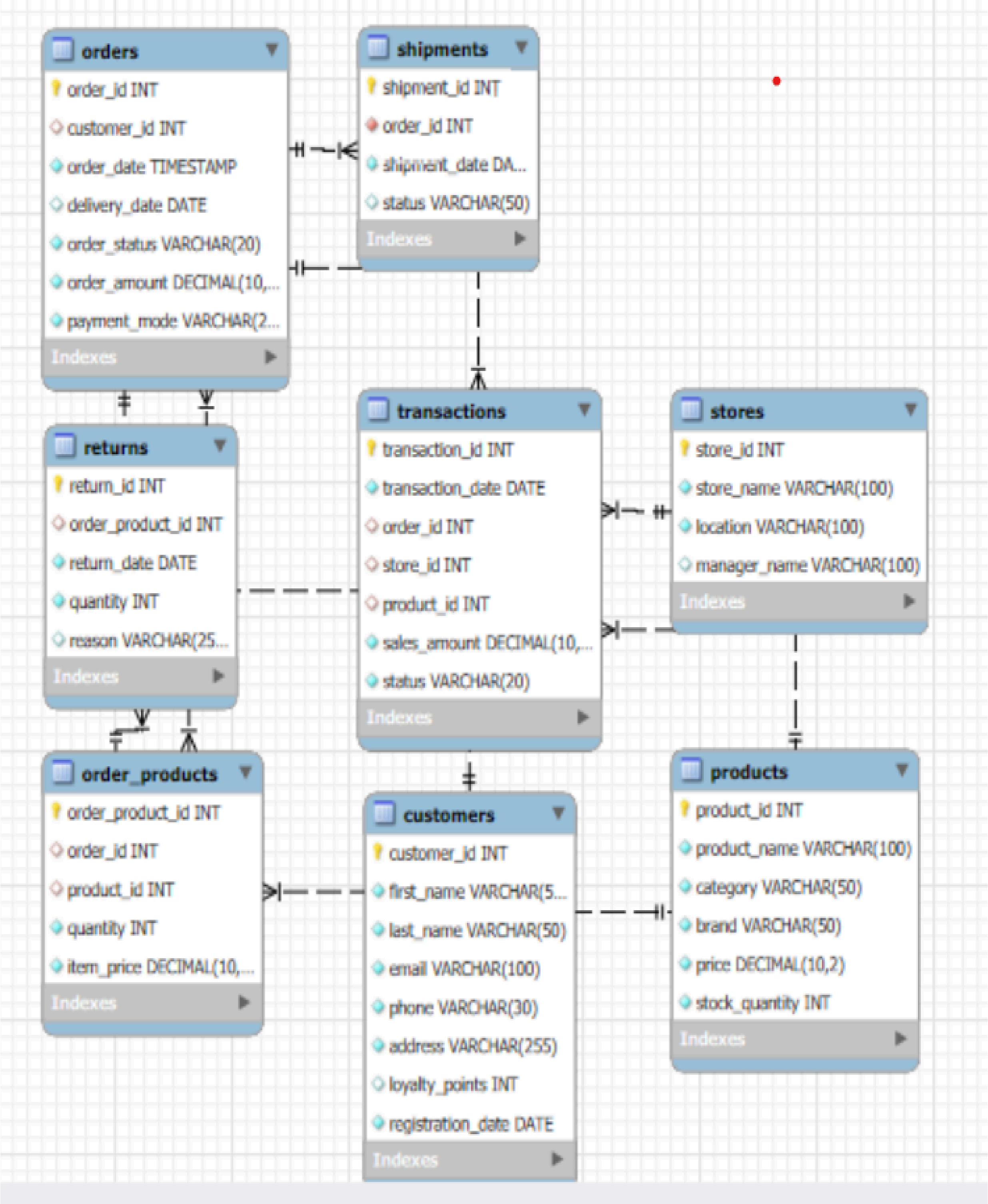
As part of our ongoing quality assurance and inventory management efforts, we need to identify products that are frequently returned by customers. To address potential quality concerns, generate a report that displays the **total number of returns** for each product.

The report should include:

- The **product name** to clearly identify the returned items.
- The **total number of returns** for each product to measure the frequency of returns.

This information will help our quality control and supply chain teams analyze trends in product returns and take necessary corrective actions, such as improving product quality, updating supplier agreements, or revising return policies.

Instructions



orders
order_id INT
customer_id INT
order_date TIMESTAMP
delivery_date DATE
order_status VARCHAR(20)
order_amount DECIMAL(10,...
payment_mode VARCHAR(2...
Indexes

shipments
shipment_id INT
order_id INT
shipment_date DA...
status VARCHAR(50)
Indexes

returns
return_id INT
order_product_id INT
return_date DATE
quantity INT
reason VARCHAR(25...
Indexes

transactions
transaction_id INT
transaction_date DATE
order_id INT
store_id INT
product_id INT
sales_amount DECIMAL(10,...
status VARCHAR(20)
Indexes

stores
store_id INT
store_name VARCHAR(100)
location VARCHAR(100)
manager_name VARCHAR(100)
Indexes

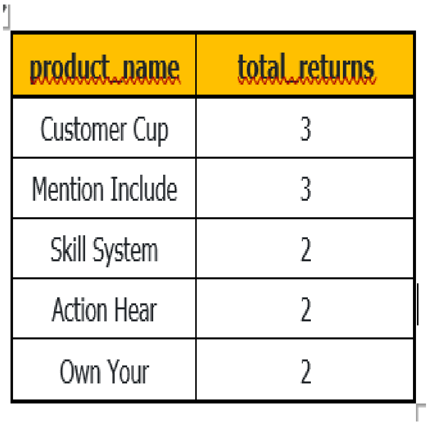
order_products
order_product_id INT
order_id INT
product_id INT
quantity INT
item_price DECIMAL(10,...
Indexes

customers
customer_id INT
first_name VARCHAR(5...
last_name VARCHAR(50)
email VARCHAR(100)
phone VARCHAR(30)
address VARCHAR(255)
loyalty_points INT
registration_date DATE
Indexes

products
product_id INT
product_name VARCHAR(100)
category VARCHAR(50)
brand VARCHAR(50)
price DECIMAL(10,2)
stock_quantity INT
Indexes

Example

Note: Sample output is provided.Your Actual Output contains 61 rows.Provide starting letter of table name in uppercase in your query(Eg.,Customers,Order_Products etc.,)



product_name	total_returns
Customer Cup	3
Mention Include	3
Skill System	2
Action Hear	2
Own Your	2

Submitted Code

```
            select p.product_name ,count(r.return_id) as total_returns from Products p join Order_Products op on p.product_id=op.product_id join Returns r on
op.order_product_id=r.order_product_id group by p.product_name order  by  count(r.return_id) desc
```

Project 4: Most Frequently Used Payment Method

Marks

Compilation:	0%	0
User Black Box:	0%	0
Evaluator Black Box:	0%	0
User White Box:	0%	0
Evaluator White Box:	0%	0

Total Marks Received	**20.00**

Plagiarism- Not Available

Code Quality- Not Available

Final Marks Received	**20.00**

Test Cases

User Black Box

Evaluator Black Box

User White Box

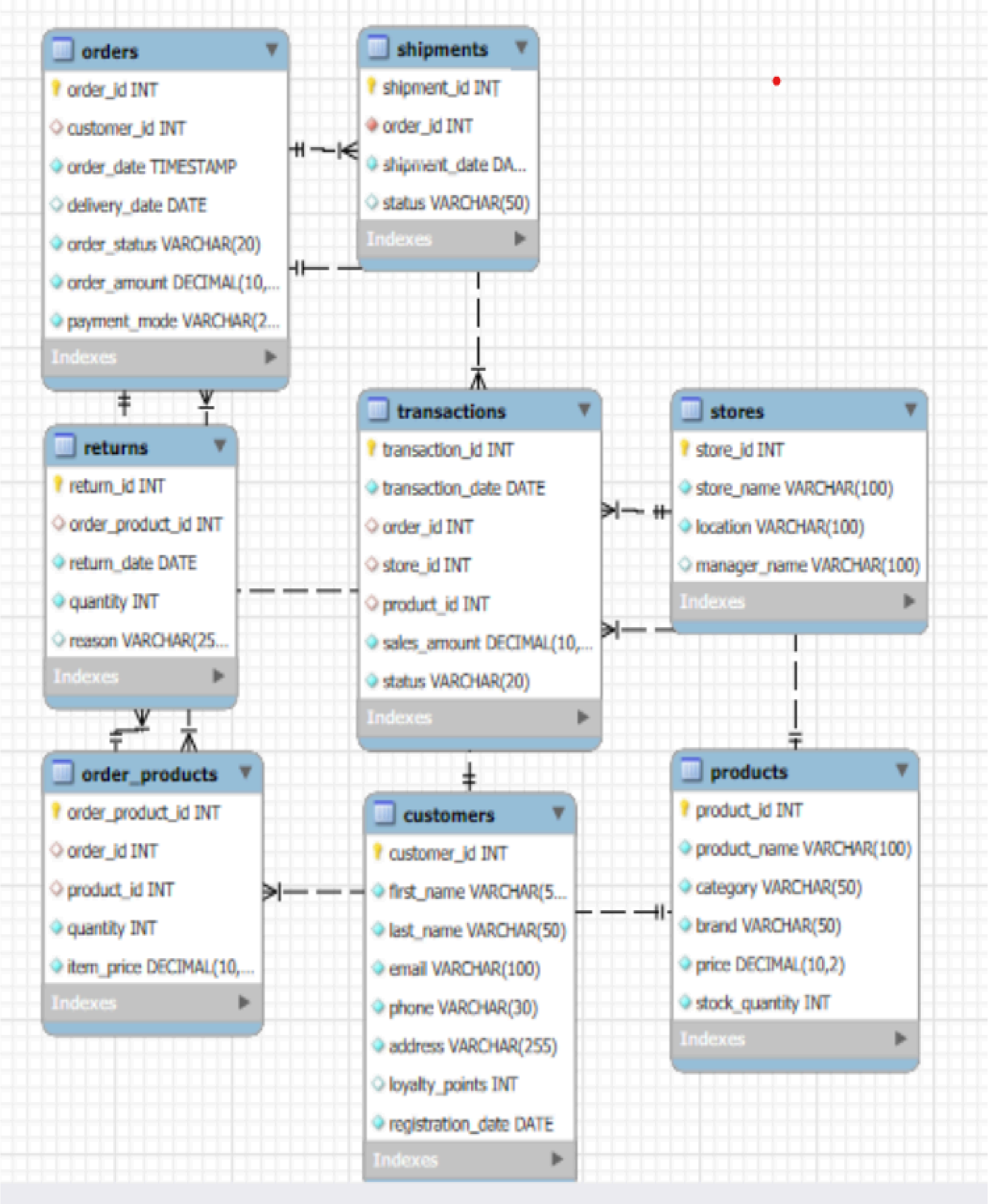
Evaluator White Box

Problem Statement

In the retail industry, understanding customer payment preferences is essential for improving sales strategies, optimizing checkout processes, and enhancing customer experience. This query helps retail businesses analyze which payment methods (e.g., credit card, debit card, cash, PayPal, digital wallets) are most frequently used by customers when placing orders.

Write a SQL query to retrieve the count of orders for each payment method, sorted in descending order of usage.Provide column alias as in output

Instructions



orders
- order_id INT
- customer_id INT
- order_date TIMESTAMP
- delivery_date DATE
- order_status VARCHAR(20)
- order_amount DECIMAL(10,...
- payment_mode VARCHAR(2...
- Indexes

shipments
- shipment_id INT
- order_id INT
- shipment_date DA...
- status VARCHAR(50)
- Indexes

returns
- return_id INT
- order_product_id INT
- return_date DATE
- quantity INT
- reason VARCHAR(25...
- Indexes

transactions
- transaction_id INT
- transaction_date DATE
- order_id INT
- store_id INT
- product_id INT
- sales_amount DECIMAL(10,...
- status VARCHAR(20)
- Indexes

stores
- store_id INT
- store_name VARCHAR(100)
- location VARCHAR(100)
- manager_name VARCHAR(100)
- Indexes

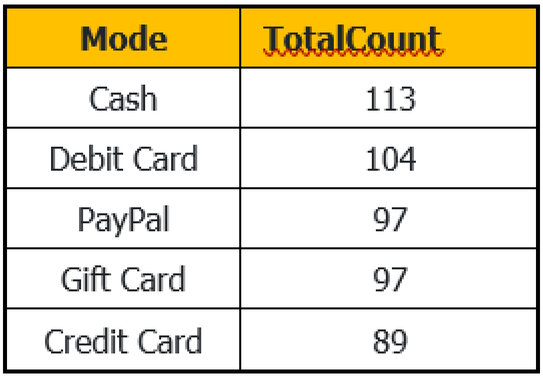
order_products
- order_product_id INT
- order_id INT
- product_id INT
- quantity INT
- item_price DECIMAL(10,...
- Indexes

customers
- customer_id INT
- first_name VARCHAR(5...
- last_name VARCHAR(50)
- email VARCHAR(100)
- phone VARCHAR(30)
- address VARCHAR(255)
- loyalty_points INT
- registration_date DATE
- Indexes

products
- product_id INT
- product_name VARCHAR(100)
- category VARCHAR(50)
- brand VARCHAR(50)
- price DECIMAL(10,2)
- stock_quantity INT
- Indexes

Example

Note : Provide starting letter of table name in uppercase in your query(Eg.,Customers)



Mode	TotalCount
Cash	113
Debit Card	104
PayPal	97
Gift Card	97
Credit Card	89

Submitted Code

```
        select payment_mode as Mode,count(payment_mode) as TotalCount  from Orders
group by payment_mode order by count(payment_mode)  desc
```

Project 5: CreditCard With Balance

Marks

Compilation:	0%	0
User Black Box:	0%	0
Evaluator Black Box:	0%	0
User White Box:	0%	0
Evaluator White Box:	0%	0

Total Marks Received 20.00

Plagiarism- Not Available

Code Quality- Not Available

Final Marks Received **20.00**

Test Cases

User Black Box

Evaluator Black Box

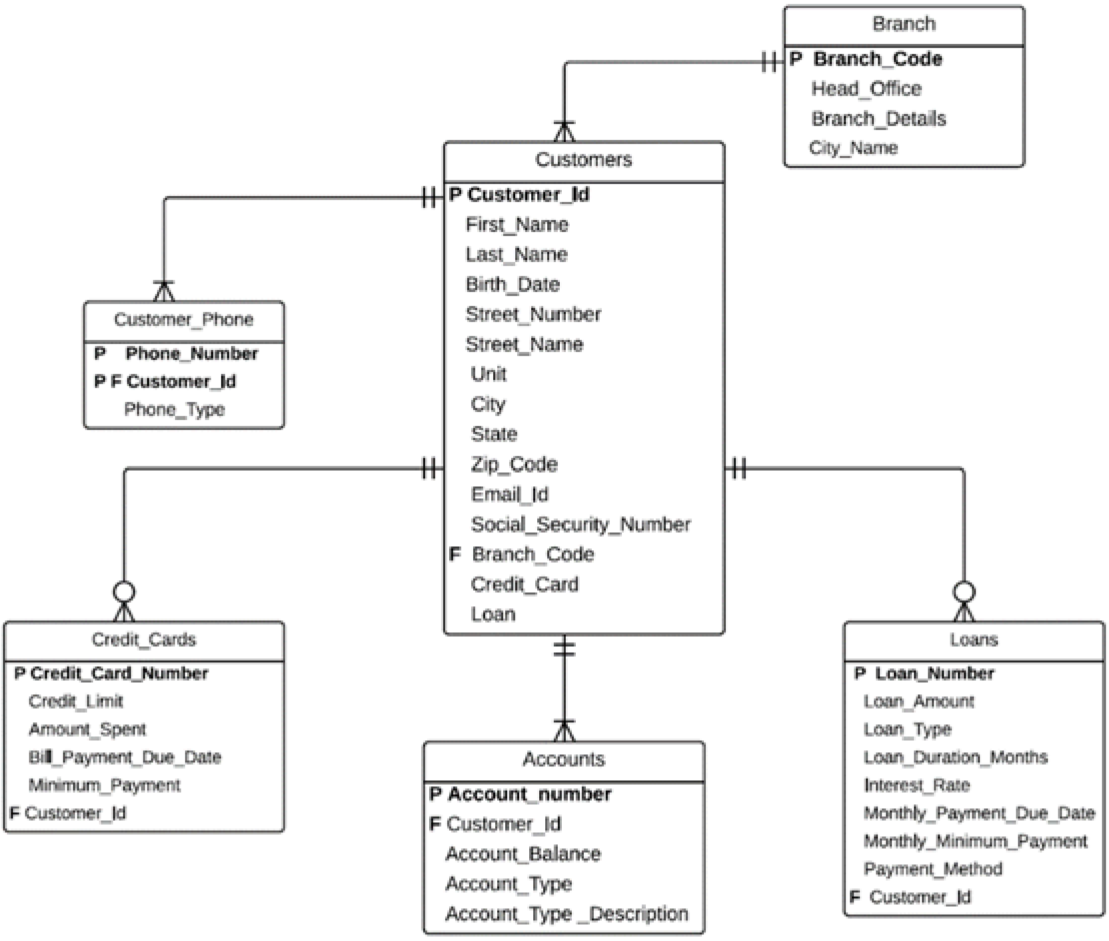
User White Box

Evaluator White Box

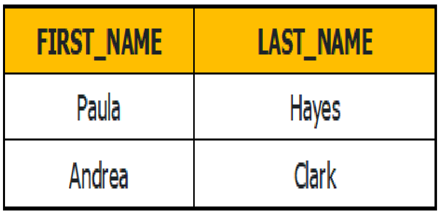
Problem Statement

Write a query to retrieve the First name and Last Name of customers who have both a credit card and at least one account with a balance greater than 9800.

Instructions



Example



FIRST_NAME	LAST_NAME
Paula	Hayes
Andrea	Clark

Submitted Code

```
        SELECT DISTINCT c.first_name, c.last_name
FROM Customers c
JOIN Accounts a ON c.customer_id = a.customer_id
JOIN Credit_cards cc ON c.customer_id = cc.customer_id
WHERE a.Account_balance > 9800;
```