

E-COMMERCE MANAGEMENT SYSTEM

NAME: SURAKSHA JADHAV
NUID: 001430368

Overview

- **Electronic Commerce** or **E-Commerce** consists of the purchasing, selling and exchanging goods and services via electronic means such as the internet or other electronic services.
- This type of trade has been growing rapidly because of the expansion of the Internet.

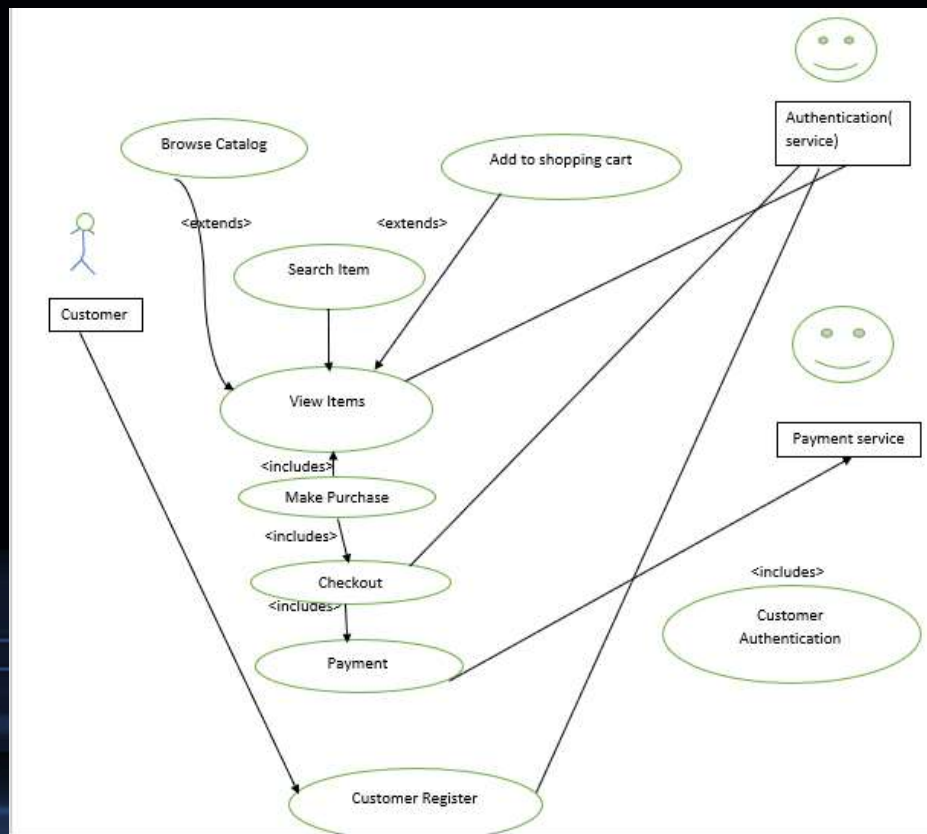
Problem Statement

- E-commerce provides an easy way to sell products to a large customer base. However, there is a lot of competition among multiple e-commerce sites. When users land on an e-commerce site, they expect to find what they are looking for quickly and easily. And the main goal for all of this is maintaining a database which will be able to handle such a large amount of data.

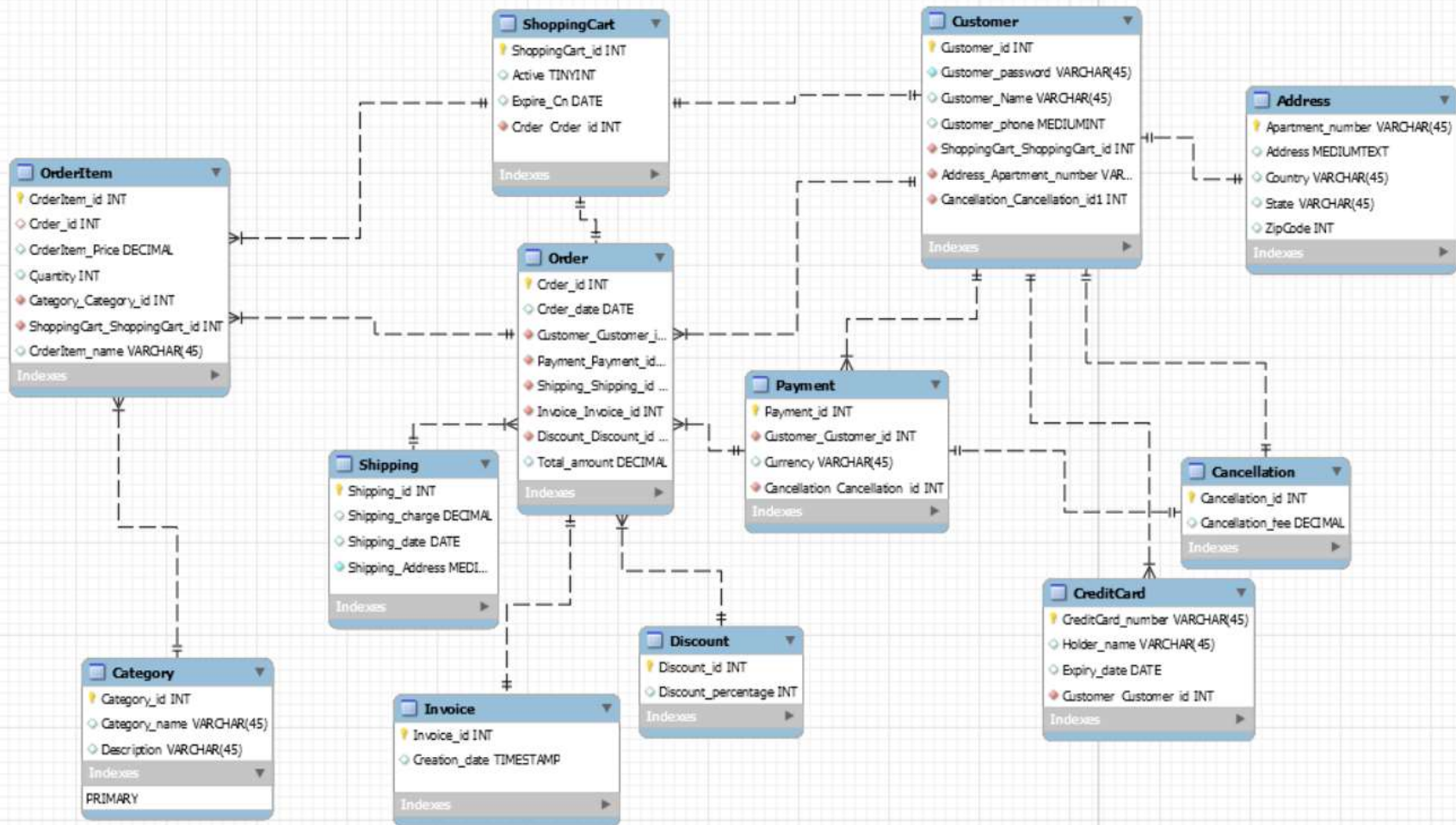
Proposed Solution

- The purpose of any e-commerce website is to help customers narrow down their broad ideas and enable them to finalize the products they want to purchase quickly.
- The database management system we construct should possess all the ACID properties.

Sample Diagram and Entities



ER Diagram



Query 1 : Trigger for changing the status

```
1  delimiter //
2  ● create trigger active_inactive_status1 before insert on ShoppingCart
3  for each row
4  begin
5  if NEW.Active='0'
6  then
7  SET NEW.Active='1';
8  end if;
9  end;
```

Limit to 1000 rows

```
1 ● insert into ShoppingCart(shoppingcart_id,Active,Expire_On,Order_Order_id) values (11,0,'2019-01-10',5);
```

Query 1 Continued..

JoinshippingorderitemcategorySQL File 5*shoppingcart

1

SELECT * FROM dmdd.shoppingcart;

Result Grid

Filter Rows:

Edit:

Export/Imp

	ShoppingCart_id	Active	Expire_On	Order_Order_id
	6	1	2018-12-15	1
	7	0	2018-11-10	2
	8	1	2018-12-20	3
	9	1	2018-12-19	4
	10	1	2019-01-02	5
	11	1	2019-01-10	5
*	NULL	NULL	NULL	NULL

Query 2: Order Discount for Products

hoppingcart orderdiscount - Routine order_discount - View address order discount customer Discount

Name: orderdiscount The name of the routine is parsed automatically from the DDL statement. The DDL is parsed automatically while you type.

DDL:

```
1 CREATE DEFINER='root'@'localhost' FUNCTION `orderdiscount` (amount int, discount i
2 BEGIN
3 DECLARE total DECIMAL(9,2);
4 SET total =amount-(amount*discount*0.01);
5
6 RETURN total;
7 END
```

hoppingcart orderdiscount - Routine order_discount - View address order discount customer Discount

Name: order_discount The name of the view is parsed automatically from the DDL statement. The DDL is parsed automatically while you type.

DDL:

```
1 CREATE
2 ALGORITHM = UNDEFINED
3 DEFINER = `root`@`localhost`
4 SQL SECURITY DEFINER
5 VIEW `order_discount` AS
6 SELECT
7 `c`.`Customer_Name` AS `Customer_name`,
8 `order`.`Total_amount` AS `Total_amount`,
9 ORDERDISCOUNT(`order`.`Total_amount`,
10 `d`.`Discount_percentage`) AS `total_discount`
11 FROM
12 ((`order`
13 JOIN `customer` `c` ON ((`c`.`Customer_id` = `order`.`Customer_Customer_id`
14 JOIN `discount` `d` ON ((`d`.`Discount_id` = `order`.`Discount_Discount_id`
```

Query 2 continued..

```
1 • SELECT distinct * FROM dmdd.order_discount;
```

<





Result Grid | | Filter Rows: | Export: | Wrap Cell Content:

	Customer_name	Total_amount	total_discount
▶	Ling	49	17.15
	Sai	150	120.00
	Liu	120	108.00
	Sue	51	38.25
	Sammy	15	6.75

Query 2 continued..

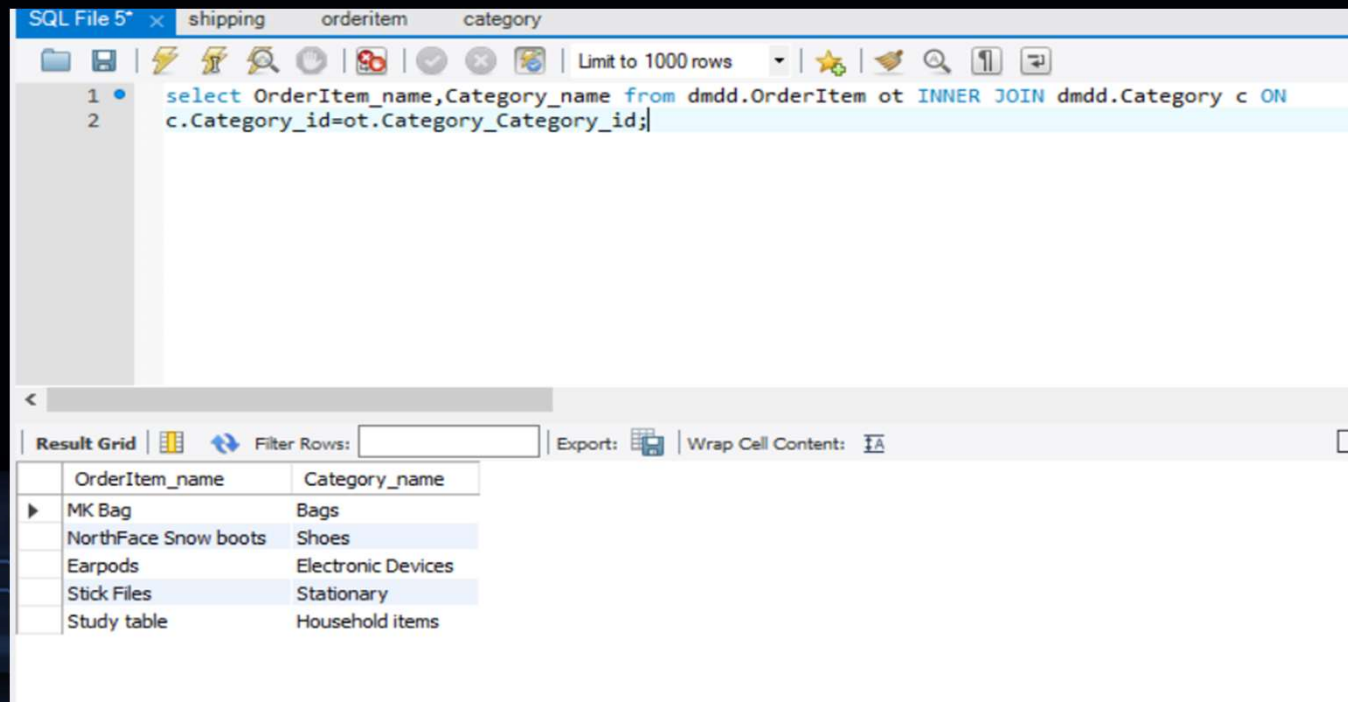
```
1 • Select sum(Total_amount)-sum(total_discount) as Discount_offer from order_discount;  
2 /*analysis for discounts given*/
```

<

Result Grid   Filter Rows: Export:  Wrap Cell Content: 

Discount_offer
251.60

Query 3: For selecting Products from respective Category



The screenshot displays a SQL query editor window with a tab labeled "SQL File 5". The query is as follows:

```
1 select OrderItem_name,Category_name from dmdd.OrderItem ot INNER JOIN dmdd.Category c ON
2 c.Category_id=ot.Category_Category_id;
```

Below the query editor, the "Result Grid" is shown, displaying the results of the query. The grid has two columns: "OrderItem_name" and "Category_name". The results are as follows:

OrderItem_name	Category_name
MK Bag	Bags
NorthFace Snow boots	Shoes
Earpods	Electronic Devices
Stick Files	Stationary
Study table	Household items

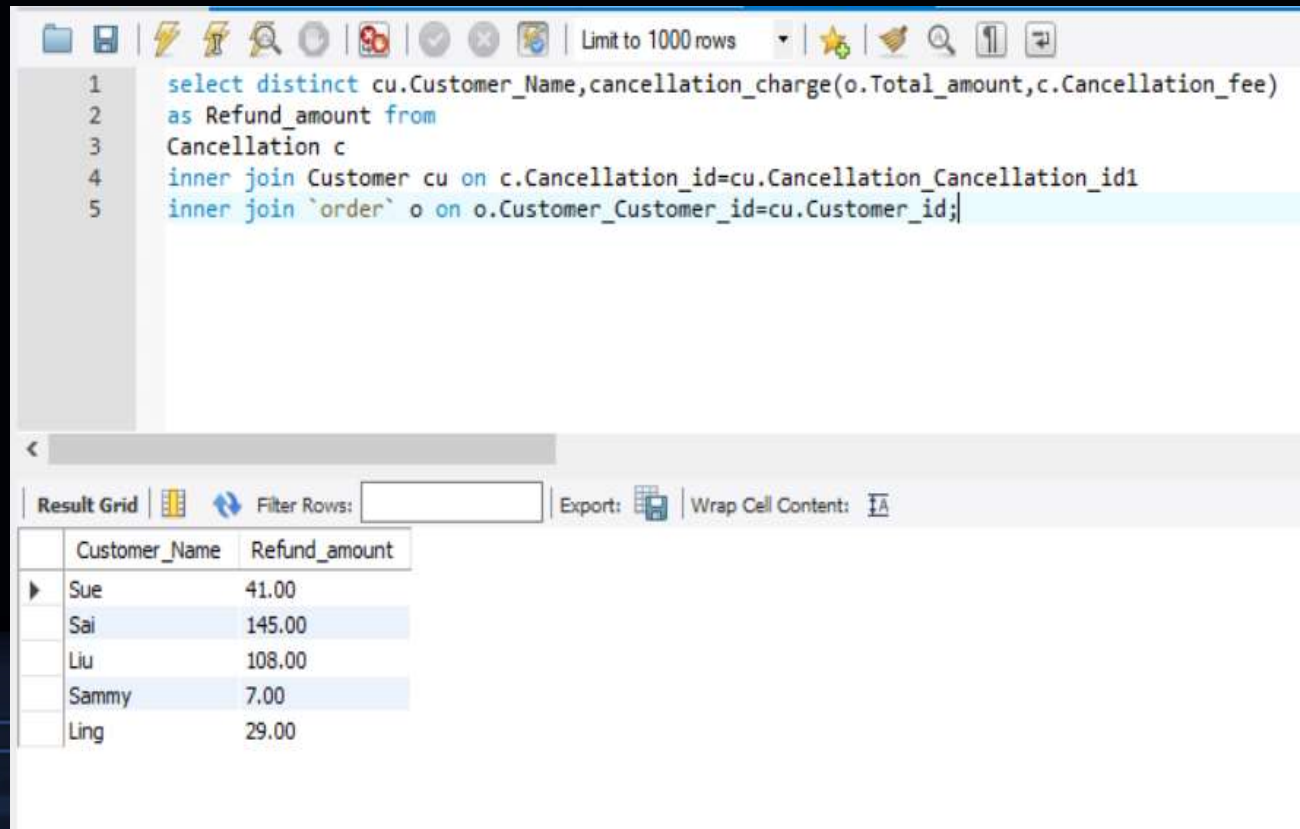
Query 4: Cancellation Fee

Name: The name of the routine is parsed automatically from the DDL statement. The DDL is parsed automatically while you type.

DDL:

```
1 CREATE DEFINER='root'@'localhost' FUNCTION `cancellation_charge`  
2 (amount int, cancellationfee decimal)  
3 RETURNS decimal(9,2)  
4 BEGIN  
5 DECLARE total DECIMAL(9,2);  
6 SET total=amount-cancellationfee;  
7  
8 RETURN total;  
9 END
```

Query 4 continued..



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

```
1 select distinct cu.Customer_Name,cancellation_charge(o.Total_amount,c.Cancellation_fee)
2 as Refund_amount from
3 Cancellation c
4 inner join Customer cu on c.Cancellation_id=cu.Cancellation_Cancellation_id1
5 inner join `order` o on o.Customer_Customer_id=cu.Customer_id;
```

Below the query editor is a result grid. The grid has two columns: Customer_Name and Refund_amount. The data is as follows:

Customer_Name	Refund_amount
Sue	41.00
Sai	145.00
Liu	108.00
Sammy	7.00
Ling	29.00

Query 5: Top Customers

The screenshot shows a database query editor interface. The top toolbar includes icons for file operations, execution, and a 'Limit to 1000 rows' dropdown. The SQL query is as follows:

```
1 SELECT o.Customer_Customer_id, count(Customer_Customer_id) as No_of_orders
2 FROM `order` as o
3 group by Customer_Customer_id
4 ORDER BY o.Customer_Customer_id DESC
5 limit 2
```

Below the query editor, the 'Result Grid' tab is active, displaying a table with the results of the query. The table has two columns: 'Customer_Customer_id' and 'No_of_orders'. The results are:

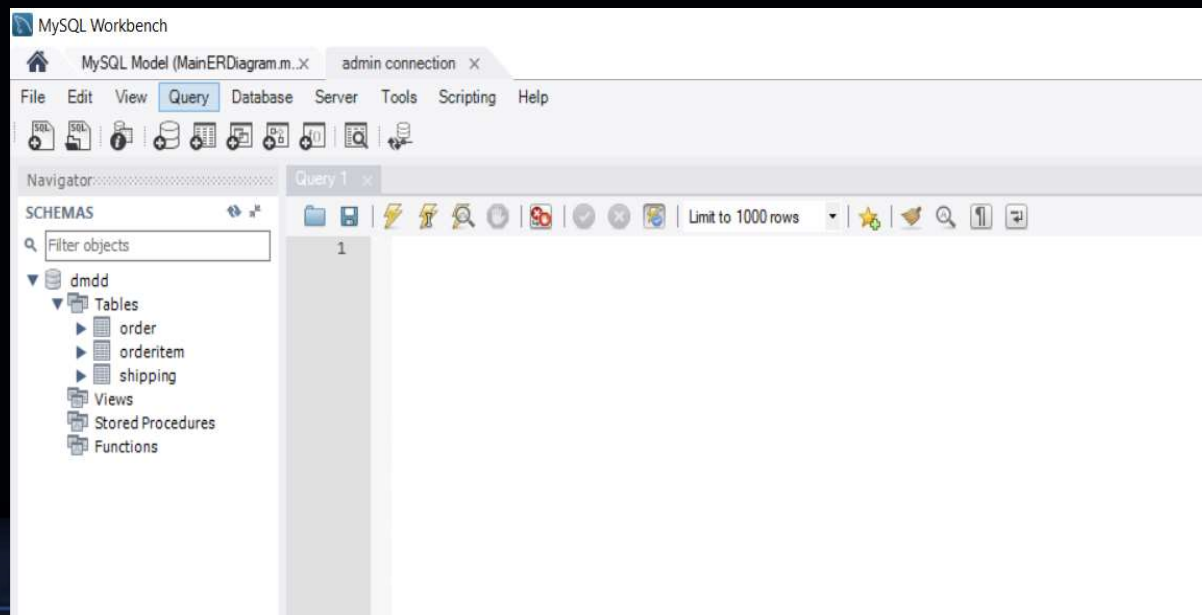
Customer_Customer_id	No_of_orders
789	3
456	3

On the right side of the interface, there are buttons for 'Result Grid' and 'Form Editor'.

Query 6: User Privileges

```
1 • create user 'admin'@'localhost' identified by 'admin';  
2  
3 • grant select on dmdd.order to 'admin'@'localhost';  
4  
5 • grant select on dmdd.shipping to 'admin'@'localhost';  
6  
7 • grant select on dmdd.orderitem to 'admin'@'localhost';|
```

Query 6 continued..





Thank you