

*Mini Project Report On*

# **social-commerce management system.**

for the course

**IT252 : Database Systems(Minor)**

*Submitted by*

**IV SEM B.Tech**

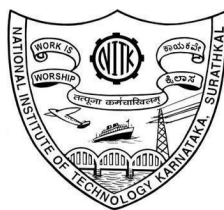
*Under the guidance of*

**PARSAPU KAMAL RAJ (201CV170)**

**Name of the Guide**  
**Dept of IT, NITK**  
**Surathkal**

*in partial fulfillment for the award of the*  
*degree of*

**Bachelor of Technology**  
**in**  
**Information Technology at**



**Department of Information Technology**  
**National Institute of Technology Karnataka, Surathkal.**

***April 2022***

## **CERTIFICATE**

This is to certify that the project entitled “**SOCIAL -COMMERCE MANAGEMENT SYSTEM**” has been submitted by PARSAPU KAMAL RAJ(201CV170) students of second year B.Tech (Civil), National Institute of Technology Karnataka, Surathkal, during the odd semester of the academic year 2021 - 2022, in partial fulfillment of the requirements for the award of the degree of Bachelor of Technology in Information Technology at NITK Surathkal.

(Signature of the Guide)

Place

:

Date:

**Department of Information Technology**  
**National Institute of Technology Karnataka, Surathkal**

*Course code : IT 252*

*Course Title: Database Systems*

*Title of the Project: **IRIS Mess and Hostel Allotment***

*Details of Project Group*

<i>Name of the Student</i>	<i>Register No.</i>	<i>Signature with Date</i>
1. PARSAPU KAMAL RAJ	201CV170	

**Name of Instructor:** Ms J R Shruti

**Signature of the Instructor:**

Place: IT Department

Date: 18-02-2022

# INDEX

1. INTRODUCTION.....
2. Normalisation.....
3. Tables.....
4. Simple Queries.....
5. Complex Queries.....
6. Views.....
7. Procedures.....
8. Functions.....
9. Triggers.....

# INTRODUCTION

## 1. PROBLEM STATEMENT:

Social commerce is a subset of electronic commerce that involves social media and online media that supports social interaction, and user contributions to assist online buying and selling of products and services. For years, China has led the world when it came to social commerce. However, during the Covid-19 pandemic, the trend has gained huge momentum in the United States, when countless shoppers moved online. Social commerce is an extension of e-commerce that involves social media and online media. User-generated content and sharing reviews also come under this category. Social commerce was first introduced with the concept of promoting ecommerce brands on other websites through advertorial content. Here, user reviews and peer to peer conversation in social media determine the shopping of a brand. This can happen without the brand being active in the social commerce front and can indicate the influence social media has on both online and Offline.

**Aim :** To develop a data model and social-commerce management system.

**The problem with the e-commerce system:** The e-commerce system has various flaws because most of the people don't use the system even though it has several advantages over traditional stores. The biggest problem is that it takes at least a day to deliver a product to the customer. While some other issues are a duplication of the product means the product is shown on the web some time differs with the original product due to which next time that customer goes to buy the product through the traditional type of stores.

**Proposed system :** Social commerce is the buying and selling of goods or services directly within a social media platform. This model moves social media beyond its traditional role in the discovery process by encouraging users to complete the entire purchase process without

leaving their preferred apps.

## **Data module :**

### **1. Registration :**

In order to use the system the users( both sellers and users) will need to register in the system, and for registration, they need to provide various information related to them such as name, address, etc.

### **2. Dashboard :**

Will mainly focus on the menu bar and search bar .

Menu bar : It will give the option to navigate through different categories .

Search bar : This provides users with an option through which they can search the product they want to purchase.

### **3. Cart :**

After selecting a product if the user wants to purchase it, later they can save the product in the cart.

### **4. Check out :**

Through this module , the user can replace their order and can choose a preferred method to make the payment .

### **5. Purchase History :**

This will show the purchase done by the user in the past.

### **6. Review/Post :**

Users will post the product reviews as a post to showcase their product as the content. Interested people who like the content marketing as what to buy for a particular purpose (like gym , traditional , marriage , etc ) can engage there itself .

### **7. Connecting with each other :**

As users/customers get engaged with each other bringing them close in the commerce marketplace via liking posts , messaging and different many features.

## 2. **ACTORS:** people who interact with the database

users/customers

social-commerce

e seller

## 3. **SOME SAMPLE QUERIES:**

### **User/Customer can**

- display all the orders from the order table issued by specific seller
- display all the orders in the cart
- display all products belongs to specific category
- delete orders from the cart
- display all orders in cart which order value greater than "X" (Any number) rupees and less than "Y" (Any number) rupees
- display all orders whose order date is "P" (Any date)
- display all orders whose shipping date is "O" (Any number)
- display all the people whose location is "XYZ" (Any location)
- count and display no. of reviews are provided for specific product
- display all products whose category is "A" (Any number) and whose price lesser than "K" (Any number)
- display the all orders whose date of payment "X" (Any date)
- display the grand total of the cart.
- display the followers and following people list
- display all the posts belongs to specific product
- display the all the posts posted by them
- display no. of participants are participate for particular conversation
- display the products which belongs to the "ABC" brand (Any brand)



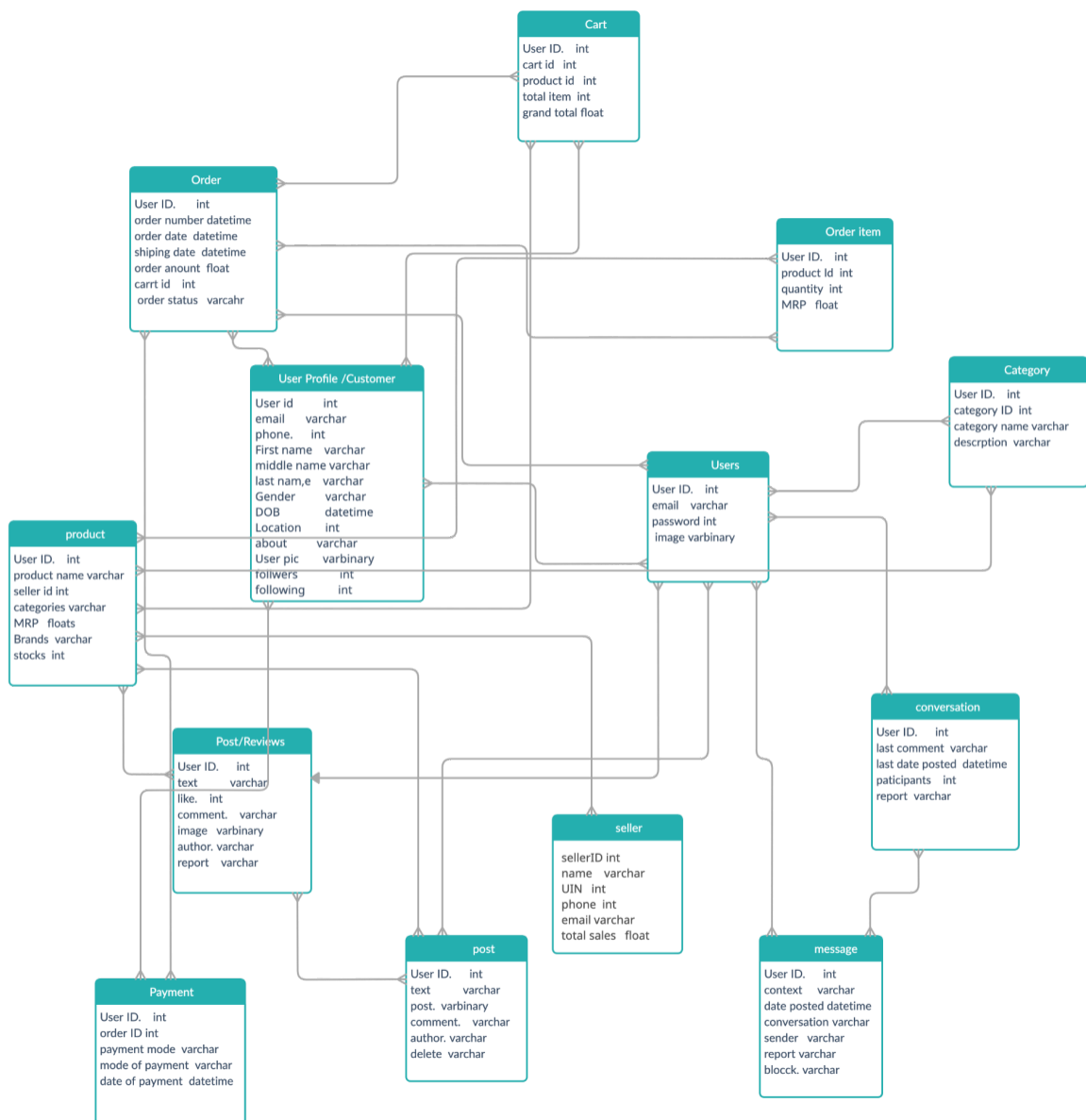
### **Seller can**

- obtain the list of are products are sold
- count and display no.of reviews are provided for each product
- count the no.of customers buying their products
- display the followers and following people list
- display all the posts belongs to specific product
- display the all the posts posted by them
- display no.of participants are participate for particular conversation

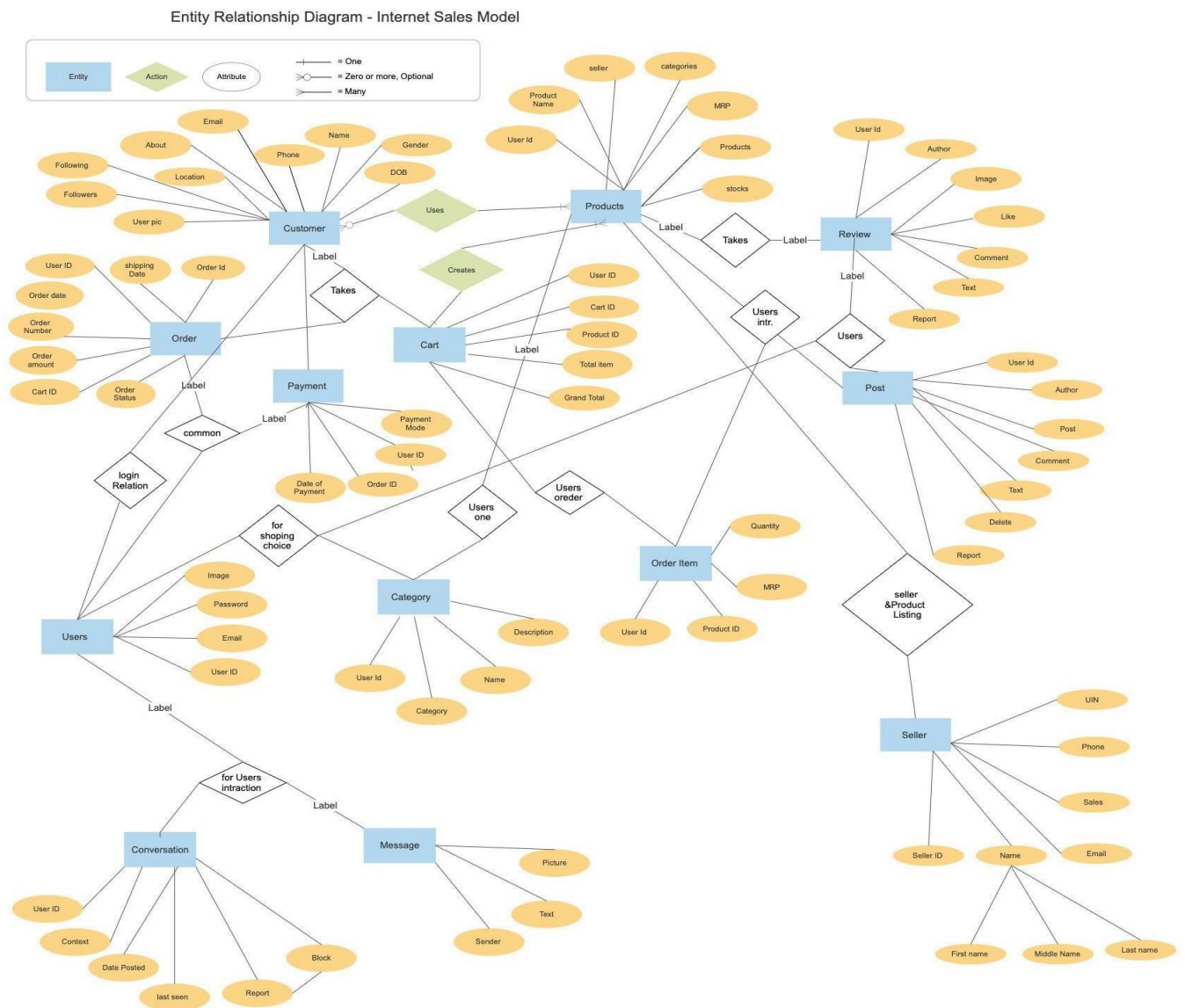
### **Social-commerce can**

- display list of sellers whose total sales cross the “X”(Any number)
- display all orders issued by specific seller whose seller ID is “Y”(Any number)
- display all the people whose followers are greater than following count
- display no.of participants are participate for particular conversation
- find the names and phone numbers of all sellers whose had more than “Y”(Any number) customers
- display no.of participants are participate for particular conversation
- update the order sum whose order sum is above “X”(Any number) rupees are reduced the order sum by “A%”(Any number) of total order sum
- display the orders ordered by customer whose difference between order date and shipping date greater than “P”(Any number) days

## 4. EER MODEL (CONCEPTUAL MODEL)



# ENTITY RELATIONSHIP DIAGRAM



## 5. GLOBAL CONCEPTUAL SCHEMA:

### USER PROFILE/CUSTOMER

User id	email	phone	name	Gender	DOB	Location	about	user pic	followers	following
---------	-------	-------	------	--------	-----	----------	-------	----------	-----------	-----------

### SELLER:

sellerID	name	UIN	phone	email	total sales
----------	------	-----	-------	-------	-------------

### ORDER:

User ID	order number	order date	shipping date	order amount	cart id	order status
---------	--------------	------------	---------------	--------------	---------	--------------

### PRODUCT:

User ID	product name	seller id	categories	MRP	Brands	stocks
---------	--------------	-----------	------------	-----	--------	--------

### POST/REVIEWS:

User ID	text	like	comment	image	author	report
---------	------	------	---------	-------	--------	--------

### PAYMENT:

User ID	order ID	payment mode	mode of payment	date of payment
---------	----------	--------------	-----------------	-----------------

### CART:

User ID	cartID	product id	total item	grand total
---------	--------	------------	------------	-------------

### ORDER ITEM:

User ID	product id	quantity	MRP
---------	------------	----------	-----

### CATEGORY:

User ID	category ID	category name	description
---------	-------------	---------------	-------------

### POST:

User ID	text	post	comment	author	delete
---------	------	------	---------	--------	--------

### MESSAGE:

User ID	context	date posted	conversation	sender	report	block
---------	---------	-------------	--------------	--------	--------	-------

### USERS:

User ID	email	password	image
---------	-------	----------	-------

### CONVERSATION:

User ID	last comment	last date posted	participants	report
---------	--------------	------------------	--------------	--------

# Normalisation:

One of the biggest issues database schema designers have faced happens to be data redundancy, which is also one of the biggest factors which gave rise to relational schema which provided an extremely intuitive & sophisticated method of removing redundancy from in our databases. Data redundancy refers to the state of the database in which the same piece of data occupies more than storage space in more than 2 separate locations. This leads to inefficient queries & wastage of a lot of storage space. The concept of Normalization was introduced to overcome the issue of data redundancy.

**Anomalies** are caused when there is too much redundancy in the database's information. Anomalies can often be caused when the tables that make up the database suffer from poor construction. There are three types of Data Anomalies: Update Anomalies, Insertion Anomalies, and Deletion Anomalies.

- **Update Anomalies** happen when the person charged with the task of keeping all the records current and accurate, is asked, for example, to change an employee's title due to a promotion. If the data is stored redundantly in the same table, and the person misses any of them, then there will be multiple titles associated with the employee. The end user has no way of knowing which is the correct title.
- **Insertion Anomalies** happen when inserting vital data into the database is not possible because other data is not already there. For example, if a system is designed to require that a customer be on file before a sale can be made to that

customer, but you cannot add a customer until they have bought something, then you have an insert anomaly. It is the classic "catch-22" situation.

- **Deletion Anomalies** happen when the deletion of unwanted information causes desired information to be deleted as well. For example, if a single database record contains information about a particular product along with information about a salesperson for the company and the salesperson quits, then information about the product is deleted along with salesperson information.

**Normalisation** is the process of minimizing redundancy from a relation or set of relations. Redundancy in relation usually causes insertion, deletion and updation anomalies as stated previously. So, it helps to minimize the redundancy in relations. Normal forms are used to eliminate or reduce redundancy in database tables.

Relational schema is a derivative of set theory in mathematics & hence it is extremely easy to analyze relational schemas by considering relations as sets & applying the concepts of set theory over them. In a set, we map a key to a particular value, such that  $t[X] = y$  & for each  $X$  there exists a  $y$ . We express this function symbolically as  $X \rightarrow y$ , where ( $\rightarrow$  stands for "determines"). This relation is known as functional dependency in DBMS, where a key is used to determine the rest of the attributes of a tuple of  $R$ .

**Closure of a Set** - The set of all those attributes which can be functionally determined from an attribute set is called a closure of that attribute set.

**Super Key** - Super Key is an attribute (or set of attributes) that is used to uniquely identify all attributes in a relation.

**Candidate Key** - Candidate key is a set of attributes (or attribute) which uniquely identify the tuples in a relation or table.

**Primary Key** - Minimal super key which is used to uniquely identify a tuple in a relation.

Every normal form should be lossless, and FD preserved.

There are three(in real, there are 5) different types of normal forms, each being a more irreducible form of the same data, such that we reach a 0% redundancy state at the end. BCNF is considered to be a 0% redundancy state & our goal is to reach BCNF of our relational schema.

**1<sup>st</sup> Normal Form:** A relation is said to be in first normal form then it should satisfy the following

- No multi-valued attribute
- No composite attribute
- Identify primary key

Here, the relationship is converted to either relation or foreign key or merging relations.

**2<sup>nd</sup> Normal Form:**

- Repeating column values are taken out and maintained in a separate table. So that change can be done only once in the new table rather than all entries in the first table. Rule is

foreign key must be on the N side else again multi-value in a column will occur.

- Identify the prime **attribute** (part of the candidate key that determines anything else), it is also called **partial dependency**, and eliminate it. Because, 2<sup>nd</sup> NF is based on **Full Functional dependency** (key should determine all other attributes in a table)
- Use foreign keys on many sides.

### 3<sup>rd</sup> Normal Form:

- Only columns with direct dependency of the primary key shall be in the entity.
- No transitive dependencies: non-prime attributes transitively depending on the key.

Example:  $A \rightarrow B \rightarrow C \implies A \rightarrow C$ . Example:  $A \rightarrow B \rightarrow C \implies A \rightarrow C$ .

$A \rightarrow B$  is a non-key attribute here,  $B \rightarrow C$  suddenly becomes a key attribute here. Because of this, we will get repeated values in a column. Therefore, it should be eliminated.

- 3<sup>rd</sup> NF should hold the condition that: if  $X \rightarrow Y$  then Either X is a super key Or Y is a prime attribute  
Following this condition will never allow transitive dependency.



# Current Database Model

## USER PROFILE/CUSTOMER

User id	email	phone	name	Gender	DOB	Location	about	user pic	followers	following
---------	-------	-------	------	--------	-----	----------	-------	----------	-----------	-----------

## SELLER:

sellerID	name	UIN	phone	email	total sales
----------	------	-----	-------	-------	-------------

## ORDER:

User ID	order number	order date	shipping date	order amount	cart id	order status
---------	--------------	------------	---------------	--------------	---------	--------------

## PRODUCT:

User ID	product name	seller id	categories	MRP	Brands	stocks
---------	--------------	-----------	------------	-----	--------	--------

## POST/REVIEWS:

User ID	text	like	comment	image	author	report
---------	------	------	---------	-------	--------	--------

## PAYMENT:

User ID	order ID	payment mode	mode of payment	date of payment
---------	----------	--------------	-----------------	-----------------

## CART:

## ORDER

## ITEM:

## CATEGOR

## Y:

## POST:

User ID	text	post	comment	author	delete
---------	------	------	---------	--------	--------

## MESSAGE:

User ID	context	date posted	conversation	sender	report	block
---------	---------	-------------	--------------	--------	--------	-------

## USERS:

User ID	email	password	image
---------	-------	----------	-------

## CONVERSATION:

User ID	last comment	last date posted	participants	report
---------	--------------	------------------	--------------	--------

Checking 1NF:

Here the necessary conditions to be in 1NF are:

1. No composite attribute should be present.
2. No multivalued attribute should be present.

Here in this database name is the attribute which has a composite attribute.

Name

First_name	Middle_name	Last_name
------------	-------------	-----------

Rest all tables remain the same as all are not having composite attributes and all have primary key.

Checking 2NF:

Necessary conditions to be in 2NF are :

1. It should be in 1NF.
2. It should not have any partial dependency.

Here all tables are in 1NF and there is no partial dependency. So, all tables are in 2NF.

Checking 3NF:

Necessary conditions to be in 3NF are:

1. It should be in 2NF.
2. It should not have transitive

dependency. Hence all tables are in 2NF  
and from table **USER PROFILE/CUSTOMER**

Userid ---> email

Email > phone

Here it is a transitive dependency

So to remove this we make them in to two tables.

#### USER PROFILE/CUSTOMER

User id	email	name	Gender	DOB	Location	about	user pic	followers	following
---------	-------	------	--------	-----	----------	-------	----------	-----------	-----------

#### Contact

Email	phone
-------	-------

Like remaining all tables are in 3NF

Checking BCNF:

Necessary conditions to be in BCNF are:

1. It should be in 3NF.
2. The LHS of FD's must be a super key of one of tables.

So now all tables are in 3NF and the Functional dependency considered in that LHS part is a key attribute so , all tables are in BCNF.

Finally the database model after normalization is :

#### USER PROFILE/CUSTOMER

User id	email	name	Gender	DOB	Location	about	user pic	followers	following
---------	-------	------	--------	-----	----------	-------	----------	-----------	-----------

#### Contact

Email			phone				
<b>SELLER:</b>							
<b>ORDER:</b>							
User ID	order number	order date	shipping date	order amount	cart id	order status	
<b>PRODUCT:</b>							
User ID	product name	seller id	categories	MRP	Brands	stocks	

**POST/REVIEW:**

User ID	text	like	comment	image	author	report
---------	------	------	---------	-------	--------	--------

**PAYMENT:**

User ID	order ID	payment mode	mode of payment	date of payment
---------	----------	--------------	-----------------	-----------------

**CART:**

User ID	cartID	product id	total item	grand total
---------	--------	------------	------------	-------------

**ORDER ITEM:**

User ID	product id	quantity	MRP
---------	------------	----------	-----

**CATEGORY:**

User ID	category ID	category name	description
---------	-------------	---------------	-------------

**POST:**

User ID	text	post	comment	author	delete
---------	------	------	---------	--------	--------

**T:**

**MESSAGE:**

User ID	context	date posted	conversation	sender	report	block
---------	---------	-------------	--------------	--------	--------	-------

**USER:**

User ID	email	password	image
---------	-------	----------	-------

**S:**

**CONVERSATION:**

User ID	last comment	last date posted	participants	report
---------	--------------	------------------	--------------	--------

# DATABASE MODEL AFTER NORMALIZATION

## USER PROFILE/CUSTOMER

User id	email	name	Gender	DOB	Location	about	user pic	followers	following
---------	-------	------	--------	-----	----------	-------	----------	-----------	-----------

## SELL

### ER:

sellerID	name	UIN	phone	email	total sales
----------	------	-----	-------	-------	-------------

## ORDER

### R:

User ID	order number	order date	shipping date	order amount	cart id	order status
---------	--------------	------------	---------------	--------------	---------	--------------

## PRODUCT:

## POST/REVIEW

## WISHLIST:

## PAYMENT:

## CART:

## ORDER ITEM:

User ID	product id	quantity	MRP
---------	------------	----------	-----

## CATEGORY:

## POST:

## MESSAGE:

## USERS:

## CONVERSATION:

## ON:

# TABLES:

```
mysql> select *from user_profile;
```

	userid	email	phone	first_name	middle_name	last_name	gender	DOB	location	about	user_Pic	followers	following
1	xyz@gmail.com	9876543212	khushal	kumar		gaurav	male	2020-01-10 00:00:00	2	student	1	123	312
2	abc@gmail.com	9823243212	khushali	kumari		solanki	female	2021-11-01 00:00:00	3	teacher	2	323	321
3	wxy@gmail.com	8876543212	naresh	kumar		shukla	male	2020-02-12 00:00:00	4	worker	3	2123	12
4	eyz@gmail.com	9866543212	kamal	raj		kamal	male	2020-01-10 00:00:00	2	student	1	123	312
5	teyz@gmail.com	4866543212	kevmal	reraj		kamal	male	2020-01-10 00:00:00	2	student	1	123	312

```
5 rows in set (0.04 sec)
```

```
mysql> desc user_profile;
```

Field	Type	Null	Key	Default	Extra
userid	decimal(25,0)	NO	PRI	NULL	
email	varchar(50)	YES		NULL	
phone	decimal(10,0)	YES		NULL	
first_name	varchar(50)	YES		NULL	
middle_name	varchar(50)	YES		NULL	
last_name	varchar(50)	YES		NULL	
gender	varchar(12)	YES		NULL	
DOB	datetime	YES		NULL	
location	int(11)	YES		NULL	
about	varchar(200)	YES		NULL	
user_Pic	int(11)	YES		NULL	
followers	int(11)	YES		NULL	
following	int(11)	YES		NULL	

```
13 rows in set (0.03 sec)
```

User ID	cartID	product id	total item	grand total						
User ID	product id	quantity	MRP							
User ID	category ID	category name	description							
User id	email	phone	name	Gender	DOB	Location	about	user pic	followers	following
sellerID	name	UIN	phone	email	total sales					
User ID	product name	seller id	categories	MRP	Brands	stocks				
User ID	text	like	comment	image	author	report				
User ID	order ID	payment mode	mode of payment	date of payment						
User ID	cartID	product id	total item	grand total						
User ID	category ID	category name	description							

User ID	text	post	comment	author	delete	
User ID	content	date posted	conversation	sender	report	block
User ID	email	password	image			
User ID	last comment	last date posted	participations	report		

## USER PROFILE/CUSTOMER

## SELLER:

```
mysql> select *from seller;
```

sellerid	name_	UIN	phone	email	total_sales
1	vishal	123	9183767223	eyw@gmail.com	1454.54
2	Paresh	132	8233767223	qwr@gmail.com	3589.83
3	khushal	515	8473667223	iru@gmail.com	2312.23

```
3 rows in set (0.05 sec)
```

```
mysql> desc seller;
```

Field	Type	Null	Key	Default	Extra
sellerid	int(11)	NO	PRI	NULL	
name_	varchar(50)	YES		NULL	
UIN	int(11)	YES		NULL	
phone	decimal(10,0)	YES		NULL	
email	varchar(50)	YES		NULL	
total_sales	float	YES		NULL	

```
6 rows in set (0.02 sec)
```

## ORDER:

```
mysql> select *from orders;
+-----+-----+-----+-----+-----+-----+-----+
| userid | ordernumber | orderdate       | shippingdate       | orderamount | cartid | order_status |
+-----+-----+-----+-----+-----+-----+-----+
| 1      | 12          | 2020-10-01 00:00:00 | 2020-10-05 00:00:00 | 123         | 1      | shipped to sellesman |
| 2      | 21          | 2021-10-01 00:00:00 | 2021-10-10 00:00:00 | 421         | 2      | arrive           |
| 3      | 31          | 2021-10-04 00:00:00 | 2021-10-10 00:00:00 | 2134        | 3      | shipped today     |
| NULL   | NULL        | 2022-05-18 11:34:29 | 2022-05-23 11:34:29 | 100         | NULL   | Shipped today     |
+-----+-----+-----+-----+-----+-----+-----+
4 rows in set (0.00 sec)

mysql> desc orders;
+-----+-----+-----+-----+-----+-----+
| Field          | Type          | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+-----+
| userid         | decimal(25,0) | YES  | MUL | NULL    |       |
| ordernumber    | decimal(20,0) | YES  |     | NULL    |       |
| orderdate      | datetime      | YES  |     | NULL    |       |
| shippingdate   | datetime      | YES  |     | NULL    |       |
| orderamount    | float         | YES  |     | NULL    |       |
| cartid         | int(11)       | YES  | MUL | NULL    |       |
| order_status   | varchar(100)  | YES  |     | NULL    |       |
+-----+-----+-----+-----+-----+-----+
7 rows in set (0.00 sec)
```

## PRODUCT:

```
mysql> select *from product;
+-----+-----+-----+-----+-----+-----+
| userid | product_name | sellerid | MRP    | brands | stocks |
+-----+-----+-----+-----+-----+-----+
| 1      | Teddy Bear   | 1        | 123    | xyz    | 231    |
| 2      | Water Bottle | 2        | 251.23 | rwt    | 31     |
| 3      | Keyboard     | 3        | 323    | ieu    | 12     |
| 4      | Gelusil      | 2        | 56.5   | Cipla  | 43     |
+-----+-----+-----+-----+-----+-----+
4 rows in set (0.00 sec)

mysql> desc product;
+-----+-----+-----+-----+-----+-----+
| Field          | Type          | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+-----+
| userid         | decimal(25,0) | YES  | MUL | NULL    |       |
| product_name   | varchar(50)   | YES  |     | NULL    |       |
| sellerid       | int(11)       | YES  | MUL | NULL    |       |
| MRP            | float         | YES  |     | NULL    |       |
| brands         | varchar(50)   | YES  |     | NULL    |       |
| stocks         | int(11)       | YES  |     | NULL    |       |
+-----+-----+-----+-----+-----+-----+
6 rows in set (0.00 sec)
```



## POST/REVIEWS:

```
mysql> select *from post;
+-----+-----+-----+-----+-----+-----+
| userid | texts   | post | comments | author  | deletes |
+-----+-----+-----+-----+-----+-----+
|      1 | Marvelus | 123  | good     | khushal | NULL    |
|      2 | great   | 212  | better   | khushali | NULL    |
|      3 | bad     | 231  | Worst    | naresh  | NULL    |
+-----+-----+-----+-----+-----+-----+
3 rows in set (0.00 sec)

mysql> desc post;
+-----+-----+-----+-----+-----+-----+
| Field      | Type                | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+-----+
| userid     | decimal(25,0)       | YES  | MUL | NULL    |       |
| texts      | varchar(200)        | YES  |     | NULL    |       |
| post       | varbinary(25)       | YES  |     | NULL    |       |
| comments   | varchar(200)        | YES  |     | NULL    |       |
| author     | varchar(25)         | YES  |     | NULL    |       |
| deletes    | varchar(200)        | YES  |     | NULL    |       |
+-----+-----+-----+-----+-----+-----+
6 rows in set (0.00 sec)
```

## PAYMENT:

```
mysql> select *from payment;
+-----+-----+-----+-----+
| userid | orderid | paymentmode | date_of_payment |
+-----+-----+-----+-----+
|      1 |      1 | EMI         | 2020-10-01 00:00:00 |
|      2 |      2 | Case on payment | 2021-10-10 00:00:00 |
|      3 |      3 | PayPal      | 2021-10-04 00:00:00 |
+-----+-----+-----+-----+
3 rows in set (0.00 sec)

mysql> desc payment;
+-----+-----+-----+-----+-----+-----+
| Field              | Type                | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+-----+
| userid             | decimal(25,0)       | YES  | MUL | NULL    |       |
| orderid            | decimal(25,0)       | YES  |     | NULL    |       |
| paymentmode        | varchar(20)         | YES  |     | NULL    |       |
| date_of_payment    | datetime            | YES  |     | NULL    |       |
+-----+-----+-----+-----+-----+-----+
4 rows in set (0.00 sec)
```

## CART:

```
mysql> select *from cart;
```

userid	cartid	productid	total_item	grand_total	new_total
1	1	1	5	123	NULL
2	2	2	3	421	NULL
3	3	3	6	2134	NULL
5	4	4	7	120	108

```
4 rows in set (0.01 sec)
```

```
mysql> desc cart;
```

Field	Type	Null	Key	Default	Extra
userid	decimal(25,0)	YES	MUL	NULL	
cartid	int(11)	NO	PRI	NULL	
productid	int(11)	YES		NULL	
total_item	int(11)	YES		NULL	
grand_total	float	YES		NULL	
new_total	int(11)	YES		NULL	

```
6 rows in set (0.00 sec)
```

## ORDER ITEM:

```
mysql> select *from order_item;
+-----+-----+-----+-----+
| userid | productid | quantity | MRP    |
+-----+-----+-----+-----+
|      1 |          1 |        21 | 88.56  |
|      2 |          2 |        12 | 2323   |
|      3 |          3 |         5 | 213    |
|      3 |          5 |         2 | 50     |
|      3 |          4 |         2 | 50     |
+-----+-----+-----+-----+
5 rows in set (0.01 sec)
```

```
mysql> desc order_item;
+-----+-----+-----+-----+-----+-----+
| Field      | Type          | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+-----+
| userid     | decimal(25,0) | YES  | MUL | NULL    |       |
| productid  | decimal(20,0) | YES  |     | NULL    |       |
| quantity   | int(11)       | YES  |     | NULL    |       |
| MRP        | float         | YES  |     | NULL    |       |
+-----+-----+-----+-----+-----+-----+
4 rows in set (0.00 sec)
```

## CATEGORY:

```
mysql> select *from category;
+-----+-----+-----+-----+
| userid | categoryid | category_name | description_ |
+-----+-----+-----+-----+
|      1 |          1 | Home         | Home Appliances |
|      2 |          2 | Electric    | electric Appliances |
|      3 |          3 | Toys        | Toys          |
+-----+-----+-----+-----+
3 rows in set (0.02 sec)
```

```
mysql> desc category;
+-----+-----+-----+-----+-----+-----+
| Field          | Type          | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+-----+
| userid         | decimal(25,0) | YES  | MUL | NULL    |       |
| categoryid     | int(11)       | YES  |     | NULL    |       |
| category_name  | varchar(50)   | YES  |     | NULL    |       |
| description_   | varchar(100)  | YES  |     | NULL    |       |
+-----+-----+-----+-----+-----+-----+
4 rows in set (0.00 sec)
```

## POST:

```
mysql> select *from post;
```

userid	texts	post	comments	author	deletes
1	Marvelus	123	good	khushal	NULL
2	great	212	better	khushali	NULL
3	bad	231	Worst	naresh	NULL

3 rows in set (0.00 sec)

```
mysql> desc post;
```

Field	Type	Null	Key	Default	Extra
userid	decimal(25,0)	YES	MUL	NULL	
texts	varchar(200)	YES		NULL	
post	varbinary(25)	YES		NULL	
comments	varchar(200)	YES		NULL	
author	varchar(25)	YES		NULL	
deletes	varchar(200)	YES		NULL	

6 rows in set (0.01 sec)

## MESSAGE:

```
mysql> desc message;
```

Field	Type	Null	Key	Default	Extra
userid	decimal(25,0)	YES	MUL	NULL	
contexts	varchar(100)	YES		NULL	
dateposted	datetime	YES		NULL	
conversation	varchar(2000)	YES		NULL	
sender	varchar(3000)	YES		NULL	
report	varchar(1000)	YES		NULL	
bloclck	varchar(10)	YES		NULL	

7 rows in set (0.00 sec)

## USERS:

```
mysql> select *from users;
```

userid	email	password_	user_pic
1	xyz@gmail.com	9876876	1
2	abc@gmail.com	4376876	2
3	wxy@gmail.com	9162576	3

```
3 rows in set (0.00 sec)
```

```
mysql> desc users;
```

Field	Type	Null	Key	Default	Extra
userid	decimal(25,0)	YES	MUL	NULL	
email	varchar(50)	YES		NULL	
password_	decimal(16,0)	YES		NULL	
user_pic	int(11)	YES		NULL	

```
4 rows in set (0.00 sec)
```

## CONVERSATION:

```
mysql> desc conversation;
```

Field	Type	Null	Key	Default	Extra
userid	decimal(25,0)	YES	MUL	NULL	
last_comment	varchar(100)	YES		NULL	
last_date_posted	datetime	YES		NULL	
report	varchar(100)	YES		NULL	
participants	int(11)	NO		NULL	

```
5 rows in set (0.02 sec)
```

# simple queries:

1) taking orders which have an order amount between 400 and 500.

mysql> select \* from orders where orderamount<=500 and orderamount>=400;

```
mysql> select * from orders where orderamount<=500 and orderamount>=400;
+-----+-----+-----+-----+-----+-----+-----+
| userid | ordernumber | orderdate       | shippingdate       | orderamount | cartid | order_status |
+-----+-----+-----+-----+-----+-----+-----+
| 2      | 21          | 2021-10-01 00:00:00 | 2021-10-10 00:00:00 | 421         | 2      | arrive       |
+-----+-----+-----+-----+-----+-----+-----+
1 row in set (0.05 sec)
```

2) Take tuples from order relation where orderdate is equal to

2020-10-01 mysql> select \* from orders where orderdate =

"2020-10-01";

```
mysql> select * from orders where orderdate = "2020-10-01";
+-----+-----+-----+-----+-----+-----+-----+
| userid | ordernumber | orderdate       | shippingdate       | orderamount | cartid | order_status |
+-----+-----+-----+-----+-----+-----+-----+
| 1      | 12          | 2020-10-01 00:00:00 | 2020-10-05 00:00:00 | 123         | 1      | shipped to sellesman |
+-----+-----+-----+-----+-----+-----+-----+
1 row in set (0.04 sec)
```

3) Take tuples from cart relations where the total number of items is

greater than 4. select cartid,grand\_total from cart where total\_item>4;

```
mysql> select cartid,grand_total from cart where total_item>4;
```

```
+-----+-----+
```

```
| cartid | grand_total |
```

```
+-----+-----+
```

```
|      1 |         123 |
```

```
|      3 |        2134 |
```

```
+-----+-----+
```

```
2 rows in set (0.02 sec)
```

```
product is "xyz". mysql> select * from product where brands = "xyz";
```

```
mysql> select * from product where brands = "xyz";
+-----+-----+-----+-----+-----+-----+
| userid | product_name | sellerid | MRP | brands | stocks |
+-----+-----+-----+-----+-----+-----+
|      1 | Teddy Bear   |      1 | 123 | xyz     |    231 |
+-----+-----+-----+-----+-----+-----+
1 row in set (0.00 sec)
```

equal to 3. `mysql> select * from user_profile where location = 3;`

```
mysql> select * from user_profile where location = 3;
```

userid	email	phone	first_name	middle_name	last_name	gender	DOB	location	about	user_Pic	followers	following
2	abc@gmail.com	9823243212	khushali	kumari	solanki	female	2021-11-01 00:00:00	3	teacher	2	323	321

1 row in set (0.01 sec)



# nested queries

1) Take tuples which contains details of product and seller and MRP of product is 323

```
mysql> select s.name_ as Name,p.product_name,p.brands  
-> from seller as s,product as p  
-> where s.sellerid = p.sellerid and p.stocks = (select stocks from product  
where MRP=323);
```

```
mysql> select * from seller;  
+-----+-----+-----+-----+-----+-----+  
| sellerid | name_ | UIN | phone | email | total_sales |  
+-----+-----+-----+-----+-----+-----+  
| 1 | vishal | 123 | 9183767223 | eyw@gmail.com | 1212.12 |  
| 2 | Paresh | 132 | 8233767223 | qwr@gmail.com | 3212.12 |  
| 3 | khushal | 515 | 8473667223 | iru@gmail.com | 2312.23 |  
+-----+-----+-----+-----+-----+-----+  
3 rows in set (0.00 sec)  
  
mysql> select * from product;  
+-----+-----+-----+-----+-----+-----+  
| userid | product_name | sellerid | MRP | brands | stocks |  
+-----+-----+-----+-----+-----+-----+  
| 1 | Teddy Bear | 1 | 123 | xyz | 231 |  
| 2 | Water Bottle | 2 | 251.23 | rwt | 31 |  
| 3 | Keyboard | 3 | 323 | ieu | 12 |  
+-----+-----+-----+-----+-----+-----+  
3 rows in set (0.00 sec)
```

```
mysql> select s.name_ as Name,p.product_name,p.brands  
-> from seller as s,product as p  
-> where s.sellerid = p.sellerid and p.stocks = (select stocks from product where MRP=323);  
+-----+-----+-----+  
| Name | product_name | brands |  
+-----+-----+-----+  
| khushal | Keyboard | ieu |  
+-----+-----+-----+  
1 row in set (0.01 sec)
```

2) Take tuples which contain details of the user and corresponding orders where cartid is equal to 2.

```
mysql> select u.userid,u.first_name as Name,o.orderdate,o.shippingdate,o.orderamount  
-> from user_profile as u,orders as o  
-> where u.userid = o.userid and o.ordernumber = (select ordernumber from  
orders where cartid=2);
```

```
mysql> select * from orders;
```

userid	ordernumber	orderdate	shippingdate	orderamount	cartid	order_status
1	12	2020-10-01 00:00:00	2020-10-05 00:00:00	123	1	shipped to sellesman
2	21	2021-10-01 00:00:00	2021-10-10 00:00:00	421	2	arrive
3	31	2021-10-04 00:00:00	2021-10-10 00:00:00	2134	3	shipped today

3 rows in set (0.00 sec)

```
mysql> select * from orders;
```

userid	ordernumber	orderdate	shippingdate	orderamount	cartid	order_status
1	12	2020-10-01 00:00:00	2020-10-05 00:00:00	123	1	shipped to sellesman
2	21	2021-10-01 00:00:00	2021-10-10 00:00:00	421	2	arrive
3	31	2021-10-04 00:00:00	2021-10-10 00:00:00	2134	3	shipped today

3 rows in set (0.00 sec)

```
mysql> select u.userid,u.first_name as Name,o.orderdate,o.shippingdate,o.orderamount  
-> from user_profile as u,orders as o  
-> where u.userid = o.userid and o.ordernumber = (select ordernumber from orders where cartid=2);
```

userid	Name	orderdate	shippingdate	orderamount
2	khushali	2021-10-01 00:00:00	2021-10-10 00:00:00	421

1 row in set (0.01 sec)

3) Take tuples which contain details of product and brands where stocks of product are greater than 200.

```
mysql> select p.product_name as Product_Name,o.productid,p.brands  
-> from product as p,order_item as o  
-> where o.MRP = p.MRP and p.brands = (select brands from product where  
stocks >= 200);
```

```
mysql> select * from order_item;
```

userid	productid	quantity	MRP
1	1	21	123
2	2	12	2323
3	3	5	213

```
3 rows in set (0.00 sec)
```

```
mysql> select * from product;
```

userid	product_name	sellerid	MRP	brands	stocks
1	Teddy Bear	1	123	xyz	231
2	Water Bottle	2	251.23	rwt	31
3	Keyboard	3	323	ieu	12

```
3 rows in set (0.00 sec)
```

```
mysql> select p.product_name as Product_Name,o.productid,p.brands  
-> from product as p,order_item as o  
-> where o.MRP = p.MRP and p.brands = (select brands from product where stocks >= 200);
```

Product_Name	productid	brands
Teddy Bear	1	xyz

```
1 row in set (0.01 sec)
```

4) Take tuples which contain details of the user and corresponding payment details where the number of followers of the user is less than 200.

```
mysql> select u.userid,u.first_name as Name,p.paymentmode,p.date_of_payment  
-> from user_profile as u,payment as p  
-> where u.userid = p.userid and email = (select email from user_profile where  
followers<200);
```

```
mysql> select * from user_profile;
```

userid	email	phone	first_name	middle_name	last_name	gender	DOB	location	about	user_Pic	followers	following
1	xyz@gmail.com	9876543212	khushal	kumar	gaurav	male	2020-01-10 00:00:00	2	student	1	123	312
2	abc@gmail.com	9823243212	khushali	kumari	solanki	female	2021-11-01 00:00:00	3	teacher	2	323	321
3	wxy@gmail.com	8876543212	naresh	kumar	shukla	male	2020-02-12 00:00:00	4	worker	3	2123	12

```
3 rows in set (0.00 sec)
```

```
mysql> select * from payment;
```

userid	orderid	paymentmode	date_of_payment
1	1	EMI	2020-10-01 00:00:00
2	2	Case on payment	2021-10-10 00:00:00
3	3	PayPal	2021-10-04 00:00:00

```
3 rows in set (0.00 sec)
```

```
mysql> select u.userid,u.first_name as Name,p.paymentmode,p.date_of_payment  
-> from user_profile as u,payment as p  
-> where u.userid = p.userid and email = (select email from user_profile where followers<200);
```

userid	Name	paymentmode	date_of_payment
1	khushal	EMI	2020-10-01 00:00:00

```
1 row in set (0.01 sec)
```

5) Take tuples which contain details of products, payment and orders where MRP of orders is greater than 200.

```
mysql> select pr.userid,pr.productid,o.orderdate,p.date_of_payment
-> from payment as p,orders as o,order_item as pr
-> where pr.userid=o.userid and pr.userid=p.userid and quantity in (select
quantity from order_item where MRP>200);
```

```
mysql> select * from order_item;
+-----+-----+-----+-----+
| userid | productid | quantity | MRP |
+-----+-----+-----+-----+
| 1 | 1 | 21 | 123 |
| 2 | 2 | 12 | 2323 |
| 3 | 3 | 5 | 213 |
+-----+-----+-----+-----+
3 rows in set (0.00 sec)

mysql> select * from payment;
+-----+-----+-----+-----+
| userid | orderid | paymentmode | date_of_payment |
+-----+-----+-----+-----+
| 1 | 1 | EMI | 2020-10-01 00:00:00 |
| 2 | 2 | Case on payment | 2021-10-10 00:00:00 |
| 3 | 3 | PayPal | 2021-10-04 00:00:00 |
+-----+-----+-----+-----+
3 rows in set (0.00 sec)

mysql> select * from orders;
+-----+-----+-----+-----+-----+-----+-----+
| userid | ordernumber | orderdate | shippingdate | orderamount | cartid | order_status |
+-----+-----+-----+-----+-----+-----+-----+
| 1 | 12 | 2020-10-01 00:00:00 | 2020-10-05 00:00:00 | 123 | 1 | shipped to sellesman |
| 2 | 21 | 2021-10-01 00:00:00 | 2021-10-10 00:00:00 | 421 | 2 | arrive |
| 3 | 31 | 2021-10-04 00:00:00 | 2021-10-10 00:00:00 | 2134 | 3 | shipped today |
+-----+-----+-----+-----+-----+-----+-----+
3 rows in set (0.00 sec)
```

```
mysql> select pr.userid,pr.productid,o.orderdate,p.date_of_payment
-> from payment as p,orders as o,order_item as pr
-> where pr.userid=o.userid and pr.userid=p.userid and quantity in (select quantity from order_item where MRP>200);
+-----+-----+-----+-----+
| userid | productid | orderdate | date_of_payment |
+-----+-----+-----+-----+
| 2 | 2 | 2021-10-01 00:00:00 | 2021-10-10 00:00:00 |
| 3 | 3 | 2021-10-04 00:00:00 | 2021-10-04 00:00:00 |
+-----+-----+-----+-----+
2 rows in set (0.01 sec)
```

# views

1) view for those users who are students .

```
mysql> create view User as (select * from user_profile  
where about="student");
```

Query OK, 0 rows affected (0.04

sec) mysql> select \* from User;

```
mysql> create view User as (select * from user_profile where about="student");  
Query OK, 0 rows affected (0.04 sec)
```

```
mysql> select * from User;
```

userid	email	phone	first_name	middle_name	last_name	gender	DOB	location	about	user_Pic	followers	following_
1	xyz@gmail.com	9876543212	khushal	kumar	gaurav	male	2020-01-10 00:00:00	2	student	1	123	312

1 row in set (0.02 sec)

```
mysql> select * from user_profile;
```

userid	email	phone	first_name	middle_name	last_name	gender	DOB	location	about	user_Pic	followers	following_
1	xyz@gmail.com	9876543212	khushal	kumar	gaurav	male	2020-01-10 00:00:00	2	student	1	123	312
2	abc@gmail.com	9823243212	khushali	kumari	solanki	female	2021-11-01 00:00:00	3	teacher	2	323	321
3	wxy@gmail.com	8876543212	naresh	kumar	shukla	male	2020-02-12 00:00:00	4	worker	3	2123	12

3 rows in set (0.00 sec)



2) view shows details of user and product also where product brand is xyz.

```
mysql> create view Product2 as (select p.userid,u.first_name
as Name,p.product_name,p.brands,p.MRP from product
as p,user_profile as u where p.userid=u.userid and
p.brands="xyz"); Query OK, 0 rows affected (0.02 sec)
```

```
mysql> select * from Product2;
```

```
mysql> create view Product2 as (select p.userid,u.first_name as Name,p.product_name,p.brands,p.MRP from product as p,user_profile as u where p.userid=u.userid and p.brands="xyz");
Query OK, 0 rows affected (0.02 sec)
```

```
mysql> select * from Product2;
```

```

+-----+-----+-----+-----+-----+
| userid | Name   | product_name | brands | MRP   |
+-----+-----+-----+-----+-----+
|      1 | khushal | Teddy Bear   | xyz    | 123   |
+-----+-----+-----+-----+-----+
1 row in set (0.01 sec)

```

```
mysql> select * from product;
```

```

  user_id | product_name | seller_id | MRP | brands | stocks |
-----+-----+-----+-----+-----+-----+
      1 | Teddy Bear   |      1    | 123 | xyz     |      231 |
      2 | Water Bottle |      2    | 251.23 | rwt    |      31 |
      3 | Keyboard     |      3    | 323 | ieu     |      12 |
3 rows in set (0.00 sec)

```

```
mysql> select * from user_profile;
```

userid	email	phone	first_name	middle_name	last_name	gender	DOB	location	about	user_pic	followers	following_
1	xyz@gmail.com	9876543212	khushal	kumar	gaurav	male	2020-01-10 00:00:00	2	student	1	123	312
2	abc@gmail.com	9823243212	khushali	kumari	solanki	female	2021-11-01 00:00:00	3	teacher	2	323	321
3	wxy@gmail.com	8876543212	naresh	kumar	shukla	male	2020-02-12 00:00:00	4	worker	3	2123	12

3 rows in set (0.00 sec)

3) view have user details and corresponding order detail and order\_status detail where cartid is greater than 1.

```
mysql> create view view2 as (select u.userid,u.first_name  
as  
Name,u.location,o.ordernumber,o.orderdate,o.shippingdate,o.order  
_stat us
```

```
-> from user_profile as u,orders as o
```

```
-> where o.userid=u.userid and o.cartid>1);
```

Query OK, 0 rows affected (0.07 sec)

```
mysql> select * from view2;
```

```
mysql> create view view2 as (select u.userid,u.first_name as Name,u.location,o.ordernumber,o.orderdate,o.shippingdate,o.order_status  
-> from user_profile as u,orders as o  
-> where o.userid=u.userid and o.cartid>1);  
Query OK, 0 rows affected (0.07 sec)
```

```
mysql> select * from view2;
```

userid	Name	location	ordernumber	orderdate	shippingdate	order_status
2	khushali	3	21	2021-10-01 00:00:00	2021-10-10 00:00:00	arrive
3	naresh	4	31	2021-10-04 00:00:00	2021-10-10 00:00:00	shipped today

2 rows in set (0.03 sec)

```
mysql> select * from user_profile;
```

userid	email	phone	first_name	middle_name	last_name	gender	DOB	location	about	user_Pic	followers	following_
1	xyz@gmail.com	9876543212	khushal	kumar	gaurav	male	2020-01-10 00:00:00	2	student	1	123	312
2	abc@gmail.com	9823243212	khushali	kumari	solanki	female	2021-11-01 00:00:00	3	teacher	2	323	321
3	wxy@gmail.com	8876543212	naresh	kumar	shukla	male	2020-02-12 00:00:00	4	worker	3	2123	12

3 rows in set (0.00 sec)

```
mysql> select * from orders;
```

userid	ordernumber	orderdate	shippingdate	orderamount	cartid	order_status
1	12	2020-10-01 00:00:00	2020-10-05 00:00:00	123	1	shipped to sellesman
2	21	2021-10-01 00:00:00	2021-10-10 00:00:00	421	2	arrive
3	31	2021-10-04 00:00:00	2021-10-10 00:00:00	2134	3	shipped today

3 rows in set (0.00 sec)



4) View has details of seller and corresponding product details where total sales of seller is greater than 2000.

```
mysql> create view view3 as (select s.sellerid,s.name_  
as Name,s.UIN,p.product_name,p.MRP,p.brands,p.stocks  
-> from seller as s,product as p  
-> where s.sellerid=p.sellerid and  
s.total_sales>2000); Query OK, 0 rows affected  
(0.07 sec)
```

```
mysql> select * from view3;
```

```
mysql> create view view3 as (select s.sellerid,s.name_  
-> from seller as s,product as p  
-> where s.sellerid=p.sellerid and s.total_sales>2000);  
Query OK, 0 rows affected (0.07 sec)  
  
mysql> select * from view3;  
+-----+-----+-----+-----+-----+-----+-----+  
| sellerid | Name | UIN | product_name | MRP | brands | stocks |  
+-----+-----+-----+-----+-----+-----+-----+  
| 2 | Paresh | 132 | Water Bottle | 251.23 | rwt | 31 |  
| 3 | khushal | 515 | Keyboard | 323 | ieu | 12 |  
+-----+-----+-----+-----+-----+-----+-----+  
2 rows in set (0.02 sec)
```

```
mysql>  
mysql> select * from seller;  
+-----+-----+-----+-----+-----+-----+  
| sellerid | name_ | UIN | phone | email | total_sales |  
+-----+-----+-----+-----+-----+-----+  
| 1 | vishal | 123 | 9183767223 | eyw@gmail.com | 1212.12 |  
| 2 | Paresh | 132 | 8233767223 | qwr@gmail.com | 3212.12 |  
| 3 | khushal | 515 | 8473667223 | iru@gmail.com | 2312.23 |  
+-----+-----+-----+-----+-----+-----+  
3 rows in set (0.00 sec)  
  
mysql> select * from product;  
+-----+-----+-----+-----+-----+-----+  
| userid | product_name | sellerid | MRP | brands | stocks |  
+-----+-----+-----+-----+-----+-----+  
| 1 | Teddy Bear | 1 | 123 | xyz | 231 |  
| 2 | Water Bottle | 2 | 251.23 | rwt | 31 |  
| 3 | Keyboard | 3 | 323 | ieu | 12 |  
+-----+-----+-----+-----+-----+-----+  
3 rows in set (0.00 sec)
```

5) view have details of comments and reviews posted by users to correspond with their products.

```
mysql> create view view4 as (select u.userid,u.first_name
as Name,p.product_name,p.brands,po.texts as Texts,po.comments
-> from user_profile as u,product as p,post as po
-> where u.userid=p.userid and p.userid=po.userid);
```

```
mysql> create view view4 as (select u.userid,u.first_name as Name,p.product_name,p.brands,po.texts as Texts,po.comments
-> from user_profile as u,product as p,post as po
-> where u.userid=p.userid and p.userid=po.userid);
Query OK, 0 rows affected (0.01 sec)

mysql> select * from view4;
+-----+-----+-----+-----+-----+-----+
| userid | Name   | product_name | brands | Texts   | comments |
+-----+-----+-----+-----+-----+-----+
| 1      | khushal | Teddy Bear   | xyz    | Marvelus | good      |
| 2      | khushali | Water Bottle | rwt    | great    | better    |
| 3      | naresh  | Keyboard     | ieu    | bad      | Worst     |
+-----+-----+-----+-----+-----+-----+
3 rows in set (0.00 sec)
```

```
mysql> select * from user_profile;
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
| userid | email      | phone      | first_name | middle_name | last_name | gender | DOB           | location | about   | user_Pic | followers | following |
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
| 1      | xyz@gmail.com | 9876543212 | khushal    | kumar       | gaurav    | male   | 2020-01-10 00:00:00 | 2        | student | 1        | 123       | 312       |
| 2      | abc@gmail.com | 9823243212 | khushali   | kumari      | solanki   | female | 2021-11-01 00:00:00 | 3        | teacher | 2        | 323       | 321       |
| 3      | wxy@gmail.com | 8876543212 | naresh     | kumar       | shukla    | male   | 2020-02-12 00:00:00 | 4        | worker  | 3        | 2123      | 12        |
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
3 rows in set (0.00 sec)

mysql> select * from post;
+-----+-----+-----+-----+-----+-----+
| userid | texts   | post      | comments | author   | deletes |
+-----+-----+-----+-----+-----+-----+
| 1      | Marvelus | 0x313233 | good     | khushal | NULL     |
| 2      | great    | 0x323132 | better   | khushali | NULL     |
| 3      | bad      | 0x323331 | Worst    | naresh  | NULL     |
+-----+-----+-----+-----+-----+-----+
3 rows in set (0.00 sec)

mysql> select * from product;
+-----+-----+-----+-----+-----+-----+
| userid | product_name | sellerid | MRP   | brands | stocks |
+-----+-----+-----+-----+-----+-----+
| 1      | Teddy Bear   | 1        | 123   | xyz    | 231    |
| 2      | Water Bottle | 2        | 251,23 | rwt    | 31     |
| 3      | Keyboard     | 3        | 323   | ieu    | 12     |
+-----+-----+-----+-----+-----+-----+
3 rows in set (0.00 sec)
```

# stored procedures

1) Procedure to give a bonus amount to seller  
by sellerid DELIMITER \$\$  
CREATE PROCEDURE Get\_Information (uid  
int) BEGIN  
SELECT  
o.ordernumber,o.orderdate,o.orderamount,up.first\_name,up.location,up.phone  
FROM `orders` AS o  
join user\_profile as up on o.userid =  
up.userid WHERE o.userid = uid;  
END \$\$  
DELIMITE  
R ;

call Get\_information (1);  
call Get\_information (2);

```
mysql> CREATE PROCEDURE Get_Information (uid int)
-> BEGIN
-> SELECT o.ordernumber,o.orderdate,o.orderamount,up.first_name,up.location,up.phone
-> FROM `orders` AS o
-> join user_profile as up on o.userid = up.userid
-> WHERE o.userid = uid;
-> END $$
Query OK, 0 rows affected (0.07 sec)

mysql> DELIMITER ;
mysql> call Get_information (1);
+-----+-----+-----+-----+-----+-----+
| ordernumber | orderdate           | orderamount | first_name | location | phone       |
+-----+-----+-----+-----+-----+-----+
|          12 | 2020-10-01 00:00:00 |          123 | khushal   |         2 | 9876543212 |
+-----+-----+-----+-----+-----+-----+
1 row in set (0.01 sec)

Query OK, 0 rows affected (0.03 sec)

mysql> call Get_information (2);
+-----+-----+-----+-----+-----+-----+
| ordernumber | orderdate           | orderamount | first_name | location | phone       |
+-----+-----+-----+-----+-----+-----+
|          21 | 2021-10-01 00:00:00 |          421 | khushali   |         3 | 9823243212 |
+-----+-----+-----+-----+-----+-----+
1 row in set (0.00 sec)

Query OK, 0 rows affected (0.01 sec)
```

## 2) Procedure for getting all the information of the user by user id

```
DELIMITER $$
CREATE PROCEDURE Give_Bonus(sid int,discout
int) BEGIN
UPDATE seller
set
total_sales = total_sales+((total_sales*discount)/100)
WHERE seller.sellerid = sid;
END $$
DELIMITE
R ;
```

```
call Give_Bonus
(1,20); select * from
seller; call
Give_Bonus (2,10);
select * from seller;
```

```
mysql> CREATE PROCEDURE Give_Bonus(sid int,discout int)
-> BEGIN
-> UPDATE seller
-> set
-> total_sales = total_sales+((total_sales*discount)/100)
-> WHERE seller.sellerid = sid;
-> END $$
Query OK, 0 rows affected (0.04 sec)

mysql> DELIMITER ;
mysql> call Give_Bonus (1,20);
Query OK, 1 row affected (0.05 sec)

mysql> select * from seller;
+-----+-----+-----+-----+-----+-----+
| sellerid | name_ | UIN | phone | email | total_sales |
+-----+-----+-----+-----+-----+-----+
| 1 | vishal | 123 | 9183767223 | eyw@gmail.com | 1454.54 |
| 2 | Paresh | 132 | 8233767223 | qwr@gmail.com | 3212.12 |
| 3 | khushal | 515 | 8473667223 | iru@gmail.com | 2312.23 |
+-----+-----+-----+-----+-----+-----+
3 rows in set (0.00 sec)

mysql> call Give_Bonus (2,10);
Query OK, 1 row affected (0.00 sec)

mysql> select * from seller;
+-----+-----+-----+-----+-----+-----+
| sellerid | name_ | UIN | phone | email | total_sales |
+-----+-----+-----+-----+-----+-----+
| 1 | vishal | 123 | 9183767223 | eyw@gmail.com | 1454.54 |
| 2 | Paresh | 132 | 8233767223 | qwr@gmail.com | 3533.33 |
| 3 | khushal | 515 | 8473667223 | iru@gmail.com | 2312.23 |
+-----+-----+-----+-----+-----+-----+
3 rows in set (0.00 sec)
```

3) procedure for giving discounts to the user and update the MRP value of the item if user have product quantity above 20

```
DELIMITER $$
CREATE PROCEDURE Give_Discount(discount
int) BEGIN
UPDATE order_item
set
MRP =
MRP-((MRP*discount)/100)
WHERE order_item.quantity>20;
END $$
DELIMITER ;
```

```
select * from order_item;
call Give_Discount(20);
select * from order_item;
call Give_Discount(10);
select * from order_item;
```

```
mysql> DELIMITER $$
mysql> CREATE PROCEDURE Give_Discount(discount int)
-> BEGIN
-> UPDATE order_item
-> set
-> MRP = MRP-((MRP*discount)/100)
-> WHERE order_item.quantity>20;
-> END $$
Query OK, 0 rows affected (0.06 sec)

mysql> DELIMITER ;
mysql> select * from order_item;call Give_Discount(20);
+-----+-----+-----+-----+
| userid | productid | quantity | MRP |
+-----+-----+-----+-----+
| 1 | 1 | 21 | 123 |
| 2 | 2 | 12 | 2323 |
| 3 | 3 | 5 | 213 |
+-----+-----+-----+-----+
3 rows in set (0.02 sec)

Query OK, 1 row affected (0.02 sec)

mysql> select * from order_item;call Give_Discount(10);select * from order_item;
+-----+-----+-----+-----+
| userid | productid | quantity | MRP |
+-----+-----+-----+-----+
| 1 | 1 | 21 | 98.4 |
| 2 | 2 | 12 | 2323 |
| 3 | 3 | 5 | 213 |
+-----+-----+-----+-----+
3 rows in set (0.00 sec)

Query OK, 1 row affected (0.01 sec)

+-----+-----+-----+-----+
| userid | productid | quantity | MRP |
+-----+-----+-----+-----+
| 1 | 1 | 21 | 88.56 |
| 2 | 2 | 12 | 2323 |
| 3 | 3 | 5 | 213 |
+-----+-----+-----+-----+
3 rows in set (0.00 sec)
```

4) procedure to check the cart detail by cartid

```
mysql> CREATE PROCEDURE view_cart(cid int)
-> BEGIN
-> SELECT c.total_item,c.grand_total,c.productid,up.first_name
-> FROM `cart` AS c
-> join user_profile as up on up.userid = c.userid
-> WHERE c.cartid = cid;
-> END $$
mysql> call view_cart(1);
mysql> call view_cart(2);
```

```
mysql> DELIMITER $$
mysql> CREATE PROCEDURE view_cart(cid int)
-> BEGIN
-> SELECT c.total_item,c.grand_total,c.productid,up.first_name
-> FROM `cart` AS c
-> join user_profile as up on up.userid = c.userid
-> WHERE c.cartid = cid;
-> END $$
```

Query OK, 0 rows affected (0.00 sec)

```
mysql> DELIMITER ;
mysql> call view_cart(1);
```

total_item	grand_total	productid	first_name
5	123	1	khushal

1 row in set (0.02 sec)

Query OK, 0 rows affected (0.03 sec)

```
mysql> call view_cart(2);
```

total_item	grand_total	productid	first_name
3	421	2	khushali

1 row in set (0.00 sec)

Query OK, 0 rows affected (0.02 sec)

5) procedure to check the payment details by userid

```
mysql> DELIMITER $$
mysql> CREATE PROCEDURE view_payment(uid int)
-> BEGIN
-> SELECT p.paymentmode,p.date_of_payment,up.first_name,o.ordernumber,o.orderamount
-> FROM `user_profile` AS up
-> join payment as p on p.userid = up.userid
-> join orders as o on o.userid = up.userid
-> WHERE up.userid = uid;
-> END $$
```

Query OK, 0 rows affected (0.01 sec)

```
mysql> DELIMITER ;
mysql> call view_payment(1);
```

```
mysql> DELIMITER $$
mysql> CREATE PROCEDURE view_payment(uid int)
-> BEGIN
-> SELECT p.paymentmode,p.date_of_payment,up.first_name,o.ordernumber,o.orderamount
-> FROM `user_profile` AS up
-> join payment as p on p.userid = up.userid
-> join orders as o on o.userid = up.userid
-> WHERE up.userid = uid;
-> END $$
```

Query OK, 0 rows affected (0.01 sec)

```
mysql> DELIMITER ;
mysql> call view_payment(1);
```

paymentmode	date_of_payment	first_name	ordernumber	orderamount
EMI	2020-10-01 00:00:00	khushal	12	123

1 row in set (0.02 sec)

Query OK, 0 rows affected (0.02 sec)

```
mysql> call view_payment(2);
```

paymentmode	date_of_payment	first_name	ordernumber	orderamount
Case on payment	2021-10-10 00:00:00	khushali	21	421

1 row in set (0.00 sec)

Query OK, 0 rows affected (0.02 sec)

# stored functions

- 1) Function which returns total number of products which a particular seller sells

```
mysql> delimiter $$
```

```
mysql> create function total_products(sId int)
```

```
-> returns numeric
```

```
-> deterministic
```

```
-> begin
```

```
-> return (select count(product_name) from Product where sellerid = sId);
```

```
-> end $$
```

```
Query OK, 0 rows affected (0.00 sec)
```

```
mysql> select total_products(3);
```

```
-> $$
```

```
mysql> create function total_products(sId int)
  -> returns numeric
  -> deterministic
  -> begin
  -> return (select count(product_name) from Product where sellerid = sId);
  -> end $$
```

```
Query OK, 0 rows affected (0.00 sec)
```

```
mysql> select total_products(3);
```

```
-> $$
```

```
+-----+
| total_products(3) |
+-----+
|                1 |
+-----+
```

```
1 row in set (0.01 sec)
```



2) Function that returns the number of products which has quantity greater than given total items

```
mysql> delimiter $$
```

```
mysql> create function products_with_greater_quantity(quantity numeric)
```

-> returns numeric

-> deterministic

-> begin

```
-> return (select count(productid) from cart where Quantity <= total_item);
```

```
-> end $$
```

Query OK, 0 rows affected (0.00 sec)

```
mysql> delimiter ;
```

```
mysql> select products_with_greater_quantity(4);
```

```
mysql> delimiter $$
mysql> create function products_with_greater_quantity(quantity numeric)
-> returns numeric
-> deterministic
-> begin
-> return (select count(productid) from cart where Quantity <= total_item);
-> end $$
```

Query OK, 0 rows affected (0.00 sec)

```
mysql> delimiter ;
```

```
mysql> select products_with_greater_quantity(4);
```

```
+-----+
| products_with_greater_quantity(4) |
+-----+
|                                     3 |
+-----+
```

```
1 row in set (0.00 sec)
```

### 3) Function that returns the total cost of the cart of a customer

```
mysql> create function totalcost(cId int)
```

```
-> returns numeric
```

```
-> deterministic
```

```
-> begin
```

```
-> return (select grand_total from Cart where userid=cId);
```

```
-> end $$
```

```
Query OK, 0 rows affected (0.00 sec)
```

```
mysql> delimiter ;
```

```
mysql> select totalcost(3);
```

```
mysql> delimiter $$
mysql> create function totalcost(cId int)
  -> returns numeric
  -> deterministic
  -> begin
  -> return (select grand_total from Cart where userid=cId);
  -> end $$
Query OK, 0 rows affected (0.00 sec)

mysql> delimiter ;
mysql> select total_cost(3);
ERROR 1146 (42S02): Table 'e_commerce.Cart_item' doesn't exist
mysql> select totalcost(3);
+-----+
| totalcost(3) |
+-----+
|          2134 |
+-----+
```

4) Function that returns the number of products of particular brand

```
mysql> delimiter $$
```

```
mysql> create function prodtype(type varchar(7))
```

```
-> returns numeric
```

```
-> deterministic
```

```
-> begin
```

```
-> return (select count(Product_name) from product where brands = type);
```

```
-> end $$
```

```
Query OK, 0 rows affected (0.00
```

```
sec) mysql> select prodtype('xyz');
```

```
mysql> delimiter $$
```

```
mysql> create function prodtype(type varchar(7))
```

```
-> returns numeric
```

```
-> deterministic
```

```
-> begin
```

```
-> return (select count(Product_name) from product where brands = type);
```

```
-> end $$
```

```
Query OK, 0 rows affected (0.00 sec)
```

```
mysql> select prodtype('xyz');
```

```
+-----+
```

```
| prodtype('xyz') |
```

```
+-----+
```

```
|                1 |
```

```
+-----+
```

```
1 row in set (0.01 sec)
```

## 5) Function to count number of products in the cart

```
mysql> delimiter $$
mysql> create function totalproducts(cartId int)
  -> returns numeric
  -> deterministic
  -> begin
  -> return (select count(productid) from cart where cartid=cartId);
  -> end $$
```

Query OK, 0 rows affected (0.00 sec)

```
mysql> select totalproducts(1);
-> $$
```

```
mysql> delimiter $$
mysql> create function totalproducts(cartId int)
  -> returns numeric
  -> deterministic
  -> begin
  -> return (select count(productid) from cart where cartid=cartId);
  -> end $$
Query OK, 0 rows affected (0.00 sec)

mysql> select totalproducts(1);
-> $$

+-----+
| totalproducts(1) |
+-----+
|                  4 |
+-----+
1 row in set (0.00 sec)
```

# triggers

1) if an item is ordered by a customer the entry is inserted in order item table then this trigger automatically inserts corresponding record in orders table make order date as current system time and shipping date is 5 days after order date

```
mysql> delimiter $$
mysql> create trigger after_order_insert
  -> after insert on order_item for each row
  -> begin
      -> insert into
orders(userid,orderdate,shippingdate,orderamount,order_status)
values(userid,now(),date_add(now(),interval 5 day),new.quantity*new.MRP,'Shipped today');
  -> end $$
Query OK, 0 rows affected (0.02 sec)
```

```
mysql> delimiter ;
mysql> insert into order_item
values(3,4,2,50); Query OK, 1 row affected
(0.01 sec)
```

```
mysql> select * from order_item;
mysql> select * from orders;
```

```
mysql> delimiter $$
mysql> create trigger after_order_insert
  -> after insert on order_item for each row
  -> begin
      -> insert into orders(userid,orderdate,shippingdate,orderamount,order_status) values(userid,now(),date_add(now(),interval 5 day),new.quantity*new.MRP,'Shipped today');
  -> end $$
Query OK, 0 rows affected (0.02 sec)

mysql> delimiter ;
mysql> insert into order_item values(3,4,2,50);
Query OK, 1 row affected (0.01 sec)

mysql> select * from order_item;
+-----+-----+-----+-----+
| userid | productid | quantity | MRP |
+-----+-----+-----+-----+
| 1 | 1 | 21 | 88.56 |
| 2 | 2 | 12 | 2323 |
| 3 | 3 | 5 | 213 |
| 3 | 5 | 2 | 50 |
| 3 | 4 | 2 | 50 |
+-----+-----+-----+-----+
5 rows in set (0.00 sec)

mysql> select * from orders;
+-----+-----+-----+-----+-----+-----+-----+
| userid | ordernumber | orderdate | shippingdate | orderamount | cartid | order_status |
+-----+-----+-----+-----+-----+-----+-----+
| 1 | 12 | 2020-10-01 00:00:00 | 2020-10-05 00:00:00 | 123 | 1 | shipped to sellesman |
| 2 | 21 | 2021-10-01 00:00:00 | 2021-10-10 00:00:00 | 421 | 2 | arrive |
| 3 | 31 | 2021-10-04 00:00:00 | 2021-10-10 00:00:00 | 2134 | 3 | shipped today |
| NULL | NULL | 2022-05-18 11:34:29 | 2022-05-23 11:34:29 | 100 | NULL | Shipped today |
+-----+-----+-----+-----+-----+-----+-----+
4 rows in set (0.00 sec)
```

2) create a trigger to 10% discount to all the products in cart

```
mysql> delimiter ;
```

```
mysql> delimiter
```

```
$$
```

```
mysql> create trigger update_totals
```

```
-> before insert on cart
```

```
-> for each row
```

```
-> begin
```

```
-> set new.new_total=(0.9)*new.grand_total ;
```

```
-> end
```

```
-> $$
```

```
Query OK, 0 rows affected (0.02 sec)
```

```
mysql> delimiter ;
```

```
mysql> insert into cart(userid,cartid,productid,total_item,grand_total)
```

```
values(5,4,4,7,120); Query OK, 1 row affected (0.01 sec)
```

```
mysql> select * from cart;
```

```
mysql> delimiter ;
mysql> delimiter $$
mysql> create trigger update_totals
  -> before insert on cart
  -> for each row
  -> begin
  -> set new.new_total=(0.9)*new.grand_total ;
  -> end
  -> $$
Query OK, 0 rows affected (0.02 sec)

mysql> delimiter ;
mysql> insert into cart(userid,cartid,productid,total_item,grand_total) values(5,4,4,7,120);
Query OK, 1 row affected (0.01 sec)

mysql> select * from cart;
+-----+-----+-----+-----+-----+-----+
| userid | cartid | productid | total_item | grand_total | new_total |
+-----+-----+-----+-----+-----+-----+
| 1 | 1 | 1 | 5 | 123 | NULL |
| 2 | 2 | 2 | 3 | 421 | NULL |
| 3 | 3 | 3 | 6 | 2134 | NULL |
| 5 | 4 | 4 | 7 | 120 | 108 |
+-----+-----+-----+-----+-----+-----+
4 rows in set (0.01 sec)
```

3) create a trigger to update the sales of seller when a product is sold(an entry is made in product table)

```
mysql> delimiter $$
mysql> create trigger update_sales
-> after insert on product
-> for each row
-> begin
-> declare id int;
-> update seller set total_sales=total_sales+NEW.mrp WHERE NEW.sellerid=sellerid;
->
-> end
-> $$
mysql> select *from seller;
-> $$
mysql> select *from product;
-> $$
```

```
mysql> delimiter $$
mysql> create trigger update_sales
-> after insert on product
-> for each row
-> begin
-> declare id int;
-> update seller set total_sales=total_sales+NEW.mrp WHERE NEW.sellerid=sellerid;
->
-> end
-> $$
Query OK, 0 rows affected (0.03 sec)
```

```
mysql> select *from seller;
-> $$
```

sellerid	name_	UIN	phone	email	total_sales
1	vishal	123	9183767223	eyw@gmail.com	1454.54
2	Paresh	132	8233767223	qwr@gmail.com	3533.33
3	khushal	515	8473667223	iru@gmail.com	2312.23

```
3 rows in set (0.04 sec)
```

```
m *sql> insert into prOduc values(4,
'Gelusil',2,50.50,'Cipla',43); Quer OK, 1 row affected
(0.03 sec)
```

sellerId	name_	UILJ	phone	email	tOtal_sale s
1	visha1	L23	91837o7223	e}tv gma11.com	1454.54
2	Paresh	132	82337o7223	qtvn gmall.	z5g9.sz
3	khusha1	515	8A7 3oo7223	iru gma11. com	23L2.23

```
3 rows In set (6.88 sec)
```

```
m *sql> select * from product;
```

use r Id	product name	se l ledid	PDRP	brand s	stoc ks
1	Teddy	1	123		2
2	Bear		x'z		3
3	Meter BOttle	2	251.23	ctvt	31
4	Keyboard	3	323	ten	12
	Gelusil	2	5ñ .5	C ip la	43

```
4 rows In set (0.00 sec)
```