RAMAIAH
UNIVERSITY
OF APPLIED SCIENCES

ASSIGNMENT

Course Code	19CSC302A
Course Name	Database Systems
Programme	B. Tech
Department	Computer Science and Engineering
Faculty	Engineering and Technology

Name of the Student	Deepak R
Reg. No	18ETCS002041
Semester/Year	5th/2020
Course Leader/s	Ami Rai E

Declaration Sheet			
Student Name	**Deepak R**		
Reg. No	**18ETCS002041**		
Programme	**B. Tech**	Semester/Year	**5th/2020**
Course Code	**19CSC302A**		
Course Title	**Database Systems**		
Course Date		to	
Course Leader	**Mrs Ami Rai E**		

Declaration

The assignment submitted herewith is a result of my own investigations and that I have conformed to the guidelines against plagiarism as laid out in the Student Handbook. All sections of the text and results, which have been obtained from other sources, are fully referenced. I understand that cheating and plagiarism constitute a breach of University regulations and will be dealt with accordingly.

Signature of the Student		Date	05/12/2020
Submission date stamp (by Examination & Assessment Section)			

Signature of the Course Leader and date	Signature of the Reviewer and date

Faculty of Engineering & Technology			
Ramaiah University of Applied Sciences			
Department	Computer Science and Engineering Engineering	**Programme**	B. Tech. Computer Science
Semester/Batch	5th/2018		
Course Code	19CSC302A	**Course Title**	Database Systems
Course Leader(s)	A. Prabhakar, Gp Capt N Rath VSM, Ami Rai E.		

Assignment - 01			
Register No.	**18ETCS002041**	Name of Student	**Deepak R**

Sections		**Marking Scheme**	**Max Marks**	**First Examiner Marks**	**Second Examiner Marks**
Part A	A.1	Merits and demerits of relational and graph databases	02		
	A.2	Justification of the stance taken and conclusion	03		
		Part-A Max Marks	**05**		
Part B1	B1.1	List of functional and data requirements	03		
	B1.2	Discussion on the entities, attributes, and relationships	02		
	B1.3	ER diagram	02		
	B1.4	Identification of any requirement that is not possible to model using ER diagram	03		
		B1 Max Marks	**10**		
Part B2	B2.1	Design of database schema	03		
	B2.2	Discussion on the constraints	02		
	B2.3	Implementation using SQL commands	02		
	B2.4	Update operations violating the schema constraints	03		
		B2 Max Marks	**10**		
		Total Assignment Marks	**25**		

Course Marks Tabulation				
Component-1(B)Assignment	**First Examiner**	**Remarks**	**Second Examiner**	**Remarks**
A				
Marks (out of 10)				
Signature of First Examiner				Signature of Second Examiner

Please note:

1. Documental evidence for all the components/parts of the assessment such as the reports, photographs, laboratory exam / tool tests are required to be attached to the assignment report in a proper order.
2. The First Examiner is required to mark the comments in RED ink and the Second Examiner's comments should be in GREEN ink.
3. The marks for all the questions of the assignment have to be written only in the **Component – CET B: Assignment** table.
4. If the variation between the marks awarded by the first examiner and the second examiner lies within +/- 3 marks, then the marks allotted by the first examiner is considered to be final. If the variation is more than +/- 3 marks then both the examiners should resolve the issue in consultation with the Chairman BoE.

Assignment - 01

Instructions to students:

1. The assignment consists of **3**questions: Part A –**1** Question, Part B- **2**Questions.
2. Maximum marks is **25**.
3. The assignment has to be neatly word processed as per the prescribed format.

Submission Date: 28/11/2020

5. **Submission after the due date is not permitted.**
6. **IMPORTANT**: It is essential that all the sources used in preparation of the assignment must be suitably referenced in the text.
7. Marks will be awarded only to the sections and subsections clearly indicated as per the problem statement/exercise/question

Solution for PART A

"Can graph databases replace relational database technologies".

Introduction

A graph database is a data management system software. The building blocks are vertices and edges. To put it in a more familiar context, a relational database is also a data management software in which the building blocks are tables. Both require loading data into the software and using a query language or APIs to access the data.

Relational databases boomed in the 1980s. Many commercial companies (i.e. Oracle, Ingres, IBM) backed the relational model (tabular organization) of data management. In that era, the main data management need was to generate reports.

Graph databases didn't see a greater advantage over relational databases until recent years, when frequent schema changes, managing explosives volume of data, real-time query response time, and more intelligent data activation requirements make people realize the advantages of the graph model.

Graph DB technology is being rapidly commoditised with platforms like Neo4J and OrientDB leading the way. I believe they will become a new defacto standard in developing all sorts of business and online applications once the inertia of 30+ years of RDBMS thinking is slowly broken down.

Merits and demerits of relational and graph databases

Advantages and Disadvantages of using relational databases

Relational Database

Advantages:

1. Ease of use: The revision of any information as tables consisting of rows and columns is much easier to understand .

2. Flexibility: Different tables from which information has to be linked and extracted can be easily manipulated by operators such as project and join to give information in the form in which it is desired.

3. Precision: The usage of relational algebra and relational calculus in the manipulation of he relations between the tables ensures that there is no ambiguity, which may otherwise arise in establishing the linkages in a complicated network type database.
4.Data Independence: Data independence is achieved more easily with normalization structure used in a relational database than in the more complicated tree or network structure.

5.. Data Manipulation Language: The possibility of responding to query by means of a language based on relational algebra and relational calculus e.g SQL is easy in the relational database approach. For data organized in other structure the query language either becomes complex or extremely limited in its capabilities.

Disadvantages :

1. Performance: A major constraint and therefore disadvantage in the use of relational database system is machine performance. If the number of tables between which relationships to be established are large and the tables themselves effect the performance in responding to the sql queries.

2. Physical Storage Consumption:With an interactive system, for example an operation like join would depend upon the physical storage also. It is, therefore common in relational databases to tune the databases and in such a case the physical data layout would be chosen so as to give good performance in the most frequently run operations. It therefore would naturally result in the fact that the lays frequently run operations would tend to become even more shared.

3. Slow extraction of meaning from data: if the data is naturally organized in a hierarchical manner and stored as such, the hierarchical approach may give quick meaning for that data.

Graph Database

Advantages:
1.Really fast queries when you are looking for relationships between nodes
2.Really fast to traverse nodes
3.Can represent multiple dimensions

Disadvantages :
1.Inappropriate for transactional information, like accounting records where relationships between records are simpler
2.Harder to do summing queries and max queries efficiently - counting queries not harder
3.Usually need to learn a new query language like CIPHER
4.Fewer vendors to choose from, and smaller user base, so harder to get support when you run into issues

Justification of the stance taken and conclusion with example

It depends on what type of data we want to store.

For example if we have financial data like income of people, their social security number and address data a relational database would be appropriate because we can store that information in simple tables and the data from one person is rarely connected in any way to the data of another person.

But if we want to create a network of information, maybe the relationships between our friends and family, a graph database would be the oppportunity to choose because we can display connections between things (nodes) in a very simple way via edges.

In a relational database you would need different tables and connect them through complicated queries.

Hierarchical datasets easily highlight the modelling and query execution differences in both relational and graph databases. However, **this is by no means an indication that you should start throwing away the one database system over the other**. Graph databases are more efficient when data relationships are at the core of our requirement.

Solution for B.1

B1.1 List of functional and data requirements

Functional Requirements for Online Furniture Shopping Management

Table 1.1 Functional Requirement 1

Requirement Tag	FR1
Requirement Description	The system should have an interface to allow a customer to enter their login credentials/ Create a new Account
Dependent on Requirements	NA
User/System interacting with the requirement	Customer

Table 1.2 Functional Requirement 2

Requirement Tag	FR2
Requirement Description	The system should have an interface to allow a customer to add Products(Furniture) in cart
Dependent on Requirements	F1
User/System interacting with the requirement	Customer

Table 1.3 Functional Requirement 3

Requirement Tag	FR3
Requirement Description	The admin must be Provided with Interface to update about Availability of Product(furniture)
Dependent on Requirements	F1
User/System interacting with the requirement	admin

Table 1.4 Functional Requirement 4

Requirement Tag	FR4
Requirement Description	The admin must be be able to view details of Customer.
Dependent on Requirements	F1
User/System interacting with the requirement	admin

Table 1.5 Functional Requirement 5

Requirement Tag	FR5
Requirement Description	The Customer must be able to delete app account if he/she wish
Dependent on Requirements	F1
User/System interacting with the requirement	User

Data Requirements for Bank Management System

Table 2.1 Data Requirement 1

Requirement Tag	DR1
Item Name	CUSTOMER_ID, Customer_pass,Customer_Name,Customer_Contact
Item Description (Where/How used)	The customer will be entering username and password to the system either to Login to Account/ Create a new Account.
Item type	Integer, varchar ,string ,Integer
User/System interacting with the item	Customer

Table 2.2 Data Requirement 2

Requirement Tag	DR2
Item Name	CUSTOMER_ID, Product_ID,Cat_Id
Item Description (Where/How used)	The customer will be entering Customer_ID,Product_ID,Category ID(Cat_ID) to add product into cart
Item type	Integer, Integer,Integer
User/System interacting with the item	Customer

Table 2.3 Data Requirement 3

Requirement Tag	DR3
Item Name	user_ID, user_Pswd , Product_Id
Item Description (Where/How used)	The user(admin) must be able to update about availability of Furniture(Product) using user_ID, user_Pswd , Product_Id
Item type	Integer, varchar , Integer
User/System interacting with the item	admin

Table 2.4 Data Requirement 4

Requirement Tag	DR4
Item Name	user_ID, Customer_Id
Item Description (Where/How used)	The user(admin) must be able to view Customer detlails using Customer_Id
Item type	Integer, Integer
User/System interacting with the item	admin

Table 2.3 Data Requirement 5

Requirement Tag	DR5
Item Name	Customer_id , Customer_Pass , Customer_Contact
Item Description (Where/How used)	The Customer must be able to delete app account by using Customer_id , Customer_Pass and through Customer_Contact Vrification
Item type	Integer, varchar , Integer
User/System interacting with the item	user

B1.2 Discussion on the entities, attributes, and relationships

Relationships are:

Cart Filled_by Customer -> 1 : 1

One Customer can fill only one product in cart at a time

Cart Contains product -> 1 : N

One Cart Contains N number of products

Products Managed_BY admins -> N: M

N number *admins can manage M number of Products*

Categories Managed_BY admins -> N: M

N number *admins can manage M number of Catogories .*

Customer Viewed_By admins -> N : M

M number of admins can view M number of customers details

Customer Selects Catagories -> N : M

N number of Customers Select M number of Categories.

Customer Buy Products -> N :M

N Customer can Buy M number of products

Discussion on the entities and attributes

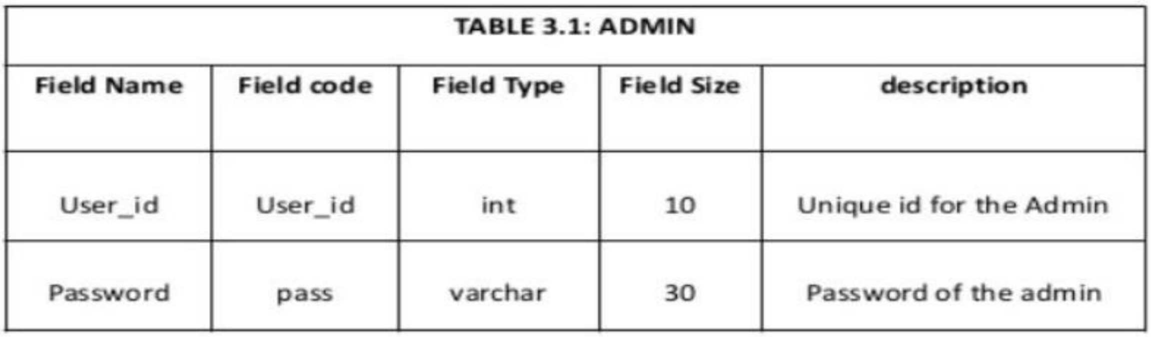


TABLE 3.1: ADMIN				
Field Name	Field code	Field Type	Field Size	description
User_id	User_id	int	10	Unique id for the Admin
Password	pass	varchar	30	Password of the admin

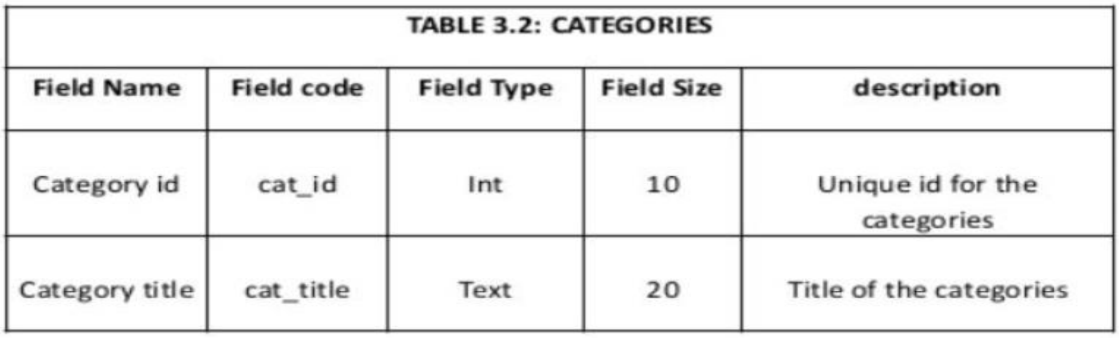


TABLE 3.2: CATEGORIES				
Field Name	Field code	Field Type	Field Size	description
Category id	cat_id	Int	10	Unique id for the categories
Category title	cat_title	Text	20	Title of the categories

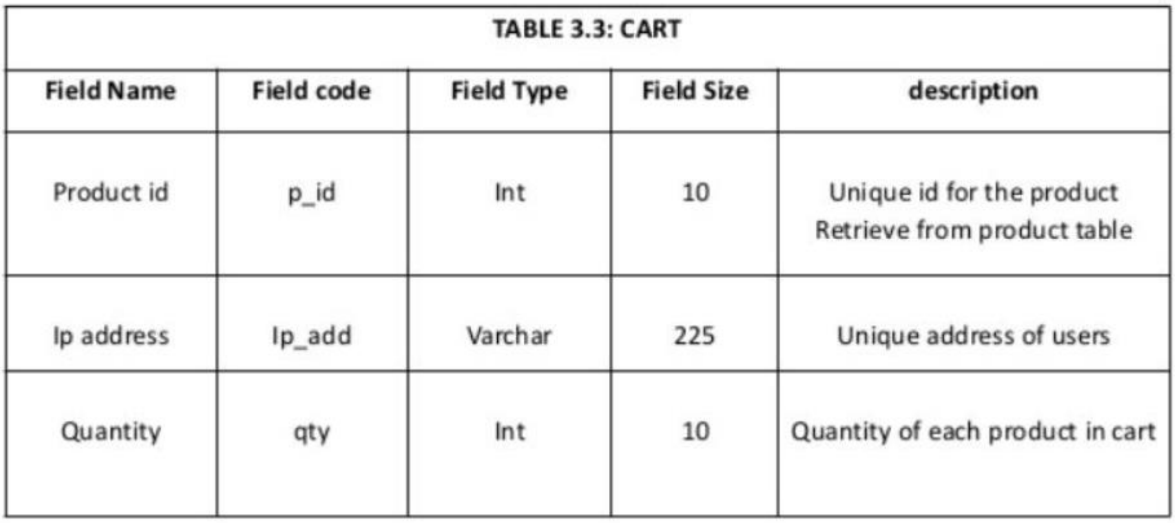


TABLE 3.3: CART				
Field Name	Field code	Field Type	Field Size	description
Product id	p_id	Int	10	Unique id for the product Retrieve from product table
Ip address	Ip_add	Varchar	225	Unique address of users
Quantity	qty	Int	10	Quantity of each product in cart

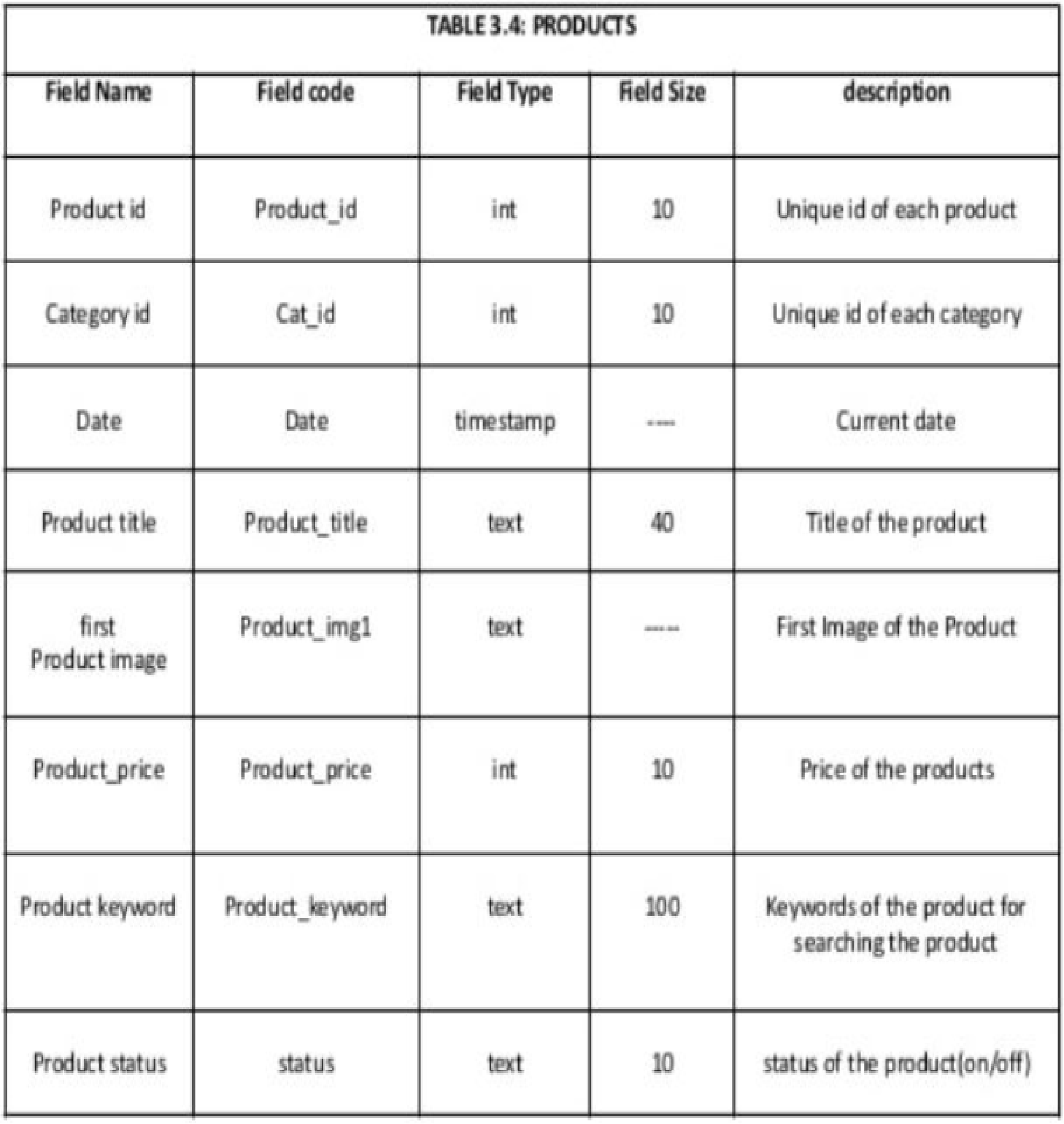


TABLE 3.4: PRODUCTS

Field Name	Field code	Field Type	Field Size	description
Product id	Product_id	int	10	Unique id of each product
Category id	Cat_id	int	10	Unique id of each category
Date	Date	timestamp	----	Current date
Product title	Product_title	text	40	Title of the product
first Product image	Product_img1	text	-----	First Image of the Product
Product_price	Product_price	int	10	Price of the products
Product keyword	Product_keyword	text	100	Keywords of the product for searching the product
Product status	status	text	10	status of the product(on/off)

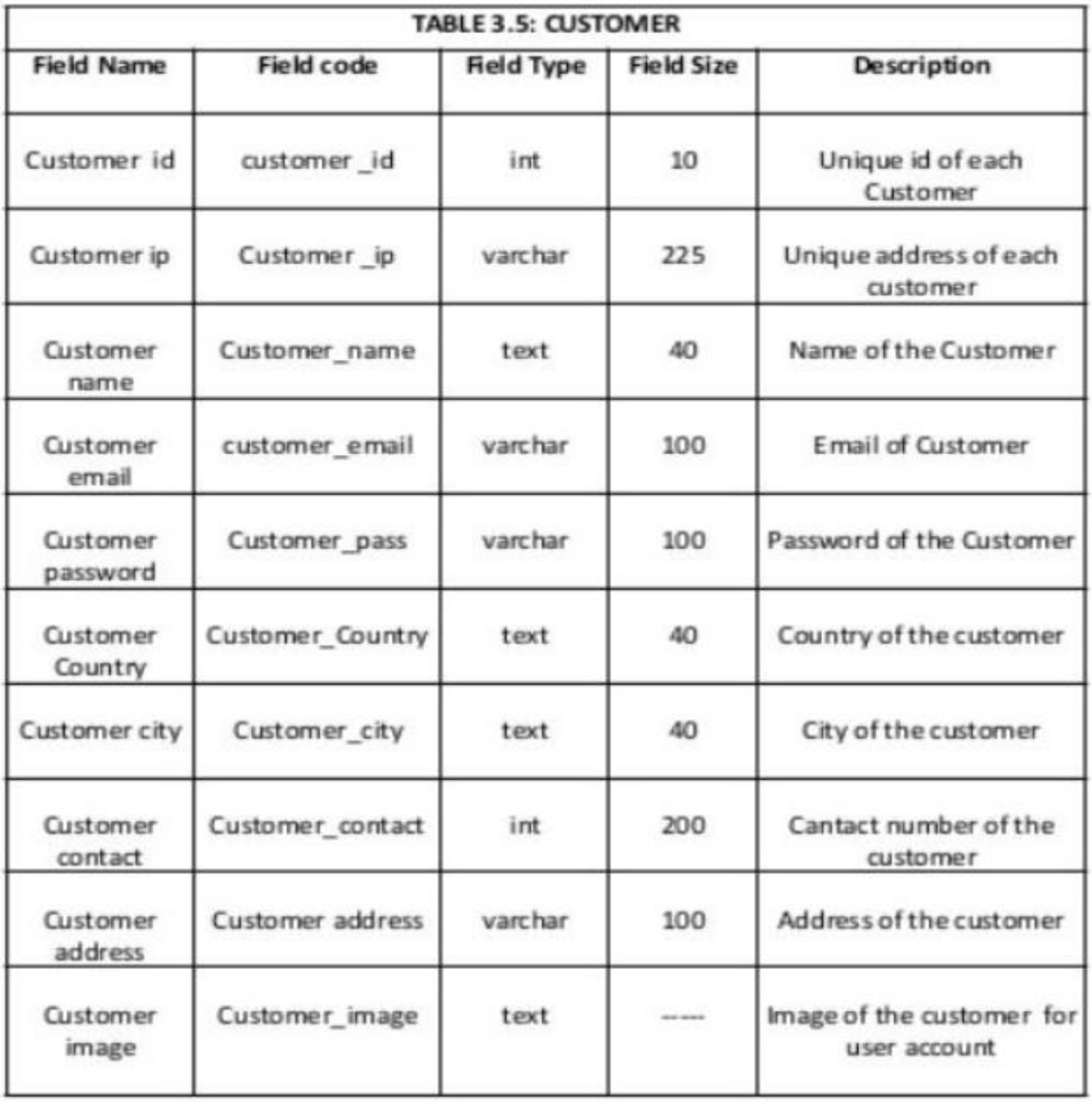
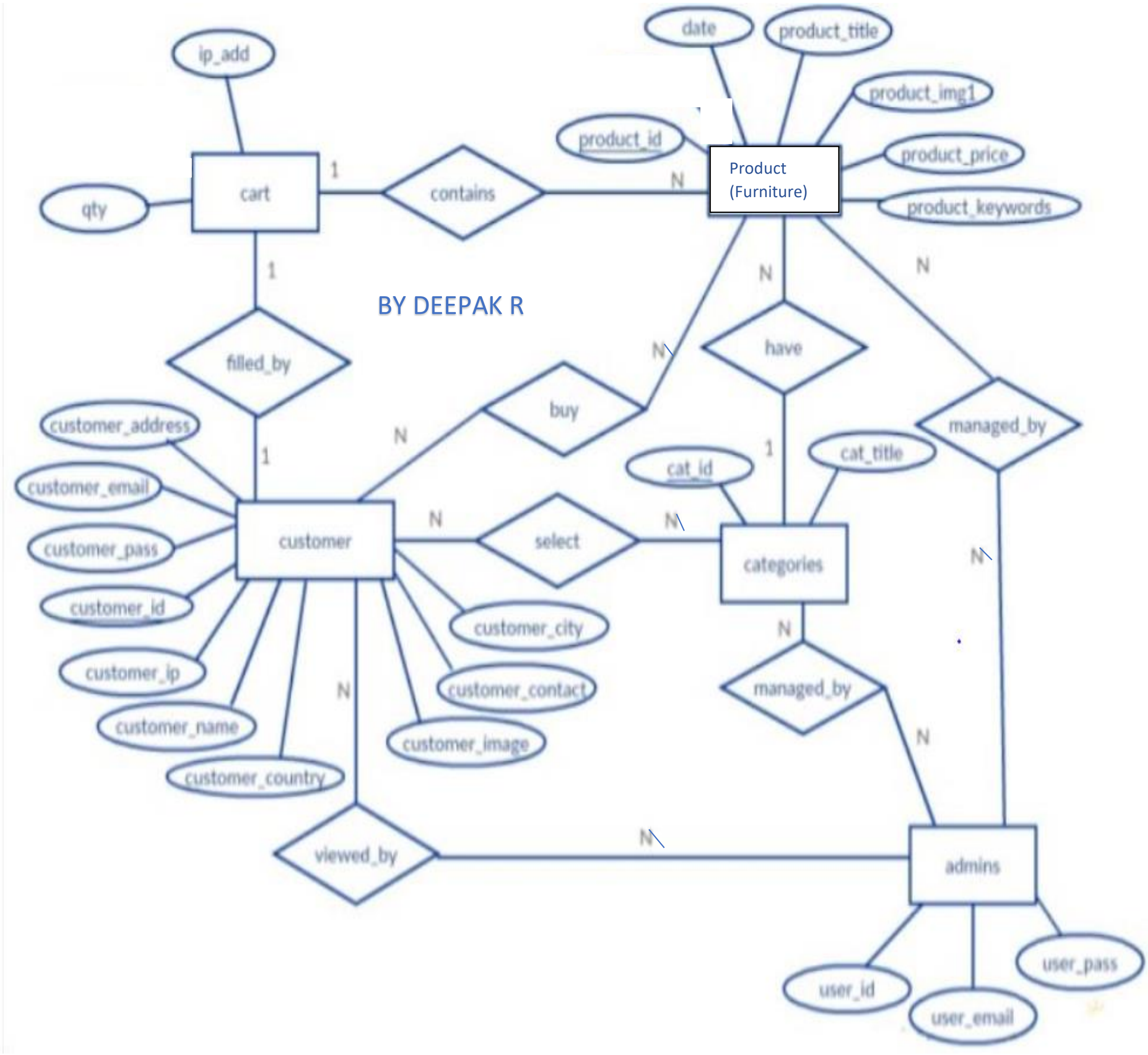


TABLE 3.5: CUSTOMER

Field Name	Field code	Field Type	Field Size	Description
Customer id	customer _id	int	10	Unique id of each Customer
Customer ip	Customer _ip	varchar	225	Unique address of each customer
Customer name	Customer_name	text	40	Name of the Customer
Customer email	customer_email	varchar	100	Email of Customer
Customer password	Customer_pass	varchar	100	Password of the Customer
Customer Country	Customer_Country	text	40	Country of the customer
Customer city	Customer_city	text	40	City of the customer
Customer contact	Customer_contact	int	200	Cantact number of the customer
Customer address	Customer address	varchar	100	Address of the customer
Customer image	Customer_image	text	-----	Image of the customer for user account

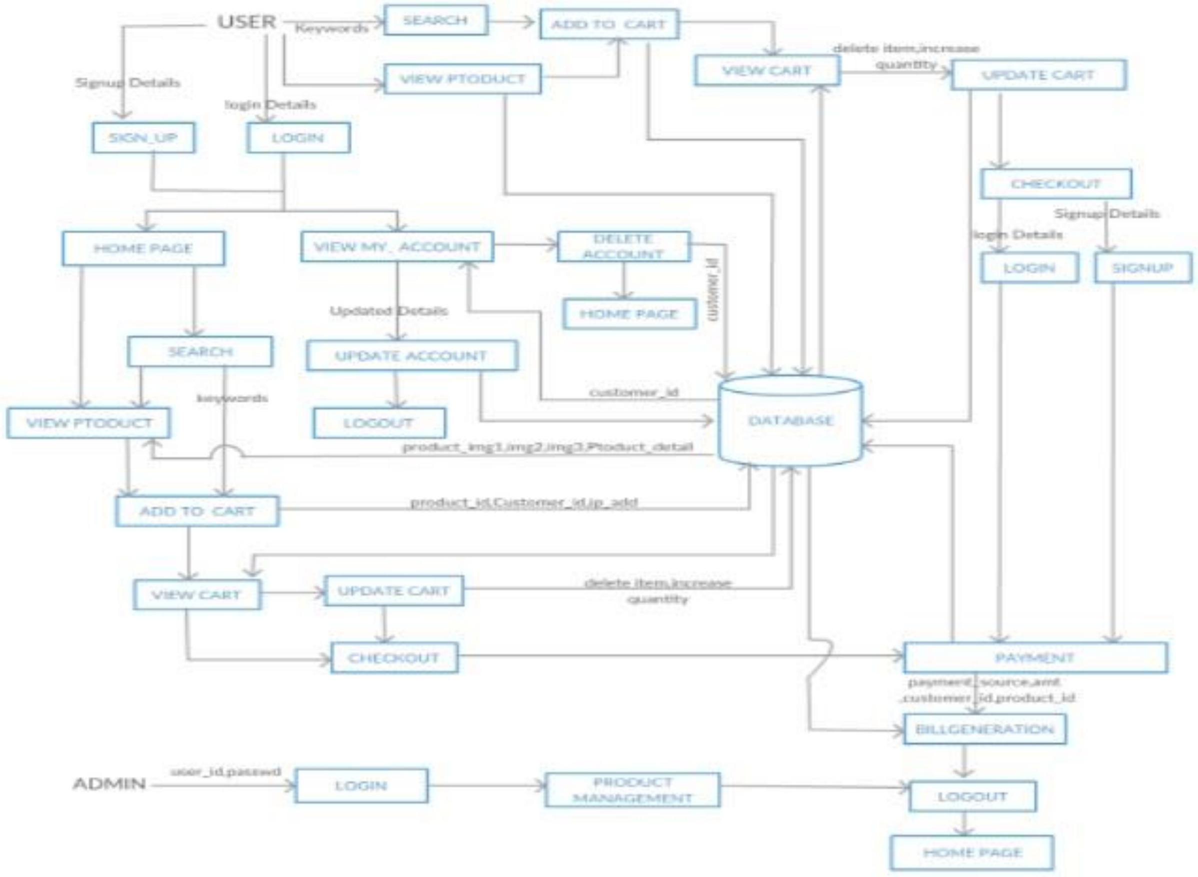
B1.3 Modelling of ER diagram



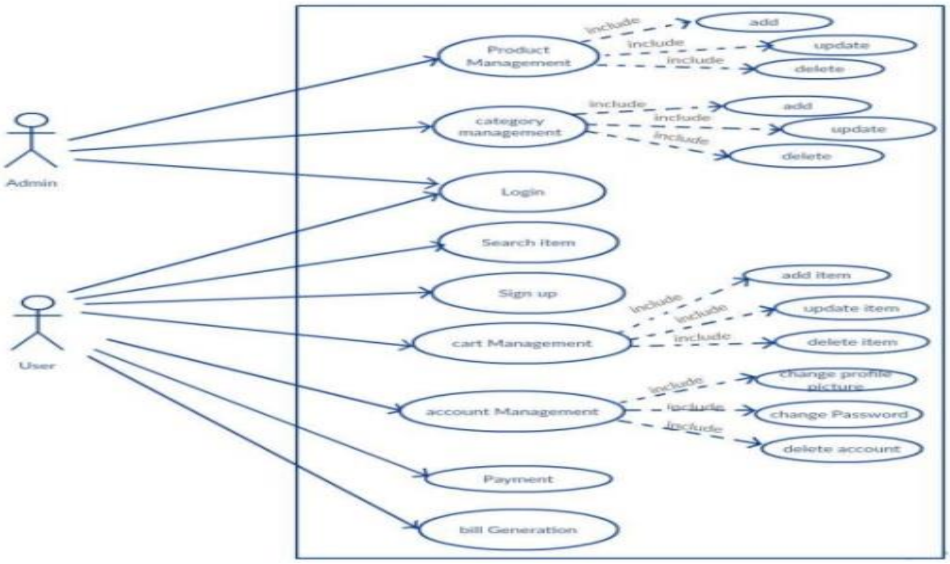
B1.4 Identification of any requirements that is not able to capture in the ER diagram and justify the way to solve it using other conceptual data models

In ER diagram we cannot represent how the customer can delete account if he don't need it or about how admins can update the non availability of Furniture or products Specifically we cannot show how the flow take place to solve this problem we can use Block diagramming and user case diagram to represent our flow of work.

Block diagramming



User case Diagram



Solution for B2

B2.1 Design of database schema

Step1:Mapping of Regular Entity Types

- ✓ **CUSTOMER**

CUSTOMER_ID	Customer_IP	Customer_Name	Customer_Email	Customer_Country	Customer_Contact	Customer_Address	Customer_Image	Customer_city	Customer_pas

- ✓ **Admin**

User_ID	User_pass	User_email

- ✓ **Categories**

Cat_ID	Cat_titte

- ✓ **Cart**

	qty	Ip_add

- ✓ **Product**

Product_ID	Product_date	Product_tittle	Product_Price	Product_Keyword	Product_imag	Product_Status

Step2:Mapping of Weak EntityTypes

- ✓ Not applicable as there are no weak entity types identified.

Step3:Mapping of Binary 1:1 Relation Types

- ✓ ***Cart Filled_by Customer -> 1 : 1***

One Customer can fill only one product in cart at a time

In this relationship, entity cart is identified as relation S and entity Customer is identified as relation T. The primary key of T i.e. **Customer_id** is now included as foreign key **Cust_id** in S.

Step4:Mapping of Binary 1:N Relationship Types

- ✓ ***Cart Contains product -> 1 : N***

One Cart Contains N number of products

In this relationship, entity Cart is identified as relation S and entity Product is identified as relation T. The primary key of T i.e. **Product_Id** is now included as foreign key **P_id** in S.

Step5:Mapping of Binary M:N Relationship Types

✓ ***Products Managed_BY admins -> N: M***

N number *admins can manage M number of Products So the Primary keys of Products and admins are introduced as foreign keys in Managed_By Relation i.e Product_id as mProduct_id and User_id as mUser_id.*

✓ **Managed_by**

mProduct_ID	mUser_id

✓ ***Customer Viewed_By admins -> N : M***

N number *admins can View M number of Customer details So the Primary keys of Customer and admins are introduced as foreign keys in Viewed_By Relation i.e Customer_id as vCustomer_id and User_id as vUser_id.*

✓ **Viewed_by**

vCustomer_ID	vUser_id

✓ ***Customer Selects Catagories -> N : M***

N number of Customers Select M number of Categories. *So the Primary keys of Customers and Categories are introduced as foreign keys in Selects Relation i.e Customer_id as sCustomer_id and Categories_id as sCategories_id.*

✓ **Selects**

sCustomer_ID	sCategories_id

✓ ***Customer Buy Products -> N :M***

N Customer can Buy M number of products. *So the Primary keys of Customers and Products are introduced as foreign keys in Selects Relation i.e Customer_id as bCustomer_id and Product_id as bProduct_id.*

✓ **Buy**

bCustomer_ID	bProduct_id

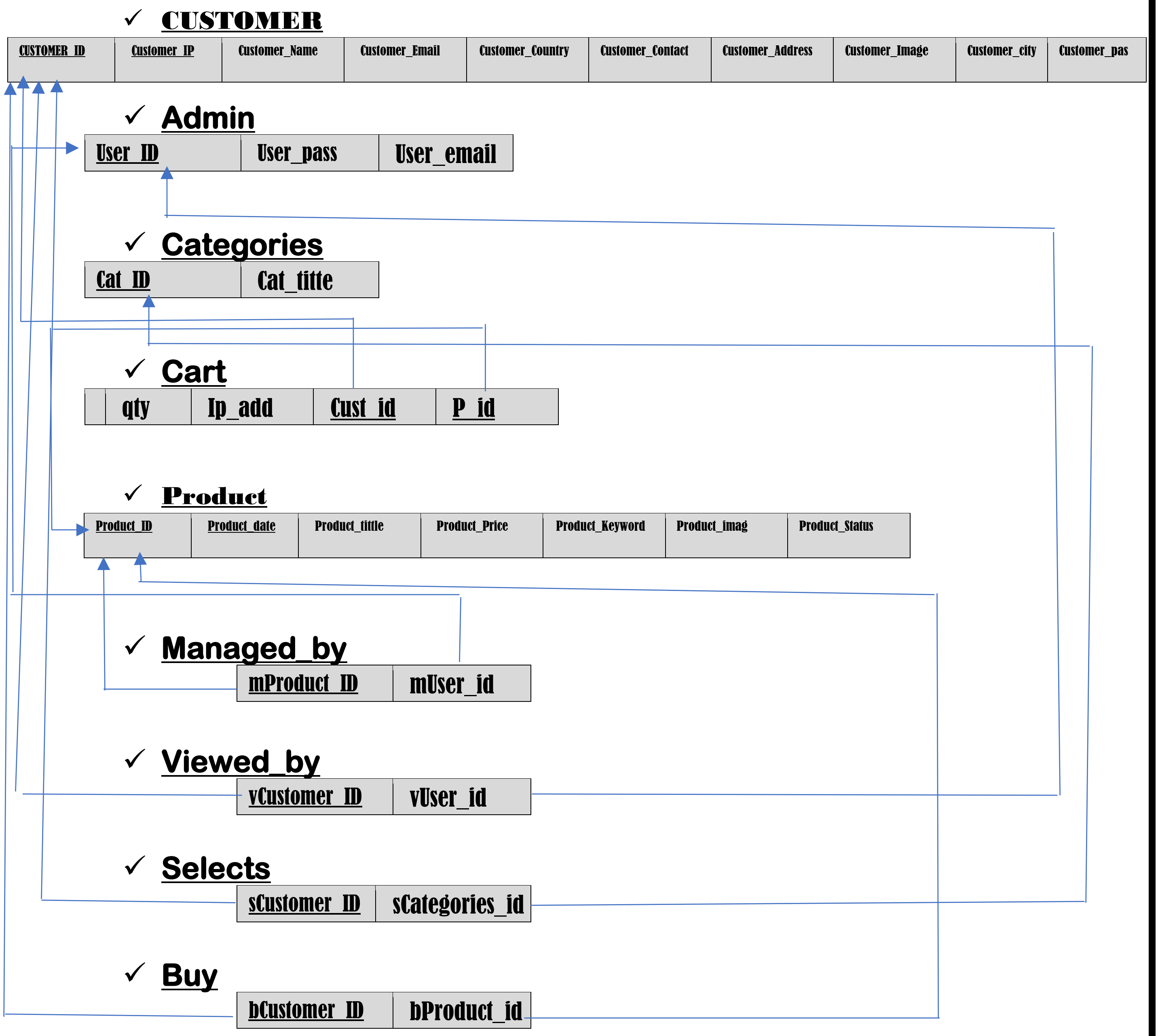
Step6:Mapping of Multivalued attributes

Not applicable as there are no Multivalated attributes identified

Step 7: Mapping of N-ary relation types

Not applicable as there are no N-ary relationships identified.

The final set of relational schemas obtained after applying the mapping algorithm are: -



✓ **CUSTOMER**

CUSTOMER_ID	Customer_IP	Customer_Name	Customer_Email	Customer_Country	Customer_Contact	Customer_Address	Customer_Image	Customer_city	Customer_pas

✓ **Admin**

User_ID	User_pass	User_email

✓ **Categories**

Cat_ID	Cat_titte

✓ **Cart**

	qty	Ip_add	Cust_id	P_id

✓ **Product**

Product_ID	Product_date	Product_tittle	Product_Price	Product_Keyword	Product_imag	Product_Status

✓ **Managed_by**

mProduct_ID	mUser_id

✓ **Viewed_by**

vCustomer_ID	vUser_id

✓ **Selects**

sCustomer_ID	sCategories_id

✓ **Buy**

bCustomer_ID	bProduct_id

B2.2 Discussion on the schema based constraints applicable for the developed schema

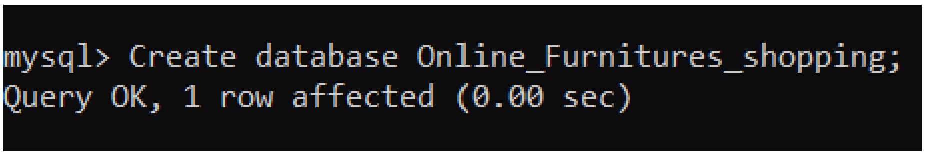
In Given Schema There are Key Constraints,Referential integrity Constraints,Entity integrity constraints and also Domain Constraints.

In key Constraints Here Primary key Cannot be Null and also it is used to identify specific tuple so its Value Should be unique In given Schema we have Customer_id,Product_id user_id as the Primary keys in customer,Product and admin respectively.

In Entity Integrity Constraints no primary key value can be NULL So Customer_id,Product_id user_id of customer,Product and admin entities respectively cannot be NULL.

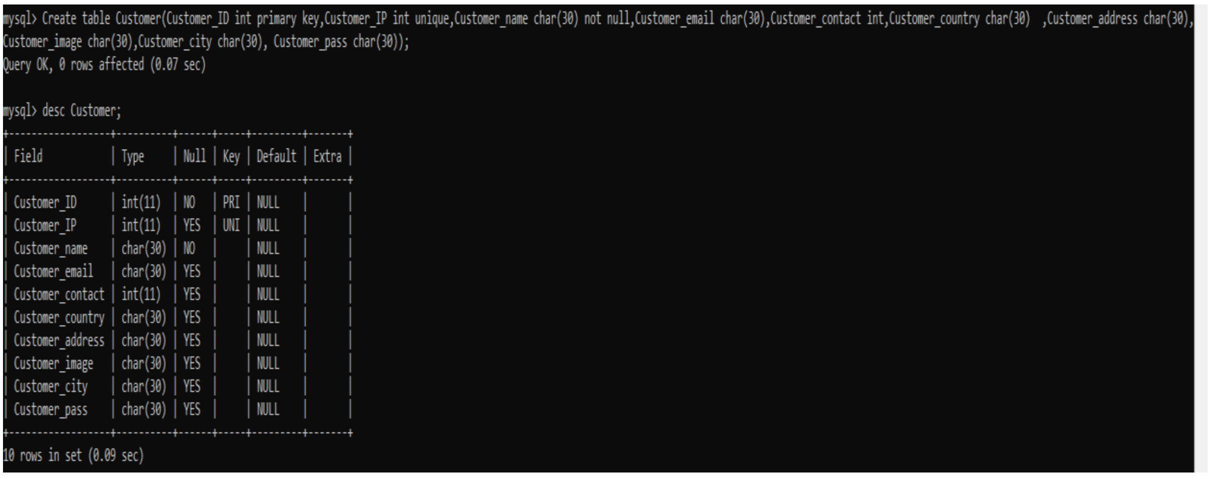
In Entity Integrity Constraints the referenced and referencing attributes Value should be Same So For Customer_id in customer"s Value should be same as SCustomer_id in Select Relation,Product_id in Product's Value Should be Same as BProduct_id values in Buys relation and s on.

B2.3 Implementation of relational database schema with appropriate attributes, and constraints using SQL commands



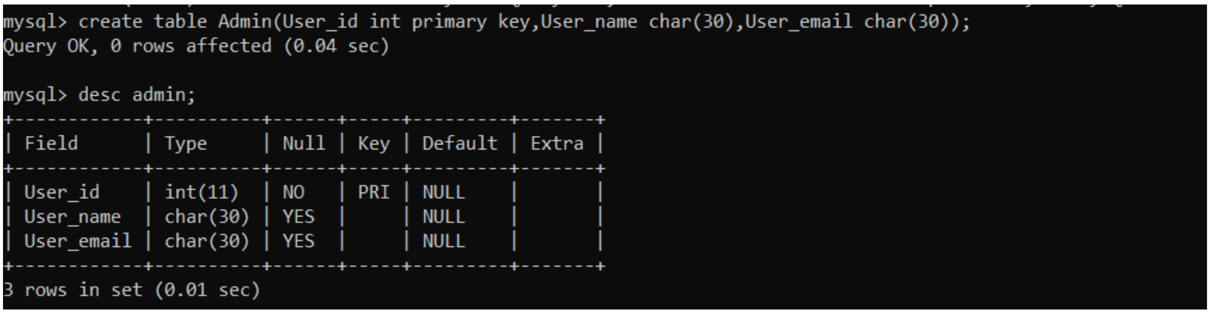
```
mysql> Create database Online_Furnitures_shopping;
Query OK, 1 row affected (0.00 sec)
```

Fig 2.3.1 Created Database named Online_Furniture_System



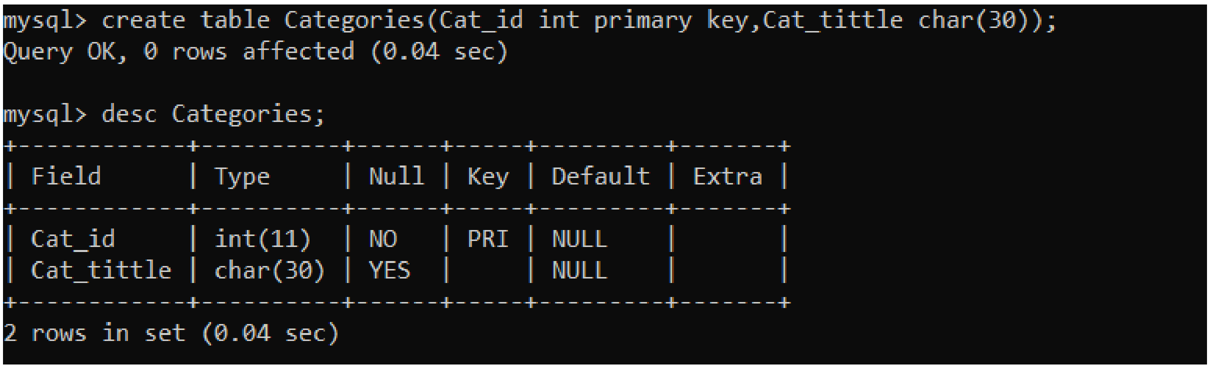
```
mysql> Create table Customer(Customer_ID int primary key,Customer_IP int unique,Customer_name char(30) not null,Customer_email char(30),Customer_contact int,Customer_country char(30)  ,Customer_address char(30),
Customer_image char(30),Customer_city char(30), Customer_pass char(30));
Query OK, 0 rows affected (0.07 sec)

mysql> desc Customer;
+------------------+----------+------+-----+---------+-------+
| Field            | Type     | Null | Key | Default | Extra |
+------------------+----------+------+-----+---------+-------+
| Customer_ID      | int(11)  | NO   | PRI | NULL    |       |
| Customer_IP      | int(11)  | YES  | UNI | NULL    |       |
| Customer_name    | char(30) | NO   |     | NULL    |       |
| Customer_email   | char(30) | YES  |     | NULL    |       |
| Customer_contact | int(11)  | YES  |     | NULL    |       |
| Customer_country | char(30) | YES  |     | NULL    |       |
| Customer_address | char(30) | YES  |     | NULL    |       |
| Customer_image   | char(30) | YES  |     | NULL    |       |
| Customer_city    | char(30) | YES  |     | NULL    |       |
| Customer_pass    | char(30) | YES  |     | NULL    |       |
+------------------+----------+------+-----+---------+-------+
10 rows in set (0.09 sec)
```

Fig 2.3.2 Created table named Customer with its respective attributes



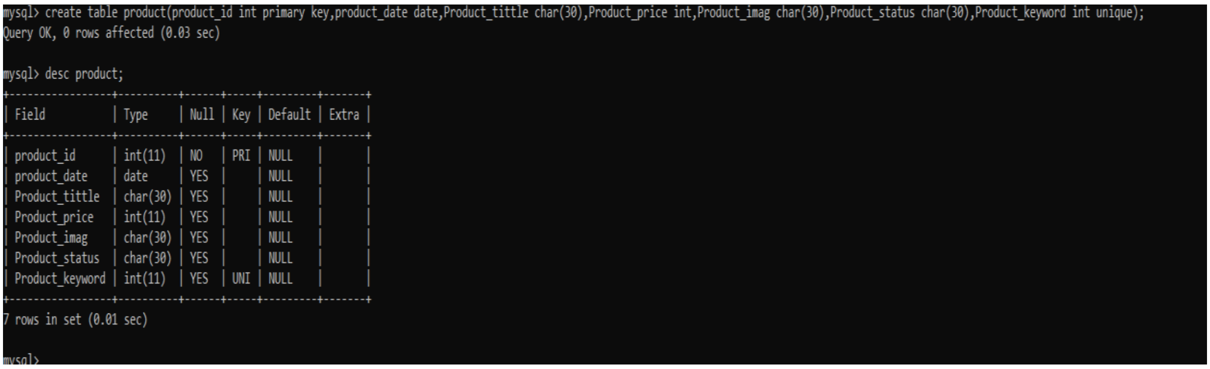
```
mysql> create table Admin(User_id int primary key,User_name char(30),User_email char(30));
Query OK, 0 rows affected (0.04 sec)

mysql> desc admin;
+------------+----------+------+-----+---------+-------+
| Field      | Type     | Null | Key | Default | Extra |
+------------+----------+------+-----+---------+-------+
| User_id    | int(11)  | NO   | PRI | NULL    |       |
| User_name  | char(30) | YES  |     | NULL    |       |
| User_email | char(30) | YES  |     | NULL    |       |
+------------+----------+------+-----+---------+-------+
3 rows in set (0.01 sec)
```

Fig 2.3.3 Created table named Admin with its respective attributes



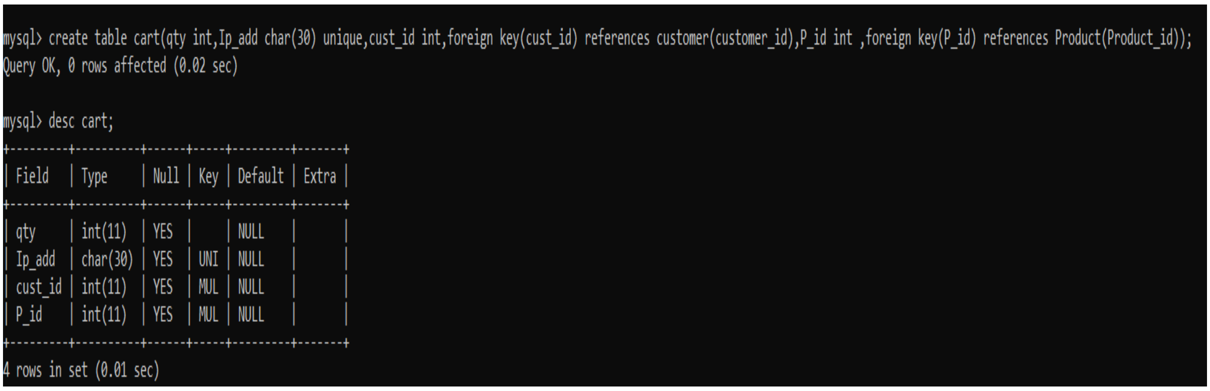
```
mysql> create table Categories(Cat_id int primary key,Cat_tittle char(30));
Query OK, 0 rows affected (0.04 sec)

mysql> desc Categories;
+------------+----------+------+-----+---------+-------+
| Field      | Type     | Null | Key | Default | Extra |
+------------+----------+------+-----+---------+-------+
| Cat_id     | int(11)  | NO   | PRI | NULL    |       |
| Cat_tittle | char(30) | YES  |     | NULL    |       |
+------------+----------+------+-----+---------+-------+
2 rows in set (0.04 sec)
```

Fig 2.3.4 Created table named Categories with its respective attributes



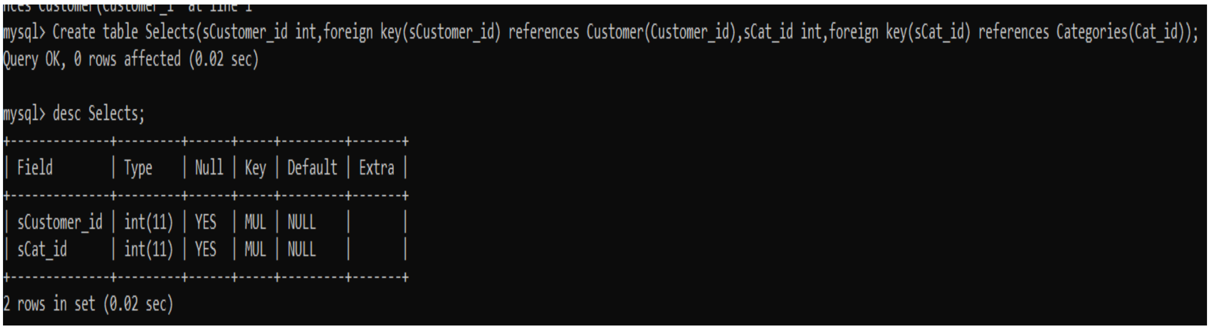
```
mysql> create table product(product_id int primary key,product_date date,Product_tittle char(30),Product_price int,Product_imag char(30),Product_status char(30),Product_keyword int unique);
Query OK, 0 rows affected (0.03 sec)

mysql> desc product;
+-----------------+----------+------+-----+---------+-------+
| Field           | Type     | Null | Key | Default | Extra |
+-----------------+----------+------+-----+---------+-------+
| product_id      | int(11)  | NO   | PRI | NULL    |       |
| product_date    | date     | YES  |     | NULL    |       |
| Product_tittle  | char(30) | YES  |     | NULL    |       |
| Product_price   | int(11)  | YES  |     | NULL    |       |
| Product_imag    | char(30) | YES  |     | NULL    |       |
| Product_status  | char(30) | YES  |     | NULL    |       |
| Product_keyword | int(11)  | YES  | UNI | NULL    |       |
+-----------------+----------+------+-----+---------+-------+
7 rows in set (0.01 sec)

mysql>
```

Fig 2.3.5 Created table named Product with its respective attributes



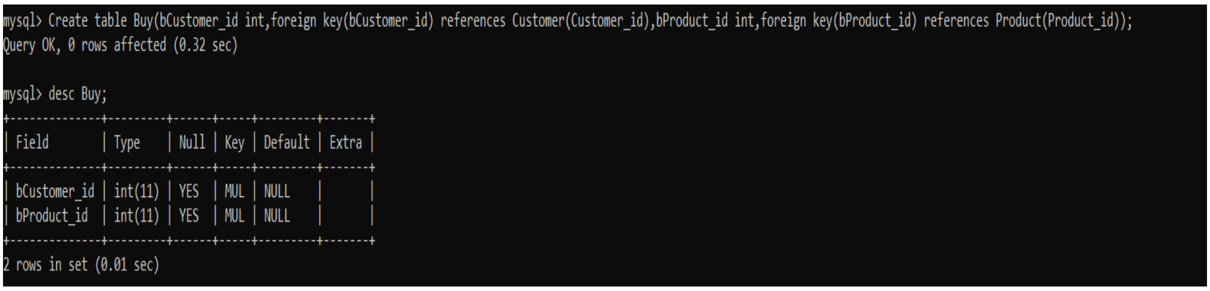
```
mysql> create table cart(qty int,Ip_add char(30) unique,cust_id int,foreign key(cust_id) references customer(customer_id),P_id int ,foreign key(P_id) references Product(Product_id));
Query OK, 0 rows affected (0.02 sec)

mysql> desc cart;
+---------+----------+------+-----+---------+-------+
| Field   | Type     | Null | Key | Default | Extra |
+---------+----------+------+-----+---------+-------+
| qty     | int(11)  | YES  |     | NULL    |       |
| Ip_add  | char(30) | YES  | UNI | NULL    |       |
| cust_id | int(11)  | YES  | MUL | NULL    |       |
| P_id    | int(11)  | YES  | MUL | NULL    |       |
+---------+----------+------+-----+---------+-------+
4 rows in set (0.01 sec)
```

Fig 2.3.6 Created table named cart with its respective attributes



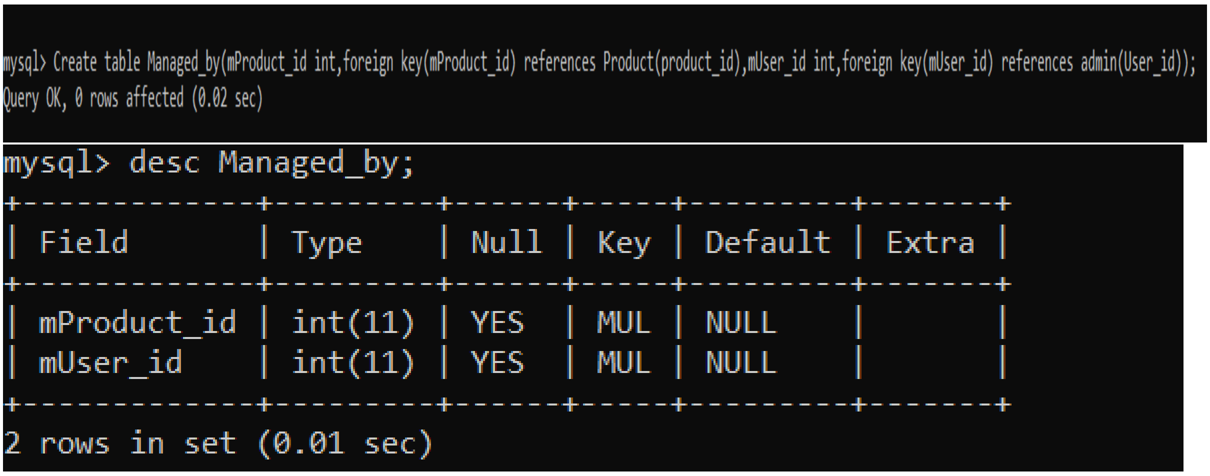
```
nces Customer(Customer_i' at line 1
mysql> Create table Selects(sCustomer_id int,foreign key(sCustomer_id) references Customer(Customer_id),sCat_id int,foreign key(sCat_id) references Categories(Cat_id));
Query OK, 0 rows affected (0.02 sec)

mysql> desc Selects;
+--------------+---------+------+-----+---------+-------+
| Field        | Type    | Null | Key | Default | Extra |
+--------------+---------+------+-----+---------+-------+
| sCustomer_id | int(11) | YES  | MUL | NULL    |       |
| sCat_id      | int(11) | YES  | MUL | NULL    |       |
+--------------+---------+------+-----+---------+-------+
2 rows in set (0.02 sec)
```

Fig 2.3.7 Created table named Selects with its respective attributes



```
mysql> Create table Buy(bCustomer_id int,foreign key(bCustomer_id) references Customer(Customer_id),bProduct_id int,foreign key(bProduct_id) references Product(Product_id));
Query OK, 0 rows affected (0.32 sec)

mysql> desc Buy;
+--------------+---------+------+-----+---------+-------+
| Field        | Type    | Null | Key | Default | Extra |
+--------------+---------+------+-----+---------+-------+
| bCustomer_id | int(11) | YES  | MUL | NULL    |       |
| bProduct_id  | int(11) | YES  | MUL | NULL    |       |
+--------------+---------+------+-----+---------+-------+
2 rows in set (0.01 sec)
```

Fig 2.3.8 Created table named Buy with its respective attributes



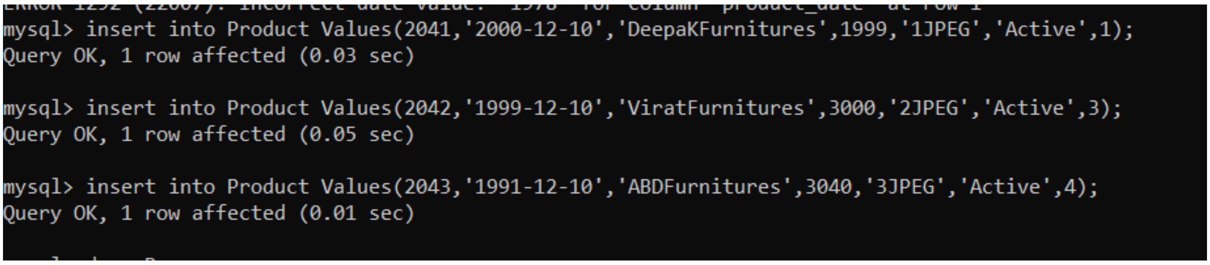
```
mysql> Create table Managed_by(mProduct_id int,foreign key(mProduct_id) references Product(product_id),mUser_id int,foreign key(mUser_id) references admin(User_id));
Query OK, 0 rows affected (0.02 sec)
```

```
mysql> desc Managed_by;
+-------------+---------+------+-----+---------+-------+
| Field       | Type    | Null | Key | Default | Extra |
+-------------+---------+------+-----+---------+-------+
| mProduct_id | int(11) | YES  | MUL | NULL    |       |
| mUser_id    | int(11) | YES  | MUL | NULL    |       |
+-------------+---------+------+-----+---------+-------+
2 rows in set (0.01 sec)
```

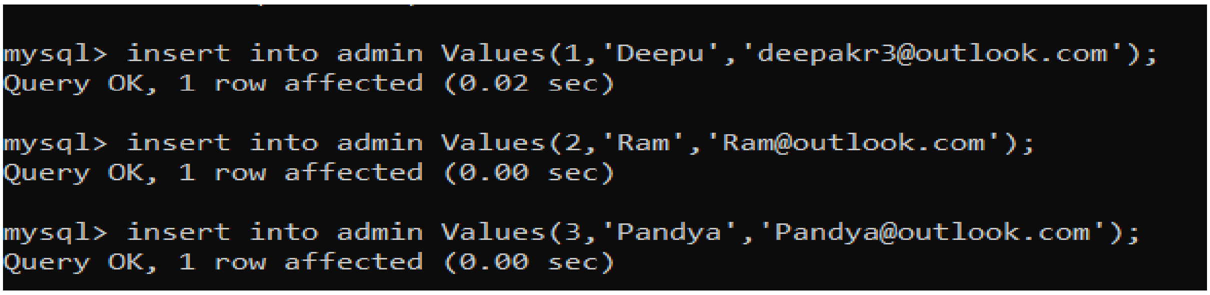
Fig 2.3.9 Created table named Managed_by with its respective attributes.

B2.4 Show how the update operations violate the schema based constraints by executing SQL commands

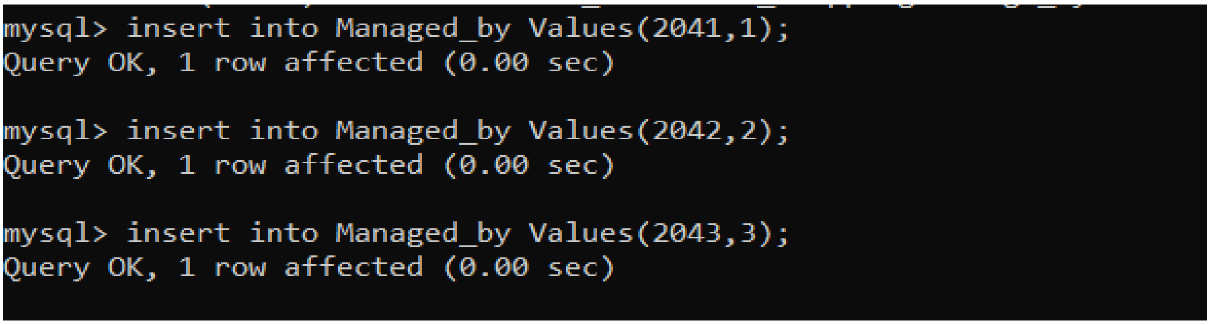
To Show where Update Operations Violates Schema Based Constraints we First inserted data into Product,admins,Mnaged_By entities.



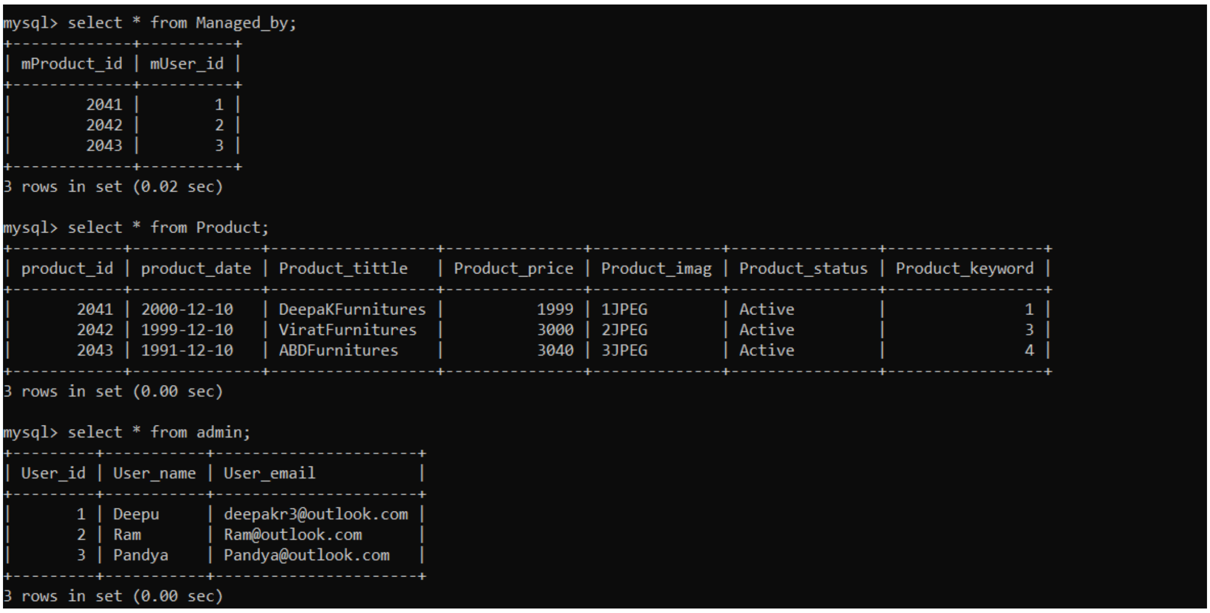
```
ERROR 1292 (22007): Incorrect date value: '1978' for column 'product_date' at row 1
mysql> insert into Product Values(2041,'2000-12-10','DeepaKFurnitures',1999,'1JPEG','Active',1);
Query OK, 1 row affected (0.03 sec)

mysql> insert into Product Values(2042,'1999-12-10','ViratFurnitures',3000,'2JPEG','Active',3);
Query OK, 1 row affected (0.05 sec)

mysql> insert into Product Values(2043,'1991-12-10','ABDFurnitures',3040,'3JPEG','Active',4);
Query OK, 1 row affected (0.01 sec)
```



```
mysql> insert into admin Values(1,'Deepu','deepakr3@outlook.com');
Query OK, 1 row affected (0.02 sec)

mysql> insert into admin Values(2,'Ram','Ram@outlook.com');
Query OK, 1 row affected (0.00 sec)

mysql> insert into admin Values(3,'Pandya','Pandya@outlook.com');
Query OK, 1 row affected (0.00 sec)
```

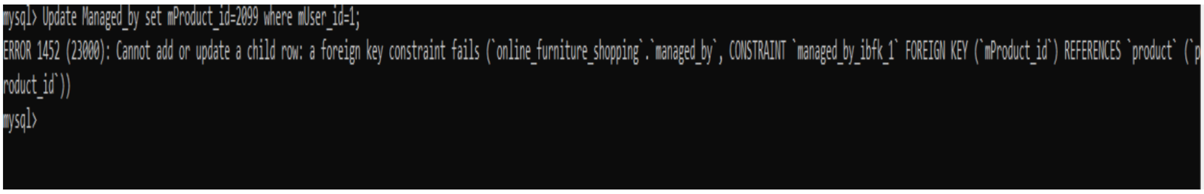


```
mysql> insert into Managed_by Values(2041,1);
Query OK, 1 row affected (0.00 sec)

mysql> insert into Managed_by Values(2042,2);
Query OK, 1 row affected (0.00 sec)

mysql> insert into Managed_by Values(2043,3);
Query OK, 1 row affected (0.00 sec)
```



```
mysql> select * from Managed_by;
+-------------+----------+
| mProduct_id | mUser_id |
+-------------+----------+
|        2041 |        1 |
|        2042 |        2 |
|        2043 |        3 |
+-------------+----------+
3 rows in set (0.02 sec)

mysql> select * from Product;
+------------+--------------+------------------+---------------+--------------+----------------+-----------------+
| product_id | product_date | Product_tittle   | Product_price | Product_imag | Product_status | Product_keyword |
+------------+--------------+------------------+---------------+--------------+----------------+-----------------+
|       2041 | 2000-12-10   | DeepaKFurnitures |          1999 | 1JPEG        | Active         |               1 |
|       2042 | 1999-12-10   | ViratFurnitures  |          3000 | 2JPEG        | Active         |               3 |
|       2043 | 1991-12-10   | ABDFurnitures    |          3040 | 3JPEG        | Active         |               4 |
+------------+--------------+------------------+---------------+--------------+----------------+-----------------+
3 rows in set (0.00 sec)

mysql> select * from admin;
+---------+-----------+----------------------+
| User_id | User_name | User_email           |
+---------+-----------+----------------------+
|       1 | Deepu     | deepakr3@outlook.com |
|       2 | Ram       | Ram@outlook.com      |
|       3 | Pandya    | Pandya@outlook.com   |
+---------+-----------+----------------------+
3 rows in set (0.00 sec)
```

Then We tried to Update Managed_by fuction'sMProduct_id which is foreign key from Products.



```
mysql> Update Managed_by set mProduct_id=2099 where mUser_id=1;
ERROR 1452 (23000): Cannot add or update a child row: a foreign key constraint fails (`online_furniture_shopping`.`managed_by`, CONSTRAINT `managed_by_ibfk_1` FOREIGN KEY (`mProduct_id`) REFERENCES `product` (`p
roduct_id`))
mysql>
```

System Displayed Error Message Saying Cannot Update a Child row.

By this we showed how update operations violate the schema based constraints by executing SQL commands.

References

Ponnaiah,P.,2007,Data Modeling Fundamentals,Wiley.