**PART-A**

**A.1)Merits and demerits of relational and graph data base**

**Merits of relational database:-**1.A relational database is much faster when operating on huge numbers of records. In a graph database.

2.Relational databases use less storage space, because they don’t store all of the relationships.

3.It is easy to access data as there is no specific path and modification of data also easy

**Demerits of graph database:** 1.In this each record has to be examined individually during a query to determine the structure of the data.

2.In this there is no Standard query language(SQL)

3.Graph databases are not as useful for operational use cases because they are not efficient at processing high volumes of transactions

**Merits of graph database:** 1.Graph database is more flexible than Relational database. It exhibits higher performance for complex deep analytics and it is also more flexible because it exibits high performances for complex transactions .

2.In graph database we can represent all the relationship between entities present and performance is high independent of data’s size.

3.Graph databases solve all problems that are both impractical and practical for relational queries

**Demerits of Relational data base:**1.Relational database has poor performance for deep analytics and it is less flexible compared to graph data base

2.Difficult to represent complex data types and also difficult to represent the heirarchies of data

3. expensive of setting up and maintaining database is costly and recovery of lost data is difficult.

A.2)

NO,the graph databases cannot replace relational databases technologies

Even the graph data bases with this there is no Standard query language(SQL)

Graph databases are not as useful for operational use cases because they are not efficient at processing high volumes of transactions

their consistency performance,agile nature, flexibility ,scalability and with their speed also with their relationshiop representation Some graph data bases can replaces only some relational data bases.

JUSTIFICATION:

BUT THERE ARE MANY REASONS WHY CANNOT WE REPLACE RELATIONAL WITH GRAPH

* As Traditional database has a SQL language to implement tables The choice purely stands on the requirements of data structure, query handling and scalability for an application also having a lot more features and it can in no way replace the traditional relational databases
* If you have constant, unchanging types of data that you are collecting, then graph may not be the most appropriate solution.Then relational data base cannot be replaced in this cases.
* Almost every technologies in the present world have data in the database is structural and transactional types are present then the graph are not all recommeñññxnded and relational database is best for it
* And also relational database uses less space compared to graph database since it shows all relations which are not usable many times. so relational data base cannot be replaced.

Conclusion:

Even though There are many advantages of using Graph data base But in practical leaving some cases like big data,low performance and flexibility compared to graph data base the relational data base cannot be replaced by graph in many cases like queries important and structured database ,containing transactions,space complexity .So graph database cannot replace relational database until it gets a standard query language and overcoming all these limitations.

**PART-B**

**B1**

**B1.1 List of functional and data requirements**

Table 2.1.1: Functional requirement 1

|  |  |
| --- | --- |
| Requirement Tag | FR1 |
| Requirement Description | The user should be able to register/login using is phone number |
| Dependent on Requirements | FR2,FR3 |
| User/System interacting with the requirement | User |

Table 2.1.2: Data requirement 1

|  |  |
| --- | --- |
| Requirement Tag | DR1 |
| Item Name | MOBILE NUMBER |
| Item Description (Where/How used) | The user enters mobile number |
| Item type | Long positive integer |
| User/System interacting with the item | User |

Table 2.1.3:Functional requirement 2

|  |  |
| --- | --- |
| Requirement Tag | FR2 |
| Requirement Description | The system should be able to display product when user searched by product name |
| Dependent on Requirements |  |
| User/System interacting with the requirement | System |

Table 2.1.4 : Data requirement 2

|  |  |
| --- | --- |
| Requirement Tag | DR2 |
| Item Name | Product Name |
| Item Description (Where/How used) | The user searches product name in system |
| Item type | String |
| User/System interacting with the item | User |

Table 2.1.5 Functional requirement 3

|  |  |
| --- | --- |
| Requirement Tag | FR3 |
| Requirement Description | The user has to provide is bank account details for online payment |
| Dependent on Requirements |  |
| User/System interacting with the requirement | User |

Table 2.1.6 : Data requirement 3

|  |  |
| --- | --- |
| Requirement Tag | DR3 |
| Item Name | Bank Account Number |
| Item Description (Where/How used) | TO PAY ONLINE ACCOUNT DETAILS USED |
| Item type | Long integer |
| User/System interacting with the item | System |

Table 2.1.7 : Functional requirement 4

|  |  |
| --- | --- |
| Requirement Tag | FR4 |
| Requirement Description | The user Can add/remove any no.of items to cart |
| Dependent on Requirements | FR2 |
| User/System interacting with the requirement | User |

Table 2.16 Data requirement 4

|  |  |
| --- | --- |
| Requirement Tag | DR4 |
| Item Name | NUMBER OF ITEMS |
| Item Description (Where/How used) | The user can know what he is ordering |
| Item type | Integer |
| User/System interacting with the item | User |

Table 2.1.9 : Functional requirement 5

|  |  |
| --- | --- |
| Requirement Tag | FR5 |
| Requirement Description | The system should able to provide an unique order number when order is placed by user |
| Dependent on Requirements | - |
| User/System interacting with the requirement | Sysem |

Table 2.2.0 : Data requirement 5

|  |  |
| --- | --- |
| Requirement Tag | DR 5 |
| Item Name | Order number |
| Item Description (Where/How used) | By entering order Number we can trace order status |
| Item type | Integer |
| User/System interacting with the item | User |

Table 2.2.1: Functional requirement 6

|  |  |
| --- | --- |
| Requirement Tag | FR6 |
| Requirement Description | System should display Total amount to pay including tax before ordering |
| Dependent on Requirements | FR5 |
| User/System interacting with the requirement | System |

Table 2.2.2 : Data requirement 6

|  |  |
| --- | --- |
| Requirement Tag | DR 6 |
| Item Name | Amount |
| Item Description (Where/How used) | Amount to be paid when ordering |
| Item type | MONEY |
| User/System interacting with the item | System |

Table 2.2.7: Functional requirement 7

|  |  |
| --- | --- |
| Requirement Tag | FR7 |
| Requirement Description | Customer should provide address to where deliver the items |
| Dependent on Requirements |  |
| User/System interacting with the requirement | User |

b Table 2.2.4 : Data requirement 7

|  |  |
| --- | --- |
| Requirement Tag | DR 7 |
| Item Name | Address |
| Item Description (Where/How used) | To deliver the order |
| Item type | String |
| User/System interacting with the item | User |

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

**ENTITIES:**

Here based on the functional requirements I have identified some of the entities to be used

They are

1)CUSTOMER: person who deals with the system and who select the products and buy the products and interact with entire online Furniture system

2)ONLINE FURNITURE SHOPPING:IT IS APP OR WEBSITE OR SYSTEM WHERE ENTIRE PROCESS RUNS

And where CUSTOMER can buy the products and where total information of his order present

3)PRODUCT:ITEMS or thing which user selects to buy

4)STOCK: the availabity of the product checked here and updated regularly

5)ORDER:Here what user wants to buy,and total price to pay can bee seen

HERE USER,ONLINE FURNITURE SHOPPING,PRODUCT,ORDER are strong entities

Because they have unque key attribute and also they doesn’t depend on any entity

BUT STOCK is entirely depend on PRODUCT and also it does not have any unique key attribute

**RELATIONSHIPS:- To connect the entities and to describe the process I have selected some important relationships**

**1)**Registers-It is between CUSTOMER & ONLINE furniture SHOPPING .It is binary because between two entities and

It is N:1 BECAUSE N users can registers to 1 online Furniture shopping and it is partial on user side

BUT full on Online furniture shopping (OFS) because customer choice register to it

2)Logins-It is between USER & ONLINE furniture SHOPPING .It is binary because between two entities and It is N:1 BECAUSE N users can registers to 1 online Furniture shopping and it is partial on user side

BUT full on Online furniture shopping (OFS) because user choice logins to it after registers also.

3)IT IS BINARY and between OFS and PRODUCT and it is m:n means m OFS categorises n products

And OFS Is partial because there may be categories or may not be categories

And full participation of Product means each product is categorised

4)Checks:It is between product and Stock and binary and it is 1:N because N PRODCTS CHEKED IN I STOCK and both are full participation

5)Add to:IT is binary and between PRODUCT and ORDER AND IT IS 1:M N products can be added to 1 order And product is full but order is half because added can also be removed

5)Delivers:-It is also binary and between order and CUSTOMER and 1:1 order delivers to 1 user and both are fully participated because order has to deliver and it is to user only.

**ATTRIBUTES:-**

**Each entity and relationship may has its own** attributes.Here I am taking only attributes of that very much needed for PROBLEM and for that entity they are

Primary key attributes are unique

1)USER:- Has to have name(product buyer),address(To deliver product),gmail(for order success and cancellation msgs)

Gmail:-Primary key for user and Phone number is multi valued attribute because he can have more than one phone number

2)Registers:- Phone No.(for proof and security) it is a primary key

3)Logins:-It has loginid(Primarykey) and password

4)OFS:-Web address(Primary key).Example:-amazon.in it is also unique

5:-)Categorise:-It has CName

6)Product:- It has productid(primary key),product name,price,material

Material is mutivalued because each product can be made of made 1 or more materials

Ex:-Table(wood and iron and plastic,wood,plastic and wood,Iron and wood etc)

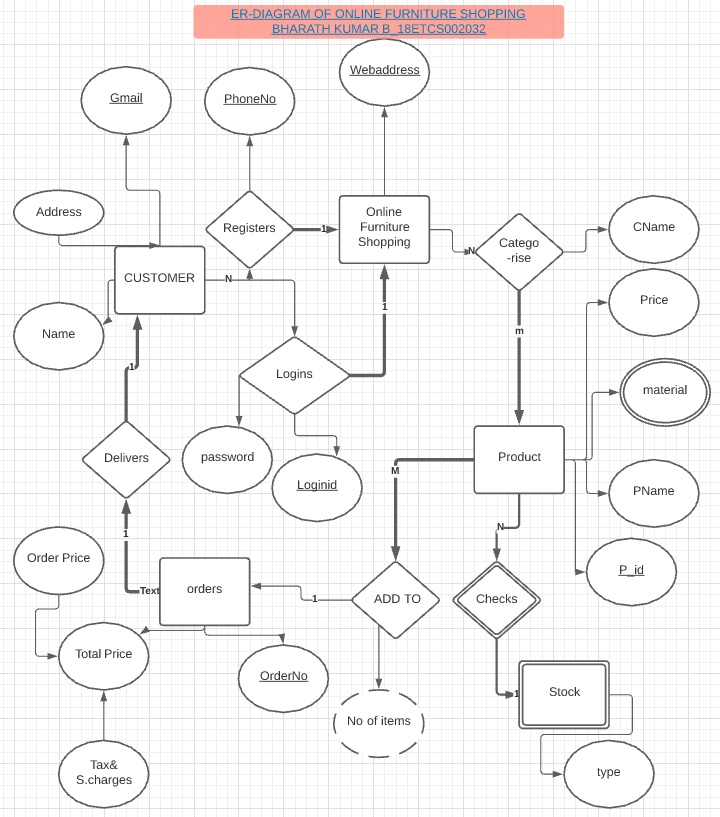
7)STOCK :- ATTRIBUTES stock\_type(partial key attribute) no key so it is week entiti

Ex:-STOCK\_TYPE(not available,old,new)

7)Add to:- It has No.of items it is DERIVED ATTRIBUTE because it depends and derived from products selected and stock so it is derived

8)ORDER:- Order number(primary key),Total price.Here Total price is composite attribute because it composed both OrderPice and Tax&S.charges.

**B1.3 Modelling of ER diagram**

****

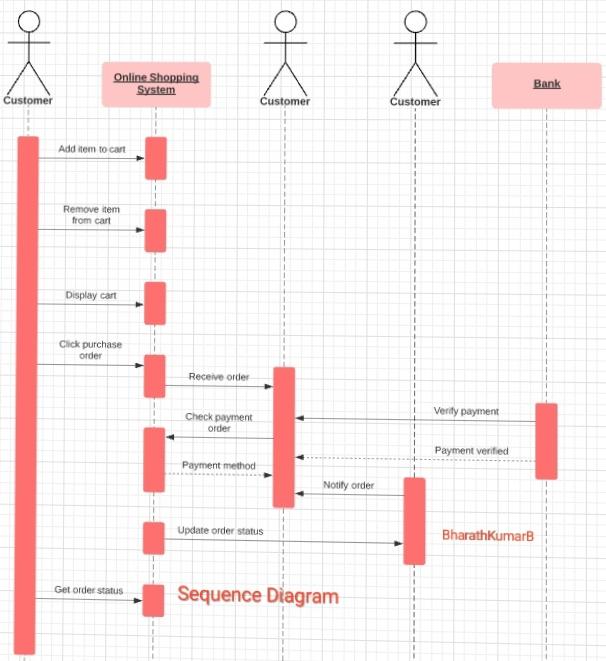
**FIGURE 1.1**

**FIGURE 1.1 shows the ER DIAGRAM OF ONLINE FURNITURE SHOPPING SYSTEM**

**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**

Here the functional requirement 3 and functional requirement 4 and Data requirement 5 are cannot be solved ER diagram

Because here adding of items and removing of items to cart has to be seen which is not possible and payment by online banking and verification of payment done or not cannot be shown .So best way to fulfill the requirements is by **SEQUENCE DIAGRAM**

****

**FIGURE 1.2**

**FIGURE 1.2 shows the SEQUENCE DIAGRAM OF ONLINE FURNITURE SHOPPING SYSTEM**

B2)

**B2.1)) Design of database schema**

**There are many steps to convert ER-DIAGRAM to Relational schema**

**Step1: MAPPING OF REGULAR ENTITIES**

**CUSTOMER**

|  |  |  |
| --- | --- | --- |
| Gmail | **Name** | **Address** |

**Online Furniture shopping**

|  |
| --- |
| Webaddress |

**PRODUCT**

|  |  |  |
| --- | --- | --- |
| P\_id | **PName** | Price |

**ORDER**

|  |  |  |
| --- | --- | --- |
| OrderNumber | Order Price | **Tax&S.charges** |

**Step2: Mapping of weak entities**

**Adding owner entity primary key as foreign key to weak entity**

**STOCK**

|  |  |
| --- | --- |
| **PP\_id** | Stock\_type |

**Step 3: mapping of 1:1 relation binary**

In this deliver relationship is 1:1 so I take and both are full participation order and customer

So I take gmail as foreign key to order

Order(order number,d\_gmail,order Price,tax& s.charges)

Step4:**mapping of 1:N binary relation**

**In** registers and logins relation nth side is customer.

So including primary key of Online furniture shopping primary key and relation attributes

To customer

Customer(gmail,d,RLwebaddress,login\_id,password,name,address,phone number)

Add to Relation also 1:N type between product and order

Here product on nth side so we add order number to product

**Product(p\_id,price,name,Aordernumber)**

**Step 5:-mapping of m:n relationships**

**Categorise**

**Categorise is m:n so new table should be created**

**Categorise**

|  |  |  |
| --- | --- | --- |
| OF\_Webaddress | **PP\_id** | **CName** |

Step 6:**Multivalued Attributess**

**Material in product is multivalued**

**P\_material**

|  |  |
| --- | --- |
| P\_id | **Material** |

**Step7:-there is no n-array relationship in the er-diagram**

**Relational schema diagram is**

**CUSTOMER**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **gmail** | **R\_L\_webaddress** | **Name** | **L\_id** | **L\_Password** | **R\_phoneNo** | **Address** |

**Online Furniture Shopping(OFS)**

|  |
| --- |
| Webaddress |

**PRODUCT**

|  |  |  |  |
| --- | --- | --- | --- |
| P\_id | **PName** | Price | Add\_OrderNo |

**ORDER**

|  |  |  |  |
| --- | --- | --- | --- |
| d\_gmail | Order Price | **Tax&S.charges** | **OrderNo** |

**STOCK**

|  |  |
| --- | --- |
| **P\_P\_id** | Stock\_type |

**Categorise**

|  |  |  |
| --- | --- | --- |
| **P\_P\_id** | **CName** | **OF\_webaddress** |

**P\_material**

|  |  |
| --- | --- |
| M\_P\_id | **Material** |

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

Schema based constraints(key constraints,domain constraints,not null,referential integrity constraint)

FOR OFS TABLE:-

Domain Constraints:-WEBADDRESSS SHOULD BE OF CHARACTERS

Not null constraints:-web address IT IS IMPORTANT TO WHOLE PROCESS SO CANNOT BE NULL

Key Constraints:-web address is prary key so should be unique

For CUSTOMER TABLE

Domain Constraints:-gmail,Name,loginid,password and address should be of characters,PhoneNo is integer

Not null constraints:-ADDRESS CANNOT BE NULL USED WHERE TO DELIVER THE PhoneNo used for proof so cannot be null,login id ,password used for login so not null constraint applied

Key Constraints:-gmail ,PhoneNo,address,login id should be unique);

Referential integrity constraint:- applied for R\_L\_webaddress so it should be one of that in Webaddress

For PRODUCT TABLE

Domain Constraints:-Product id is integer,Price can be int or float,PName is of characters

Not null constraints:-PRODUCT ,PRICE ,PNAME used for searchingCANNOT BE NULL.

Key Constraints:- product id should be unique since it is primary key

Referential integrity constraint;-AOrderno should be the one in Order tables order no

FOR ORDER

Domain Constraints:-Order no should be int,order price and Tax should be float or int,

Not null constraints:-OrderNo cannot be null,Order price,d\_gmail cannot be null

Key Constraints:-orderno primary key should be unique

Referential integrity constraint;-d\_gmail is a foreign key so it is the one of ail that given

In gmail of customer table

STOCK

Domain Constraints:-TYPE SHOULD BE IF CHARACTERS and P\_P\_id should be int

Not null constraints:-TYPE cannot be null used to check for item present or not and p\_p\_id

Cannot be null

Key Constraints:-p\_p\_id is primary key should be unique

Referential integrity constraint;-applied for p\_p\_id It should be one in p\_id of product

For categorise TABLE

Domain Constraints:-Cname should be characters

Not null constraints:-OF\_WEBADDRESS,CName ,P\_P\_ID CANNOT BE NULL

Key Constraints:-OF\_WEBADDRESS AND P\_P\_id combinedly forms primary key individually can be same by and combinedly should be unique

Referential integrity constraint;-OF\_WEBADDRESS SHOULD BE FROM WEB ADDRESS OF OFS and P\_P\_id from P\_id of product

For p\_material TABLE

Domain Constraints:-material should be characters and m\_p\_id is int

Not null constraints:material cannot be null

Key Constraints:-

Referential integrity constraint;-m&p\_id should be one from p\_id from prodeuct table

**B2.3 Implementation of relational database schema with appropriate attributes, and**

**Constraints using SQL commands**

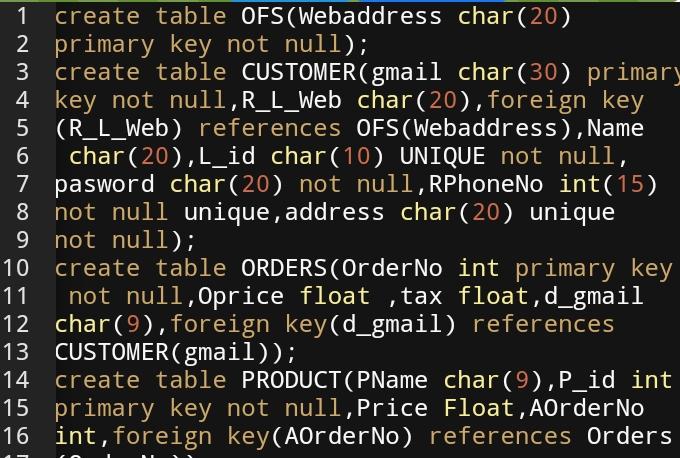
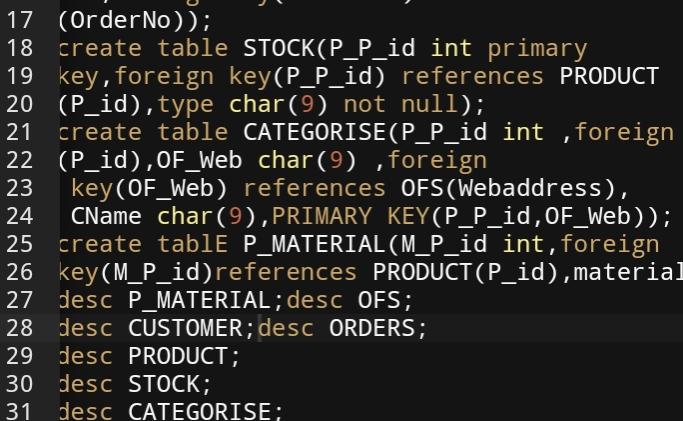
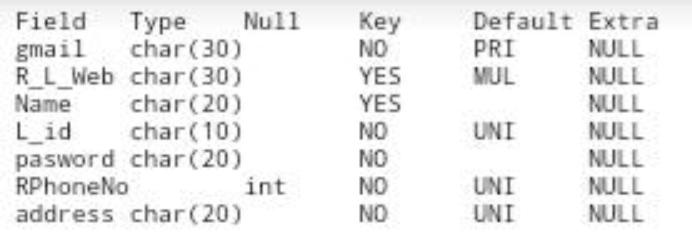


FIGURE **1.3**

**FIGURE 1.3 shows the screenshot of SQL QUERIES FOR OFS RELATIONAL SCHEMA**

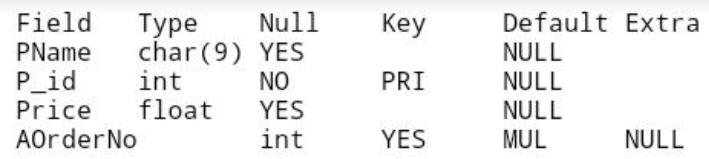
****

**FIGURE 1.4**

**FIGORE 1.4 SHOWS THE STRUCTURE,CONSTRAINTS OF THE TABLE CUSTOMER**

****

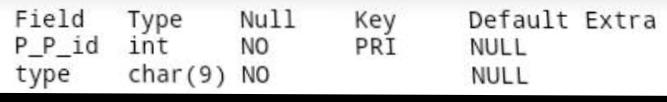
**FIGURE 1.5**

**FIGURE 1.5 SHOWS THE STRUCTURE,CONSTRAINTS OF THE TABLE OFS**

**FIGURE 1.6**

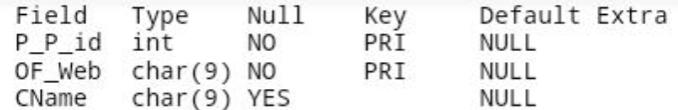
**FIGURE 1.6 SHOWS THE STRUCTURE,CONSTRAINTS OF THE TABLE PRODUCT**

**FIGURE 1.7**

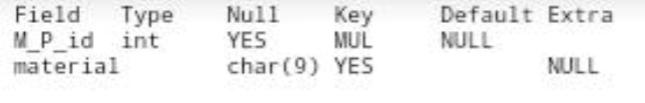
**FIGURE 1.7 SHOWS THE STRUCTURE,CONSTRAINTS OF THE TABLE ORDERS**

**FIGURE 1.8**

**FIGURE 1.8 SHOWS THE STRUCTURE,CONSTRAINTS OF THE TABLE STOCK**



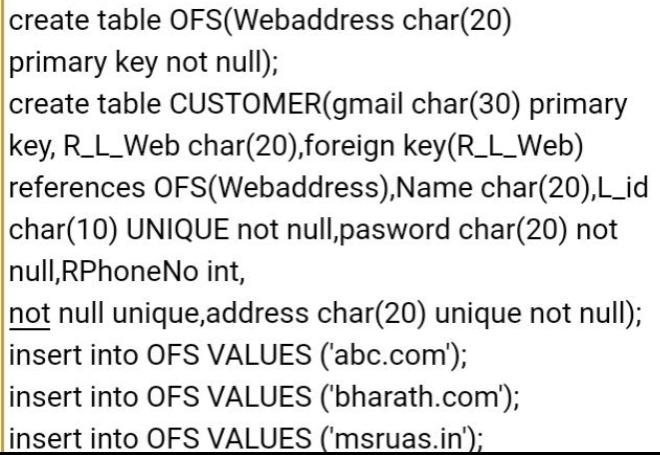
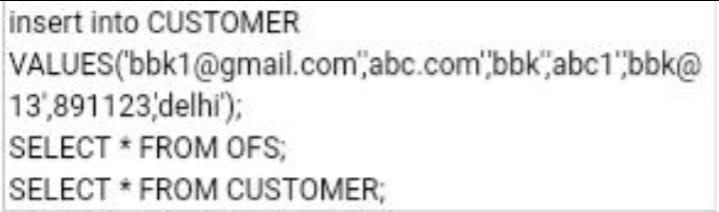
**FIGURE 1.9**

**FIGURE 1.9 SHOWS THE STRUCTURE,CONSTRAINTS OF THE TABLE CATEGORISE**

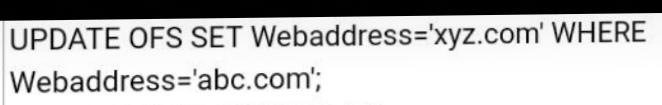
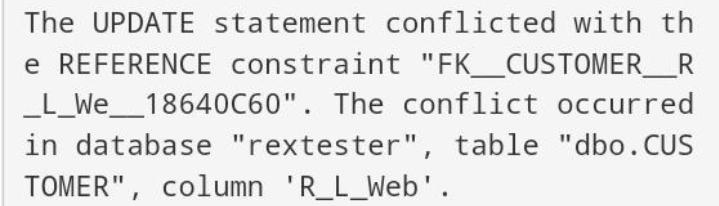
**FIGURE 2.0**

**FIGURE 2.0 SHOWS THE STRUCTURE,CONSTRAINTS OF THE TABLE P\_MATERIAL**

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

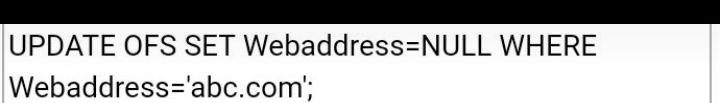
**SQL commands**

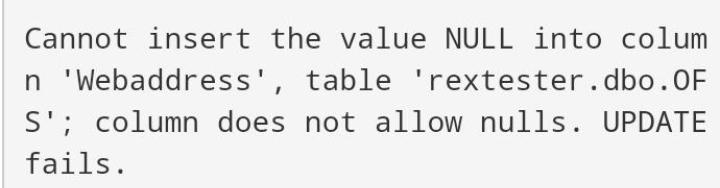
**FIGURE 2.1**

**FIGURE 2.1 SHOWS INSERT ITEMS INTO OFS AND CUSTOMER**

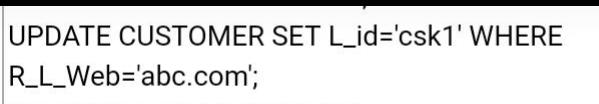
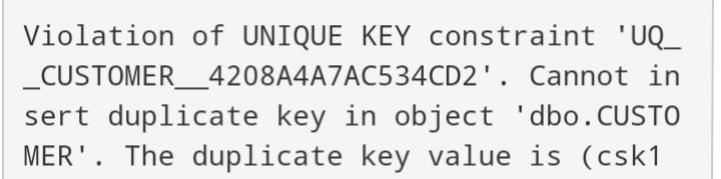
**FIGURE 2.2**

**FIGURE 2.2 SHOWS QUERY AND CONSTRAINT WHEN UPDATE OF FOREIGN KEY(REFERENTIAL INTEGRITY CONSTRAINT)**

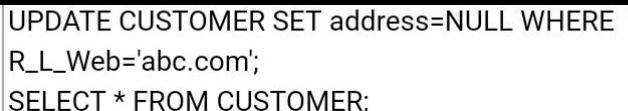
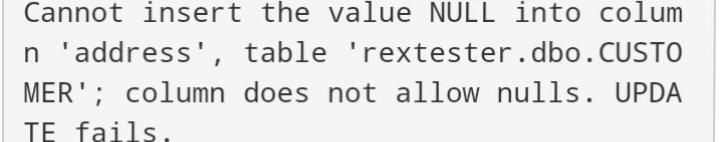
****

****

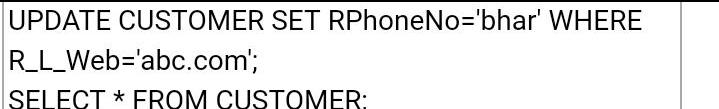
**FIGURE 2.3**

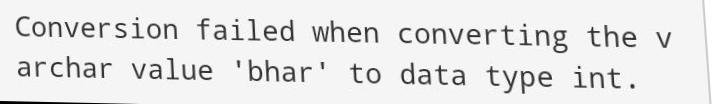
**FIGURE 2.3 SHOWS QUERY AND CONSTRAINT WHEN UPDATE OF PRIMARY KEY**

**FIGURE 2.4**

**FIGURE 2.4 SHOWS QUERY AND CONSTRAINT WHEN UPDATE OF UNIQUE KEY**

**FIGURE 2.5**

**FIGURE 2.5 SHOWS QUERY AND CONSTRAINT WHEN UPDATE OF NOT NULL CONSTRAINT**

****

**FIGURE 2.6**

**FIGURE 2.6 SHOWS QUERY AND CONSTRAINT WHEN UPDATE OF DOMAIN CONSTRAINT**