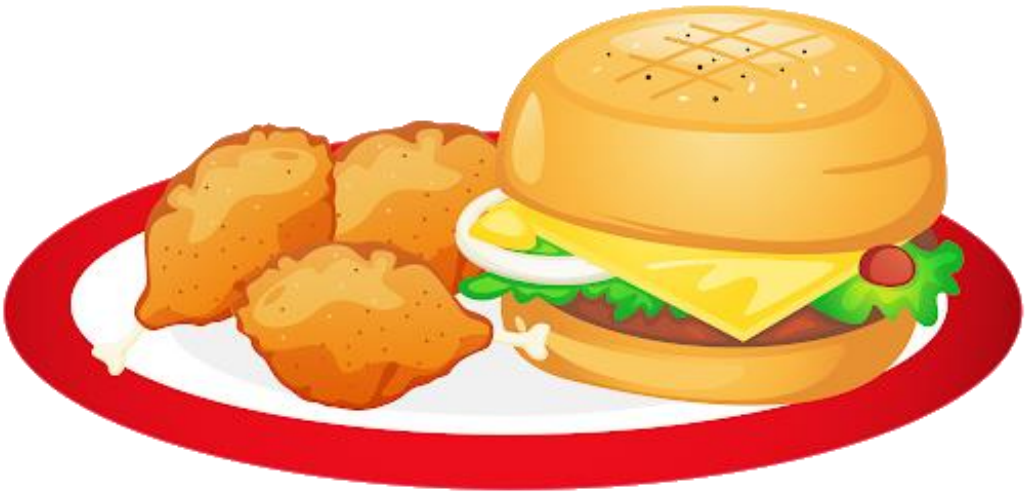
REPORT
DATABASE SYSTEMS
Second Semester (2020-21)

Food - Ordering Application



Group Members:-

2019A7PS0233U - Abshar Mohammed Aslam
2019A7PS0124U - Tannishtha Roy
2019A7PS0212U - Inas Riyaz Mohammed
2019A7PS0136U - Madhumita Bera
2019A7PS0215U - Ruman Althaf Ahamed

ACKNOWLEDGEMENT

We would like to sincerely thank our instructors Dr. Tamizharasan Periswamy and Dr. Sapna Sadhwani for their invaluable guidance without which this project would not have been completed.

TABLE OF CONTENTS:-

INTRODUCTION:-

Through our project, we've tried to implement an Online-Food Delivery Database System.

- **Through this system, a User is granted the flexibility to order food from the Restaurant of their choice.**
- **A Delivery Man will be assigned the responsibility to deliver the food to said User (Customer).**
- **The User can add items to his/her Cart and may also utilise the offers available to them that have been provided by the restaurant.**
- **Users may also store his/her Credit Card information on the database for faster transactions.**
- **Restaurants may also add/update/delete items from their Menus.**
- **Multiple Restaurants from the same franchise can exist on the database (KFC, McDonalds, etc.)**

Food ordering online is becoming a norm for restaurants which offer takeout and delivery orders. It is designed as it is cost effective yet an efficient system to satisfy the restaurant's and the user's needs. The system is also designed for its ultimate flexibility and performance. The customers will be able to access the company existing website and browse at their menu and select and place their orders on what they desire.

The online ordering system also enables customers to order days beforehand and the system will execute the order at the specified time. The system has been built to handle large amounts of traffic simultaneously to prevent a system overload.

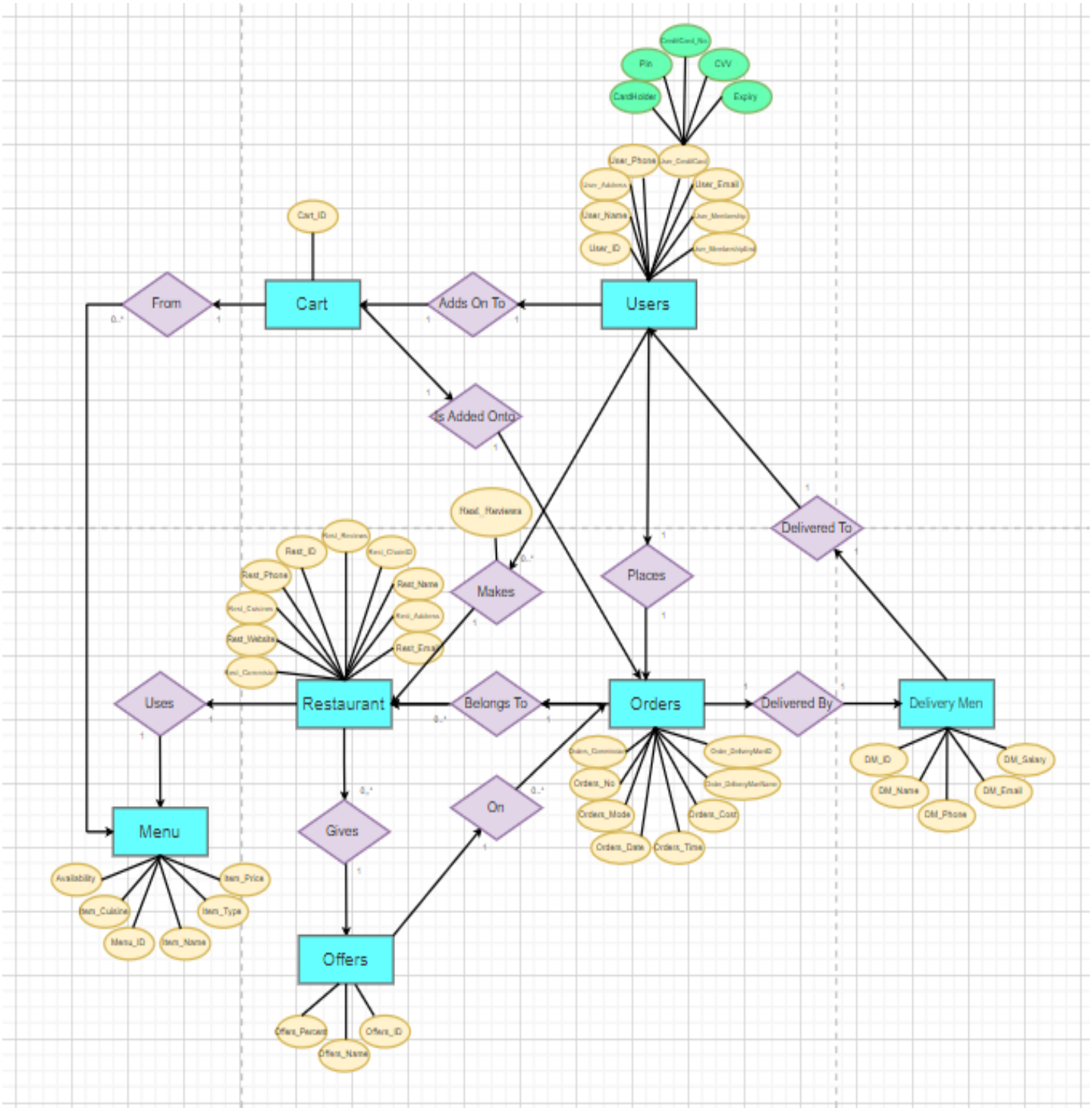
This system is also flexible in a way, whereby customers are able to place online orders quickly, with just a 'click' compared to the tradition where orders were taken through the phone. More time and cost will be saved ordering online as a phone bill is charged according to the time the phone is on the line. The system was used intended to meet the requirements of all of the clients.

There will be fewer errors on the orders or miscommunication amongst the customers and the person taking orders. Besides, it provides timely service. The software examines all food orders prior to the completion and corrects human errors.

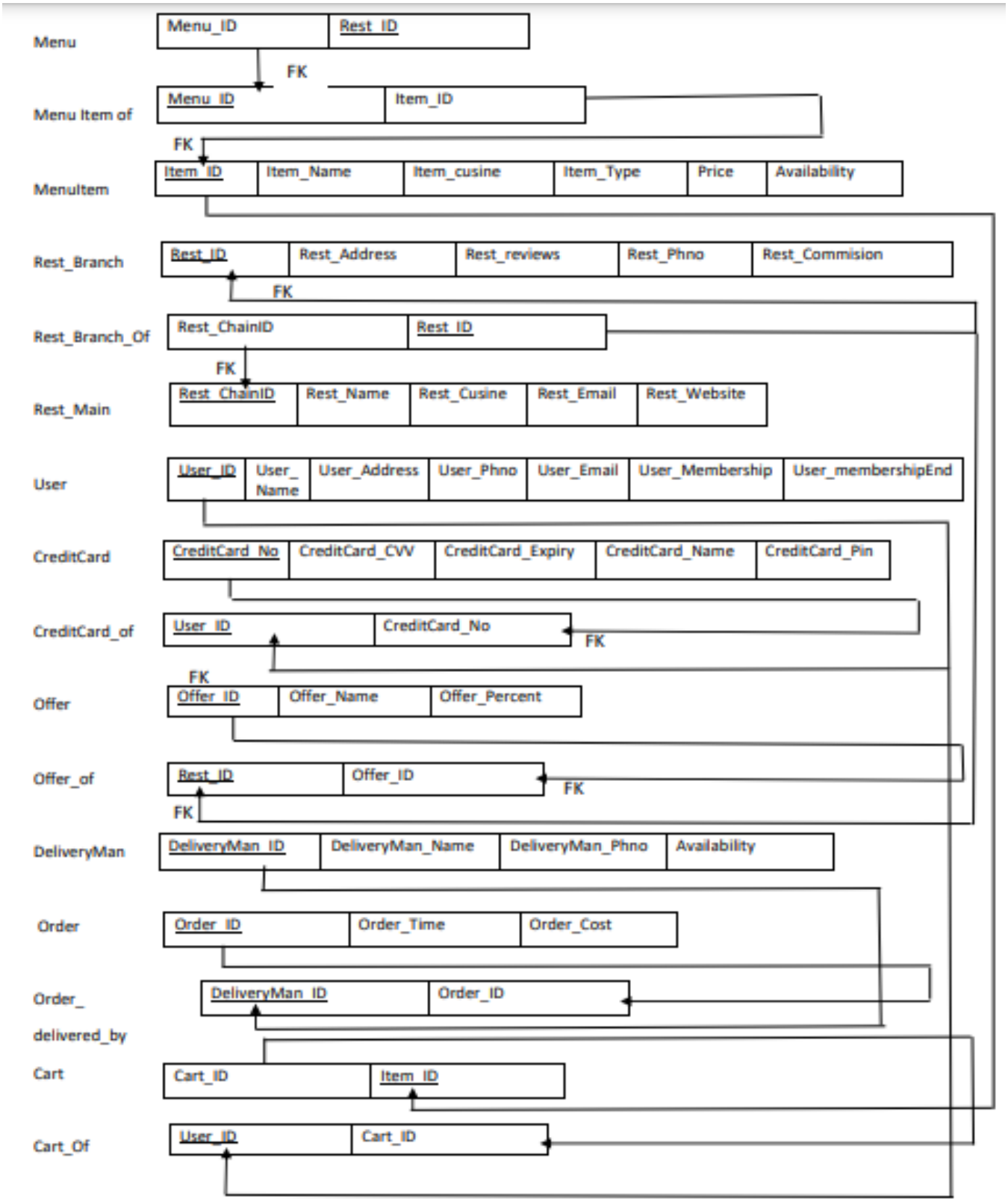
ASSUMPTIONS:-

- We assume that a user can only place orders at one restaurant at a time.
- The first available Delivery Man will be assigned to the order in progress.

ER/EER DIAGRAM (Conceptual Design)



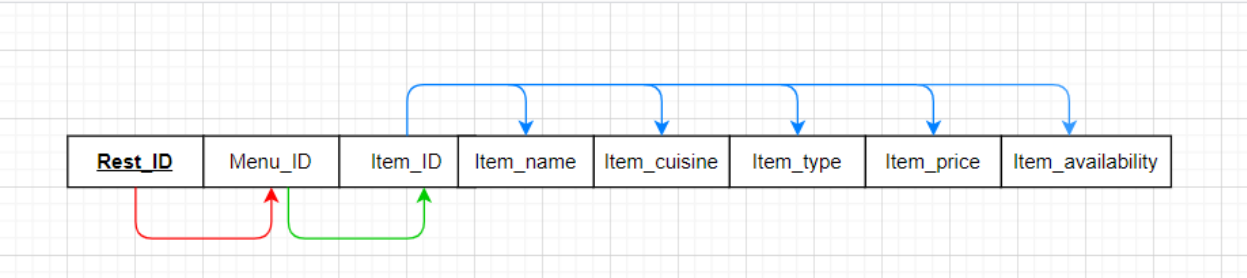
RELATIONAL MODEL



NORMAL FORMS

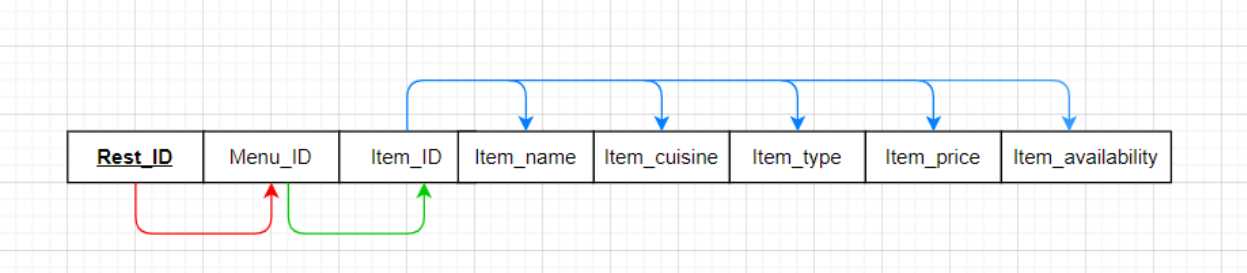
Menu Table:-

- **1NF (First Normal Form)**



No multivalued attributes so it is at least in 1NF.

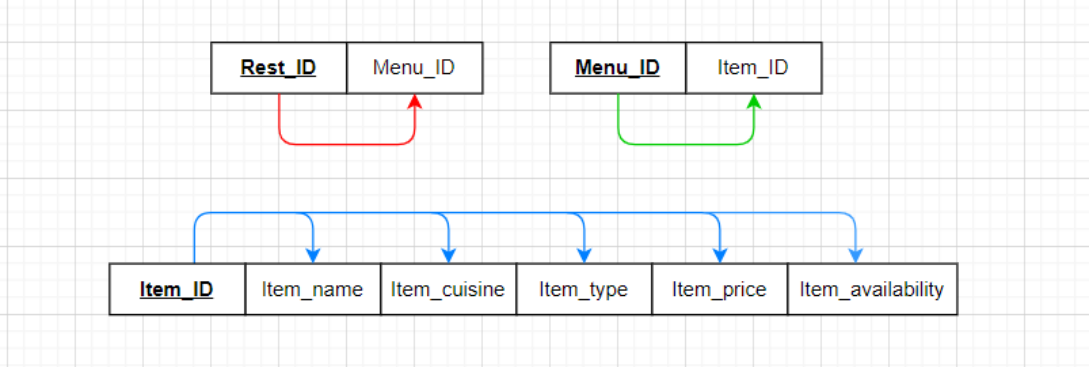
- **2NF (Second Normal Form)**



No partial dependencies so it is 2NF.

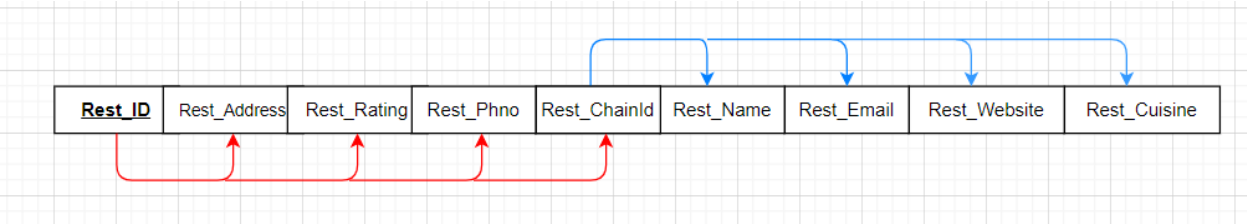
- **3NF (Third Normal Form)**

- Since transitivity exists it is not in 3NF so we split the table as shown:-
- Table split into Menu, Menu_Item_Of & Menu_Item respectively.



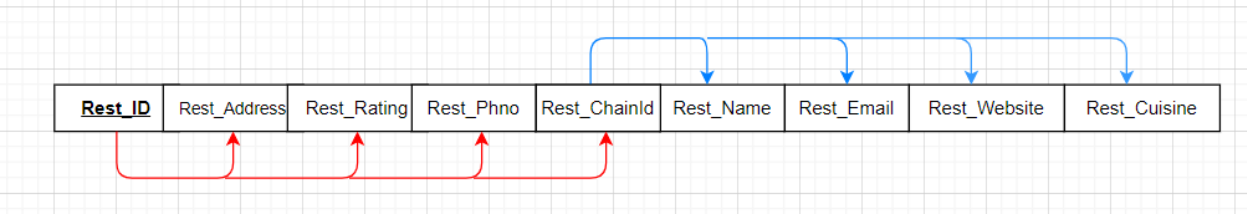
Restaurant Table:-

- **1NF (First Normal Form)**



No multivalued attributes so it is at least 1NF.

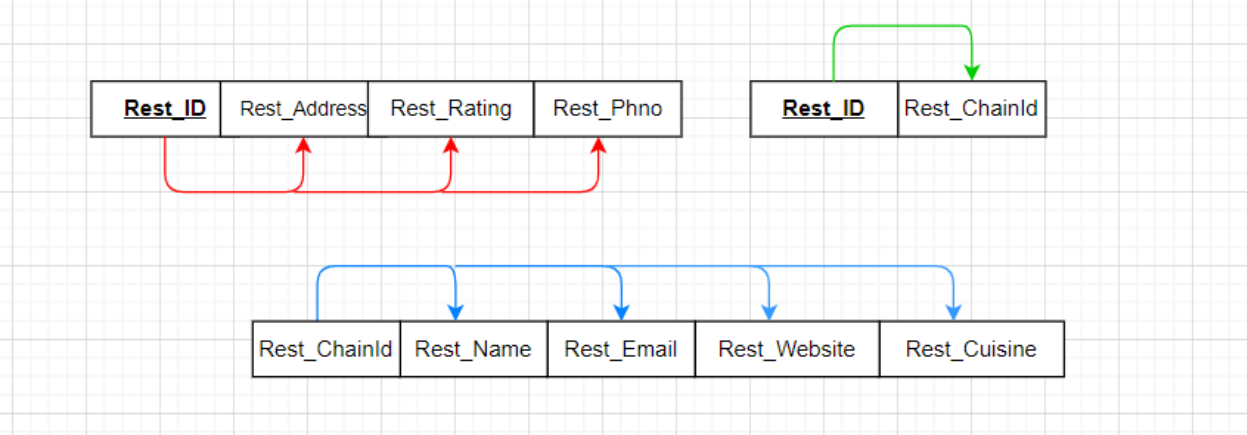
- **2NF (Second Normal Form)**



No partial dependencies so it is 2NF.

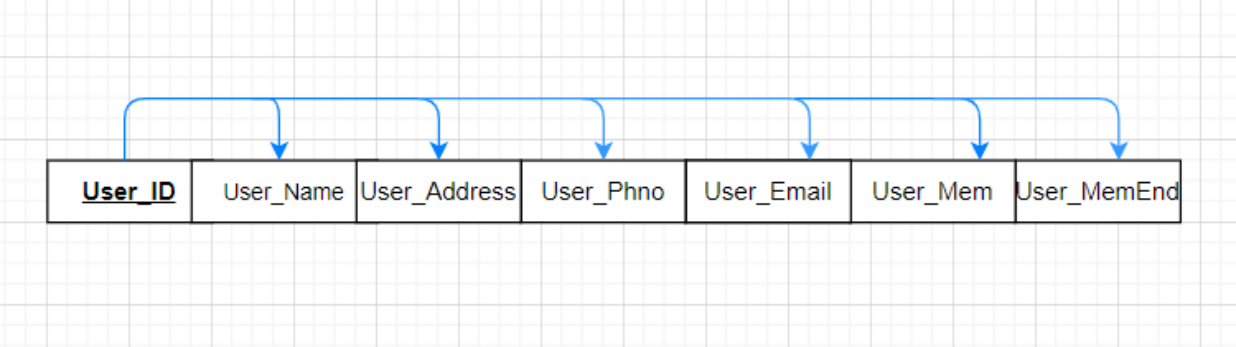
- **3NF (Third Normal Form)**

- Since transitivity exists it is not in 3NF so we split the table as shown:-
- Table split into Rest_Branch, Rest_Main & Rest_Branch_Of respectively.



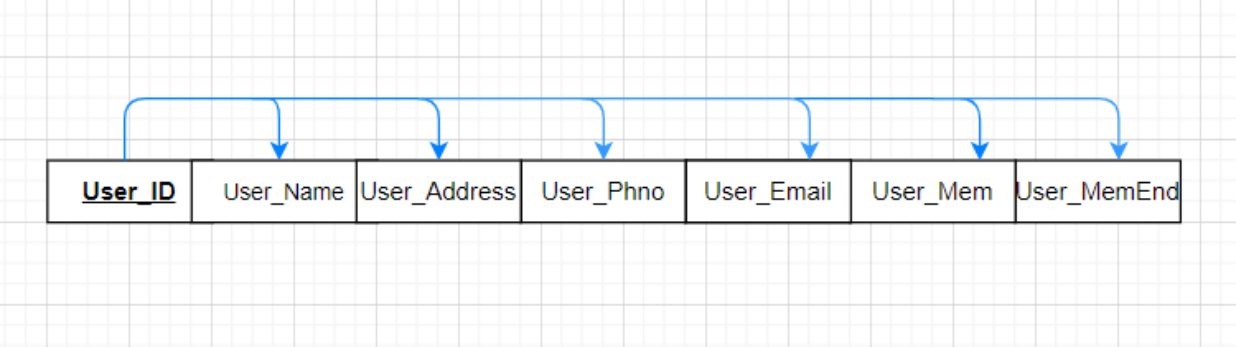
User Table:-

- **1NF (First Normal Form)**



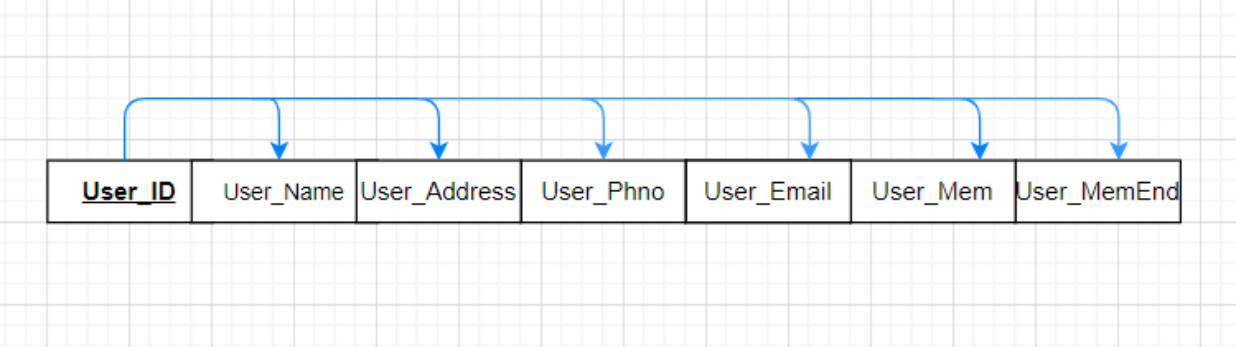
No multivalued attributes so it is at least 1NF.

- **2NF (Second Normal Form)**



No partial dependencies so it is 2NF.

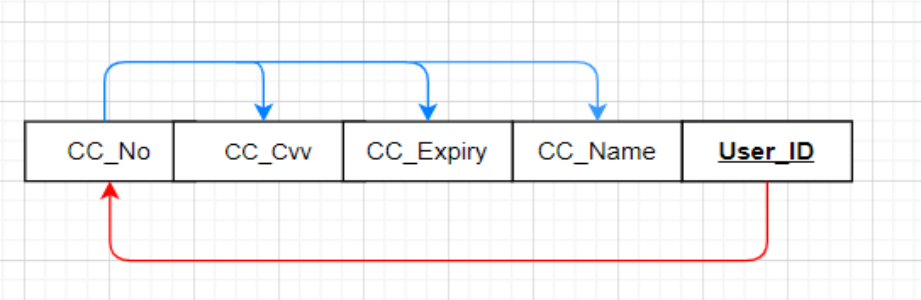
- **3NF (Third Normal Form)**



No transitivity therefore it is 3NF.

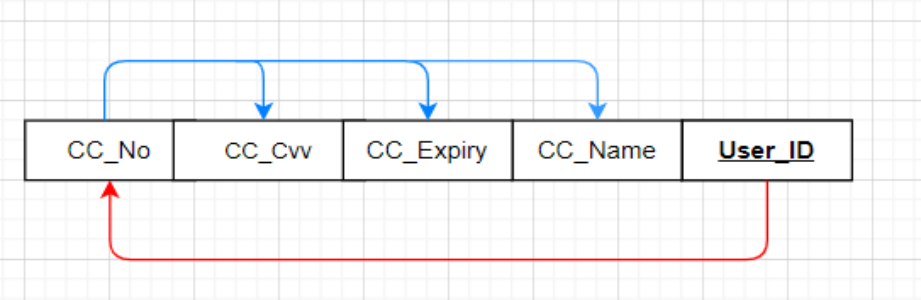
Credit Card Table:-

- **1NF (First Normal Form)**



No multivalued attributes so it is at least 1NF.

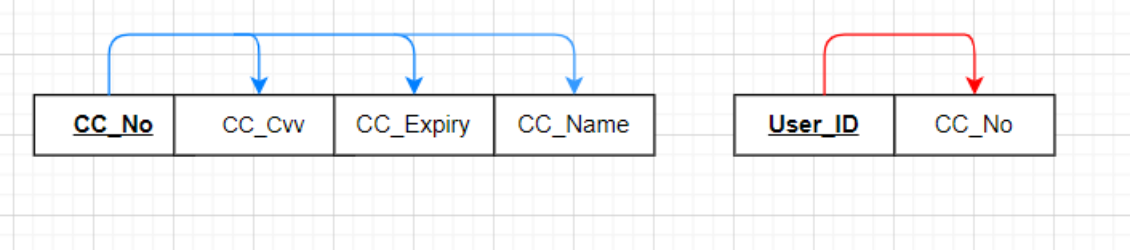
- **2NF (Second Normal Form)**



No partial dependencies so it is 2NF.

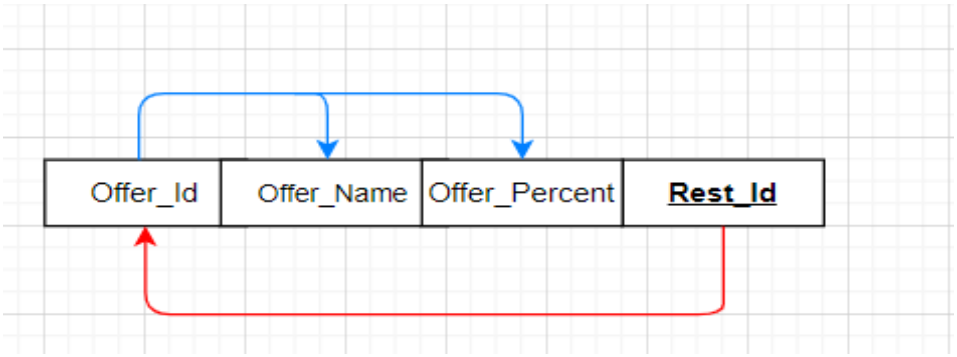
- **3NF (Third Normal Form)**

- Since transitivity exists it is not in 3NF so we split the table as shown:-
- Table split into CreditCard & Credit_Card_Of respectively.



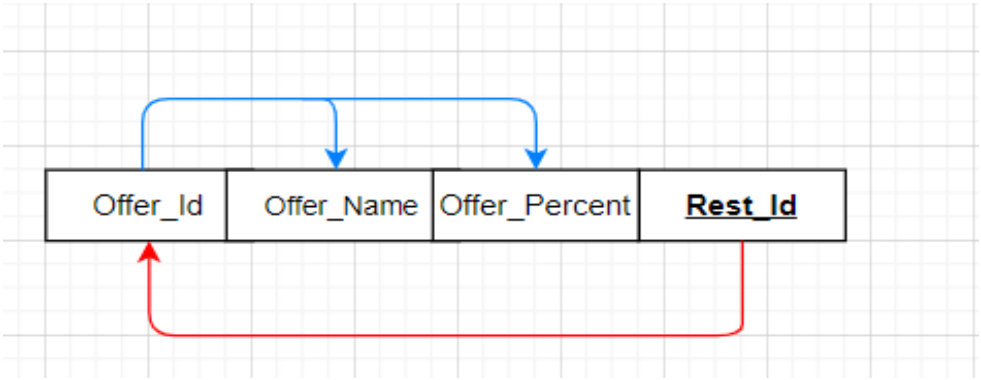
Offers Table:-

- **1NF (First Normal Form)**



No multivalued attributes so it is at least 1NF.

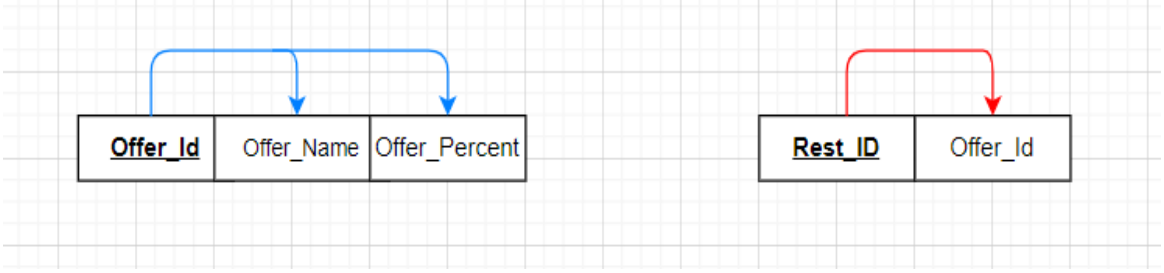
- **2NF (Second Normal Form)**



No partial dependencies so it is 2NF.

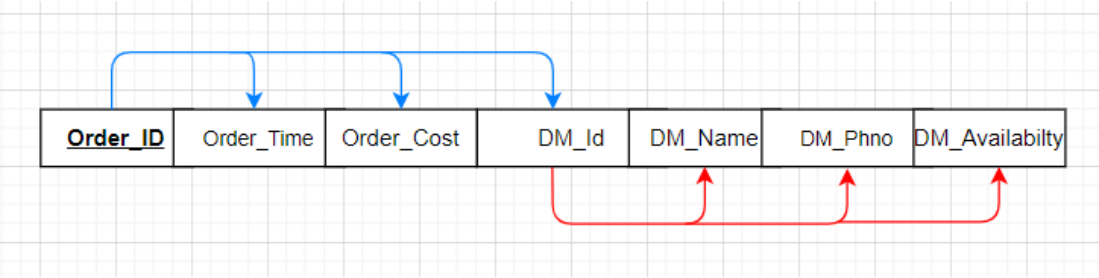
- **3NF (Third Normal Form)**

- Since transitivity exists it is not in 3NF so we split the table as shown:-
- Table split into Offer & Offer_Of respectively.



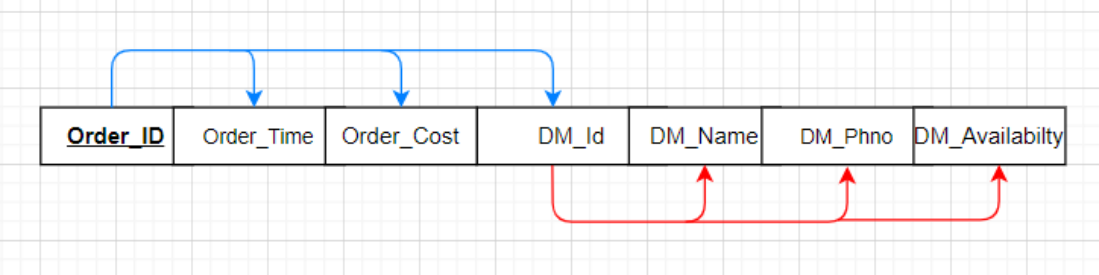
Orders Table:-

1NF (First Normal Form)



No multivalued attributes so it is at least 1NF.

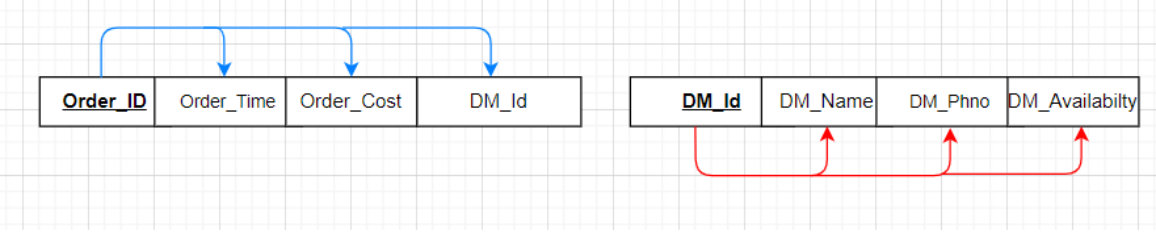
2NF (Second Normal Form)



No partial dependencies so it is 2NF.

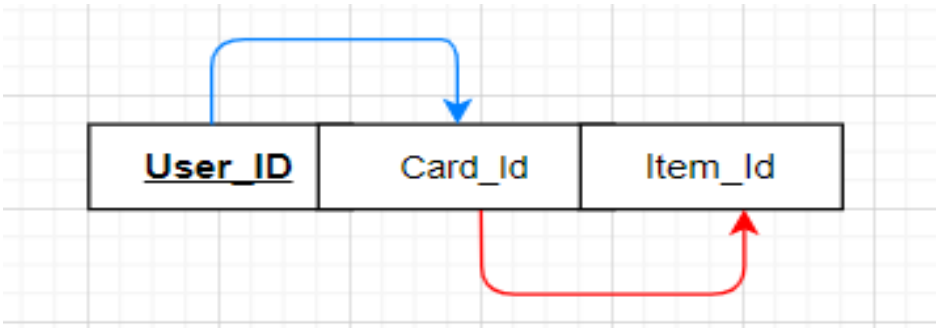
3NF (Third Normal Form)

- Since transitivity exists it is not in 3NF so we split the table as shown:-
- Table split into Orders & Order_Delivered_By respectively.



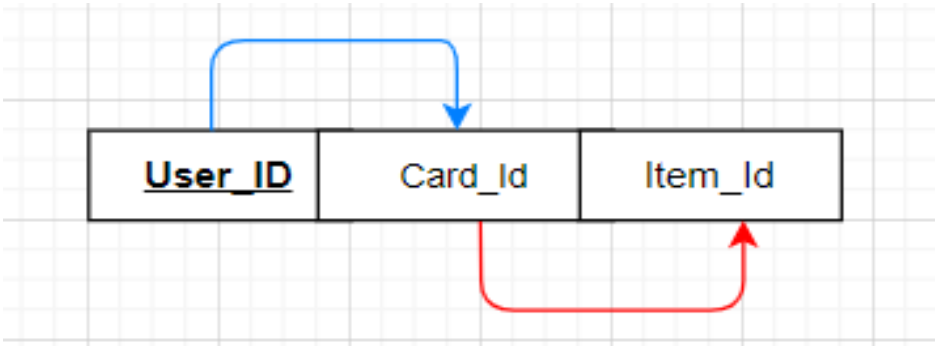
Cart Table:-

- **1NF (First Normal Form)**



No multivalued attributes so it is at least 1NF.

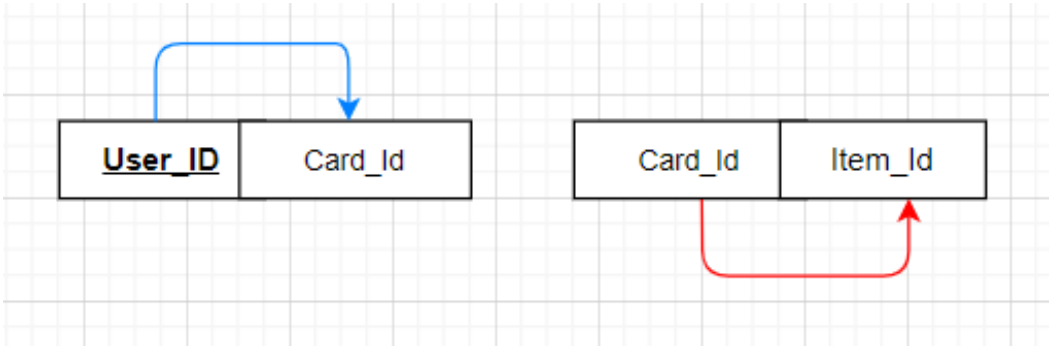
- **2NF (Second Normal Form)**



No partial dependencies so it is 2NF.

- **3NF (Third Normal Form)**

- Since transitivity exists it is not in 3NF so we split the table as shown:-
- Table split into Cart_Of & Cart respectively.



SQL TABLES

Restaurant Table:-

- CREATE TABLE REST_BRANCH_OF(
  REST_ID VARCHAR(5) UNIQUE NOT NULL,
  REST_CHAINID VARCHAR(6) UNIQUE NOT NULL,
  PRIMARY KEY(REST_ID));

  INSERT INTO REST_BRANCH_OF VALUES("RI001","RCI001");
  INSERT INTO REST_BRANCH_OF VALUES("RI002","RCI002");
  INSERT INTO REST_BRANCH_OF VALUES("RI003","RCI003");
  INSERT INTO REST_BRANCH_OF VALUES("RI004","RCI004");
  INSERT INTO REST_BRANCH_OF VALUES("RI005","RCI005");

Rest_Branch_Of	
Rest_Id	*Rest_ChainId*
RI001	RCI001
RI002	RCI002
RI003	RCI003
RI004	RCI004
RI005	RCI005

- CREATE TABLE REST_BRANCH (
  REST_ID VARCHAR(6) NOT NULL,
  REST_ADDRESS VARCHAR(20),
  REST_RATING DECIMAL (2, 1),
  REST_PHNO INT,
  PRIMARY KEY(REST_ID),
  FOREIGN KEY (REST_ID) REFERENCES REST_BRANCH_OF(REST_ID));

  INSERT INTO REST_BRANCH VALUES("RI001","Ajman",4.2,666000);
  INSERT INTO REST_BRANCH VALUES("RI002","Dubai",4.8,777888);
  INSERT INTO REST_BRANCH VALUES("RI003","Sharjah",3.9,999222);
  INSERT INTO REST_BRANCH VALUES("RI004","Fujairah",4.5,444111);
  INSERT INTO REST_BRANCH VALUES("RI005","Dubai",4.3,222333);

Rest_Branch			
Rest_Id	Rest_Address	Rest_Rating	Rest_Phno
RI001	Ajman	4.2	666000
RI002	Dubai	4.8	777888
RI003	Sharjah	3.9	999222
RI004	Fujairah	4.5	444111
RI005	Dubai	4.3	222333

- CREATE TABLE REST_MAIN(
  REST_CHAINID VARCHAR(6) NOT NULL,
  REST_NAME VARCHAR(20) NOT NULL,
  REST_CUSINE VARCHAR(20) NOT NULL,
  REST_EMAIL VARCHAR(20) NOT NULL,
  REST_WEBSITE VARCHAR(20) NOT NULL,
  FOREIGN KEY (REST_CHAINID) REFERENCES REST_BRANCH_OF(REST_CHAINID));

  INSERT INTO REST_MAIN VALUES("RCI001"," China Town"," Chinese"," CT@gmail.com"," ChinaTown.com");
  INSERT INTO REST_MAIN VALUES("RCI002"," India Palace"," Indian"," IP@gmail.com" ,"IndianPalace.com");
  INSERT INTO REST_MAIN VALUES("RCI003"," Al Hallab"," Arabic"," AH@gmail.com"," AlHallab.com");
  INSERT INTO REST_MAIN VALUES("RCI004"," Mexi House"," Mexican"," MH@gmail.com"," MexiHouse.com");
  INSERT INTO REST_MAIN VALUES("RCI005"," SukhoThai"," Thai"," ST@gmail.com"," SukhoThai.com");

Rest_Main				
Rest_ChainId	Rest_Name	Rest_Cuisine	Rest_Email	Rest_Website
RCI001	China Town	Chinese	CT@gmail.com	ChinaTown.com
RCI002	India Palace	Indian	IP@gmail.com	IndianPalace.com
RCI003	Al Hallab	Arabic	AH@gmail.com	AlHallab.com

RCI004	Mexi House	Mexican	MH@gmail.com	MexiHouse.com
RCI005	SukhoThai	Thai	ST@gmail.com	SukhoThai.com

Orders Table:-

- CREATE TABLE ORDERS(
  ORDER_ID VARCHAR(6),
  ORDER_TIME DATETIME,
  ORDER_COST DOUBLE,
  USER_ID VARCHAR(6) NOT NULL,
  PRIMARY KEY(ORDER_ID));

  INSERT INTO ORDERS VALUES("OI001", "2021-03-02 8:00:00", 100, "UID001");
  INSERT INTO ORDERS VALUES("OI002", "2021-03-17 9:45:00", 50, "UID006");

Orders			
Order_Id	Order_Time	Order_Cost	USER_ID
OI001	2021-03-02 8:00:00	100	UID001
OI002	2021-03-17 9:45:00	50	UID006

- CREATE TABLE ORDER_DELIVERED_BY (
  ORDER_ID VARCHAR(6),
  DM_ID VARCHAR(6),
  PRIMARY KEY(ORDER_ID),
  FOREIGN KEY (ORDER_ID) REFERENCES ORDERS(ORDER_ID),
  FOREIGN KEY (DM_ID) REFERENCES DELIVERYMAN(DM_ID));

  INSERT INTO ORDER_DELIVERED_BY VALUES("OI001", "DMI001");
  INSERT INTO ORDER_DELIVERED_BY VALUES("OI002", "DMI003");

Order_Delivered_By	
Order_Id	*DM_Id*
OI001	DMI001
OI002	DMI003

Menu Table:-

- CREATE TABLE MENU(
  REST_ID CHAR(5) NOT NULL,
  MENU_ID CHAR(5) UNIQUE NOT NULL,
  PRIMARY KEY(REST_ID));

  INSERT INTO MENU VALUES("RI001", "MI001");
  INSERT INTO MENU VALUES("RI002", "MI002");
  INSERT INTO MENU VALUES("RI003", "MI003");
  INSERT INTO MENU VALUES("RI004", "MI004");
  INSERT INTO MENU VALUES("RI005", "MI005");

Menu	
Rest_Id	Menu_Id
RI001	MI001
RI002	MI002
RI003	MI003
RI004	MI004
RI005	MI005

  - CREATE TABLE MENU_ITEM_OF(
    MENU_ID CHAR(5) NOT NULL,
    ITEM_ID CHAR(5) NOT NULL);

    INSERT INTO MENU_ITEM_OF VALUES("MI001", "II101");
    INSERT INTO MENU_ITEM_OF VALUES("MI001", "II102");
    INSERT INTO MENU_ITEM_OF VALUES("MI001", "II103");
    INSERT INTO MENU_ITEM_OF VALUES("MI002", "II201");
    INSERT INTO MENU_ITEM_OF VALUES("MI002", "II202");
    INSERT INTO MENU_ITEM_OF VALUES("MI002", "II203");
    INSERT INTO MENU_ITEM_OF VALUES("MI003", "II301");
    INSERT INTO MENU_ITEM_OF VALUES("MI003", "II302");
    INSERT INTO MENU_ITEM_OF VALUES("MI003", "II303");
    INSERT INTO MENU_ITEM_OF VALUES("MI004", "II401");
    INSERT INTO MENU_ITEM_OF VALUES("MI004", "II402");

INSERT INTO MENU_ITEM_OF VALUES("MI004", "II403");
INSERT INTO MENU_ITEM_OF VALUES("MI005", "II501");
INSERT INTO MENU_ITEM_OF VALUES("MI005", "II502");
INSERT INTO MENU_ITEM_OF VALUES("MI005", "II503");

Menu_Item_Of	
Menu_Id	Item_Id
MI001	II101
MI001	II102
MI001	II103
MI002	II201
MI002	II202
MI002	II203
MI003	II301
MI003	II302
MI003	II303
MI004	II401
MI004	II402
MI004	II403
MI005	II501
MI005	II502
MI005	II503

- CREATE TABLE MENU_ITEM(
  ITEM_ID CHAR(5) NOT NULL,
  ITEM_NAME VARCHAR(20) NOT NULL,
  ITEM_CUISINE VARCHAR(20),
  ITEM_TYPE VARCHAR(20),
  ITEM_PRICE FLOAT NOT NULL,
  ITEM_AVAILABILITY BOOLEAN NOT NULL,
  PRIMARY KEY(ITEM_ID));

  INSERT INTO MENU_ITEM VALUES("II101", "Noodles", "Chinese", "Main Course", 30, TRUE);
  INSERT INTO MENU_ITEM VALUES("II102", "Fried Rice", "Chinese", "Main Course", 25, TRUE);
  INSERT INTO MENU_ITEM VALUES("II103", "Momos", "Chinese", "Starter", 20, TRUE);

```
INSERT INTO MENU_ITEM VALUES("II201", "Chicken Tikka", "Indian", "Main Course", 35, TRUE);
INSERT INTO MENU_ITEM VALUES("II202", "Butter Chicken", "Indian", "Main Course", 40, FALSE);
INSERT INTO MENU_ITEM VALUES("II203", "Naan", "Indian", "Main Course", 5, TRUE);
INSERT INTO MENU_ITEM VALUES("II301", "Hummus", "Arabic", "Side", 15, TRUE);
INSERT INTO MENU_ITEM VALUES("II302", "Shawerma", "Arabic", "Starter", 5, TRUE);
INSERT INTO MENU_ITEM VALUES("II303", "Shish Tawook", "Arabic", "Main Course", 35, TRUE);
INSERT INTO MENU_ITEM VALUES("II401", "Tacos", "Mexican", "Side", 35, TRUE);
INSERT INTO MENU_ITEM VALUES("II402", "Quesadillas", "Mexican", "Appetizer", 35, TRUE);
INSERT INTO MENU_ITEM VALUES("II403", "Lasagne", "Mexican", "Main Course", 40, TRUE);
INSERT INTO MENU_ITEM VALUES("II501", "Green Curry", "Thai", "Main Course", 45, TRUE);
INSERT INTO MENU_ITEM VALUES("II502", "Sticky Rice", "Thai", "Main Course", 25, FALSE);
INSERT INTO MENU_ITEM VALUES("II503", "Nasi Goreng", "Thai", "Main Course", 40, TRUE);
```

Menu_Item					
Item_Id	Item_Name	Item_Cuisine	Item_Type	Item_Price	Item_Availablity
II101	Noodles	Chinese	Main Course	30	TRUE
II102	Fried Rice	Chinese	Main Course	25	TRUE
II103	Momos	Chinese	Starter	20	TRUE
II201	Chicken tikka	Indian	Main Course	35	TRUE
II202	Butter Chicken	Indian	Main Course	40	FALSE
II203	Naan	Indian	Main Course	5	TRUE
II301	Hummus	Arabic	Side	15	TRUE
II302	Shawerma	Arabic	Starter	5	TRUE
II303	Shish Tawook	Arabic	Main Course	35	TRUE
II401	Tacos	Mexican	Side	35	TRUE
II402	Quesadillas	Mexican	Appetizer	35	TRUE
II403	Lasagne	Mexican	Main Course	40	TRUE

II501	Green Curry	Thai	Main Course	45	TRUE
II502	Sticky Rice	Thai	Main Course	25	FALSE
II503	Nasi Goreng	Thai	Main Course	40	TRUE

User Table:-

- CREATE TABLE USER(
  USER_ID VARCHAR(6) NOT NULL,
  USER_NAME VARCHAR(10) NOT NULL,
  USER_ADDRESS VARCHAR(10),
  USER_PHNO INT,
  USER_EMAIL VARCHAR(20),
  USER_MEM BOOLEAN NOT NULL,
  USER_MEMEND DATE,
  PRIMARY KEY(USER_ID));

  INSERT INTO USER
  VALUES("UID001","RAJ","AJMAN",12345,"RAJ@GMAIL.COM",TRUE,"2023-10-09");
  INSERT INTO USER
  VALUES("UID002","OMAR","DUBAI",98765,"OMAR@HOTMAIL.COM",FALSE,"2022-02-18");
  INSERT INTO USER
  VALUES("UID003","PARTH","FUJAIRAH",45678,"PARTH@OUTLOOK.COM",TRUE,"2023-09-08");
  INSERT INTO USER
  VALUES("UID004","DIVIJA","SHARJAH",23456,"DIVI@GMAIL.COM",FALSE,"2021-10-09");
  INSERT INTO USER VALUES("UID005","RAKSHAN","ABU
  DHABI",67892,"RAKS@GMAIL.COM",TRUE,"2023-09-10");
  INSERT INTO USER
  VALUES("UID006","INAYA","SHARJAH",83456,"INAYA@OUTLOOK.COM",FALSE,"2022-01-04");

User						
User_Id	User_Name	User_Address	User_Phno	User_Email	User_Mem	User_MemEnd
UID001	Raj	Ajman	12345	raj@gmail.com	TRUE	9-10-2023
UID002	Omar	Dubai	98765	omar@hotmail.com	FALSE	18-02-2022
UID003	Parth	Fujairah	45678	parth@outlook.com	TRUE	08-09-2023
UID004	Divija	Sharjah	23456	divi@gmail.com	FALSE	9-10-2021
UID005	Rakshan	Abu Dhabi	67892	raks@gmail.com	TRUE	10-09-2023
UID006	Inaya	Sharjah	83456	inaya@outlook.com	FALSE	04-01-2022

Delivery Man Table:-

- CREATE TABLE DELIVERYMAN(
  DM_ID VARCHAR(6),
  DM_NAME VARCHAR(20),
  DM_PHNO INT,
  DM_AVAILABILITY BOOLEAN,
  PRIMARY KEY(DM_ID));

  INSERT INTO DELIVERYMAN VALUES("DMI001","RAJESH", 60890, TRUE);
  INSERT INTO DELIVERYMAN VALUES("DMI002","MAHESH", 76126, FALSE);
  INSERT INTO DELIVERYMAN VALUES("DMI003","NILESH", 76198, TRUE);
  INSERT INTO DELIVERYMAN VALUES("DMI004","SNEHANSHU", 98123, TRUE);

DeliveryMan			
DM_Id	DM_Name	DM_Phno	DM_Availability
DMI001	Rajesh	60890	TRUE
DMI002	Mahesh	76126	FALSE
DMI003	Nilesh	76198	TRUE
DMI004	Snehanshu	98123	TRUE

Credit Card Table:-

- CREATE TABLE CREDITCARD_OF(
  USER_ID VARCHAR(6) NOT NULL,
  CC_NO CHAR(30) UNIQUE NOT NULL,
  PRIMARY KEY(USER_ID),
  FOREIGN KEY(USER_ID) REFERENCES USER(USER_ID));

  INSERT INTO CREDITCARD_OF VALUES("UID001","44850295807598228");
  INSERT INTO CREDITCARD_OF VALUES("UID003","4024007199137174");
  INSERT INTO CREDITCARD_OF VALUES("UID004","5130874015983067");
  INSERT INTO CREDITCARD_OF VALUES("UID006","5164444244374535");

CreditCard_Of	
User_Id	*CC_No*
UID001	4485029580759228
UID003	4024007199137174
UID004	5130874015983067
UID006	5164444244374535

- CREATE TABLE CREDITCARD (
  CC_NO CHAR(30),
  CC_CVV INT,
  CC_EXPIRY DATE,
  CC_NAME VARCHAR(20),
  PRIMARY KEY(CC_NO),
  FOREIGN KEY (CC_NO) REFERENCES CREDITCARD_OF(CC_NO));

  INSERT INTO CREDITCARD VALUES("44850295807598228", 204, "2021-04-22", "Raj");
  INSERT INTO CREDITCARD VALUES("4024007199137174", 947, "2021-09-25", "Parth");
  INSERT INTO CREDITCARD VALUES("5130874015983067", 453, "2021-10-24", "Divija");
  INSERT INTO CREDITCARD VALUES("5164444244374535", 239, "2021-11-21", "Inaya");

CreditCard			
CC_No	CC_Cvv	CC_Expiry	CC_Name
4485029580759228	204	04/22	Raj
4024007199137174	947	09/25	Parth
5130874015983067	453	10/24	Divija
5164444244374535	239	11/21	Inaya

Offers:-

- CREATE TABLE OFFER (
  OFFER_ID VARCHAR(6),
  OFFER_NAME VARCHAR(20),
  OFFER_PERCENT DECIMAL(3,2),
  PRIMARY KEY(OFFER_ID));

  INSERT INTO OFFER VALUES("OFI101", "50OFF", 0.5);
  INSERT INTO OFFER VALUES("OFI202", "25OFF", 0.25);
  INSERT INTO OFFER VALUES("OFI303", "30OFF", 0.3);

Offer		
Offer_Id	Offer_Name	Offer_Percent
OFI101	50OFF	0.5
OFI202	25OFF	0.25
OFI303	30OFF	0.3

- CREATE TABLE OFFER_OF(
  REST_ID VARCHAR(5) NOT NULL,
  OFFER_ID VARCHAR(6) UNIQUE NOT NULL,
  PRIMARY KEY(REST_ID),
  FOREIGN KEY(REST_ID) REFERENCES REST_BRANCH_OF(REST_ID),
  FOREIGN KEY(OFFER_ID) REFERENCES OFFER(OFFER_ID));

INSERT INTO OFFER_OF VALUES("RI001","OFI101");
INSERT INTO OFFER_OF VALUES("RI002","OFI202");
INSERT INTO OFFER_OF VALUES("RI003","OFI303");

Offer_Of	
Rest_Id	*Offer_Id*
RI001	OFI101
RI002	OFI202
RI003	OFI303

Cart:-

- CREATE TABLE CART_OF (
  USER_ID VARCHAR(6),
  CART_ID VARCHAR(6) UNIQUE NOT NULL,
  PRIMARY KEY(USER_ID),
  FOREIGN KEY (USER_ID) REFERENCES USER(USER_ID));

  INSERT INTO CART_OF VALUES("UID001","CI001");
  INSERT INTO CART_OF VALUES("UID002","CI002");
  INSERT INTO CART_OF VALUES("UID004","CI003");

Cart_Of	
User_Id	Cart_Id
UID001	CI001
UID002	CI002
UID004	CI003

- CREATE TABLE CART (
  CART_ID VARCHAR(6),
  ITEM_ID CHAR(5),
  PRIMARY KEY(ITEM_ID),
  FOREIGN KEY (ITEM_ID) REFERENCES MENU_ITEM(ITEM_ID));

```
INSERT INTO CART VALUES("CI001", "II101");
INSERT INTO CART VALUES("CI001", "II102");
INSERT INTO CART VALUES("CI002", "II301");
INSERT INTO CART VALUES("CI002", "II303");
```

Cart	
Cart_Id	**Item_Id**
CI001	II101
CI001	II102
CI002	II301
CI002	II303

QUERIES

#QUERY TO FIND THE DETAILS OF THE DELIVERY MAN FOR EVERY ORDER

- SELECT O.ORDER_ID, D.DM_ID, D.DM_NAME, D.DM_PHNO FROM ORDER_DELIVERED_BY O NATURAL JOIN DELIVERYMAN D;

QUERY TO FIND THE TOTAL NUMBER OF ORDERS FOR EACH USER

- SELECT USER_ID, COUNT(*) AS NO_OF_ORDERS FROM ORDERS GROUP BY USER_ID HAVING USER_ID IN (SELECT USER_ID FROM USER);

FUNCTIONS, PROCEDURES AND TRIGGERS

- #Function for counting the number of rests above N star rating

```
Delimiter $$
CREATE FUNCTION RATINGRANGE(RATING INT)
        RETURNS INTEGER
        DETERMINISTIC
        BEGIN
                DECLARE C INT;
                SELECT COUNT(REST_RATING) INTO C FROM REST_BRANCH WHERE
REST_RATING >=RATING ;
        RETURN C;
END; $$
DELIMITER ;
```

- # Function to check whether User has Membership

```
Delimiter $$
CREATE FUNCTION HASMEMBERSHIP(ID VARCHAR(6))
        RETURNS BOOLEAN
        DETERMINISTIC
        BEGIN
                DECLARE C BOOLEAN;
                SELECT USER_MEMEND INTO C FROM USER WHERE USER.USER_ID=ID;
        RETURN C;
```

```
END; $$
DELIMITER ;
```

- # Function to Check Out from Cart

```
Delimiter $$
CREATE FUNCTION CHECKOUT(CID VARCHAR(6))
    RETURNS VARCHAR(50)
     DETERMINISTIC
    BEGIN
                                        DECLARE ORDERTOT FLOAT;
                                        DECLARE CURRTIME DATETIME;
                                        DECLARE NEXTIND VARCHAR(6);
                                        DECLARE CURRUSER VARCHAR(6);
        DECLARE AVDEL VARCHAR(6);
                                        SET CURRUSER = (SELECT USER_ID FROM CART_OF
WHERE CART_ID=CID);
                                        SET CURRTIME = CURRENT_TIMESTAMP();
                                        SET ORDERTOT = TOTAL(CID);
                                        SET NEXTIND = NEXTVAL();
        CALL RETAVAILABLE(1,AVDEL);
                                        INSERT INTO ORDERS VALUES(NEXTIND,
CURRTIME, ORDERTOT, CURRUSER);
        INSERT INTO ORDER_DELIVERED_BY VALUES(NEXTIND, AVDEL);
        DELETE FROM CART;
    RETURN "ORDER PLACED";
```

```
END; $$

DELIMITER ;
```

- # Function to get Total Of Cart

```
Delimiter $$
CREATE FUNCTION TOTAL( CID VARCHAR(6))
        RETURNS DOUBLE
        DETERMINISTIC
        BEGIN
                DECLARE C DOUBLE;
                SELECT SUM(I.ITEM_PRICE) INTO C FROM CART CT NATURAL JOIN
MENU_ITEM I WHERE CT.CART_ID = CID;

        RETURN C;
END; $$
DELIMITER ;
```

- # Function to find Next Index

```
Delimiter $$
CREATE FUNCTION NEXTVAL()
        RETURNS VARCHAR(6)
        DETERMINISTIC
        BEGIN
                        DECLARE LASTID VARCHAR(6);
                        DECLARE LASTINT DECIMAL(3,0);
                        DECLARE NEWVAL DECIMAL(3,0);
                        DECLARE NEWVALCHAR VARCHAR(3);
                        DECLARE NEWID VARCHAR(6);
                        SET LASTID = (SELECT ORDER_ID FROM ORDERS ORDER BY
ORDER_TIME DESC LIMIT 1);
                        SET LASTINT = (SELECT CAST((SELECT SUBSTRING(LASTID, 3, 5))
AS DECIMAL(3,0)));
                        SET NEWVAL = LASTINT + 1;
                        SET NEWVALCHAR= (SELECT CAST(NEWVAL AS CHAR));
                        SET NEWID = (SELECT
CONCAT(CONCAT((SUBSTRING(LASTID,1,2)), "00"),NEWVALCHAR));
                RETURN NEWID;
```

```
END; $$
DELIMITER ;
```

- # Trigger for applying discount for members

```
DELIMITER $$
CREATE TRIGGER MEMBER_DISCOUNT
        BEFORE INSERT ON ORDERS
         FOR EACH ROW
                BEGIN
                        DECLARE ISMEM BOOLEAN;
                        SET ISMEM = HASMEMBERSHIP(NEW.USER_ID);
                        IF ISMEM = 1 THEN
                                SET NEW.ORDER_COST = NEW.ORDER_COST-
(NEW.ORDER_COST)*0.25;
                        END IF;
                END$$
DELIMITER ;
```

- # Procedure to get first available Delivery Man

```
DELIMITER $$
CREATE PROCEDURE RETAVAILABLE(IN VAL INT, OUT ID VARCHAR(6))
        BEGIN
        SELECT DM_ID INTO ID FROM DELIVERYMAN WHERE DM_AVAILABILITY=1 LIMIT
1;
        END ; $$
DELIMITER ;
```