Data Book Part – 1

Swiggy Database narrative

Database name: Swiggy db

An online platform for food delivery, swiggy essentially acts as a go-between for customers and restaurants. Through the Swiggy website or mobile app, customers can place meal orders from a variety of nearby restaurants. Many things need to work well in the backend for this action to occur. For Swiggy to effectively run its business and offer a flawless user experience, database maintenance is vital.

To manage the order data, which is the Order Management system, it first requires a database. The items requested, the delivery address, the order status, and the payment information for the customer's order must all be stored in the database.

Secondly, it needs the database to manage the restaurant and menu data that is the restaurant and menu management. The database needs to stores information about partner restaurants, their menus, pricing, and availability. This helps Swiggy display up-to-date menu options to customers and manage restaurant partnerships.

Thirdly, it needs the database to manage the customers data that is the customer management. The database needs to store customer data, including profiles, order history, and preferences. This allows Swiggy to personalize the user experience and recommendations.

Fourthly, we'll need the database to store the delivery executive information that is the delivery executive management. This includes their availability, current location, and delivery history. This helps Swiggy to assign orders to available delivery partners and track deliveries.

Fifth, the database would be required to hold data regarding Swiggy's discounts, promos, and coupon codes. This enables Swiggy to monitor and evaluate the success of marketing efforts, as well as apply discounts to orders.

With these considerations in mind, the database for the swiggy delivery consists of a total of 12 tables: RESTAURANT, MENU_ITEM, CUSTOMER, ORDER, ORDER_ITEM, DELIVERY_EXECUTIVE, PROMOTION, CUSTOMER_ADDRESS, PAYMENT, REVIEW, CUSTOMERCARE_INTERACTION,

ENTITY DESCRIPTION

1. RESTAURANT: Represents a food place offering food to the customers

2. MENU_ITEM: Describes the food item available in the menu for order.

3. CUSTOMER: Represents an individual who places an order online.

4. ORDER: Represents a request made by a customer through the food delivery app.

5. ORDER_ITEM: Represents a specific item that is part of the order.

6. DELIVERY_EXECUTIVE: Represents an individual assigned to deliver an order

7. PROMOTION: Describes a special offer or discount provided by a restaurant to customers.

8. CUSTOMER_ADDRESS: Represents the location where an order should be delivered.

9. PAYMENT: Represents a financial transaction made by a customer to pay for an order.

10. REVIEW: Represents feedback or comments provided by a customer about their experience with a restaurant and the delivery

11. CUSTOMERCARE_INTERACTION: Represents an interaction between a customer and customer care services.

Table Name	Attribute Name	Contents	Data Type	Format	Range	Required	PK/FK	Reference
RESTAURANT	REST_ID	Unique identifier for a restaurant	INT(6)	999999	1-999…	Y	Primary Key	
	REST_NAME	Name of the restaurant	VARCHAR(35)	Xxxxxxxxxx		Y		
	REST_CUISINE	Type of cuisine served	VARCHAR(35)	Xxxxxxxxxx		Y		
	REST_RATING	Rating of the restaurant	DECIMAL(2,1)	9.9	0-5	y		
	REST_LOCATION	Address of the restaurant	VARCHAR(50)	Xxxxxxx		Y		
	REST_OPENHRS	Opening hours of the restaurant	TIME	Hh:mm:ss		Y		
	REST_CLOSEHRS	Closing hours of the restaurant	TIME	Hh:mm:ss		Y		
MENU_ITEM	MENUITEM_ID	Unique identifier for a menu item	INT(6)	999999	1-999…	Y	Primary Key	
	REST_ID	Identifier for the restaurant	INT(6)	999999	1-999..	Y	Foreign Key	RESTAURANT
	MENUITEM_NAME	Name of the menu item	VARCHAR(30)	Xxxxxx		Y		
	MENUITEM_DES	Description of the menu item	VARCHAR(100)	Xxxxxxxx		N		
	MENUITEM_PRICE	Price of the menu item	DECIMAL(6,2)	9999.99	1.00-9999.99	Y		
	MENUITEM_CTG	Category of the menu item	VARCHAR(20)	Xxxxxx		Y		
CUSTOMER	CUST_ID	Unique identifier for a customer	INT(5)	99999	1-999	Y	Primary Key	
	CUST_NAME	Name of the customer	VARCHAR(25)	Xxxxx		Y		
	CUST_EMAIL	Email address of the customer	VARCHAR(50)	Xxx@xxx		Y		
	CUST_PHONE	Phone number of the customer	CHAR(12)	999-999-9999		Y		
	CUST_DOB	Data of birth of the customer	DATE	Yyyy-mm-dd		Y		
ORDER	ORDER_ID	Unique identifier for an order	INT(5)	99999	1-999..	Y	Primary Key	
	CUST_ID	Identifier for the customer	INT(5)	99999	1-999..	Y	Foreign Key	CUSTOMER
	MENUITEM_ID	Unique identifier for a menu item	INT(6)	999999	1-999..	Y	Foreign Key	MENU_ITEM
	ORDER_DATE	Date and time of the order	DATE TIME	Yyyy-mm-dd hh:mm		Y		
	ORDER_DELIVERY_ADD	Address for order delivery	VARCHAR(50)	Xxxxxx		Y		
	ORDER_STATUS	Status of the order	VARCHAR(20)	Xxxxxxx		Y		
ORDER_ITEM	ORDERITEM_ID	Unique identifier for an order item	INT(4)	9999	0-999..	Y	Primary Key	
	ORDER_ID	Identifier for the order	INT(5)	99999	1-999..	Y	Foreign Key	ORDER
	MENUITEM_ID	Identifier for the menu item	INT(6)	999999	1-999..	Y	Foreign Key	MENU_ITEM
	ORDERITEM_QUAN	Quantity of the menu item	INT(2)	99	1-99..	Y		
	ORDERITEM_PRICE	Price of the order item	DECIMAL(6,2)	9999.99	1.00-9999.99	Y		
DELIVERY_EXECUTIVE	DELIVERYEXU_ID	Unique identifier for a delivery executive	INT(5)	99999	1-999..	Y	Primary Key	
	ORDER_ID	Unique Identifier for the order	INT(5)	99999	1-999..	Y	Foreign Key	ORDER
	REST_ID	Identifier for the restaurant	INT(6)	999999	1-999..	Y	Foreign Key	RESTAURANT
	DELIVERYEXU_NAME	Name of the delivery executive	VARCHAR(20)	Xxxxx		Y		

	DELIVERYEXU_PHONE	Phone number of the delivery executive	CHAR(12)	999-999-9999		Y		
	DELIVERYEXU_LOCATION	Current location of the delivery executive	VARCHAR(30)	Xxxxxxxx		Y		
	DELIVERYEXU_AVAL	Availability status of the delivery executive	BOOL	0/1	0-1	Y		
PROMOTION	PROM_ID	Unique identifier for a promotion	INT(5)	99999	1-999..	Y	Primary Key	
	ORDER_ID	Unique Identifier for the order	INT(5)	99999	1-999..	Y	Foreign Key	ORDER
	PROM_CODE	Promotion code	VARCHAR(5)	Xxxxxx		Y		
	PROM_DES	Description of the promotion	VARCHAR(25)	Xxxxxxxx		N		
	PROM_DISCOUNT	Discount amount for the promotion	DECIMAL(4,2)	99.99	1-99.99	Y		
	PROM_VALIDFR	Start date of the promotion	Date	Yyyy-mm-dd		Y		
	PROM_VALIDTO	End date of the promotion	Date	Yyyy-mm-dd		Y		
CUSTOMER_ADDRESS	CUST_ADD_ID	Unique identifier for a customer address	INT(6)	999999	1-999..	Y	Primary Key	
	CUST_ID	Identifier for the customer	INT(5)	99999	1-999..	Y	Foreign Key	CUSTOMER
	CUST_ADD	Address of the customer	VARCHAR(50)	Xxxxxxx		Y		
	CUST_DEFAULT	Indicates if the address is the default	BOOL	0/1	0-1	Y		
PAYMENT	PAY_ID	Unique identifier for a payment	INT(5)	99999	1-999..	Y	Primary Key	
	ORDER_ID	Identifier for the order	INT(5)	99999	1-999..	Y	Foreign Key	ORDER
	PAY_AMOUNT	Amount of the payment	DECIMAL(6,2)	9999.99	1.00-9999.99	Y		
	PAY_DATE	Date and time of the payment	DATE TIME	Yyyy-mm-dd hh:mm		Y		
	PAY_METHOD	Payment method used	VARCHAR(20)	Xxxxxxxx		Y		
REVIEW	REVIEW_ID	Unique identifier for a review	INT(4)	9999	1-999..	Y	Primary Key	
	CUST_ID	Identifier for the customer	INT(5)	99999	1-999..	Y	Foreign Key	CUSTOMER
	REST_ID	Identifier for the restaurant	INT(6)	999999	1-999..	Y	Foreign Key	RESTAURANT
	REVIEW_COMMENT	Comment on the restaurant	VARCHAR(100)	Xxxxxxxx		N		
	REVIEW_DATE	Date of the review	DATE	Yyyy-mm-dd		Y		
CUSTOMERCARE_INERACTION	INTER_ID	Unique identifier for an interaction	INT(5)	99999	0-999	Y	Primary Key	
	CUST_ID	Identifier for the customer	INT(5)	99999	1-999	Y	Foreign Key	CUSTOMER
	INTER_DATE	Date and time of the interaction	DATE TIME	Yyyy-mm-dd hh:mm		Y		
	INTER_DES	Description of the interaction	VARCHAR(100)	Xxxxxxxxxxx		N		
	INTER_STATUS	Status of the issue (open OR closed)	BOOL	0/1	0-1	Y		

Entity Relationship Model (ERM):

ENTITY	RELATIONSHIP	CONNECTIVITY	ENTITY
RESTAURANT	HAS	(1:M)	DELIVERY_EXECUTIVE
RESTAURANT	OFFERS	(1:M)	MENU_ITEM
CUSTOMER	SUBMIT	(0:M)	REVIEW
ORDER	IS PAID VIA	(1:M)	PAYMENT
ORDER	IS FOR	(1:1)	CUSTOMER
ORDER_ITEM	INCLUDES	(M:M)	MENU_ITEM
DELIVERY_EXECUTIVE	DELIVERS	(1:M)	ORDER
CUSTOMER_ADDRESS	IS FOR	(1:1)	CUSTOMER
PROMOTION	APPLIED	(1:1)	ORDER
PAYMENT	IS FOR	(1:1)	ORDER
REVIEW	VIEWED BY	(M:1)	CUSTOMER
CUSTOMERCARE_INTERACTION	INTERACTED WITH	(1:M)	CUSTOMER

NOTE: BRIDGE TABLE IMPLEMENTING M:M RELATIONS

ORDER_ITEM	BELONGS TO	(1:M)	ORDER
MENU_ITEM	OWENED BY	(1:M)	ORDER

Based on the relationships between the entities, here are the business rules:

RESTAURANT/ DELIVERY_EXECUTIVE

- A restaurant can have one or more delivery executives, but a delivery executive is associated with only one restaurant at a time. (1:M)

RESTAURANT/ MENU_ITEM

- A restaurant offers one or more menu items, but a menu item is offered by only one restaurant. (1:M)

CUSTOMER / REVIEW

- A customer can submit zero or more reviews, but each review is submitted by only one customer. (0:M)

ORDER/ PAYMENT

An order is paid via many payment methods, but a payment can be for one order. (1:M)

ORDER / CUSTOMER

- An order is for one customer, but a customer can have one or more orders. (1:1)

ORDER_ITEM /MENU_ITEM

- An order item includes one or more menu items, and a menu item can be included in one or more order items. (M:M)

DELIVERY_EXECUTIVE /ORDER

- A delivery executive delivers one or more orders, but each order is delivered by only one delivery executive. (1:M)

CUSTOMER_ADDRESS / CUSTOMER

- A customer address is for one customer, and a customer can have only one address. (1:1)

PROMOTION / ORDER

- A promotion is applied to one order, and an order can have only one promotion applied. (1:1)

PAYMENT / ORDER

- A payment is for one order, and an order can have only one or more payment method. (1:M)

REVIEW /CUSTOMER

- Many review can be viewed by one customer, but a customer can view zero or more reviews. (M:1)

CUSTOMERCARE_INTERACTION /CUSTOMER

- A customer care interaction is interacted with by one customer, but a customer can have one or more interactions with different customers. (1:M)

MANY TO MANY (M:M) RELATIONS AND BRIDGE ENTITIES IMPLEMENTING THE RESPECTIVE RELATIONS

ORDER_ITEM /ORDER

- An order item belongs to one order, and an order can have one or more order items. (1:M)

MENU_ITEM /ORDER

- A menu item can be part of one or more orders, and an order can contain one or more menu items. (MENU_ITEM CAN BE (1:M) ORDER)

Shreya Reddy Swiggy Entity Relationship Diagram

SHORT HAND RELATION SCHEMA:

RESTAURANT (**REST_ID**, REST_NAME, REST_CUISINE, REST_RATING, REST_LOCATION, REST_OPENHRS, REST_CLOSEHRS)

MENU_ITEM (**MENUITEM_ID**, REST_ID, MENUITEM_NAME, MENUITEM_DES, MENUITEM_PRICE, MENUITEM_CTG)

CUSTOMER (**CUST_ID**, CUST_NAME, CUST_EMAIL, CUST_PHONE, CUST_DOB)

ORDER (**ORDER_ID**, CUST_ID, REST_ID, ORDER_DATE, ORDER_DELIVERY_ADD, ORDER_STATUS)

ORDER_ITEM (**ORDERITEM_ID**, ORDER_ID, MENUITEM_ID, ORDERITEM_QUAN, ORDERITEM_PRICE)

DELIVERY_EXECUTIVE (**DELIVERYEXU_ID,** DELIVERYEXU_NAME, DELIVERYEXU_PHONE, DELIVERYEXU_LOCATION, DELIVERYEXU_AVAL)

PROMOTION (**PROM_ID**, PROM_CODE, PROM_DES, PROM_DISCOUNT, PROM_VALIDFR, PROM_VALIDTO)

CUSTOMER_ADDRESS (**CUST_ADD_ID**, CUST_ID, CUST_ADD, CUST_DEFAULT)

PAYMENT (**PAY_ID**, ORDER_ID, PAY_AMOUNT, PAY_DATE, PAY_METHOD)

REVIEW (**INTER_ID**, CUST_ID, REST_ID, REVIEW_COMMENT, REVIEW_DATE)

CUSTOMERCARE_INERACTION (**INTER_ID**, CUST_ID, INTER_DATE, INTER_DES, INTER_STATUS)

DELIVERY_EXECUTIVE_ASSIGNMENT (**INTER_ID**, ASSIGNMENT_STATUS)

1NF DEPENDENCY AND DIAGRAM

1NF (**ORDER_ID, CUST_ID, REST_ID,** ORDER_DATE, ORDER_STATUS, CUST_EMAIL,CUST_ADD_ID,CUST_ADD,REST_LOCATION, REST_NAME)

PARTIAL DEPENDENCIES:

(ORDER_ID → ORDER_DATE, ORDER_STATUS)

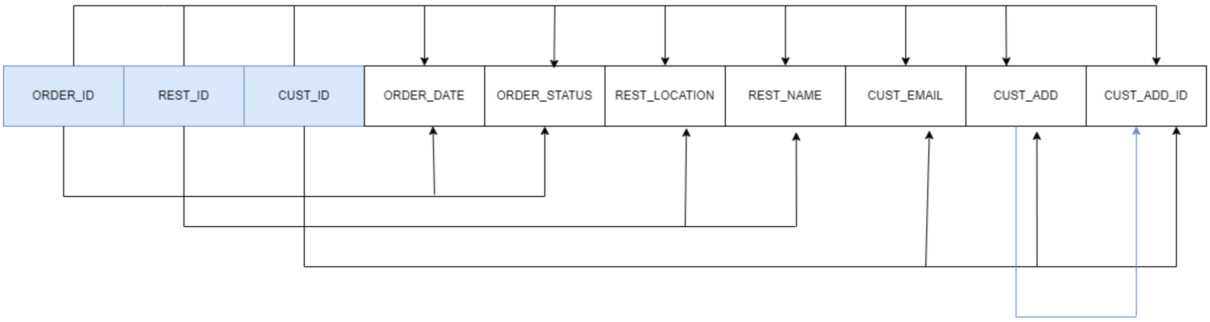
(CUST_ID → CUST_EMAIL, CUST_ADD_ID, CUST_ADD)

(REST_ID → REST_LOCATION, REST_NAME)

TRANSITIVE DEPENDENCY

CUST_ADD→ CUST_ADD_ID

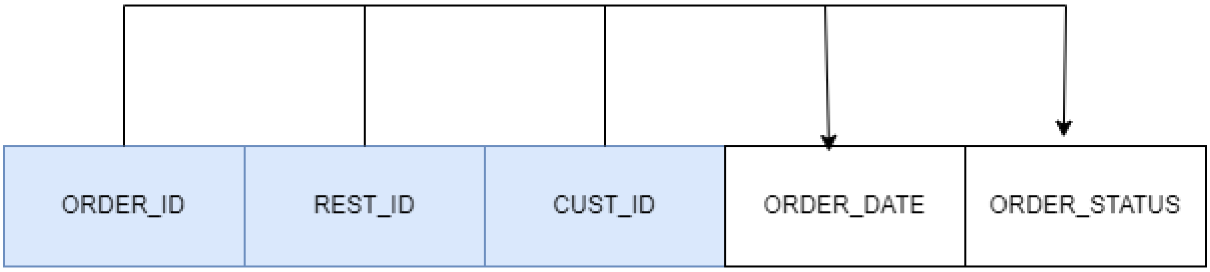
I NF

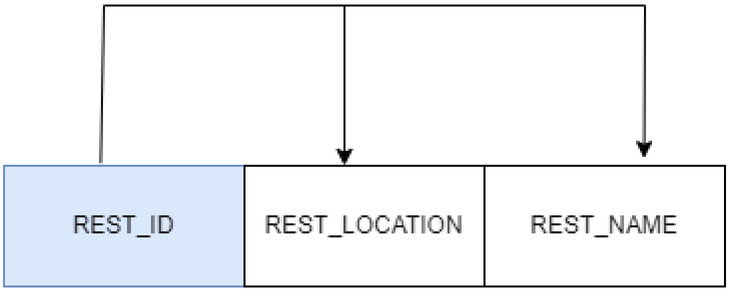


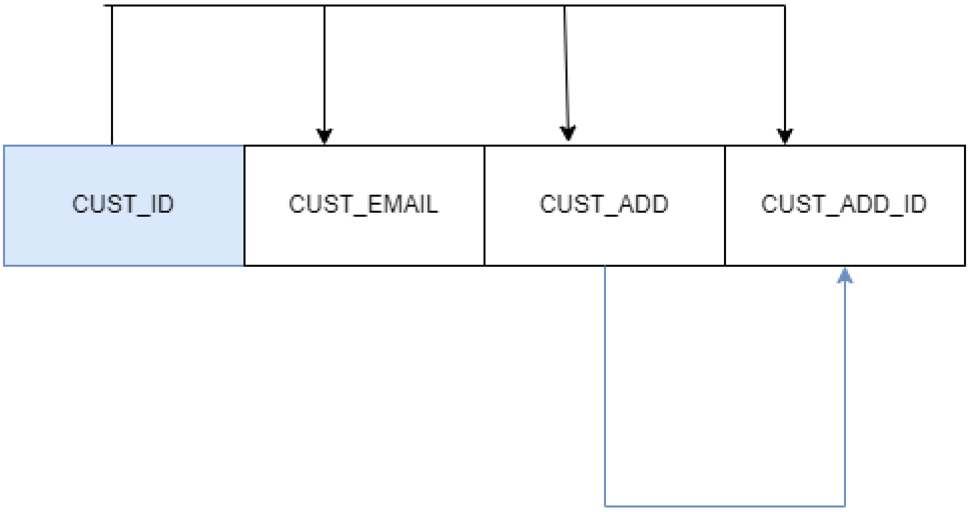
To achieve Second Normal Form (2NF) we need to remove Partial Dependencies

2NF AND DEPENDENCY DIAGRAMS

2 NF







ORDER (**<u>ORDER_ID, REST_ID, CUST_ID</u>**, ORDER_DATE, ORDER_STATUS)

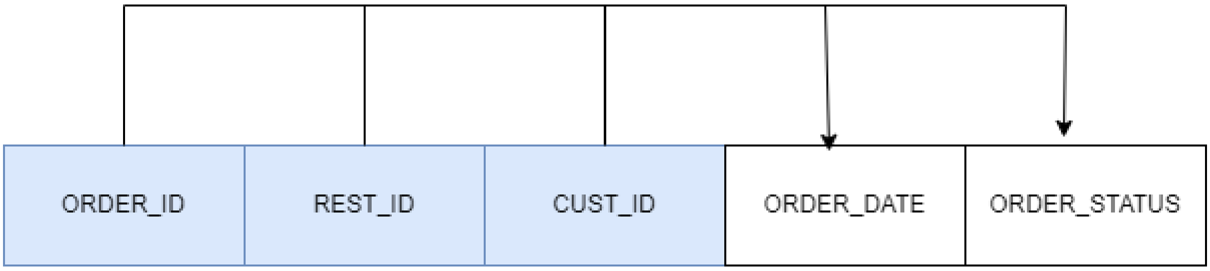
RESTAURANT (**<u>REST_ID</u>**, REST_LOCATION, REST_NAME)

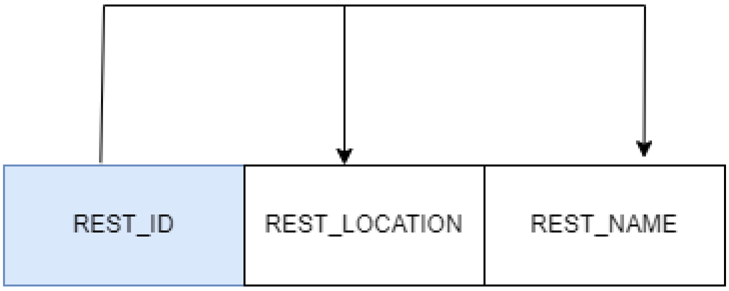
CUSTOMER (**CUST_ID**, CUST_EMAIL, CUST_ADD, CUST_ADD_ID

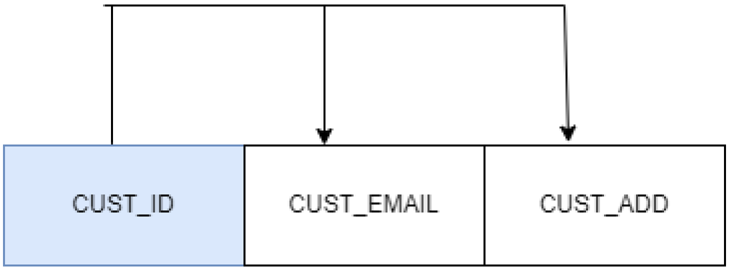
2NF: Here we removed partial dependencies and reduced by putting them into separate tables. But ORDER_DATE AND ORDER_STATUS are still dependent on ORDER_ID, REST_ID, CUST_ID

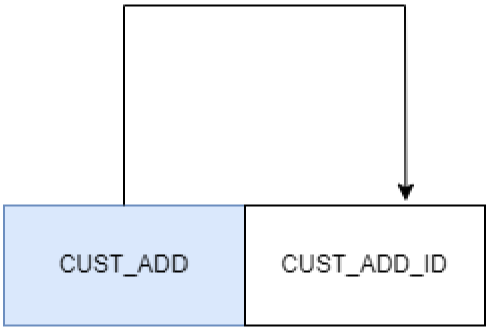
There is also Transitive dependency that should be separated only then we can achieve 3NF.

3NF AND DEPENDENCY DIAGRAMS









ORDER (**ORDER_ID, REST_ID, CUST_ID**, ORDER_DATE, ORDER_STATUS)

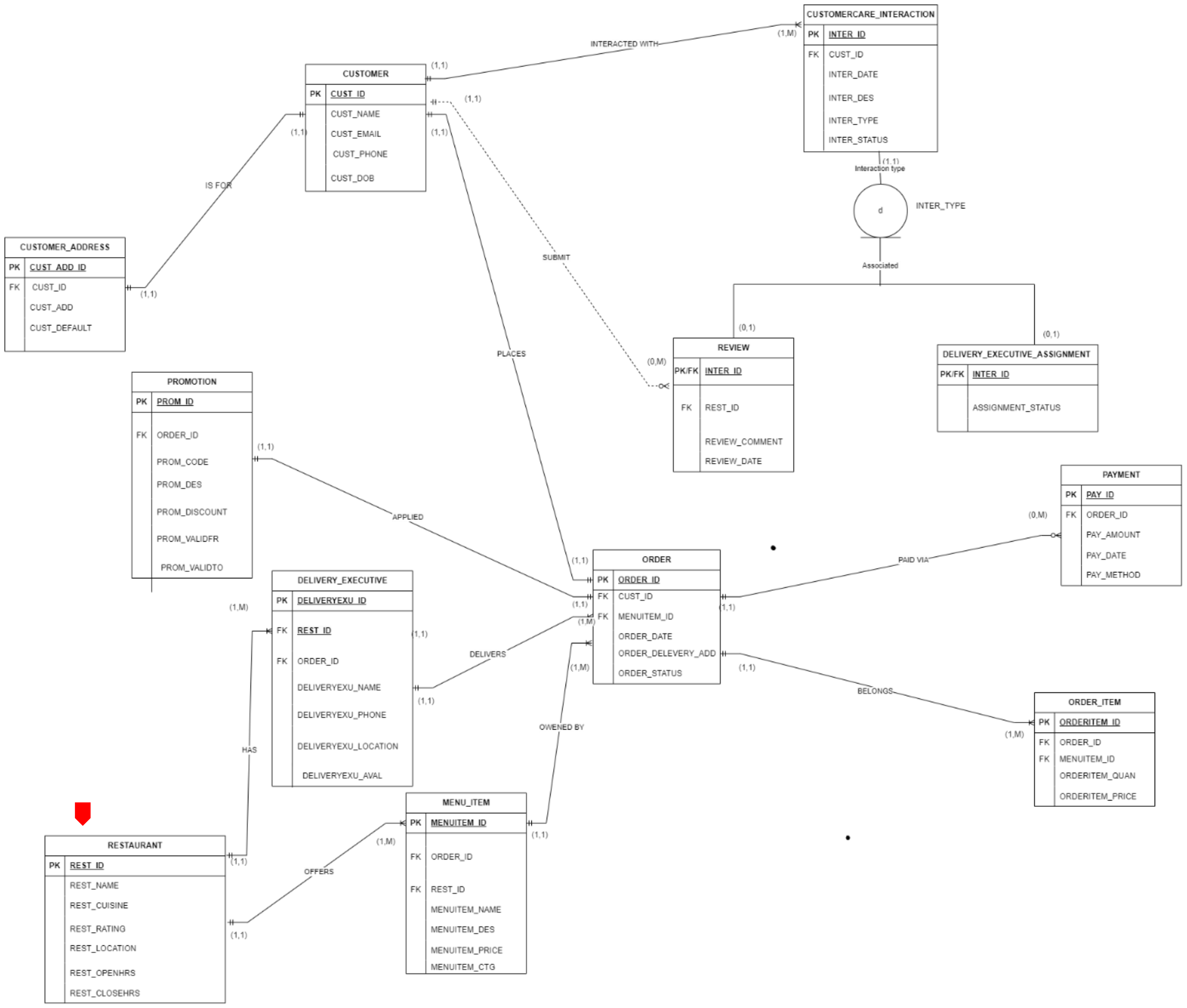
RESTAURANT (**REST_ID**, REST_LOCATION, REST_NAME)

CUSTOMER (**CUST_ID**, CUST_EMAIL, CUST_ADD,)

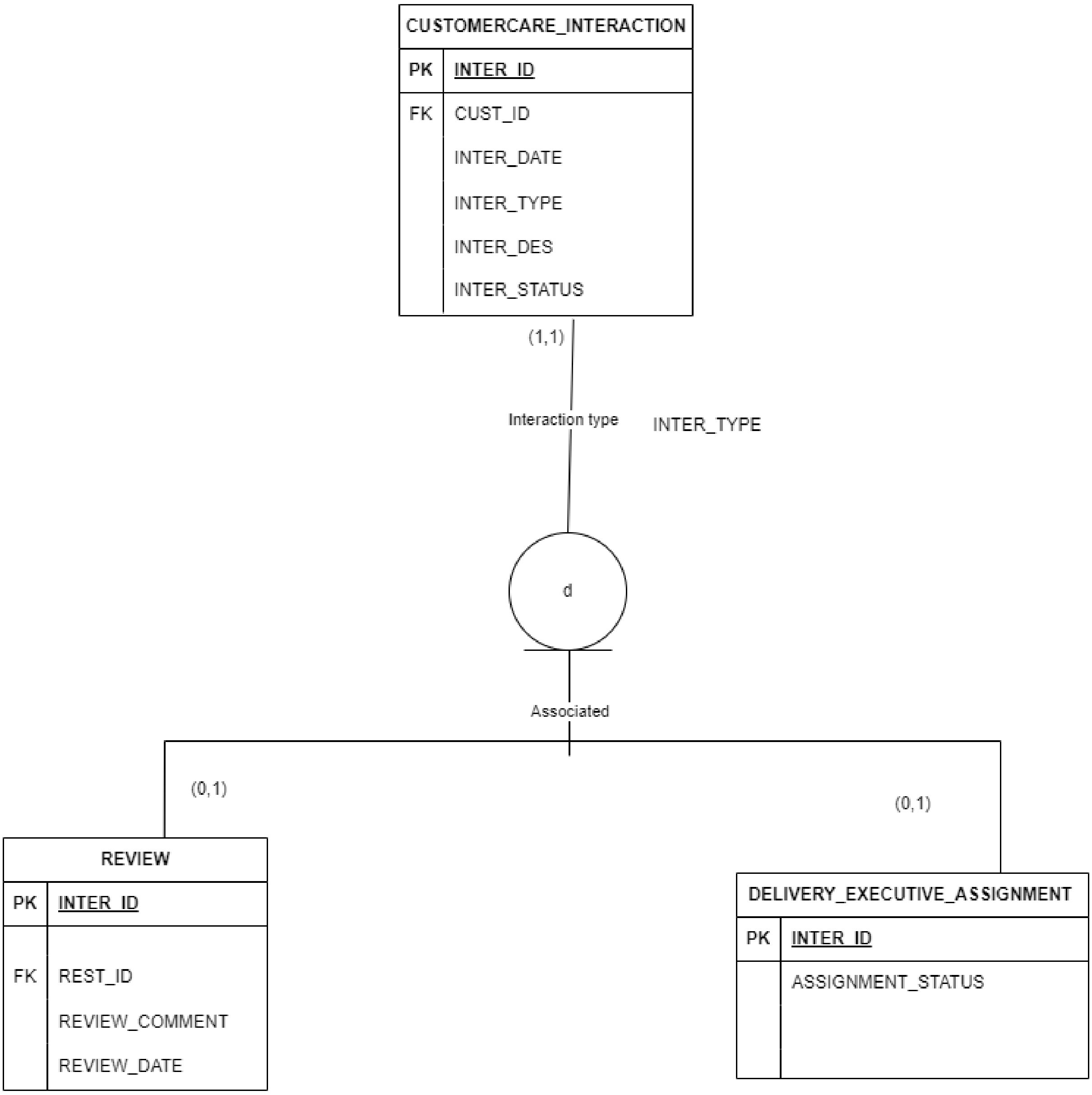
CUSTOMER_ADD (**CUST_ADD**, CUST_ADD_ID)

Here CUSTOMER_ADD given their own table, transitive dependency is removed and 3NF is achieved.

ERD USING CROWS FOOT NOTATION



SUPER TYPE AND SUBTYPE



subtypes and supertypes refer to a relationship between entities where one entity (supertype) represents a general category, and another entity (subtype) represents a more specific type of the supertype.

Here are the potential subtypes and supertype:

Supertype:

CUSTOMERCARE_INERACTION is the supertype, representing a general category of interactions with customers.

Subtypes:

REVIEW is a subtype of, CUSTOMERCARE_INERACTION representing a specific type of interaction where customers leave reviews about restaurants.

DELIVERY_EXECUTIVE_ASSIGNMENT is another subtype of Customer Interaction, representing interactions related to assigning delivery executives to orders.

subtypes are disjoint or non-overlapping, it means that each entity instance of the supertype can only belong to one of the subtypes
In the above we can see that Interaction type cannot be both review and delivery executive assignment. It can only be one

Since each occurrence of a supertype must belong to at least one subtype, it is a Total Completeness.

QUERIES

1.Query pulling data from one table:

Can you provide a list of restaurants and their locations in the USA?

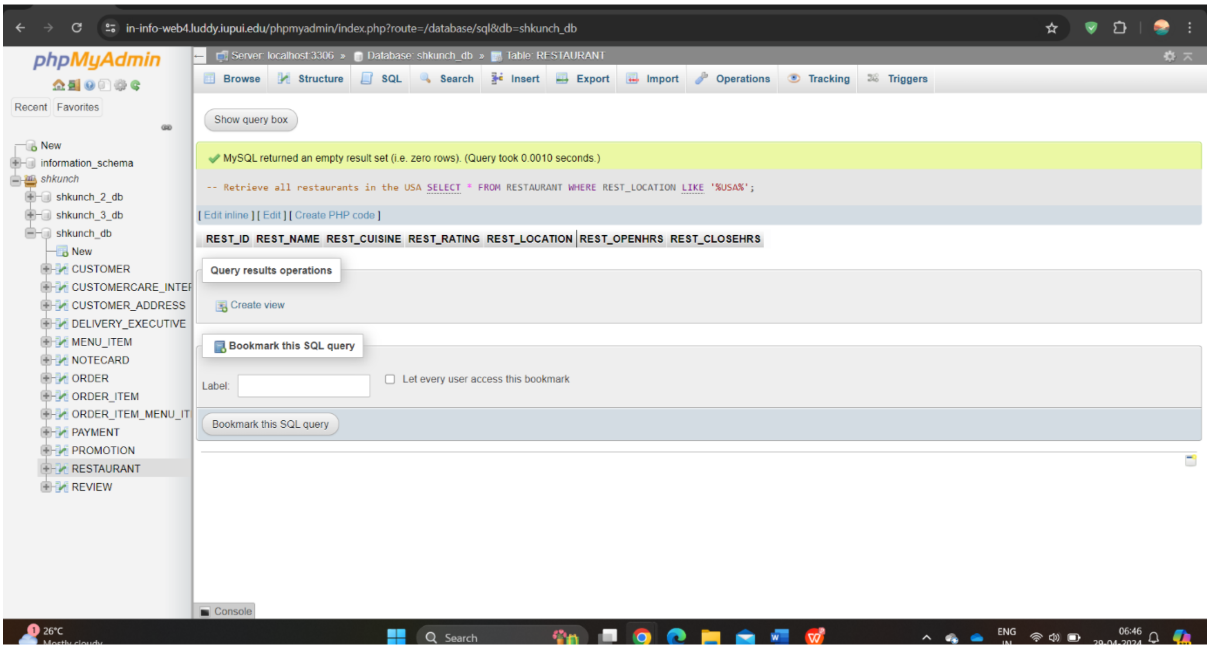
Code

```
-- Retrieve all restaurants in the USA

SELECT *

FROM RESTAURANT

WHERE REST_LOCATION LIKE '%USA%';
```



✓ MySQL returned an empty result set (i.e. zero rows). (Query took 0.0010 seconds.)

```
-- Retrieve all restaurants in the USA SELECT * FROM RESTAURANT WHERE REST_LOCATION LIKE '%USA%';
```

REST_ID	REST_NAME	REST_CUISINE	REST_RATING	REST_LOCATION	REST_OPENHRS	REST_CLOSEHRS

2. Query pulling data from two tables (with a join):

What are the menu items offered by the restaurant named "Restaurant_Name"?

Code

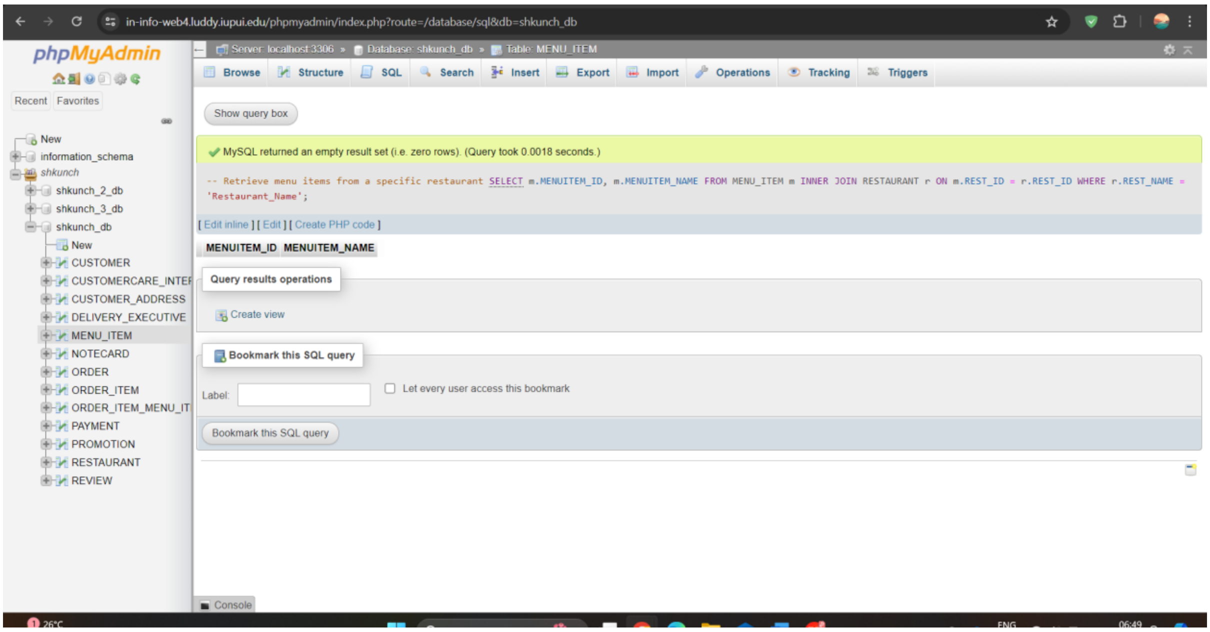
```
-- Retrieve menu items from a specific restaurant

SELECT m.MENUITEM_ID, m.MENUITEM_NAME

FROM MENU_ITEM m

INNER JOIN RESTAURANT r ON m.REST_ID = r.REST_ID
```

WHERE r.REST_NAME = 'Restaurant_Name';



✓ MySQL returned an empty result set (i.e. zero rows). (Query took 0.0018 seconds.)

```
-- Retrieve menu items from a specific restaurant SELECT m.MENUITEM_ID, m.MENUITEM_NAME FROM MENU_ITEM m INNER JOIN RESTAURANT r ON m.REST_ID = r.REST_ID WHERE r.REST_NAME =
'Restaurant_Name';
```

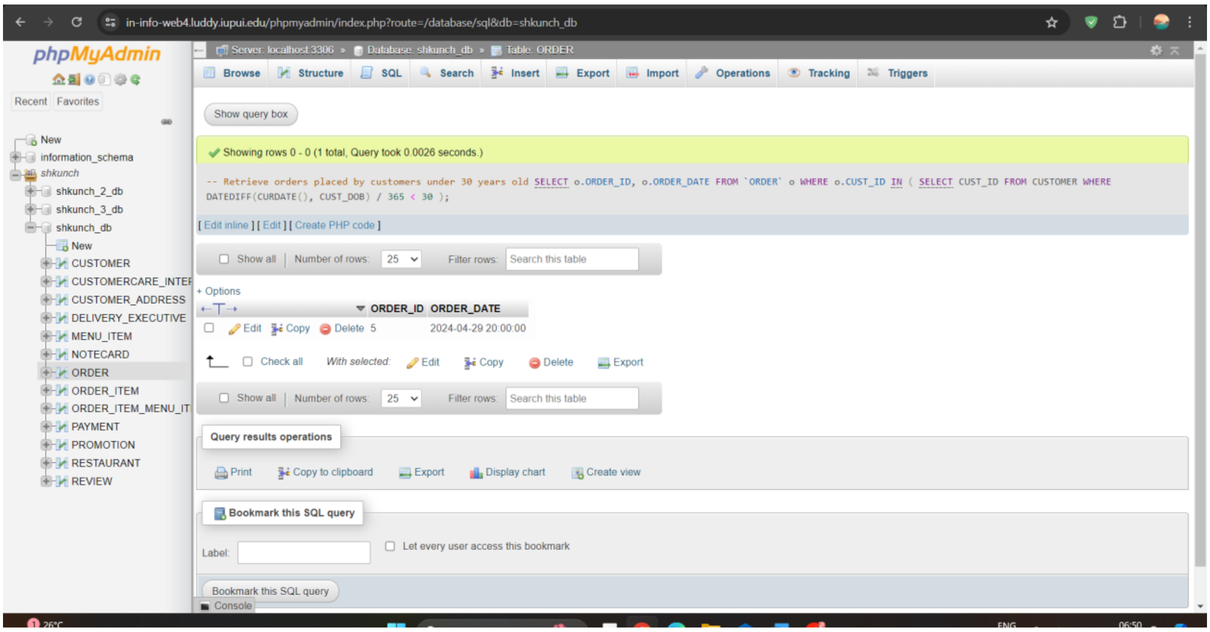
MENUITEM_ID	MENUITEM_NAME

Query with a subquery:

How many orders were placed by customers under the age of 30?

Code

```
-- Retrieve orders placed by customers under 30 years old
SELECT o.ORDER_ID, o.ORDER_DATE
FROM `ORDER` o
WHERE o.CUST_ID IN (
    SELECT CUST_ID
    FROM CUSTOMER
    WHERE DATEDIFF(CURDATE(), CUST_DOB) / 365 < 30
);
```



✓ Showing rows 0 - 0 (1 total, Query took 0.0026 seconds.)

```
-- Retrieve orders placed by customers under 30 years old SELECT o.ORDER_ID, o.ORDER_DATE FROM `ORDER` o WHERE o.CUST_ID IN ( SELECT CUST_ID FROM CUSTOMER WHERE
DATEDIFF(CURDATE(), CUST_DOB) / 365 < 30 );
```

[Edit inline] [Edit] [Create PHP code]
☐ Show all
Number of rows: 25
Filter rows: Search this table
+ Options

	ORDER_ID	ORDER_DATE
☐ Edit Copy Delete	5	2024-04-29 20:00:00