**Assignment 3**

**Group Members:**

● Monika Agarwal

● Rishi Shrivastava

● Snehal Bailmare

● Prateek Agrawal

● Smita Damani

1. **Design the member order management database.**
2. **List the entities, attributes, and relationships from the sample data.**

**ENTITIES:**

1. **RESTAURANT:**
   1. RestaurantID
   2. RestaurantName
   3. RestaurantTypeID
   4. IncomePercentage
   5. *CityID*
2. **CITY:**
   1. CityID
   2. City
3. **ORDER:**
   1. OrderID
   2. Date
   3. Time
   4. TotalOrder
   5. *MemberID*
   6. *RestaurantID*
4. **ORDER\_DETAIL:**
   1. Order\_Detail\_ID
   2. OrderID
   3. MealID
5. **MEMBER:**
   1. MemberID
   2. FirstName
   3. LastName
   4. Gender
   5. Email
   6. MonthlyBudget
   7. *CityID*
6. **MEAL:**
   1. MealID
   2. HotCold
   3. MealName
   4. Price
   5. *RestaurantID*
   6. *ItemTypeID*
   7. *MealTypeID*
7. **MEAL\_TYPE:**
   1. MealTypeID
   2. MealType
8. **ITEM\_TYPE:**
   1. ItemTypeID
   2. ItemType
9. **MEMBER\_MONTHLY\_TOTAL:**
   1. MemberID
   2. FirstName
   3. LastName
   4. Gender
   5. Email
   6. City
   7. Year
   8. Month
   9. OrderCount
   10. MealCount
   11. TotalExpense
   12. Balance
   13. Commission
   14. MonthlyBudget

**RELATIONSHIPS:**

1. **One-To-Many (1:N):**
   1. CITY to RESTAURANT
   2. RESTAURANT to ORDER
   3. MEMBER to ORDER
   4. CITY to MEMBER
   5. MEAL\_TYPE to MEAL
   6. ITEM\_TYPE to MEAL
   7. RESTAURANT to MEAL
   8. MEMBER to MEMBER\_MONTHLY\_TOTAL
   9. ORDER to ORDER\_DETAIL
   10. MEAL to ORDER\_DETAIL
2. **Draw an entity-relationship (E-R) diagram using the Crow’s feet or Chen notation.**

**Start with the model that you reverse engineer from the given data. Then, make modifications to improve the model. State how you improved the model that was inherent in the data.**

1. **Entity-relationship (E-R) diagram using the Crow’s feet notation**

Diagram

Description automatically generated

1. **Modifications from original dataset :**
2. Made restaurantTypeId as foreign key in RESTAURANT table.
3. Added a new table RESTAURANT\_TYPE.
4. ORDER\_DETAIL is the result of a many to many relationship between ORDER and MEAL tables, so dropped its primary key.
5. Dropped MEMBER\_MONTHLY\_TOTAL table.

**Assumptions for above mentioned Modifications:**

1. For better understanding and designing of the database we have introduced a new table RESTAURANT\_TYPE with attributes RestaurantTypeID, RestaurantType with a real word business requirement where we have multiple restaurant serving different authentic and cultural cuisines.
2. With reference to above table (RESTAURANT\_TYPE) we have added a foreign key constraint in RESTAURANT table for the column ‘RestaurantTypeID’.
3. With the given set of entities and attributes, we can easily anticipate that we have an many to many association between ORDER and MEAL. This relationship will help us creating a new table called ‘ORDER\_DETAIL’ having composite primary key as (OrderID, MealID). With this concept we have learnt in class we do not need to have a separate unique identifier called ‘OrderDetailID’. Hence, we have removed it.
4. Finally, while doing our analysis and having our observations over the real world data provided in spreadsheets, we have identified that we do not need a separate table called ‘MEMBER\_MONTHLY\_INCOME’. To justify our assumptions below are the key points –
5. MemberID, FirstName, LastName, Gender, Email are the redundant columns from MEMBER table, thus we can avoid this redundancy.
6. We can derive OrderCount directly from the ORDER table and MealCount from ORDER\_DETAIL table having join with other parent tables we already have in our structure.
7. As we have some messy data from the real word entities where we need to introduce data integrity, MonthlyBudget and TotalExpense can be used interchangeably. Also, these are derived columns as well we can easily calculate from the ORDER table where we already have month-wise date for all the members.
8. Balance can be derived from point C data which is difference of Monthly Budget and Expenditure done by a member at course of time.
9. Again, ‘Commission’ is a derived attribute which can be calculated using ORDER Table columns TotalOrder and RestaurantID. Further we can utilize RESTAURANT table to get the ‘Income Percentages’ via RestaurantID’s. We have interpreted that it is a commission earned by the restaurant using the client’s membership program. We can also derive the values with respect to year, months as well.
10. **Transform the E-R diagram into a relational database design that includes all the keys, data types, and constraints (e.g., null or not-null values). Show how the transformation rules are applied.**
11. **Relational Model:**

**RESTAURANT** (RestaurantID, RestaurantName, IncomePercentage, *RestaurantTypeID, CityID*)

**RESTAURANT\_TYPE** (RestaurantTypeID, RestaurantType)

**CITY** (CityID, City)

**ORDER** (OrderID, Date, Time, TotalOrder, *MemberID*, *RestaurantID*)

**ORDER\_DETAIL** (OrderID, MealID)

**MEMBER** (MemberID, FirstName, LastName, Gender, Email, MonthlyBudget, *CityID*)

**MEAL** (MealID, HotCold, MealName, Price, *RestaurantID*, *ItemTypeID*, *MealTypeID*)

**MEAL\_TYPE** (MealTypeID, MealType)

**ITEM\_TYPE** (ItemTypeID, ItemType)

1. **Database Schema:**

**RESTAURANT(**RestaurantID, RestaurantName, IncomePercentage, Rating, *RestaurantTypeID, CityID*)

Table

Description automatically generated

**RESTAURANT\_TYPE**(RestaurantTypeID, RestaurantType)

Graphical user interface, text, application

Description automatically generated

**CITY** (CityID, City)

Table

Description automatically generated

**MEMBER**(MemberID, FirstName, LastName, Gender, Email, MonthlyBudget, *CityID*)

Table

Description automatically generated

**ORDER**(OrderID, Date, Time, TotalOrder, *MemberID*, *RestaurantID*)

Table

Description automatically generated

**MEAL\_TYPE**(MealTypeID, MealType)

Graphical user interface, text, application

Description automatically generated

**ITEM\_TYPE**(ItemTypeID, ItemType)

Graphical user interface, application

Description automatically generated

**MEAL**(MealID, HotCold, MealName, Price, *RestaurantID*, *ItemTypeID*, *MealTypeID*)

Table

Description automatically generated

**ORDER\_DETAIL**(OrderID, MealID)

Graphical user interface, table

Description automatically generated

**Database Schema Diagram**

**Diagram

Description automatically generated**

1. Implement the database. Use Oracle, which is provided by GSU. Use a portion of the given data to populate the database; e.g., 15-20 entries for each relation for which the data exists. Show the data by using the Select \*; command.

**Table 1: RESTAURANT**

Table

Description automatically generated

**Table 2: RESTAURANT\_TYPE**

Graphical user interface, text, application

Description automatically generated

**Table 3: CITY**

Graphical user interface, text, application, email

Description automatically generated

**Table 4: MEMBER**

Graphical user interface, table

Description automatically generated

**Table 5: MEAL\_TYPE**

Graphical user interface, text, application, email

Description automatically generated

**Table 6: ITEM\_TYPE**

Graphical user interface, text, application, email

Description automatically generated

**Table 7: MEAL**

Table

Description automatically generated

**Table 8: ORDERR**

Table

Description automatically generated with medium confidence

**Table 9: ORDER\_DETAIL**

Graphical user interface

Description automatically generated with medium confidence

1. **Identify 10 important, non-trivial queries for this database. Write the queries in  
   English and state the importance of each query. Write the queries in SQL and run them  
   against the database. Show the SQL commands. Provide screen shots of the results obtained.  
   Trivial queries are, for example, ‘retrieve the names of the customers.’ At least 7 of the  
   queries should require a join operation.**
2. Retrieve the names of all restaurants in a specific city.

This query is important to help customers find restaurants within a specific city they are interested in visiting.

**SELECT** RestaurantName

**FROM** RESTAURANT

**WHERE** CityID **=** 2**;**

Graphical user interface, application

Description automatically generated

1. Retrieve the names of all restaurants that offer a specific type of meal.

This query is important for customers looking for a specific type of meal, such as vegetarian or vegan options.

**SELECT** RestaurantName

**FROM** RESTAURANT

**INNER** **JOIN** MEAL **ON** RESTAURANT**.**RestaurantID **=** MEAL**.**RestaurantID

**INNER** **JOIN** MEAL\_TYPE **ON** MEAL**.**MealTypeID **=** MEAL\_TYPE**.**MealTypeID

**WHERE** MEAL\_TYPE**.**MealType **=** 'Vegan'**;**

Graphical user interface, text, application, email

Description automatically generated

1. Retrieve the names of all members who have a monthly budget greater than a specific amount.

This query is important to help restaurants target customers who are more likely to spend money on their meals.

**SELECT** FirstName**,** LastName

**FROM** MEMBER

**WHERE** MonthlyBudget **>** 500**;**

Graphical user interface, text, application, email

Description automatically generated

1. Retrieve the names and prices of all meals offered by a specific restaurant.

This query is important for customers looking for a specific meal or price range at a restaurant.

**SELECT** MealName**,** Price

**FROM** MEAL

**WHERE** RestaurantID **=** 1**;**

Graphical user interface, text, application

Description automatically generated

1. Retrieve the names of all members who have placed an order at a specific restaurant.

This query is important for restaurants to target their marketing efforts towards customers who have already shown interest in their business.

**SELECT** **DISTINCT** FirstName**,** LastName

**FROM** MEMBER

**INNER** **JOIN** ORDERR **ON** MEMBER**.**MemberID **=** ORDERR**.**MemberID

**WHERE** ORDERR**.**RestaurantID **=** 4**;**

Graphical user interface, text, application, email

Description automatically generated

1. Retrieve the names and prices of all meals that are offered by a specific type of restaurant.

This query is important for customers looking for a specific type of restaurant and meal.

**SELECT** MealName**,** Price

**FROM** MEAL

**INNER** **JOIN** RESTAURANT **ON** MEAL**.**RestaurantID **=** RESTAURANT**.**RestaurantID

**INNER** **JOIN** RESTAURANT\_TYPE **ON** RESTAURANT**.**RestaurantTypeID **=** RESTAURANT\_TYPE**.**RestaurantTypeID

**WHERE** RESTAURANT\_TYPE**.**RestaurantType **=** 'American'**;**

Graphical user interface, text, application, email

Description automatically generated

1. Retrieve the names of all cities that have at least one restaurant.

This query is important for customers to easily identify which cities have restaurants they can visit.

**SELECT** MealName**,** Price

**FROM** MEAL

**INNER** **JOIN** ITEM\_TYPE **ON** MEAL**.**ItemTypeID **=** ITEM\_TYPE**.**ItemTypeID

**WHERE** ITEM\_TYPE**.**ItemType **=** 'Desert'**;**

Graphical user interface

Description automatically generated

1. Retrieve the income percentage earned by a specific restaurant.

This query is important for the restaurant to track their earnings and to plan for future investments.

**SELECT** MealName**,** Price

**FROM** MEAL

**INNER** **JOIN** MEAL\_TYPE **ON** MEAL**.**MealTypeID **=** MEAL\_TYPE**.**MealTypeID

**INNER** **JOIN** RESTAURANT **ON** MEAL**.**RestaurantID **=** RESTAURANT**.**RestaurantID

**INNER** **JOIN** RESTAURANT\_TYPE **ON** RESTAURANT**.**RestaurantTypeID **=** RESTAURANT\_TYPE**.**RestaurantTypeID

**WHERE** MEAL\_TYPE**.**MealType **=** 'Vegan' **AND** RESTAURANT\_TYPE**.**RestaurantType **=** 'American'**;**

Graphical user interface, text, application, Word, email

Description automatically generated

1. Retrieve the total count of meals ordered, the total expense, and the average price per meal for a specific restaurant type and meal type, for a given city.

This query is important for restaurants to analyze their sales performance by meal type and to identify popular menu items.

**SELECT** MEAL\_TYPE**.**MealType**,** MEAL**.**MealName**,** **COUNT(DISTINCT** ORDER\_DETAIL**.**MealID**)** **AS** MealCount**,** **SUM(**Price**)** **AS** TotalExpense**,** **AVG(**Price**)** **AS** AvgPrice

**FROM** MEAL

**INNER** **JOIN** MEAL\_TYPE **ON** MEAL**.**MealTypeID **=** MEAL\_TYPE**.**MealTypeID

**INNER** **JOIN** ITEM\_TYPE **ON** MEAL**.**ItemTypeID **=** ITEM\_TYPE**.**ItemTypeID

**INNER** **JOIN** RESTAURANT **ON** MEAL**.**RestaurantID **=** RESTAURANT**.**RestaurantID

**INNER** **JOIN** RESTAURANT\_TYPE **ON** RESTAURANT**.**RestaurantTypeID **=** RESTAURANT\_TYPE**.**RestaurantTypeID

**INNER** **JOIN** CITY **ON** RESTAURANT**.**CityID **=** CITY**.**CityID

**INNER** **JOIN** ORDER\_DETAIL **ON** MEAL**.**MealID **=** ORDER\_DETAIL**.**MealID

**INNER** **JOIN** ORDERR **ON** ORDER\_DETAIL**.**OrderID **=** ORDERR**.**OrderID

**WHERE** CITY**.**City **=** 'Ramat Hasharon' **AND** RESTAURANT\_TYPE**.**RestaurantType **=** 'American' **AND** MEAL\_TYPE**.**MealType **=** 'Vegan'

**GROUP** **BY** MEAL\_TYPE**.**MealType**,** MEAL**.**MealName**;**

Graphical user interface, text, application

Description automatically generated

1. **Discuss the real-world significance of this database. You will need to research and understand the purposes of customer relationship management systems. Then, based on your general knowledge of the application domain, and your research, identify what additional constructs should be included in the database and state how they would be useful. This question requires well-conceived and well-written arguments that reflect the role of data and its analysis in a company.**

Databases & Customer relationship management systems are critical to the success of organizations in today's data-driven world. By providing a centralized location for data, enabling efficient decision-making, and improving collaboration and efficiency, databases & CRM can help organizations achieve their goals and drive growth and innovation. Below are some of the real-world significances for the online ordering startup database:

1. Member Data Management: The CRM system will provide a centralized location for managing member data. This includes contact information, order history, preferences, and other meal ordering attributes. The system will allow the startup to segment members based on different criteria such as frequency of orders, preferred restaurants, and cuisine preferences.

Improved Data Management will improve data accuracy, accessibility, and security. By having a structured and organized database, the Client can manage data more efficiently and effectively.

1. Analytics and Insights: The CRM system will provide analytics and insights into member behavior. This includes data on order patterns, frequency of orders, and preferred restaurants. The CRM will also provide insights on which promotions and offers are most effective in increasing member engagement and retention.
2. Efficient Decision-Making: It will be quick and easy to find patterns and perform trend analysis especially when the data set is very large because of 30 restaurants in 5 cities. This can enable Client to make informed decisions based on real-time data, rather than relying on intuition or incomplete information. By having access to accurate and up-to-date data, better decisions can be made quicker.
3. Marketing Automation: It will be more effective for the startup to automate marketing campaigns based on member data. This includes personalized promotions and offers based on member preferences and behavior. The system will also provide tools for email marketing, social media marketing, and SMS marketing.
4. Loyalty Program Management: The Client or the member management team will be able to manage its loyalty program. This includes tracking member activities, calculating rewards, and managing reward redemption. It will become more effective to find patterns that which rewards are most effective in increasing member engagement and retention. This will help to increase member satisfaction and encourage them to continue ordering from the platform.
5. Personalized Customer Experience: Member management team can analyze customer data based on the orders and the behavioral patterns, which will facilitate Client to provide personalized experiences for their customers, such as targeted marketing campaigns, personalized recommendations, and customized meals and discounts.
6. Real-Time Communication: The database can provide real-time updates to members on their order status, delivery time, and any changes to the menu or restaurant availability. This can improve communication between the online ordering platform, participating restaurants, and delivery partners, and provide a better customer experience.
7. Scalability: As the startup continues to expand its operations, the database can help to ensure scalability. The startup can use the database to track member activities across multiple cities and restaurants, which can provide insights into which locations and restaurants are most popular among members. This can help the startup to make informed decisions on where to expand its operations and which restaurants to partner with.

These real-world applications will help online meal ordering startup to manage member data, analyze customer behavior, and improve customer engagement. Thus, the startup can provide a better customer experience and increase customer retention.

Below are the additional constructs which should be included in the database to achieve more refined and sophisticated results:

1. Order Status: This attribute would track the status of each member's order, including whether it is being prepared, in transit, or has been delivered. And this functionality can be clubbed in the App for the customer to see such status using the mobile phone. This information would be useful in providing real-time updates to members and improving communication between the online ordering platform and the participating restaurants and delivery partners.
2. Special instructions: This will help customers to provide specific or additional instructions on how their meal should d be prepared.
3. Feedback and Ratings: This attribute would allow customers to provide feedback and ratings on their ordering experience, the participating restaurants, and delivery partners. This information can be used to improve the quality of the online ordering platform and identify areas for improvement.
4. Dietary Restrictions: This attribute would allow members to specify any dietary restrictions or preferences, such as vegetarian, gluten-free, or low-carb. This information can be used to provide personalized menu recommendations and ensure that members receive appropriate menu items.
5. Inventory Data: The database can include data on restaurant inventory levels, such as the availability of menu items and ingredients. This data can be used to manage inventory and reduce waste, as well as to inform members of menu item availability and substitutions.
6. Referral Source: This attribute would track how each member found out about the online ordering platform, such as through social media, advertising, or word-of-mouth. This information can be used to identify the most effective marketing channels and optimize marketing campaigns.

Overall, these additional attributes can provide valuable insights into the behavior and preferences of customers and improve the quality and personalization of the online ordering platform. By including these attributes in the database, the online meal ordering startup can better understand its customers, provide a better customer experience, and increase customer engagement and retention.