CS 4-5543 Database Systems Project **Submitted by: Nasrin Sultana Nipa** **Student ID: 50618423** **Submission Date: 11/21/2023**	**Final project report : It includes the contents/requirements from -** **Phase I** • Project description • Functionalities (that user can use for this project) • E/R diagrams **Phase II** Database schema and relationships **Phase III** Implementation (in PhaseIII.sql file)

Project Overview: RestoWave - Elevating the Dining Experience

INTRODUCTION:

The RESTOWAVE project is a comprehensive Restaurant Management System designed to streamline various aspects of restaurant operations. It provides a centralized platform for managing customers, dishes, restaurants, and their associated data. The system facilitates efficient restaurant management and enhances the overall dining experience for both customers and staff.

KEY FUNCTIONALITIES:

1. Restaurant Management:

- Track essential restaurant information such as name, address, phone number, and operating hours.

- Assign managers to restaurants using the ManagerRunsRestaurant relation.

- Manage and update restaurant menus using the RestaurantServesDish relation, including dish prices.

2. Customer Interaction:

- Maintain customer profiles with details such as name, contact information, and favorite dishes.

- Track customer visits to different restaurants and their favorite dishes using CustomerFrequentVisits and CustomerLikesDish relations.

5. Social Connections:

- Identify married couples and buddies among customers using the MarriedCouples and Buddies relations.

- Enhance the social experience for customers by considering their connections.

3. Dish Information:

- Store comprehensive information about each dish, including name, description, and calorie count.

- Establish relationships with customers who like specific dishes through the CustomerLikesDish relation.

4. Manager Details:

- Keep track of manager details, including manager ID, name, and contact information.

- Assign managers to restaurants through the ManagerRunsRestaurant relation.

5. Operational Hours:

- Record the operational hours of each restaurant for accurate scheduling and planning.

- Ensure proper opening and closing times on specific days of the week.

6. Customer Preferences:

- Record dishes that customers like and their favorite dishes.

- Capture data on frequent customer visits to restaurants.

8. Transactional Support:

- Execute transactions to update, insert, or delete data, ensuring database consistency using foreign key constraints.

- Utilize stored procedures for specific data retrieval tasks.

9. View Creation:

- Create views for simplified data presentation, such as customer-favorite dish associations.

- Enhance user accessibility to relevant data subsets.

10. Triggers and Stored Procedures:

- Implement triggers to automate specific actions, such as updating a customer's favorite dish.

- Develop stored procedures for custom data processing tasks.

These key functionalities collectively contribute to the efficient management of restaurant-related data and customer interactions within the "RESTOWAVE" database.

RELATIONSHIPS BETWEEN THE ENTITIES:

1. Restaurant-Manager Relationship (Many-to-One):

- Each restaurant (one) is associated with a manager (one), but a manager can oversee multiple restaurants. Therefore, it's a many-to-one relationship.

2. Customer-Favorite Dish Relationship (Many-to-One):

- A customer (one) can have a favorite dish (one), but a particular dish may be the favorite for multiple customers. This forms a many-to-one relationship.

3. Customer-Liked-Dish Relationship (Many-to-Many):

- A customer (many) can like multiple dishes, and a dish (many) can be liked by multiple customers. This forms another many-to-many relationship.

4. Manager-Restaurant Relationship (One-to-Many):

- A manager (one) can oversee multiple restaurants (many), but a restaurant is managed by exactly one manager. This creates a one-to-many relationship.

5. Restaurant-Dish Relationship (Many-to-Many):

- A restaurant (one) can serve multiple dishes (many), and that dish can be found in other restaurants also. This establishes a many-to-many relationship.

6. Customer-Frequent-Visits Relationship (Many-to-Many):

- A customer (many) can frequently visit multiple restaurants, and a restaurant (many) can be frequently visited by multiple customers. This forms another many-to-many relationship.

7. Married-Couples Relationship (one-to-one):

- The MarriedCouples relation represents a one-to-one relationship, as each customer in a couple can be connected to just his or her spouse.

8. Buddies Relationship (Many-to-Many):

- Similarly, the Buddies relation signifies a many-to-many relationship, as each customer in a buddy pair can be part of multiple buddy connections.

These relationship descriptions provide insights into how the entities in your database schema are interconnected and the cardinality of those connections.

Project Benefits:

Efficient Restaurant Management: The system streamlines restaurant operations by providing a centralized platform for managing key data.

Enhanced Customer Experience: Personalized recommendations and social connections enhance the overall dining experience for customers.

Data Integrity: Relationships and constraints ensure data integrity, preventing inconsistencies in the database.

Entity-Relationship (E/R) Diagram :

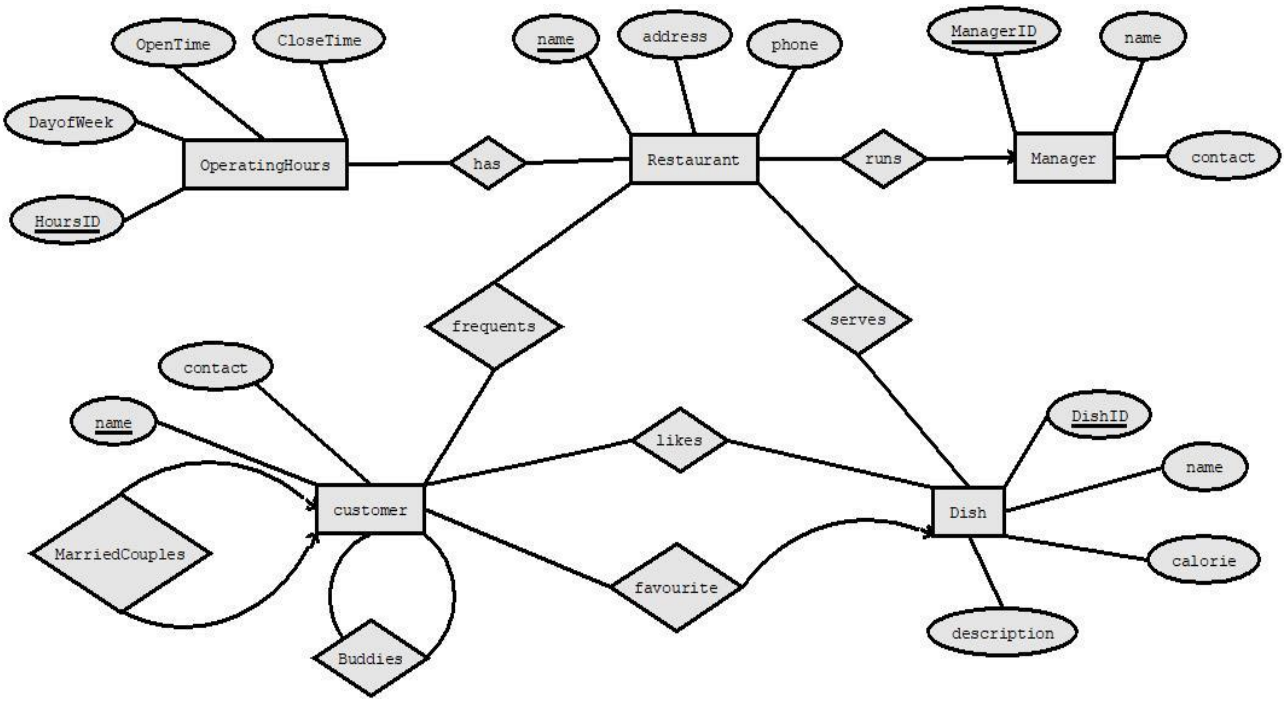

Figure: E/R diagram of RestoWave Database System using Dia

E/R Diagrams to relations:

1) Restaurant (Name, Address, Phone, ManagerID(FK) (References Manager))
2) Customer (Name, Contact, FavouriteDishID (FK) (References Dish))
3) Manager (ManagerID, Name, Contact)
4) Dish (DishID, Name, Description, Calorie)

5) OperatingHours (HoursID, DayofWeek, OpenTime, CloseTime, RestaurantName (FK)) - RestaurantName (References Restaurant)

6) ManagerRunsRestaurant (RestaurantName, ManagerID) [composite primary key]
- RestaurantName (References Restaurant)
- ManagerID (References Manager)

7) RestaurantServesDish (RestaurantName, DishID, Price) [composite primary key]
- RestaurantName (References Restaurant)
- DishID (References Dish)

8) CustomerFrequentVisits (CustomerName, RestaurantName, DishID) [composite primary key]
- CustomerName (References Customer)
- RestaurantName (References Restaurant)
- DishID (References Dish)

9) CustomerLikesDish (CustomerName, DishID) [composite primary key]
- CustomerName (References Customer)
- DishID (References Dish)

10) CustomerFavouritesDish (CustomerName, DishID) [composite primary key]
- CustomerName (References Customer)
- DishID (References Dish)

11) MarriedCouples (CustomerName, CustomerName) [composite primary key]
- CustomerName1 (References Customer)
- CustomerName2 (References Customer)

12) Buddies (CustomerName, CustomerName) [composite primary key]
- CustomerName1 (References Customer)
- CustomerName2 (References Customer)

Conclusion:

RESTOWAVE is designed to be a user-friendly and efficient solution for modern restaurant management, catering to the diverse needs of both customers and restaurant staff.