ONLINE ORDERING FOR UNIVERSITY OF WASHINGTON DINING

University of Washington's campus dining services is the main business or organization used for the database application being created. The university has multiple dining locations with a variety of different cuisines across the campus. Currently, the customers can go to the dining locations, stand in queue, and order their food. If the preparation time is longer then the customers must wait till the time the food the prepared.

The goal of the project is to create a database that mainly focuses on providing information about all the restaurants on the campus allowing customers to order online. Students, faculty and all the customers can order ahead and save time by browsing menus, placing orders, and paying online. The customers have an option to pick up orders without waiting in the queue. Multiple payment methods such as dining account, Husky card or credit card would be beneficial. Overall, the aim is to enhance the dining experience on campus by providing convenient ordering process.

Revised entities that were identified -

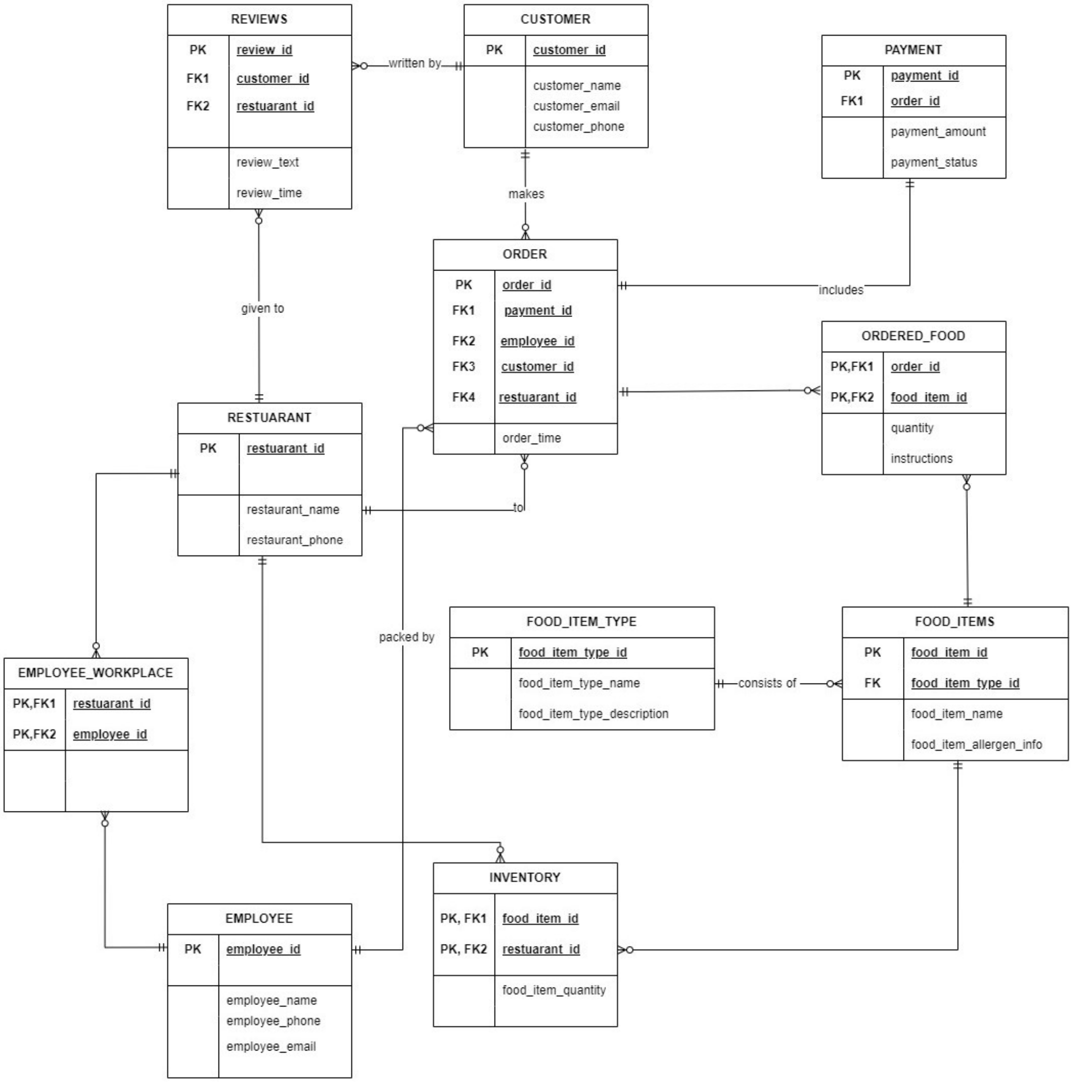
CUSTOMER	Represents individuals who visit or order from the restaurant.
FOOD_ITEMS	Represents the various dishes and items available on the restaurant's menu.
ORDER	Represents information about specific customer orders.
RESTAURANT	Represents the individual restaurants or dining locations.
INVENTORY	Represents the many-to-many relationship between restaurants and food items.
PAYMENT	Represents the various methods by which customers can make payments for their orders.
REVIEWS	Stores customer feedback and ratings for both the restaurant and specific food items.
FOOD_ITEM_TYPE	The type of food item for example beverage, fast food, etc.
EMPLOYEE	Represents the staff working at the restaurant.
EMPLOYEE_WORKPLACE	Represents the many-to-many relationship between restaurants and employees.
ORDERED_FOOD	Represents the many-to-many relationship between order and food items.

Revised Business Rules –

Each customer can place multiple orders.
One order can be placed by one customer.
Each customer can give multiple reviews.
Each review can be given by one customer.
Multiple food items can be ordered in one order.
Multiple orders can consist of one food item.
Multiple orders can be placed in one restaurant.
One order can be placed in one restaurant.
Each order can have one payment.

One payment will be linked to an order.
One restaurant can have multiple reviews.
Each review will be associated with one restaurant.
One food item type would have multiple food items.
Each food item would be a of a food item type.
One employee can handle many orders.
Each order will be handled by one employee.
Each employee can work in many restaurants.
Each restaurant can hire at many employees.
Each restaurant can have many food items.
Each food item can be part of many restaurants.

Revised ER Diagram –



1. Why are these changes necessary?

Changing the relationship of restaurant and employee from one to many to many to many was necessary as employees can work at many restaurants. It enables us to accommodate evolving business requirements and future changes in processes without necessitating significant alterations to the underlying structure.

2. What benefits will your stakeholders have because of these changes?

This design accommodates more use cases and clearly explains the functioning of the system. The primary keys and foreign keys will help the stakeholders better understand the dynamics between the entities.

3. What were the underlying best practices that you were following? In other words, how was the ERD better able to accommodate the business needs of the organization (both current as well in the future)?

I aimed to minimize data redundancy and have data integrity throughout the database structure. This involved having proper many-to-many relationships and distinguishing between entities and attributes. It provides a solid foundation for accommodating future adjustments to business processes without compromising the overall structure or functionality of the database.

4. Introduce a few concepts related the Enhanced ERD such as Supertype/Subtype, Generalization, Specialization etc.

The revisions to the ERD focused on fundamental principles of relational database design. The emphasis was on enhancing data integrity by accurately defining entities, relationships, and characteristics within the ERD. We did not need any enhanced ERD such as Supertype/Subtype or Generalization for this use case.