# UTD Dining and Meal Plan System

## Final Phase

### Course Title: Database Foundations for Business Analytics (BUAN 6320.004)

### Semester Project Group- 8

**Group Members:**

Jagdish Umashankar Vishwakarma (juv230000@utdallas.edu)

Uchaansh Lodhi (uxl230001@utdallas.edu)

Sonakshi Handoo (sxh230044@utdallas.edu)

Aditya Chaganti (vxc230016@utdallas.edu)

Utsav Mukesh Maisheri (uxm230000@utdallas.edu)

# Abstract

This project report outlines developing and implementing a SQL-based Dining and Meal Plan System for The University of Texas at Dallas. The current dining system at UTD lacks a centralized platform, resulting in inefficiencies in meal planning, underutilization of meal plans, and insufficient accommodation of dietary preferences. The proposed system addresses these challenges by providing a comprehensive solution for students and administrators.

The UTD dining and meal plan system, built on SQL, offers a centralized database for managing meal balances, dietary preferences, and transaction histories. This platform enables students to easily track and manage their meal plans, switch plans if desired, and receive personalized food recommendations. Simultaneously, administrators gain tools to manage dining options and meal subscriptions and analyze real-time data to understand student preferences and demand.

The project emphasizes the creation of several database entities, including Students, MealPlans, DiningHalls, Menus, MenuItems, DietaryPreferences, StudentTransactions, Feedback, and Coupons. Each entity is crucial in providing a seamless and personalized dining experience.

Team responsibilities have been equitably distributed, ensuring effective collaboration and timely completion of tasks. The objective is to enhance the efficiency and personalization of the dining experience at UTD, promoting a smoother interaction between students and the university's dining services.

## Content

# Phase I – Proposal

**Narrative Description:** Our proposed database project is to design and implement a system for the University of Texas at Dallas' (UTD) Dining and Meal Plan to make eating on campus easier and more efficient for its students. Right now, it's a bit tricky for students to know how many meals they have left, change their meal plan, or even find foods that fit their dietary needs. Our project i.e., UTD dining and meal plan system based on SQL, which is a type of computer language to manage information, will help to solve this problem.

With this new system, everything about campus dining will be in one place. Students can easily see and manage their meal balances, pick a different plan if they want, and even get suggestions on where to find certain foods. This means no more guessing or wasting time figuring things out. The idea is to make the dining experience at UTD smoother and more personalized for everyone. The system will help administrators manage available dining options, meal plan subscriptions, and student/staff dining transactions.

**Problem/Opportunity:** The current UTD dining system lacks a centralized platform, leading to inefficiencies in meal planning, underutilization of meal plans, and inadequate addressing of dietary preferences. Implementing a SQL-based Dining and Meal Plan System offers an opportunity to streamline meal plan management, optimize food preparation based on real-time demand, and cater to diverse dietary needs, enhancing the overall dining experience for students.

**Information Needs**

To solve this problem and capitalize on this opportunity, our system would need:

Detailed data on different dining options available at UTD.
Real-time tracking of meal balances for students.
Dietary preference and restriction information for each student.
Feedback mechanism for students to rate and review dining experiences.
Analysis tools for the administration to understand demand and preferences.

**Initial List of Entities (Tables)**

**Students:** Records of students personal information, meal plan, dietary preferences, and balance.
**MealPlans:** Details of various meal plans, including plan name, price, and meals per day.

**DiningHalls**: Information about dining halls, such as name, location, opening and closing times.

**Menus:** Records of menus offered at dining halls, including the date and meal type.

**MenuItems:** Detailed information about menu items, including dish name, ingredients, calories.

**DietaryPreferences:** Various dietary preferences with names and descriptions.

**StudentTransactions:** Transaction history of students, including which dining hall, menu item, and date of transaction.

**Feedback:** Feedback provided by students, including content, rating, and date.
**Coupon**: Different types of coupon code on different meal plan or combo meal.

**Responsibilities:** The responsibilities for this group project have been equally distributed among the team members with the goal of ensuring hard work and proper efforts. Not to mention, the group has ensured to consistently communicate with each other and adhere to the deadlines by taking care of it ahead of time.

Phase II – Design and Modelling

## Content

## 1. Executive Summary

Our proposed database project is to design and implement a system for the University of Texas at Dallas' (UTD) Dining and Meal Plan to make eating on campus easier and more efficient for its students. Right now, it's tricky for students to know how many meals they have left, change their meal plan, or even find foods that fit their dietary needs. Our project, i.e., UTD dining and meal plan system based on SQL, a type of computer language to manage information, will help solve this problem. With this new system, everything about campus dining will be in one place. Students can easily see and manage their meal balances, pick a different plan, and even get suggestions for finding certain foods. This means no more guessing or wasting time figuring things out. The idea is to make the dining experience at UTD smoother and more personalized for everyone. The system will help administrators manage available dining options, meal plan subscriptions, and student/staff dining transactions.

## 2. Problem Description

The current UTD dining system lacks a centralized platform, leading to inefficiencies in meal planning, underutilization, and inadequate addressing of dietary preferences. Implementing a SQL-based Dining and Meal Plan System offers an opportunity to streamline meal plan management, optimize food preparation based on real-time demand, and cater to diverse dietary needs, enhancing the overall dining experience for students.

## 3. Conceptual Design

Here is the EER diagram generated based on our project description and real-life experiences.

### 3.1 EER diagram with all assumptions

**ER Diagram Link:**

**https://app.diagrams.net/#G12149uuTS4q8384m_hZavfXBz7Pn4AXER**

**(Click on the link and select "Diagrams.net" option to view the ER diagram clearly )**

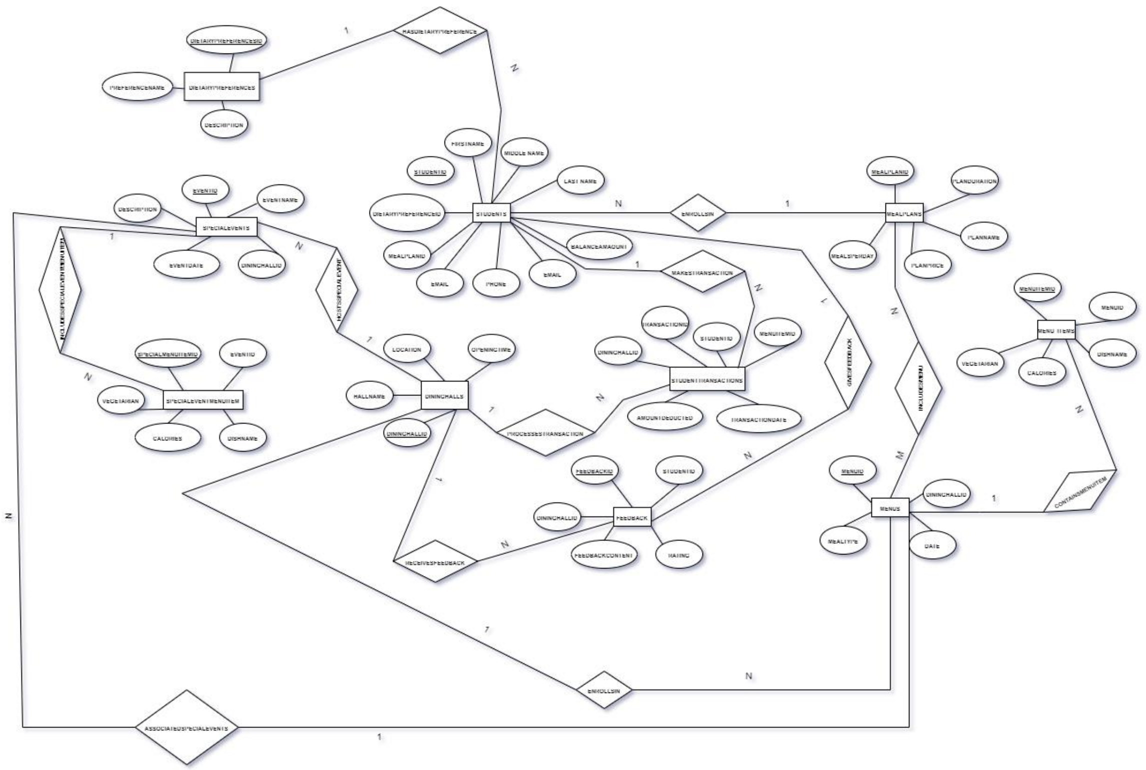


Figure 1. EER Diagram

### 3.2 (Min, Max) Notation for Relationship

In this part we discuss the (min, max) notation for several important relationships that exist in our EER diagram. Table 1 clearly specifies how the numerical expression corresponds to the relationship between two entities.

Table 1. Explanation for (Min, Max) Notation

| Relationship | Cardinality | Explanation |
|---|---|---|
| Students - MealPlans | Students(0, 1) to (0, N) MealPlans | (One student can have one meal plan, and one meal plan is associated with many student.) |
| Students - DietaryPreferences | Students(0, 1) to (1, N)DietaryPreferences | (A student may not have a dietary preference, but if they do, they have exactly one dietary preference.<br>One DietararyPreference is associated with one or many students) |
| Students - StudentTransactions: | Students (1, N) to (1, 1) StudentTransactions | (A student must have at least one transaction but can have multiple transactions.<br>A transaction is associated with only one student.) |
| Students - Feedback: | Students (1, N) to (1,1) Feedback | (A student must provide at least one feedback but can provide multiple feedback entries.<br>but a feedback entry is associated with only one student.) |

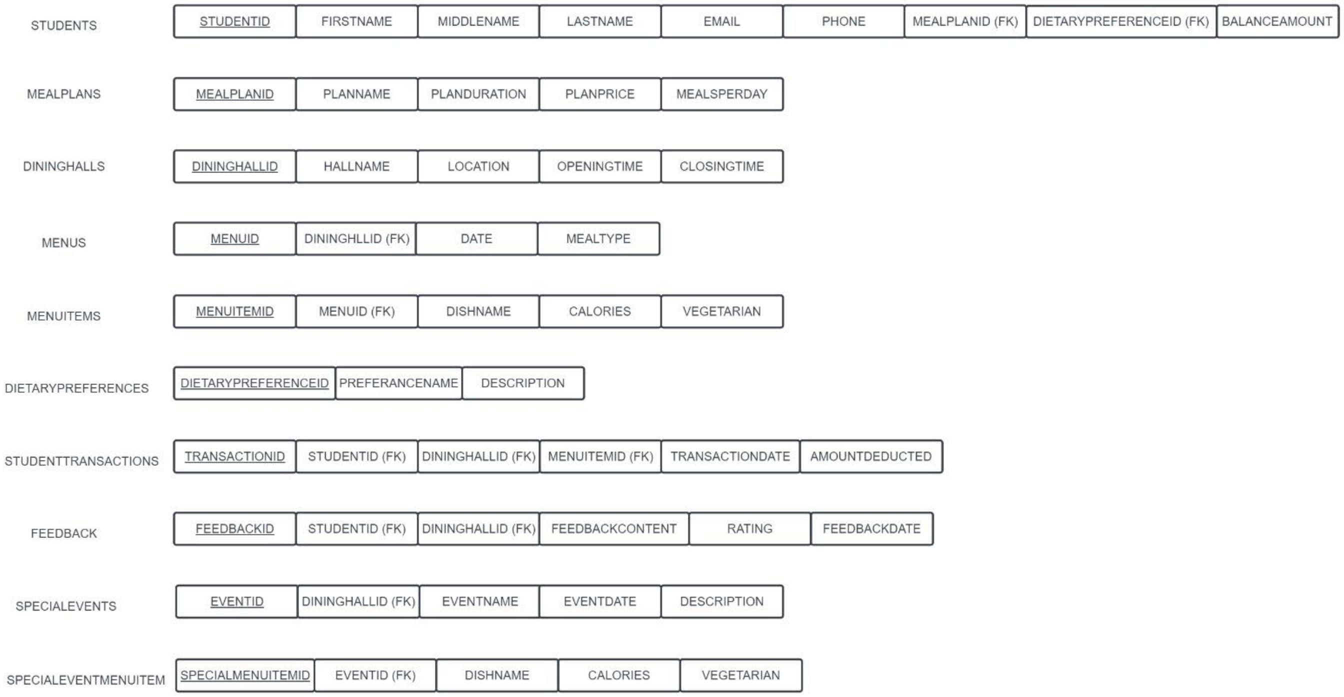
| | | |
|---|---|---|
| MealPlans - Menus: | MealPlans (1, N) to (1, M) Menus | (A meal plan may have one menu, or it can have multiple menus, and a menu can be associated with one or multiple meal plans.) |
| DiningHalls - Menus | DiningHalls (1, N) to (1, 1) Menus | (A dining hall must have at least one menu, but can have multiple menus, and a menu is associated with exactly one dining hall.) |
| Menus - MenuItems: | Menus (1, N) to (1, N) MenuItems | (A menu must have at least one menu item, but can have multiple menu items, and a menu item is associated with one or many menu.) |
| DiningHalls - StudentTransactions: | DiningHalls (1, N) to (1,1) StudentTransactions | (A dining hall must have at least one transaction, but can have multiple transactions, and a transaction is associated with exactly one dining hall.) |
| DiningHalls - Feedback: | DiningHalls (1, N) to (1,1) Feedback | (A dining hall must have at least one feedback entry, but can have multiple entries, and a feedback entry is associated with exactly one dining hall.) |
| DiningHalls - SpecialEvents: | DiningHalls (1, N) to (1,1) SpecialEvents | (A dining hall must have at least one special event, but can have multiple events, and a special event is associated with exactly one dining hall.) |

| | | |
|---|---|---|
| SpecialEvents - SpecialEventMenuItems: | SpecialEvents (1, N) to (1, N) SpecialEventMenuItems | (A special event must have at least one menu item, but can have multiple items, and a menu item is associated with one or many special event.) |
| Menus - SpecialEvents: | Menus (0, N) to (0, N) SpecialEvents | (A menu be associated with zero or more special events, and a special event is associated with zero or more menus.) |

# 4. Relational Schema

## 4.1 Relational Schema

Our Relational Schema has no weak entities



| | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| STUDENTS | STUDENTID | FIRSTNAME | MIDDLENAME | LASTNAME | EMAIL | PHONE | MEALPLANID (FK) | DIETARYPREFERENCEID (FK) | BALANCEAMOUNT |
| MEALPLANS | MEALPLANID | PLANNAME | PLANDURATION | PLANPRICE | MEALSPERDAY | | | | |
| DININGHALLS | DININGHALLID | HALLNAME | LOCATION | OPENINGTIME | CLOSINGTIME | | | | |
| MENUS | MENUID | DININGHLLID (FK) | DATE | MEALTYPE | | | | | |
| MENUITEMS | MENUITEMID | MENUID (FK) | DISHNAME | CALORIES | VEGETARIAN | | | | |
| DIETARYPREFERENCES | DIETARYPREFERENCEID | PREFERANCENAME | DESCRIPTION | | | | | | |
| STUDENTTRANSACTIONS | TRANSACTIONID | STUDENTID (FK) | DININGHALLID (FK) | MENUITEMID (FK) | TRANSACTIONDATE | AMOUNTDEDUCTED | | | |
| FEEDBACK | FEEDBACKID | STUDENTID (FK) | DININGHALLID (FK) | FEEDBACKCONTENT | RATING | FEEDBACKDATE | | | |
| SPECIALEVENTS | EVENTID | DININGHALLID (FK) | EVENTNAME | EVENTDATE | DESCRIPTION | | | | |
| SPECIALEVENTMENUITEM | SPECIALMENUITEMID | EVENTID (FK) | DISHNAME | CALORIES | VEGETARIAN | | | | |

Figure 2: Relational Schema

## 4.2 Data Format for Every Relation

Table 2. Data Format for Each Relation

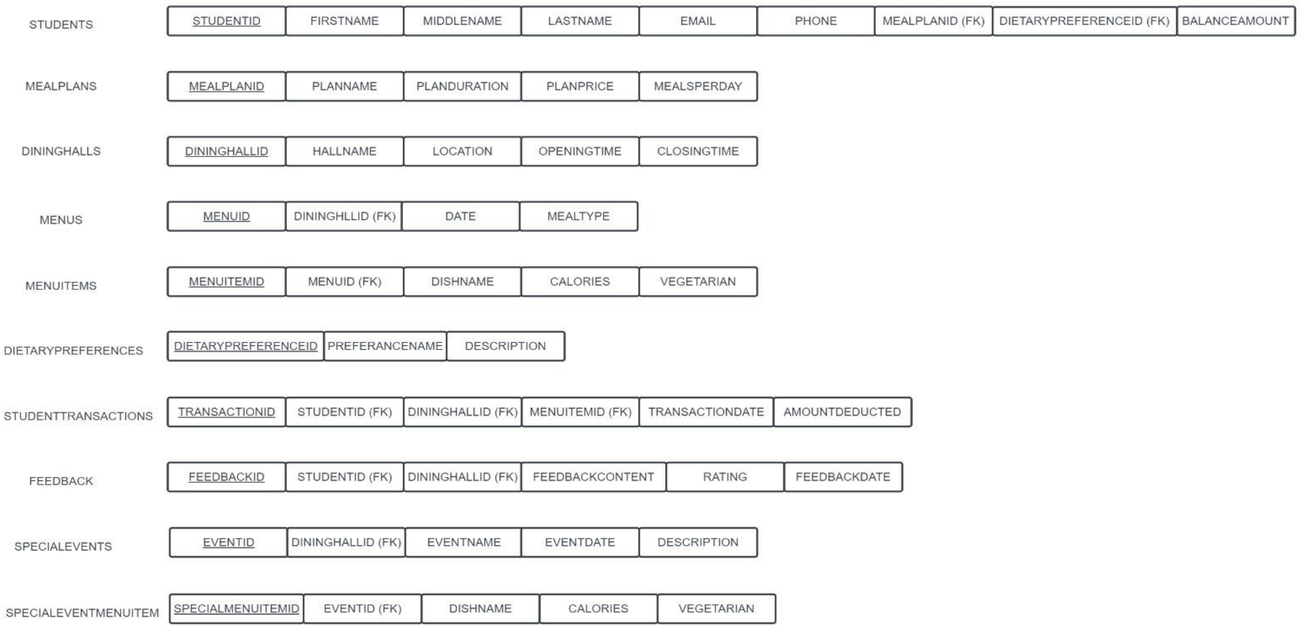
| **Relation Names** | **Attributes** | **Data Type** | **Length (Max) or Size** |
|---|---|---|---|
| Students | StudentID (PK) | String (Primary Key) | 10 |
| | FullName | String | 50 |
| | Email | String | 255 |
| | Phone | String | 12 |
| | MealPlanID (FK) | String (Foreign Key) | 10 |
| | DietaryPreferenceID (FK) | String (Foreign Key) | 10 |
| | BalanceAmount | Decimal | - |
| MealPlans | MealPlanID (PK) | String (Primary Key) | 10 |
| | PlanName | String | 50 |
| | PlanDuration | String | 20 |
| | PlanPrice | Decimal | - |
| | MealsPerDay | Integer | - |
| DiningHalls | DiningHallID (PK) | String (Primary Key) | 10 |

| | | | |
|---|---|---|---|
| | HallName | String | 50 |
| | Location | String | 50 |
| | OpeningTime | Time | - |
| | ClosingTime | Time | - |
| Menus | MenuID (PK) | String (Primary Key) | 10 |
| | DiningHallID (FK) | String (Foreign Key) | 10 |
| | Date | Date | - |
| | MealType | String | 20 |
| MenuItems | MenuItemID (PK) | String (Primary Key) | 10 |
| | MenuID (FK) | String (Foreign Key) | 10 |
| | DishName | String | 50 |
| | | | |
| | Calories | Decimal | - |
| | IsVegetarian | Boolean | - |
| DietaryPreferences | DietaryPreferenceID (PK) | String (Primary Key) | 10 |
| | PreferenceName | String | 50 |
| | Description | String | - |
| StudentTransactions | TransactionID (PK) | String (Primary Key) | 10 |
| | StudentID (FK) | String (Foreign Key) | 10 |
| | DiningHallID (FK) | String (Foreign Key) | 10 |
| | MenuItemID (FK) | String (Foreign Key) | 10 |
| | TransactionDate | Date | - |
| | AmountDeducted | Decimal | - |

| | | | |
|---|---|---|---|
| Feedback | FeedbackID (PK) | String (Primary Key) | 10 |
| | StudentID (FK) | String (Foreign Key) | 10 |
| | DiningHallID (FK) | String (Foreign Key) | 10 |
| | FeedbackContent | String | - |
| | Rating | Integer | - |
| | FeedbackDate | Date | - |
| SpecialEvents | EventID (PK) | String (Primary Key) | 10 |
| | DiningHallID (FK) | String (Foreign Key) | 10 |
| | EventName | String | 50 |
| | EventDate | Date | - |
| | Description | String | - |
| SpecialEventMenuItems | SpecialMenuItemID (PK) | String (Primary Key) | 10 |
| | EventID (FK) | String (Foreign Key) | 10 |
| | DishName | String | 50 |
| | | | |
| | Calories | Decimal | - |
| | IsVegetarian | Boolean | - |

# 5. Normalization

In this part, we apply the principles of normalization to ensure all the tables conform to 3NF. To do this, we document all functional dependencies and indicate how the normalization is performed.



| | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| STUDENTS | STUDENTID | FIRSTNAME | MIDDLENAME | LASTNAME | EMAIL | PHONE | MEALPLANID (FK) | DIETARYPREFERENCEID (FK) | BALANCEAMOUNT |
| MEALPLANS | MEALPLANID | PLANNAME | PLANDURATION | PLANPRICE | MEALSPERDAY | | | | |
| DININGHALLS | DININGHALLID | HALLNAME | LOCATION | OPENINGTIME | CLOSINGTIME | | | | |
| MENUS | MENUID | DININGHLLID (FK) | DATE | MEALTYPE | | | | | |
| MENUITEMS | MENUITEMID | MENUID (FK) | DISHNAME | CALORIES | VEGETARIAN | | | | |
| DIETARYPREFERENCES | DIETARYPREFERENCEID | PREFERANCENAME | DESCRIPTION | | | | | | |
| STUDENTTRANSACTIONS | TRANSACTIONID | STUDENTID (FK) | DININGHALLID (FK) | MENUITEMID (FK) | TRANSACTIONDATE | AMOUNTDEDUCTED | | | |
| FEEDBACK | FEEDBACKID | STUDENTID (FK) | DININGHALLID (FK) | FEEDBACKCONTENT | RATING | FEEDBACKDATE | | | |
| SPECIALEVENTS | EVENTID | DININGHALLID (FK) | EVENTNAME | EVENTDATE | DESCRIPTION | | | | |
| SPECIALEVENTMENUITEM | SPECIALMENUITEMID | EVENTID (FK) | DISHNAME | CALORIES | VEGETARIAN | | | | |

Figure 3: Normalized Relational Schema

## 6. Conclusion

In this report, we discuss and design the relational schema of the UTD Dining Database. Our EER diagram and its associated relational schema show the conceptual and logical designs of the system. We also define data types and formats for each attribute in the relation. The next step is to implement this database. We may change some designs in the future due to practical difficulties and other requirements.

## 7. List of Figures

## 8. List of Tables

Phase III – Implementation

## Content

# 1. List of Figures

## 1.1 EER diagram with all assumptions

**ER Diagram Link:**

**[https://app.diagrams.net/#G12149uuTS4q8384m_hZavfXBz7Pn4AXER](https://app.diagrams.net/#G12149uuTS4q8384m_hZavfXBz7Pn4AXER)**

**(Click on the link and select "Diagrams.net" option to view the ER diagram clearly)**

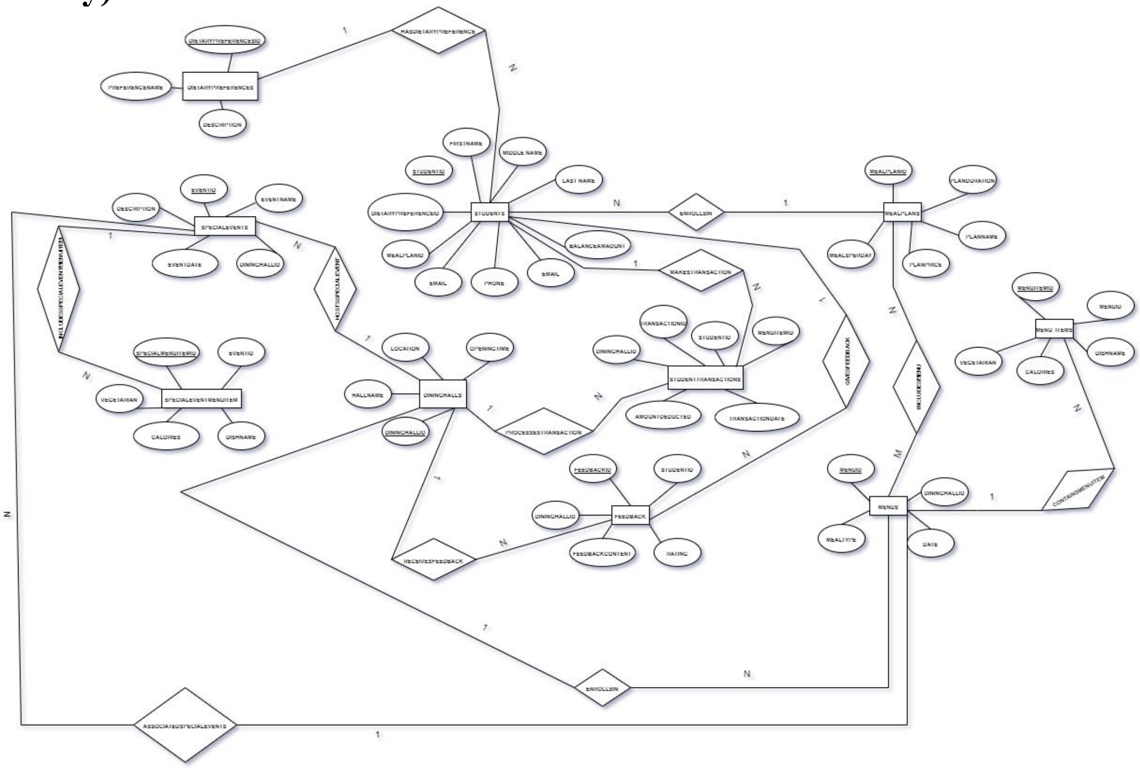
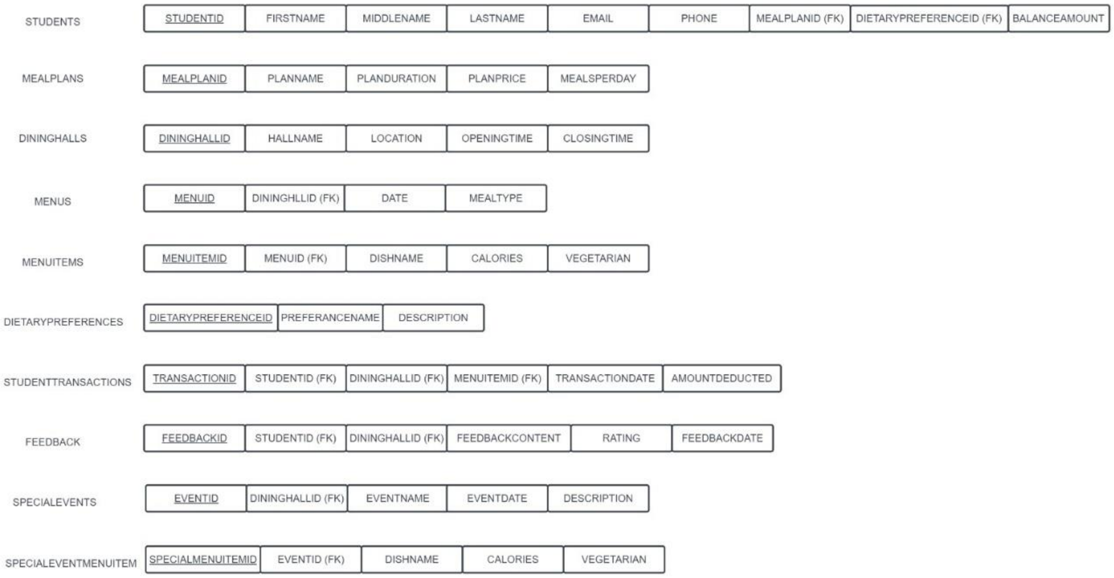


Figure 1. EER Diagram

## 1.2 Relational Schema

Our Relational Schema has no weak entities

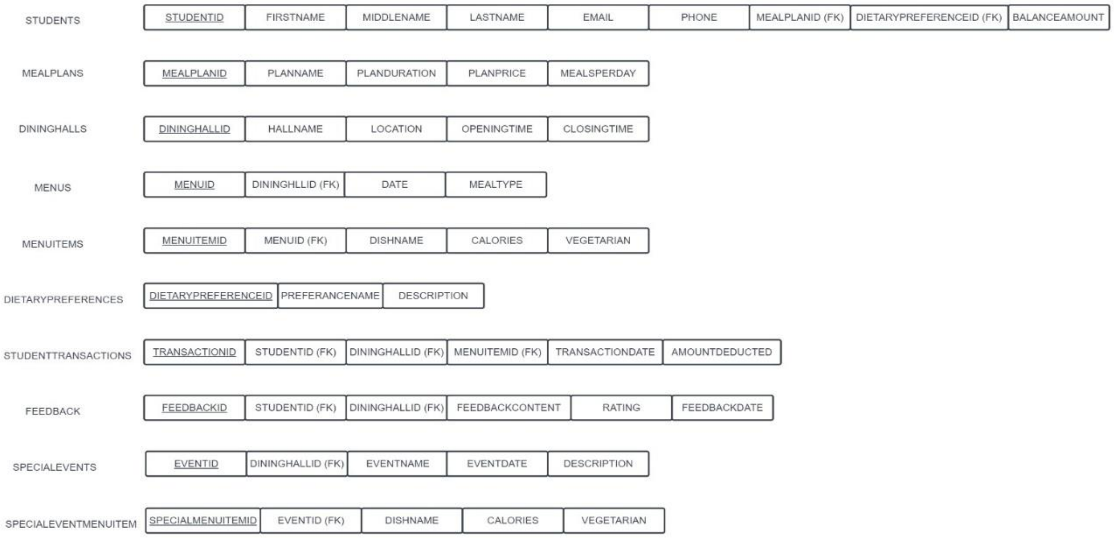


| | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| STUDENTS | STUDENTID | FIRSTNAME | MIDDLENAME | LASTNAME | EMAIL | PHONE | MEALPLANID (FK) | DIETARYPREFERENCEID (FK) | BALANCEAMOUNT |
| MEALPLANS | MEALPLANID | PLANNAME | PLANDURATION | PLANPRICE | MEALSPERDAY | | | | |
| DININGHALLS | DININGHALLID | HALLNAME | LOCATION | OPENINGTIME | CLOSINGTIME | | | | |
| MENUS | MENUID | DININGHLLID (FK) | DATE | MEALTYPE | | | | | |
| MENUITEMS | MENUITEMID | MENUID (FK) | DISHNAME | CALORIES | VEGETARIAN | | | | |
| DIETARYPREFERENCES | DIETARYPREFERENCEID | PREFERANCENAME | DESCRIPTION | | | | | | |
| STUDENTTRANSACTIONS | TRANSACTIONID | STUDENTID (FK) | DININGHALLID (FK) | MENUITEMID (FK) | TRANSACTIONDATE | AMOUNTDEDUCTED | | | |
| FEEDBACK | FEEDBACKID | STUDENTID (FK) | DININGHALLID (FK) | FEEDBACKCONTENT | RATING | FEEDBACKDATE | | | |
| SPECIALEVENTS | EVENTID | DININGHALLID (FK) | EVENTNAME | EVENTDATE | DESCRIPTION | | | | |
| SPECIALEVENTMENUITEM | SPECIALMENUITEMID | EVENTID (FK) | DISHNAME | CALORIES | VEGETARIAN | | | | |

Figure 2: Relational Schema

# 1.3. Normalization

In this part, we apply the principles of normalization to ensure all the tables conform to 3NF. To do this, we document all functional dependencies and indicate how the normalization is performed.



| | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| STUDENTS | STUDENTID | FIRSTNAME | MIDDLENAME | LASTNAME | EMAIL | PHONE | MEALPLANID (FK) | DIETARYPREFERENCEID (FK) | BALANCEAMOUNT |
| MEALPLANS | MEALPLANID | PLANNAME | PLANDURATION | PLANPRICE | MEALSPERDAY | | | | |
| DININGHALLS | DININGHALLID | HALLNAME | LOCATION | OPENINGTIME | CLOSINGTIME | | | | |
| MENUS | MENUID | DININGHLLID (FK) | DATE | MEALTYPE | | | | | |
| MENUITEMS | MENUITEMID | MENUID (FK) | DISHNAME | CALORIES | VEGETARIAN | | | | |
| DIETARYPREFERENCES | DIETARYPREFERENCEID | PREFERANCENAME | DESCRIPTION | | | | | | |
| STUDENTTRANSACTIONS | TRANSACTIONID | STUDENTID (FK) | DININGHALLID (FK) | MENUITEMID (FK) | TRANSACTIONDATE | AMOUNTDEDUCTED | | | |
| FEEDBACK | FEEDBACKID | STUDENTID (FK) | DININGHALLID (FK) | FEEDBACKCONTENT | RATING | FEEDBACKDATE | | | |
| SPECIALEVENTS | EVENTID | DININGHALLID (FK) | EVENTNAME | EVENTDATE | DESCRIPTION | | | | |
| SPECIALEVENTMENUITEM | SPECIALMENUITEMID | EVENTID (FK) | DISHNAME | CALORIES | VEGETARIAN | | | | |

Figure 3: Normalized Relational Schema

# 2. List of Tables

## 2.1 (Min, Max) Notation for Relationship

In this part we discuss the (min, max) notation for several important relationships that exist in our EER diagram. Table 1 clearly specifies how the numerical expression corresponds to the relationship between two entities.

Table 1. Explanation for (Min, Max) Notation

| Relationship | Cardinality | Explanation |
|---|---|---|
| Students - MealPlans | Students (0, 1) to (0, N) MealPlans | (One student can have one meal plan, and one meal plan is associated with many students.) |
| Students - DietaryPreferences | Students (0, 1) to (1, N)DietaryPreferences | (A student may not have a dietary preference, but if they do, they have exactly one dietary preference.<br>One DietararyPreference is associated with one or many students) |
| Students - StudentTransactions: | Students (1, N) to (1, 1) StudentTransactions | (A student must have at least one transaction but can have multiple transactions.<br>A transaction is associated with only one student.) |
| Students - Feedback: | Students (1, N) to (1,1) Feedback | (A student must provide at least one feedback but can provide multiple feedback entries.<br>but a feedback entry is associated with only one student.) |
| MealPlans - Menus: | MealPlans (1, N) to (1, M) Menus | (A meal plan may have one menu, or it can have multiple menus, and a menu can be associated with one or multiple meal plans.) |
| DiningHalls - Menus | DiningHalls (1, N) to (1, 1) Menus | (A dining hall must have at least one menu, but can have multiple menus, and a menu is associated with exactly one dining hall.) |

| | | |
|---|---|---|
| Menus - MenuItems: | Menus (1, N) to (1, N) MenuItems | (A menu must have at least one menu item, but can have multiple menu items, and a menu item is associated with one or many menus.) |
| DiningHalls - StudentTransactions: | DiningHalls (1, N) to (1,1) StudentTransactions | (A dining hall must have at least one transaction, but can have multiple transactions, and a transaction is associated with exactly one dining hall.) |
| DiningHalls - Feedback: | DiningHalls (1, N) to (1,1) Feedback | (A dining hall must have at least one feedback entry, but can have multiple entries, and a feedback entry is associated with exactly one dining hall.) |
| DiningHalls - SpecialEvents: | DiningHalls (1, N) to (1,1) SpecialEvents | (A dining hall must have at least one special event, but can have multiple events, and a special event is associated with exactly one dining hall.) |
| SpecialEvents - SpecialEventMenuItems: | SpecialEvents (1, N) to (1, N) SpecialEventMenuItems | (A special event must have at least one menu item, but can have multiple items, and a menu item is associated with one or many special events.) |
| Menus - SpecialEvents: | Menus (0, N) to (0, N) SpecialEvents | (A menu is associated with zero or more special events, and a special event is associated with zero or more menus.) |

## 2.2 Data Format for Every Relation

Table 2. Data Format for Each Relation

| Relation Names | Attributes | Data Type | Length (Max) or Size |
|---|---|---|---|
| Students | StudentID (PK) | String (Primary Key) | 10 |
| | FullName | String | 50 |
| | Email | String | 255 |
| | Phone | String | 12 |
| | MealPlanID (FK) | String (Foreign Key) | 10 |
| | DietaryPreferenceID (FK) | String (Foreign Key) | 10 |
| | BalanceAmount | Decimal | - |
| MealPlans | MealPlanID (PK) | String (Primary Key) | 10 |
| | PlanName | String | 50 |
| | PlanDuration | String | 20 |
| | PlanPrice | Decimal | - |
| | MealsPerDay | Integer | - |
| DiningHalls | DiningHallID (PK) | String (Primary Key) | 10 |
| | HallName | String | 50 |
| | Location | String | 50 |
| | OpeningTime | Time | - |
| | ClosingTime | Time | - |
| Menus | MenuID (PK) | String (Primary Key) | 10 |
| | DiningHallID (FK) | String (Foreign Key) | 10 |
| | Date | Date | - |
| | MealType | String | 20 |
| MenuItems | MenuItemID (PK) | String (Primary Key) | 10 |
| | MenuID (FK) | String (Foreign Key) | 10 |
| | DishName | String | 50 |
| | | | |
| | Calories | Decimal | - |
| | IsVegetarian | Boolean | - |
| DietaryPreferences | DietaryPreferenceID (PK) | String (Primary Key) | 10 |

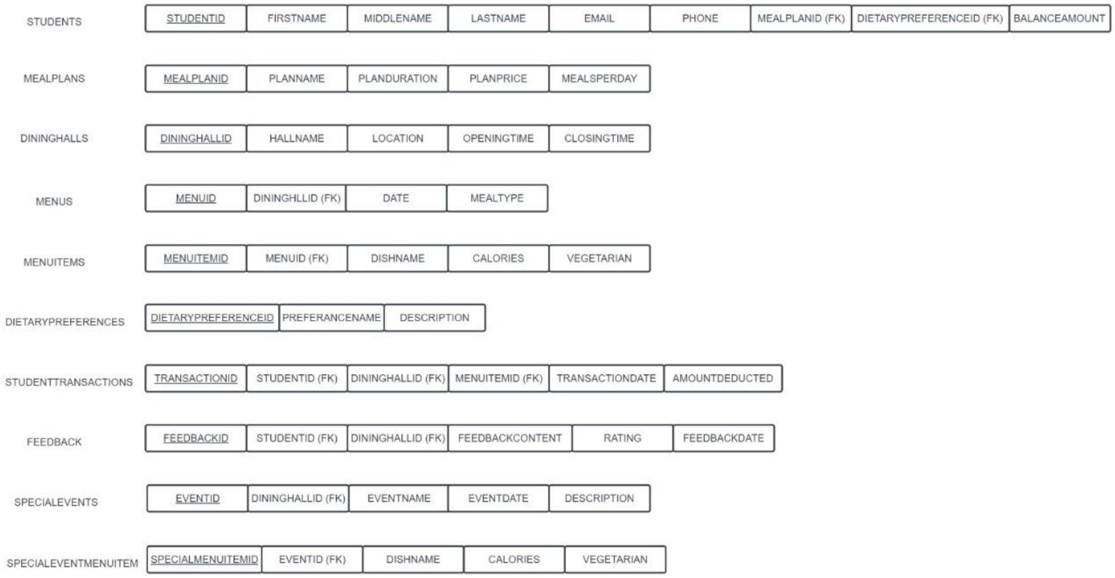
| | | | |
|---|---|---|---|
| | PreferenceName | String | 50 |
| | Description | String | - |
| StudentTransactions | TransactionID (PK) | String (Primary Key) | 10 |
| | StudentID (FK) | String (Foreign Key) | 10 |
| | DiningHallID (FK) | String (Foreign Key) | 10 |
| | MenuItemID (FK) | String (Foreign Key) | 10 |
| | TransactionDate | Date | - |
| | AmountDeducted | Decimal | - |
| Feedback | FeedbackID (PK) | String (Primary Key) | 10 |
| | StudentID (FK) | String (Foreign Key) | 10 |
| | DiningHallID (FK) | String (Foreign Key) | 10 |
| | FeedbackContent | String | - |
| | Rating | Integer | - |
| | FeedbackDate | Date | - |
| SpecialEvents | EventID (PK) | String (Primary Key) | 10 |
| | DiningHallID (FK) | String (Foreign Key) | 10 |
| | EventName | String | 50 |
| | EventDate | Date | - |
| | Description | String | - |
| SpecialEventMenuItems | SpecialMenuItemID (PK) | String (Primary Key) | 10 |
| | EventID (FK) | String (Foreign Key) | 10 |
| | DishName | String | 50 |
| | | | |
| | Calories | Decimal | - |
| | IsVegetarian | Boolean | - |

# 3. Pre-Illumination

This report outlines the implementation phase of the database project, focusing on the creation of the database, table setup, data population and SQL queries. Our project utilizes the MySQL database management system. Part 1 is the modified relational schema. Part 2 is the creation of the database, including tables, all other structures as well as constraints, data type and format, Part 3 is the query scenario design and implementation.

# 4. Relational Schema

The relational schema is shown in Figure 1.



| | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| STUDENTS | STUDENTID | FIRSTNAME | MIDDLENAME | LASTNAME | EMAIL | PHONE | MEALPLANID (FK) | DIETARYPREFERENCEID (FK) | BALANCEAMOUNT |
| MEALPLANS | MEALPLANID | PLANNAME | PLANDURATION | PLANPRICE | MEALSPERDAY | | | | |
| DININGHALLS | DININGHALLID | HALLNAME | LOCATION | OPENINGTIME | CLOSINGTIME | | | | |
| MENUS | MENUID | DININGHLLID (FK) | DATE | MEALTYPE | | | | | |
| MENUITEMS | MENUITEMID | MENUID (FK) | DISHNAME | CALORIES | VEGETARIAN | | | | |
| DIETARYPREFERENCES | DIETARYPREFERENCEID | PREFERANCENAME | DESCRIPTION | | | | | | |
| STUDENTTRANSACTIONS | TRANSACTIONID | STUDENTID (FK) | DININGHALLID (FK) | MENUITEMID (FK) | TRANSACTIONDATE | AMOUNTDEDUCTED | | | |
| FEEDBACK | FEEDBACKID | STUDENTID (FK) | DININGHALLID (FK) | FEEDBACKCONTENT | RATING | FEEDBACKDATE | | | |
| SPECIALEVENTS | EVENTID | DININGHALLID (FK) | EVENTNAME | EVENTDATE | DESCRIPTION | | | | |
| SPECIALEVENTMENUITEM | SPECIALMENUITEMID | EVENTID (FK) | DISHNAME | CALORIES | VEGETARIAN | | | | |

The following section shows how we created our database in MySQL.

# 5. Creation of Database with SQL Statements

## 5.1 Table Creation

First, we created the tables using the following SQL statement:

- **Meal Plans Table:**

```
CREATE TABLE MealPlans (
    MealPlanID VARCHAR(10) PRIMARY KEY,
    PlanName VARCHAR(50),
    PlanDuration VARCHAR(20),
    PlanPrice DECIMAL,
    MealsPerDay INT);
```

- **Dining Halls Table:**

```
CREATE TABLE DiningHalls (
    DiningHallID VARCHAR(10) PRIMARY KEY,
    HallName VARCHAR(50),
    Location VARCHAR(50),
    OpeningTime TIME,
    ClosingTime TIME);
```

- **Dietary PreferencesTable:**

```
CREATE TABLE DietaryPreferences (
    DietaryPreferenceID VARCHAR(10) PRIMARY KEY,
    PreferenceName VARCHAR(50),
    Description VARCHAR(255));
```

- **Menus Table:**

```
CREATE TABLE Menus (
    MenuID VARCHAR(10) PRIMARY KEY,
    DiningHallID VARCHAR(10),
    Date DATE,
    MealType VARCHAR(20),

FOREIGN KEY (DiningHallID) REFERENCES DiningHalls(DiningHallID));
```

- **Special Events Table:**

```
CREATE TABLE SpecialEvents (
    EventID VARCHAR(10) PRIMARY KEY,
    DiningHallID VARCHAR(10),
    EventName VARCHAR(50),
    EventDate DATE,
    Description VARCHAR(255),

FOREIGN KEY (DiningHallID) REFERENCES DiningHalls(DiningHallID));
```

- **Menu Items Table:**

```
 CREATE TABLE MenuItems (
    MenuItemID VARCHAR(10) PRIMARY KEY,
    MenuID VARCHAR(10),
    DishName VARCHAR(50),
    Calories DECIMAL,
    IsVegetarian BOOLEAN,

FOREIGN KEY (MenuID) REFERENCES Menus(MenuID));
```

- **Special Event Menu Items Table:**

```
CREATE TABLE SpecialEventMenuItems (
    SpecialMenuItemID VARCHAR(10) PRIMARY KEY,
    EventID VARCHAR(10),
    DishName VARCHAR(50),
    Calories DECIMAL,
    IsVegetarian BOOLEAN,

FOREIGN KEY (EventID) REFERENCES SpecialEvents(EventID));
```

- **Students Table:**

```
CREATE TABLE Students (
    StudentID VARCHAR(10) PRIMARY KEY,
    FullName VARCHAR(50),
    Email VARCHAR(255),
    Phone VARCHAR(12),
    MealPlanID VARCHAR(10),
    DietaryPreferenceID VARCHAR(10),
    BalanceAmount DECIMAL,

FOREIGN KEY (MealPlanID) REFERENCES MealPlans(MealPlanID),
      FOREIGN KEY (DietaryPreferenceID) REFERENCES
DietaryPreferences(DietaryPreferenceID));
```

- **Student Transactions Table:**

```
CREATE TABLE StudentTransactions (
    TransactionID VARCHAR(10) PRIMARY KEY,
    StudentID VARCHAR(10),
    DiningHallID VARCHAR(10),
    MenuItemID VARCHAR(10),
    TransactionDate DATE,
    AmountDeducted DECIMAL,
FOREIGN KEY (StudentID) REFERENCES Students(StudentID),
FOREIGN KEY (DiningHallID) REFERENCES DiningHalls(DiningHallID),
FOREIGN KEY (MenuItemID) REFERENCES MenuItems(MenuItemID));
```

- **Feedback Table:**

```
----------------------------------------------------------------------------------------------------

CREATE TABLE Feedback (
    FeedbackID VARCHAR(10) PRIMARY KEY,
    StudentID VARCHAR(10),
    DiningHallID VARCHAR(10),
    FeedbackContent VARCHAR(255),
    Rating INT,
    FeedbackDate DATE,
   FOREIGN KEY (StudentID) REFERENCES Students(StudentID),
   FOREIGN KEY (DiningHallID) REFERENCES DiningHalls(DiningHallID));
----------------------------------------------------------------------------------------------------
```

## 5.2 A Database State

To ensure the database is populated for testing and development purposes, sample data was inserted into each table. The following records were added to each table, maintaining data consistency and validity.

- **Insertion Of MealPlans**

```
----------------------------------------------------------------------------------------------------INSERT

INTO MealPlans (MealPlanID, PlanName, PlanDuration, PlanPrice,

MealsPerDay)

VALUES
    ('MP001', 'Basic Plan', '30 days', 50.00, 3),
    ('MP002', 'Standard Plan', '60 days', 75.00, 2),
    ('MP003', 'Premium Plan', '90 days', 100.00, 3),
     ('MP004', 'Family Plan', '45 days', 120.00, 4),
    ('MP005', 'Vegetarian Plan', '30 days', 60.00, 2);
```

- **Insertion Of Dining Halls**

```
----------------------------------------------------------------------------------------------------
INSERT INTO DiningHalls (DiningHallID, HallName, Location, OpeningTime,
ClosingTime)
VALUES
    ('DH001', 'The Market', 'Engineering and Computer Science West
(ECSW)', '09:00:00', '16:00:00'),
    ('DH002', 'Novel Brew', 'Eugene McDermott Library (MC)', '09:00:00',
'14:00:00'),
    ('DH003', 'The Market', 'Naveen Jindal School of Management (JSOM)',
'08:00:00', '21:00:00'),
    ('DH004', 'Einstein Bros Bagels', 'Parking Structure 3 (PS3)',
'07:30:00', '16:00:00'),
    ('DH005', 'Taco Bell Cantina', 'Parking Structure 3 (PS3)',
'08:00:00', '00:00:00'),
    ('DH006', 'Dining Hall West', 'Residence Hall West (RHW)', '07:00:00',
'10:00:00'),
    ('DH007', 'Papa John’s', 'Residence Hall West (RHW)', '11:00:00',
'00:00:00'),
    ('DH008', 'The Market', 'Residence Hall West (RHW)', '11:00:00',
'00:00:00'),
    ('DH009', 'The Market', 'Sciences Building (SCI)', '09:00:00',
'17:00:00'),
    ('DH010', 'The Market', 'Student Services Building Addition (SSA)',
'09:00:00', '16:00:00'),
    ('DH011', 'Chick-Fil-A', 'Student Union (SU)', '07:30:00',
'20:00:00'),
    ('DH012', 'Firehouse Subs', 'Student Union (SU)', '11:00:00',
'16:00:00'),
    ('DH013', 'Kalachandji’s Express', 'Student Union (SU)', '11:00:00',
'16:00:00'),
    ('DH014', 'Moe’s', 'Student Union (SU)', '11:00:00', '18:00:00'),
    ('DH015', 'Panda Express', 'Student Union (SU)', '11:00:00',
'18:00:00'),
    ('DH016', 'Smoothie King', 'Student Union (SU)', '07:30:00',
'16:00:00'),
    ('DH017', 'Starbucks', 'Student Union (SU)', '07:30:00', '22:00:00'),
    ('DH018', 'The Halal Shack', 'Student Union (SU)', '11:00:00',
'20:00:00'),
    ('DH019', 'The Market', 'Student Union (SU)', '09:00:00', '17:00:00'),
    ('DH020', 'Bookstore Coffee Shop', 'Visitor Center and University
Bookstore (VCB)', '08:00:00', '17:00:00');
```

- ## Insertion Of Dietary Preferences

```
INSERT INTO DietaryPreferences (DietaryPreferenceID, PreferenceName,
Description)
VALUES
    ('DP001', 'Vegetarian', 'Does not consume meat or fish'),
    ('DP002', 'Vegan', 'Avoids all animal products, including dairy and
eggs'),
    ('DP003', 'Pescatarian', 'Eats fish and other seafood but avoids other
meats'),
    ('DP004', 'Gluten-Free', 'Avoids foods that contain gluten'),
    ('DP005', 'Lactose-Free', 'Avoids dairy products containing lactose'),
    ('DP006', 'Halal', 'Permissible according to Islamic law'),
    ('DP007', 'Kosher', 'Conforms to the dietary laws of Jewish
tradition');
```

- ## Insertion Of Menus

```
INSERT INTO Menus (MenuID, DiningHallID, Date, MealType)
```

```
VALUES
    ('M001', 'DH001', '2023-12-01', 'Breakfast'),
    ('M002', 'DH001', '2023-12-01', 'Lunch'),
    ('M003', 'DH002', '2023-12-01', 'Breakfast'),
    ('M004', 'DH003', '2023-12-01', 'Everyday'),
    ('M005', 'DH004', '2023-12-01', 'Breakfast'),
    ('M006', 'DH004', '2023-12-01', 'Lunch'),
    ('M007', 'DH005', '2023-12-01', 'Everyday'),
    ('M008', 'DH006', '2023-12-01', 'Breakfast'),
    ('M009', 'DH007', '2023-12-01', 'Everyday'),
    ('M010', 'DH008', '2023-12-01', 'Everyday'),
    ('M011', 'DH009', '2023-12-01', 'Breakfast'),
    ('M012', 'DH009', '2023-12-01', 'Lunch'),
    ('M013', 'DH010', '2023-12-01', 'Breakfast'),
    ('M014', 'DH010', '2023-12-01', 'Lunch'),
    ('M015', 'DH011', '2023-12-01', 'Breakfast'),
    ('M016', 'DH011', '2023-12-01', 'Lunch'),
    ('M017', 'DH012', '2023-12-01', 'Lunch'),
    ('M018', 'DH013', '2023-12-01', 'Lunch'),
    ('M019', 'DH014', '2023-12-01', 'Lunch'),
    ('M020', 'DH015', '2023-12-01', 'Lunch'),
    ('M021', 'DH016', '2023-12-01', 'Breakfast'),
   ('M022', 'DH016', '2023-12-01', 'Lunch'),
    ('M023', 'DH017', '2023-12-01', 'Everyday'),
    ('M024', 'DH018', '2023-12-01', 'Lunch'),
    ('M025', 'DH018', '2023-12-01', 'Dinner'),
   ('M026', 'DH019', '2023-12-01', 'Breakfast'),
    ('M027', 'DH019', '2023-12-01', 'Lunch'),
    ('M028', 'DH020', '2023-12-01', 'Breakfast'),
    ('M029', 'DH020', '2023-12-01', 'Lunch');
```

- **Insertion Of Special Events**

```
INSERT INTO SpecialEvents (EventID, DiningHallID, EventName, EventDate,
Description)
VALUES
    ('EVT001', 'DH001', 'Orientation Day', '2023-12-10', 'Welcome new
students to campus and provide essential information.'),
    ('EVT002', 'DH002', 'Homecoming Celebration', '2023-12-12', 'Join the
UTD community in celebrating school spirit and traditions.'),
    ('EVT003', 'DH004', 'Career Fair', '2023-12-15', 'Connect with
potential employers and explore career opportunities.'),
    ('EVT004', 'DH004', 'Science Expo', '2023-12-18', 'Showcase scientific
achievements and projects from various departments.'),
    ('EVT005', 'DH007', 'Cultural Festival', '2023-12-20', 'Experience
diverse cultures through food, performances, and activities.'),
    ('EVT006', 'DH007', 'Coding Competition', '2023-12-22', 'Participate
in a coding competition with students from different disciplines.'),
    ('EVT007', 'DH009', 'Student Appreciation Day', '2023-12-25',
'Celebrate students\' achievements and contributions to the campus.'),
    ('EVT008', 'DH011', 'Hackathon', '2023-12-28', 'Collaborate with
fellow students to create innovative tech solutions.'),
    ('EVT009', 'DH011', 'Health and Wellness Expo', '2023-12-30', 'Promote
health and well-being through informational sessions and activities.'),
    ('EVT010', 'DH013', 'Leadership Summit', '2024-01-02', 'Engage in
discussions and workshops focused on leadership skills.'),
    ('EVT011', 'DH015', 'Graduation Ceremony', '2024-01-05', 'Congratulate
and honor graduating students on their academic achievements.');
```

- **Insertion Of Menu Items**

```
INSERT INTO MenuItems (MenuItemID, MenuID, DishName, Calories,
IsVegetarian)
VALUES
    ('MI001', 'M001', 'Classic Breakfast Combo(Omelette and Toast)',
300.5, FALSE),
    ('MI002', 'M002', 'Grilled Chicken Caesar Salad', 450.2, FALSE),
    ('MI003', 'M003', 'Fruit and Yogurt Parfait', 112.1, TRUE),
    ('MI004', 'M004', 'Chicken Shawarma Bowl with Hummus', 256.9, FALSE),
    ('MI005', 'M005', 'Shrimp Stir-Fry with Rice', 111.7, FALSE),
    ('MI006', 'M006', 'Pancakes with Syrup', 207.7, TRUE),
    ('MI007', 'M007', 'Grilled Cheese Sandwich with Tomato Soup', 508.4,
TRUE),
    ('MI008', 'M008', 'Classic Burger with Sweet Potato Fries', 457.8,
TRUE),
    ('MI009', 'M009', 'Chicken Shawarma Bowl with Hummus', 256.9, FALSE),
    ('MI010', 'M010', 'Grilled Cheese Sandwich with Tomato Soup', 508.4,
TRUE),
    ('MI011', 'M011', 'Classic Breakfast Combo(Omelette and Toast)',
300.5, FALSE),
    ('MI012', 'M012', 'Shrimp Stir-Fry with Rice', 111.7, FALSE),
    ('MI013', 'M013', 'Classic Breakfast Combo(Omelette and Toast)',
300.5, TRUE),
    ('MI014', 'M014', 'Vegetarian Pasta Primavera', 324.1, TRUE),
    ('MI015', 'M015', 'Fruit and Yogurt Parfait', 112.1, TRUE),
    ('MI016', 'M016', 'Grilled Chicken Caesar Salad', 450.2, FALSE),
    ('MI017', 'M017', 'Grilled Chicken Caesar Salad', 450.2, FALSE),
    ('MI018', 'M018', 'Grilled Chicken Caesar Salad', 450.2, FALSE),
    ('MI019', 'M019', 'Vegetarian Pasta Primavera', 324.1, TRUE),
    ('MI020', 'M020', 'Shrimp Stir-Fry with Rice', 111.7, FALSE),
    ('MI021', 'M021', 'Fruit and Yogurt Parfait', 112.1, TRUE),
    ('MI022', 'M022', 'Vegetarian Pasta Primavera', 324.1, TRUE),
    ('MI023', 'M023', 'Classic Burger with Sweet Potato Fries', 457.8,
TRUE),
    ('MI024', 'M024', 'Shrimp Stir-Fry with Rice', 111.7, FALSE),
    ('MI025', 'M025', 'Chicken Alfredo', 210.7, FALSE),
    ('MI026', 'M026', 'Pancakes with Syrup', 207.7, TRUE),
    ('MI027', 'M027', 'Shrimp Stir-Fry with Rice', 111.7, FALSE),
    ('MI028', 'M028', 'Classic Breakfast Combo(Omelette and Toast)',
300.5, FALSE),
    ('MI029', 'M029', 'Vegetarian Pasta Primavera', 324.1, TRUE);
```

## • Insertion Of Special Events Menu Items

```
------------------------------------------------------------------------------------------------------------
INSERT  INTO  SpecialEventMenuItems  (SpecialMenuItemID,  EventID,  DishName,
Calories, IsVegetarian)
VALUES
    ('SMI001', 'EVT001', 'Fancy Appetizer Platter', 300.5, TRUE),
    ('SMI002', 'EVT002', 'Chocolate Cake', 700.2, FALSE),
    ('SMI003', 'EVT003', 'Tiramisu', 892, FALSE),
    ('SMI004', 'EVT004', 'New York Cheesecake',  623.2, FALSE),
    ('SMI005', 'EVT005', 'Breakfast Burrito', 598.1, FALSE),
    ('SMI006', 'EVT006', 'Fancy Appetizer Platter', 300.5, TRUE),
    ('SMI007', 'EVT007', 'Chocolate Cake', 700.2, FALSE),
    ('SMI008', 'EVT008', 'Tiramisu', 892, FALSE),
    ('SMI009', 'EVT009', 'New York Cheesecake', 623.2, FALSE),
    ('SMI010', 'EVT010', 'Breakfast Burrito', 598.1, FALSE),
    ('SMI011', 'EVT011', 'Fancy Appetizer Platter', 700.2, TRUE);
```

## • Insertion Of Students

```
------------------------------------------------------------------------------------------------------------
INSERT INTO Students (StudentID, FullName, Email, Phone, MealPlanID,
DietaryPreferenceID, BalanceAmount)
VALUES
    ('S001', 'John Doe', 'john.doe@utdallas.edu', '123-456-7890', 'MP001',
'DP001', 200.00),
    ('S002', 'Jane Smith', 'jane.smith@utdallas.edu', '987-654-3210',
'MP002', 'DP002', 150.00),
    ('S003', 'Robert Johnson', 'robert.johnson@utdallas.edu', '456-789-
0123', 'MP003', 'DP003', 300.00),
    ('S004', 'Emily White', 'emily.white@utdallas.edu', '789-012-3456',
'MP004', 'DP004', 180.00),
    ('S005', 'Daniel Adams', 'daniel.adams@utdallas.edu', '234-567-8901',
'MP005', 'DP005', 250.00),
    ('S006', 'Sophia Brown', 'sophia.brown@utdallas.edu', '567-890-1234',
'MP001', 'DP006', 350.00),
    ('S007', 'William Clark', 'william.clark@utdallas.edu', '890-123-
4567', 'MP002', 'DP007', 120.00),
    ('S008', 'Olivia Davis', 'olivia.davis@utdallas.edu', '123-456-7890',
'MP003', 'DP001', 400.00),
    ('S009', 'James Evans', 'james.evans@utdallas.edu', '987-654-3210',
'MP004', 'DP002', 280.00),
    ('S010', 'Emma Fisher', 'emma.fisher@utdallas.edu', '456-789-0123',
'MP005', 'DP003', 320.00),
('S011', 'Aiden Garcia', 'aiden.garcia@utdallas.edu', '234-567-8901',
'MP001', 'DP006', 280.00),
    ('S012', 'Mia Hall', 'mia.hall@utdallas.edu', '567-890-1234', 'MP002',
'DP007', 150.00),
    ('S013', 'Lucas Hill', 'lucas.hill@utdallas.edu', '890-123-4567',
'MP003', 'DP001', 200.00),
    ('S014', 'Ava Jenkins', 'ava.jenkins@utdallas.edu', '123-456-7890',
'MP004', 'DP002', 350.00),
    ('S015', 'Liam King', 'liam.king@utdallas.edu', '987-654-3210',
'MP005', 'DP003', 180.00),
    ('S016', 'Isabella Lewis', 'isabella.lewis@utdallas.edu', '456-789-
0123', 'MP001', 'DP004', 300.00),
    ('S017', 'Jackson Martin', 'jackson.martin@utdallas.edu', '789-012-
3456', 'MP002', 'DP005', 250.00),
    ('S018', 'Sophie Mitchell', 'sophie.mitchell@utdallas.edu', '123-456-
7890', 'MP003', 'DP006', 400.00),
    ('S019', 'Oliver Nelson', 'oliver.nelson@utdallas.edu', '987-654-
3210', 'MP004', 'DP007', 320.00),
    ('S020', 'Emma Owen', 'emma.owen@utdallas.edu', '456-789-0123',
'MP005', 'DP001', 150.00),
    ('S021', 'Carter Parker', 'carter.parker@utdallas.edu', '789-012-
3456', 'MP001', 'DP002', 280.00),
    ('S022', 'Chloe Perry', 'chloe.perry@utdallas.edu', '123-456-7890',
'MP002', 'DP003', 350.00),
    ('S023', 'Ethan Reed', 'ethan.reed@utdallas.edu', '987-654-3210',
'MP003', 'DP004', 200.00),
    ('S024', 'Aria Richardson', 'aria.richardson@utdallas.edu', '234-567-
8901', 'MP004', 'DP005', 180.00),
    ('S025', 'Lucas Ross', 'lucas.ross@utdallas.edu', '567-890-1234',
'MP005', 'DP006', 320.00),
    ('S026', 'Zoe Russell', 'zoe.russell@utdallas.edu', '890-123-4567',
'MP001', 'DP007', 150.00),
    ('S027', 'Leo Simmons', 'leo.simmons@utdallas.edu', '123-456-7890',
'MP002', 'DP001', 250.00),
    ('S028', 'Ava Stewart', 'ava.stewart@utdallas.edu', '987-654-3210',
'MP003', 'DP002', 300.00),
    ('S029', 'Logan Taylor', 'logan.taylor@utdallas.edu', '234-567-8901',
'MP004', 'DP003', 400.00),
    ('S030', 'Lily Turner', 'lily.turner@utdallas.edu', '567-890-1234',
'MP005', 'DP004', 280.00),
    ('S031', 'Elijah Adams', 'elijah.adams@utdallas.edu', '234-567-8901',
'MP001', 'DP001', 200.00),
    ('S032', 'Avery Bennett', 'avery.bennett@utdallas.edu', '567-890-
1234', 'MP002', 'DP002', 150.00),
    ('S033', 'Elena Campbell', 'elena.campbell@utdallas.edu', '890-123-
4567', 'MP003', 'DP003', 300.00),
    ('S034', 'Connor Diaz', 'connor.diaz@utdallas.edu', '123-456-7890',
'MP004', 'DP004', 180.00),
    ('S035', 'Gabriella Foster', 'gabriella.foster@utdallas.edu', '987-
654-3210', 'MP005', 'DP005', 250.00),
    ('S036', 'Henry Gray', 'henry.gray@utdallas.edu', '456-789-0123',
'MP001', 'DP006', 350.00),
    ('S037', 'Addison Hayes', 'addison.hayes@utdallas.edu', '789-012-
3456', 'MP002', 'DP007', 120.00),
    ('S038', 'Isaac Ingram', 'isaac.ingram@utdallas.edu', '123-456-7890',
'MP003', 'DP001', 400.00),
    ('S039', 'Aria Jones', 'aria.jones@utdallas.edu', '987-654-3210',
'MP004', 'DP002', 280.00),
    ('S040', 'Eli Knight', 'eli.knight@utdallas.edu', '456-789-0123',
'MP005', 'DP003', 320.00),
    ('S041', 'Nora Long', 'nora.long@utdallas.edu', '234-567-8901',
'MP001', 'DP004', 180.00),
    ('S042', 'Oscar Miller', 'oscar.miller@utdallas.edu', '567-890-1234',
'MP002', 'DP005', 250.00),
    ('S043', 'Penelope Nelson', 'penelope.nelson@utdallas.edu', '890-123-
4567', 'MP003', 'DP006', 350.00),
    ('S044', 'Quinn Olson', 'quinn.olson@utdallas.edu', '123-456-7890',
'MP004', 'DP007', 120.00),
    ('S045', 'Riley Parker', 'riley.parker@utdallas.edu', '987-654-3210',
'MP005', 'DP001', 400.00),
    ('S046', 'Sawyer Quinn', 'sawyer.quinn@utdallas.edu', '456-789-0123',
'MP001', 'DP002', 280.00),
    ('S047', 'Taylor Turner', 'taylor.turner@utdallas.edu', '789-012-
3456', 'MP002', 'DP003', 320.00),
    ('S048', 'Uma Vaughn', 'uma.vaughn@utdallas.edu', '123-456-7890',
'MP003', 'DP004', 180.00),
    ('S049', 'Vincent Wallace', 'vincent.wallace@utdallas.edu', '987-654-
3210', 'MP004', 'DP005', 250.00),
    ('S050', 'Willow Xavier', 'willow.xavier@utdallas.edu', '456-789-
0123', 'MP005', 'DP006', 350.00);
```

- **Insertion Of Student Transactions**

------------------------------------------------------------------------------------------------------------

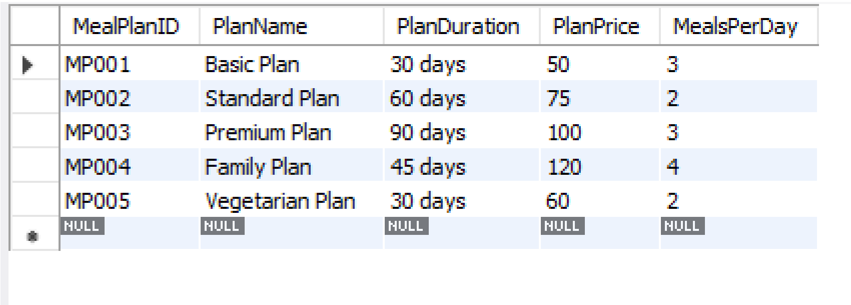
```
INSERT INTO StudentTransactions (TransactionID, StudentID, DiningHallID,
MenuItemID, TransactionDate, AmountDeducted)
VALUES
('T001', 'S001', 'DH001', 'MI001', '2023-11-10', 10.50),
    ('T002', 'S002', 'DH002', 'MI002', '2023-11-10', 15.75),
    ('T003', 'S003', 'DH003', 'MI003', '2023-11-03', 8.20),
    ('T004', 'S004', 'DH004', 'MI004', '2023-11-04', 12.30),
    ('T005', 'S005', 'DH005', 'MI005', '2023-11-05', 5.80),
    ('T006', 'S006', 'DH006', 'MI006', '2023-11-06', 9.60),
    ('T007', 'S007', 'DH007', 'MI007', '2023-11-07', 14.50),
    ('T008', 'S008', 'DH008', 'MI008', '2023-11-08', 12.75),
    ('T009', 'S009', 'DH009', 'MI009', '2023-11-09', 6.90),
    ('T010', 'S010', 'DH010', 'MI010', '2023-11-10', 11.25),
    ('T011', 'S011', 'DH011', 'MI011', '2023-11-11', 9.60),
    ('T012', 'S012', 'DH012', 'MI012', '2023-11-11', 14.40),
    ('T013', 'S013', 'DH013', 'MI013', '2023-11-13', 8.10),
    ('T014', 'S014', 'DH014', 'MI014', '2023-11-14', 12.30),
    ('T015', 'S015', 'DH015', 'MI015', '2023-11-15', 10.20),
    ('T016', 'S016', 'DH016', 'MI016', '2023-11-16', 15.80),
    ('T017', 'S017', 'DH017', 'MI017', '2023-11-17', 7.50),
    ('T018', 'S018', 'DH018', 'MI018', '2023-11-18', 10.80),
    ('T019', 'S019', 'DH019', 'MI019', '2023-11-19', 13.50),
    ('T020', 'S020', 'DH020', 'MI020', '2023-11-20', 8.40),
    ('T021', 'S021', 'DH001', 'MI021', '2023-11-21', 12.30),
    ('T022', 'S022', 'DH002', 'MI022', '2023-11-22', 15.75),
    ('T023', 'S023', 'DH003', 'MI023', '2023-11-23', 9.20),
    ('T024', 'S024', 'DH004', 'MI024', '2023-11-24', 14.40),
    ('T025', 'S025', 'DH005', 'MI025', '2023-11-25', 7.10),
    ('T026', 'S026', 'DH006', 'MI026', '2023-11-26', 11.50),
    ('T027', 'S027', 'DH007', 'MI027', '2023-11-27', 13.80),
    ('T028', 'S028', 'DH008', 'MI028', '2023-11-28', 10.75),
    ('T029', 'S029', 'DH009', 'MI029', '2023-11-29', 16.20),
    ('T030', 'S030', 'DH010', 'MI006', '2023-11-30', 12.60),
    ('T031', 'S031', 'DH011', 'MI011', '2023-11-30', 9.60),
    ('T032', 'S032', 'DH012', 'MI012', '2023-10-10', 14.40),
    ('T033', 'S033', 'DH013', 'MI013', '2023-10-02', 8.10),
    ('T034', 'S034', 'DH014', 'MI014', '2023-10-03', 12.30),
    ('T035', 'S035', 'DH015', 'MI015', '2023-10-04', 10.20),
    ('T036', 'S036', 'DH016', 'MI016', '2023-10-05', 15.80),
    ('T037', 'S037', 'DH017', 'MI017', '2023-10-06', 7.50),
    ('T038', 'S038', 'DH018', 'MI018', '2023-10-07', 10.80),
    ('T039', 'S039', 'DH019', 'MI019', '2023-10-08', 13.50),
    ('T040', 'S040', 'DH020', 'MI020', '2023-10-09', 8.40),
    ('T041', 'S041', 'DH001', 'MI021', '2023-10-10', 12.30),
    ('T042', 'S042', 'DH002', 'MI022', '2023-10-11', 15.75),
    ('T043', 'S043', 'DH003', 'MI023', '2023-10-11', 9.20),
    ('T044', 'S044', 'DH004', 'MI024', '2023-10-13', 14.40),
    ('T045', 'S045', 'DH005', 'MI025', '2023-10-14', 7.10),
    ('T046', 'S046', 'DH006', 'MI026', '2023-10-15', 11.50),
    ('T047', 'S047', 'DH007', 'MI027', '2023-10-16', 13.80),
    ('T048', 'S048', 'DH008', 'MI028', '2023-10-17', 10.75),
    ('T049', 'S049', 'DH009', 'MI029', '2023-10-18', 16.20),
    ('T050', 'S050', 'DH010', 'MI004', '2023-10-19', 12.60);
```

- ## Insertion Of Feedbacks

```
INSERT INTO Feedback (FeedbackID, StudentID, DiningHallID,
FeedbackContent, Rating, FeedbackDate)
VALUES
    ('F001', 'S001', 'DH001', 'The Market has a great selection of fresh
produce.', 4, '2023-11-11'),
    ('F002', 'S002', 'DH002', 'Love the coffee at Novel Brew!', 5, '2023-
11-02'),
    ('F003', 'S003', 'DH003', 'The Market at JSOM is my go-to for lunch.',
4, '2023-11-03'),
    ('F004', 'S004', 'DH004', 'Einstein Bros Bagels serves the best bagels
in town.', 5, '2023-11-04'),
    ('F005', 'S005', 'DH005', 'Taco Bell Cantina is perfect for late-night
cravings.', 4, '2023-11-05'),
    ('F006', 'S006', 'DH006', 'Dining Hall West breakfast is a must-try.',
5, '2023-11-06'),
    ('F007', 'S007', 'DH007', 'Papa John’s pizza never disappoints.', 4,
'2023-11-07'),
    ('F008', 'S008', 'DH008', 'The Market at RHW is convenient for
snacks.', 5, '2023-11-08'),
    ('F009', 'S009', 'DH009', 'The Market at SCI has a variety of
options.', 4, '2023-11-09'),
    ('F010', 'S010', 'DH010', 'Enjoyed the sandwiches at The Market
(SSA).', 5, '2023-11-10'),
    ('F011', 'S011', 'DH011', 'Chick-Fil-A service is quick and
efficient.', 4, '2023-11-11'),
    ('F012', 'S012', 'DH012', 'Firehouse Subs never compromises on
quality.', 5, '2023-11-11'),
    ('F013', 'S013', 'DH013', 'Kalachandji’s Express has unique and tasty
options.', 4, '2023-11-13'),
    ('F014', 'S014', 'DH014', 'Moe’s is a great place for a quick bite.',
5, '2023-11-14'),
    ('F015', 'S015', 'DH015', 'Panda Express never disappoints!', 4,
'2023-11-15'),
    ('F016', 'S016', 'DH016', 'Smoothie King is perfect for a refreshing
drink.', 5, '2023-11-16'),
    ('F017', 'S017', 'DH017', 'Starbucks coffee is a must-have!', 4,
'2023-11-17'),
    ('F018', 'S018', 'DH018', 'The Halal Shack offers delicious halal
options.', 5, '2023-11-18'),
    ('F019', 'S019', 'DH019', 'The Market at SU has a variety of
choices.', 4, '2023-11-19'),
    ('F020', 'S020', 'DH020', 'Great coffee at the Bookstore Coffee
Shop.', 5, '2023-11-20'),
    ('F021', 'S021', 'DH011', 'Chick-Fil-A service is quick and
efficient.', 4, '2023-11-21'),
    ('F022', 'S022', 'DH012', 'Firehouse Subs never compromises on
quality.', 5, '2023-11-22'),
    ('F023', 'S023', 'DH013', 'Kalachandji’s Express has unique and tasty
options.', 4, '2023-11-23'),
    ('F024', 'S024', 'DH014', 'Moe’s is a great place for a quick bite.',
5, '2023-11-24'),
    ('F025', 'S025', 'DH015', 'Panda Express never disappoints!', 4,
'2023-11-25'),
    ('F026', 'S026', 'DH016', 'Smoothie King is perfect for a refreshing
drink.', 5, '2023-11-26'),
    ('F027', 'S027', 'DH017', 'Starbucks coffee is a must-have!', 4,
'2023-11-27'),
    ('F028', 'S028', 'DH018', 'The Halal Shack offers delicious halal
options.', 5, '2023-11-28'),
    ('F029', 'S029', 'DH019', 'The Market at SU has a variety of
choices.', 4, '2023-11-29'),
    ('F030', 'S030', 'DH020', 'Great coffee at the Bookstore Coffee
Shop.', 5, '2023-11-30'),
    ('F031', 'S031', 'DH011', 'Chick-Fil-A service is quick and
efficient.', 4, '2023-11-29'),
    ('F032', 'S032', 'DH012', 'Firehouse Subs never compromises on
quality.', 5, '2023-11-11'),
    ('F033', 'S033', 'DH013', 'Kalachandji’s Express has unique and tasty
options.', 4, '2023-11-02'),
    ('F034', 'S034', 'DH014', 'Moe’s is a great place for a quick bite.',
5, '2023-11-03'),
    ('F035', 'S035', 'DH015', 'Panda Express never disappoints!', 4,
'2023-11-04'),
    ('F036', 'S036', 'DH016', 'Smoothie King is perfect for a refreshing
drink.', 5, '2023-11-05'),
    ('F037', 'S037', 'DH017', 'Starbucks coffee is a must-have!', 4,
'2023-11-06'),
    ('F038', 'S038', 'DH018', 'The Halal Shack offers delicious halal
options.', 5, '2023-11-07'),
    ('F039', 'S039', 'DH019', 'The Market at SU has a variety of
choices.', 4, '2023-11-08'),
    ('F040', 'S040', 'DH020', 'Great coffee at the Bookstore Coffee
Shop.', 5, '2023-11-09'),
    ('F041', 'S041', 'DH011', 'Chick-Fil-A service is quick and
efficient.', 4, '2023-11-10'),
    ('F042', 'S042', 'DH012', 'Firehouse Subs never compromises on
quality.', 5, '2023-11-11'),
    ('F043', 'S043', 'DH013', 'Kalachandji’s Express has unique and tasty
options.', 4, '2023-11-11'),
    ('F044', 'S044', 'DH014', 'Moe’s is a great place for a quick bite.',
5, '2023-11-13'),
    ('F045', 'S045', 'DH015', 'Panda Express never disappoints!', 4,
'2023-11-14'),
    ('F046', 'S046', 'DH016', 'Smoothie King is perfect for a refreshing
drink.', 5, '2023-11-15'),
    ('F047', 'S047', 'DH017', 'Starbucks coffee is a must-have!', 4,
'2023-11-16'),
    ('F048', 'S048', 'DH018', 'The Halal Shack offers delicious halal
options.', 5, '2023-11-17'),
    ('F049', 'S049', 'DH019', 'The Market at SU has a variety of
choices.', 4, '2023-11-18'),
    ('F050', 'S050', 'DH020', 'Great coffee at the Bookstore Coffee
Shop.', 5, '2023-11-19');
```

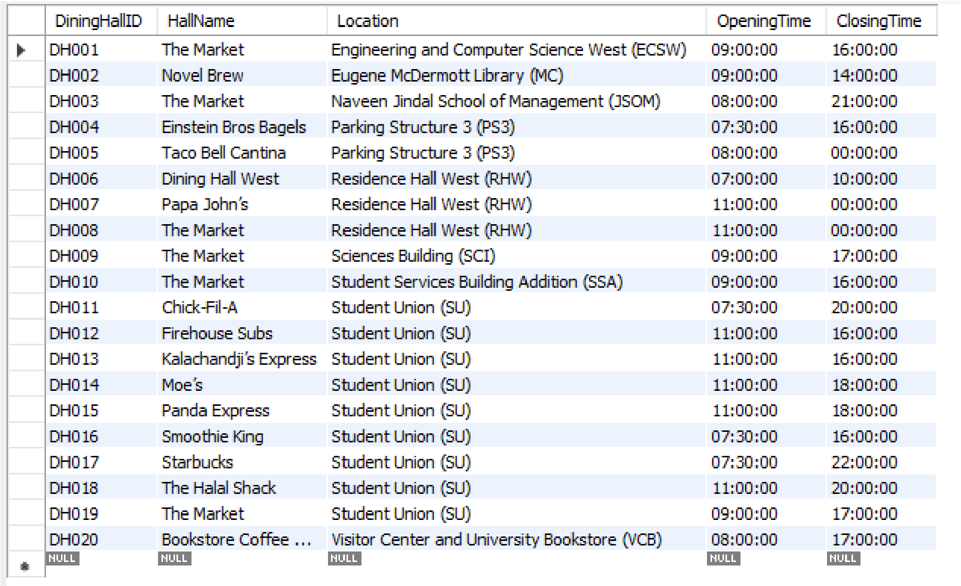
-----------------------------------------------------------------------------------------------------------------------

**Table 1. Meal Plans**



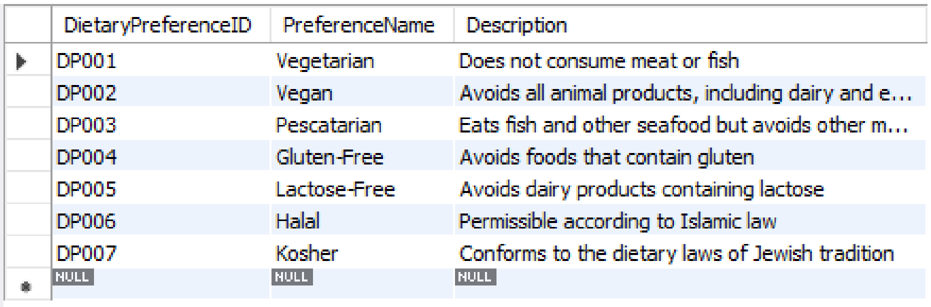
| MealPlanID | PlanName | PlanDuration | PlanPrice | MealsPerDay |
|---|---|---|---|---|
| MP001 | Basic Plan | 30 days | 50 | 3 |
| MP002 | Standard Plan | 60 days | 75 | 2 |
| MP003 | Premium Plan | 90 days | 100 | 3 |
| MP004 | Family Plan | 45 days | 120 | 4 |
| MP005 | Vegetarian Plan | 30 days | 60 | 2 |
| NULL | NULL | NULL | NULL | NULL |

**Table 2. Dining Halls**



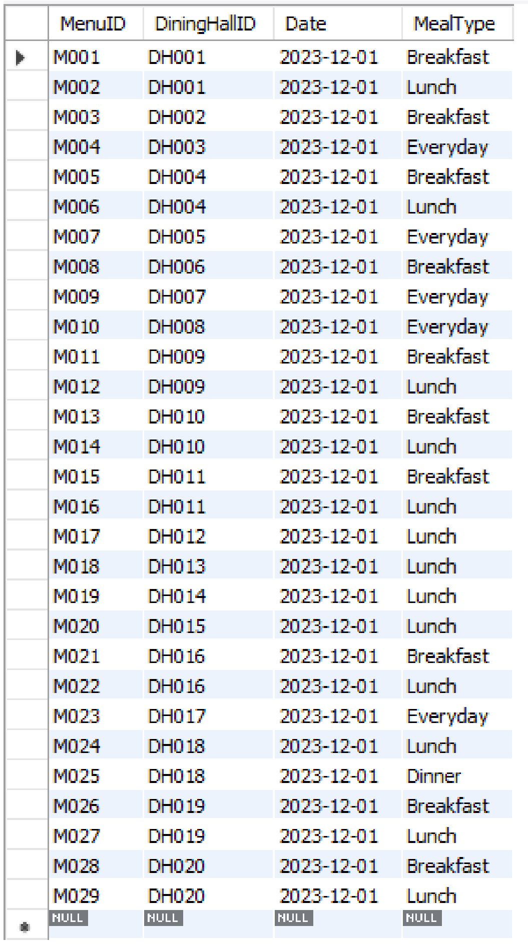
| DiningHallID | HallName | Location | OpeningTime | ClosingTime |
|---|---|---|---|---|
| DH001 | The Market | Engineering and Computer Science West (ECSW) | 09:00:00 | 16:00:00 |
| DH002 | Novel Brew | Eugene McDermott Library (MC) | 09:00:00 | 14:00:00 |
| DH003 | The Market | Naveen Jindal School of Management (JSOM) | 08:00:00 | 21:00:00 |
| DH004 | Einstein Bros Bagels | Parking Structure 3 (PS3) | 07:30:00 | 16:00:00 |
| DH005 | Taco Bell Cantina | Parking Structure 3 (PS3) | 08:00:00 | 00:00:00 |
| DH006 | Dining Hall West | Residence Hall West (RHW) | 07:00:00 | 10:00:00 |
| DH007 | Papa John's | Residence Hall West (RHW) | 11:00:00 | 00:00:00 |
| DH008 | The Market | Residence Hall West (RHW) | 11:00:00 | 00:00:00 |
| DH009 | The Market | Sciences Building (SCI) | 09:00:00 | 17:00:00 |
| DH010 | The Market | Student Services Building Addition (SSA) | 09:00:00 | 16:00:00 |
| DH011 | Chick-Fil-A | Student Union (SU) | 07:30:00 | 20:00:00 |
| DH012 | Firehouse Subs | Student Union (SU) | 11:00:00 | 16:00:00 |
| DH013 | Kalachandji's Express | Student Union (SU) | 11:00:00 | 16:00:00 |
| DH014 | Moe's | Student Union (SU) | 11:00:00 | 18:00:00 |
| DH015 | Panda Express | Student Union (SU) | 11:00:00 | 18:00:00 |
| DH016 | Smoothie King | Student Union (SU) | 07:30:00 | 16:00:00 |
| DH017 | Starbucks | Student Union (SU) | 07:30:00 | 22:00:00 |
| DH018 | The Halal Shack | Student Union (SU) | 11:00:00 | 20:00:00 |
| DH019 | The Market | Student Union (SU) | 09:00:00 | 17:00:00 |
| DH020 | Bookstore Coffee ... | Visitor Center and University Bookstore (VCB) | 08:00:00 | 17:00:00 |
| NULL | NULL | NULL | NULL | NULL |

**Table 3. Dietary Preferences**



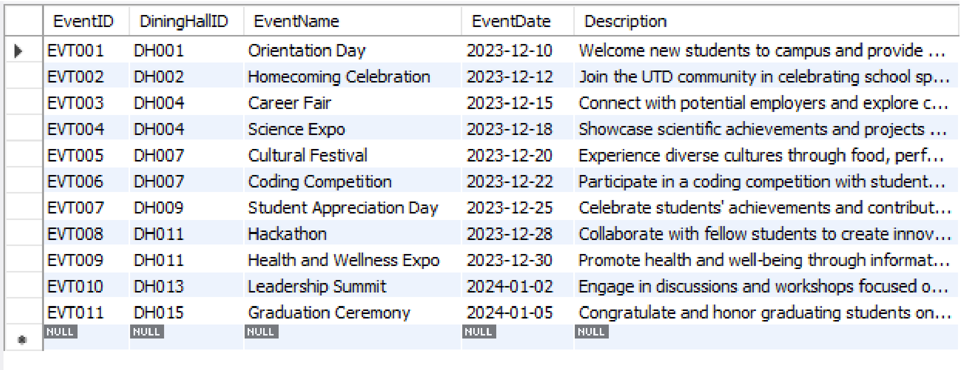
| DietaryPreferenceID | PreferenceName | Description |
|---|---|---|
| DP001 | Vegetarian | Does not consume meat or fish |
| DP002 | Vegan | Avoids all animal products, including dairy and e... |
| DP003 | Pescatarian | Eats fish and other seafood but avoids other m... |
| DP004 | Gluten-Free | Avoids foods that contain gluten |
| DP005 | Lactose-Free | Avoids dairy products containing lactose |
| DP006 | Halal | Permissible according to Islamic law |
| DP007 | Kosher | Conforms to the dietary laws of Jewish tradition |
| NULL | NULL | NULL |

## Table 4. Menus



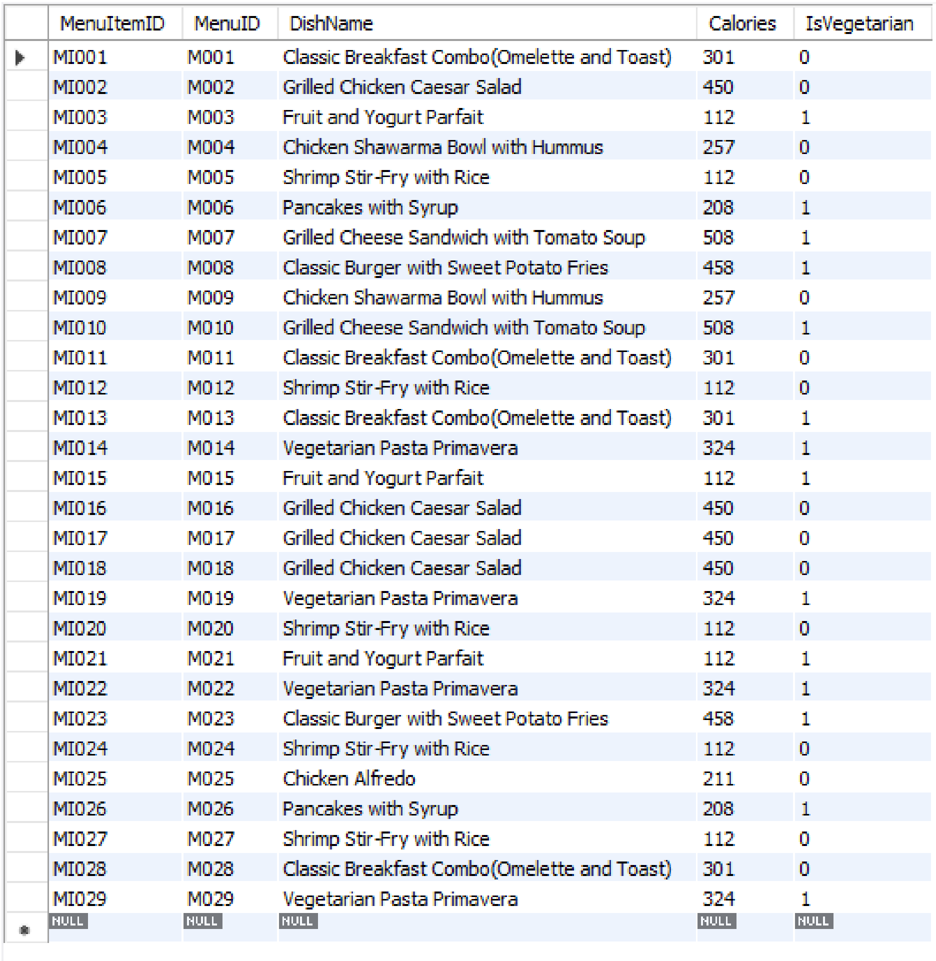
| MenuID | DiningHallID | Date | MealType |
|---|---|---|---|
| M001 | DH001 | 2023-12-01 | Breakfast |
| M002 | DH001 | 2023-12-01 | Lunch |
| M003 | DH002 | 2023-12-01 | Breakfast |
| M004 | DH003 | 2023-12-01 | Everyday |
| M005 | DH004 | 2023-12-01 | Breakfast |
| M006 | DH004 | 2023-12-01 | Lunch |
| M007 | DH005 | 2023-12-01 | Everyday |
| M008 | DH006 | 2023-12-01 | Breakfast |
| M009 | DH007 | 2023-12-01 | Everyday |
| M010 | DH008 | 2023-12-01 | Everyday |
| M011 | DH009 | 2023-12-01 | Breakfast |
| M012 | DH009 | 2023-12-01 | Lunch |
| M013 | DH010 | 2023-12-01 | Breakfast |
| M014 | DH010 | 2023-12-01 | Lunch |
| M015 | DH011 | 2023-12-01 | Breakfast |
| M016 | DH011 | 2023-12-01 | Lunch |
| M017 | DH012 | 2023-12-01 | Lunch |
| M018 | DH013 | 2023-12-01 | Lunch |
| M019 | DH014 | 2023-12-01 | Lunch |
| M020 | DH015 | 2023-12-01 | Lunch |
| M021 | DH016 | 2023-12-01 | Breakfast |
| M022 | DH016 | 2023-12-01 | Lunch |
| M023 | DH017 | 2023-12-01 | Everyday |
| M024 | DH018 | 2023-12-01 | Lunch |
| M025 | DH018 | 2023-12-01 | Dinner |
| M026 | DH019 | 2023-12-01 | Breakfast |
| M027 | DH019 | 2023-12-01 | Lunch |
| M028 | DH020 | 2023-12-01 | Breakfast |
| M029 | DH020 | 2023-12-01 | Lunch |
| NULL | NULL | NULL | NULL |

## Table 5. Special Events



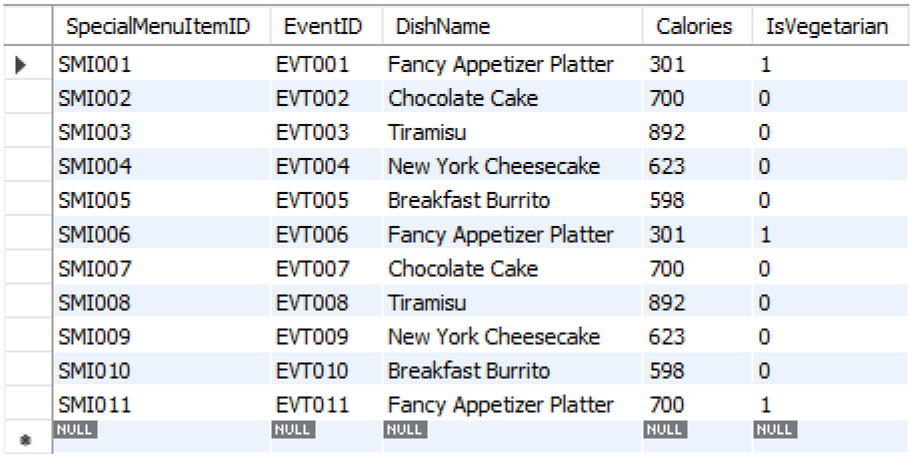
| EventID | DiningHallID | EventName | EventDate | Description |
|---|---|---|---|---|
| EVT001 | DH001 | Orientation Day | 2023-12-10 | Welcome new students to campus and provide ... |
| EVT002 | DH002 | Homecoming Celebration | 2023-12-12 | Join the UTD community in celebrating school sp... |
| EVT003 | DH004 | Career Fair | 2023-12-15 | Connect with potential employers and explore c... |
| EVT004 | DH004 | Science Expo | 2023-12-18 | Showcase scientific achievements and projects ... |
| EVT005 | DH007 | Cultural Festival | 2023-12-20 | Experience diverse cultures through food, perf... |
| EVT006 | DH007 | Coding Competition | 2023-12-22 | Participate in a coding competition with student... |
| EVT007 | DH009 | Student Appreciation Day | 2023-12-25 | Celebrate students' achievements and contribut... |
| EVT008 | DH011 | Hackathon | 2023-12-28 | Collaborate with fellow students to create innov... |
| EVT009 | DH011 | Health and Wellness Expo | 2023-12-30 | Promote health and well-being through informat... |
| EVT010 | DH013 | Leadership Summit | 2024-01-02 | Engage in discussions and workshops focused o... |
| EVT011 | DH015 | Graduation Ceremony | 2024-01-05 | Congratulate and honor graduating students on... |
| NULL | NULL | NULL | NULL | NULL |

## Table 6. Menu Items



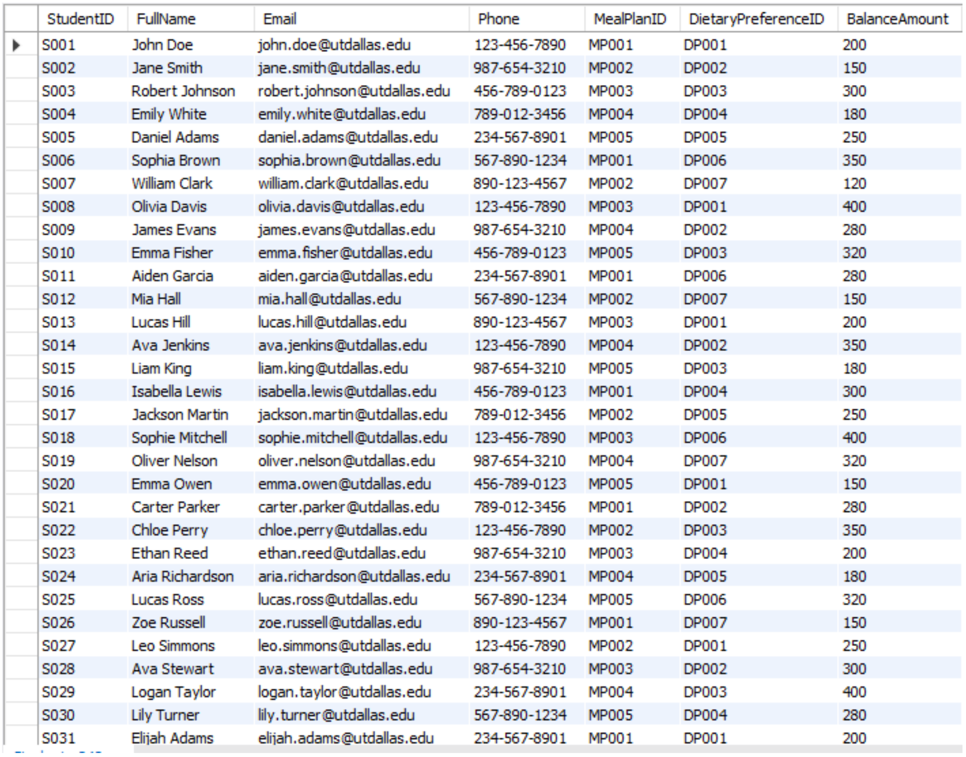
| MenuItemID | MenuID | DishName | Calories | IsVegetarian |
|---|---|---|---|---|
| MI001 | M001 | Classic Breakfast Combo(Omelette and Toast) | 301 | 0 |
| MI002 | M002 | Grilled Chicken Caesar Salad | 450 | 0 |
| MI003 | M003 | Fruit and Yogurt Parfait | 112 | 1 |
| MI004 | M004 | Chicken Shawarma Bowl with Hummus | 257 | 0 |
| MI005 | M005 | Shrimp Stir-Fry with Rice | 112 | 0 |
| MI006 | M006 | Pancakes with Syrup | 208 | 1 |
| MI007 | M007 | Grilled Cheese Sandwich with Tomato Soup | 508 | 1 |
| MI008 | M008 | Classic Burger with Sweet Potato Fries | 458 | 1 |
| MI009 | M009 | Chicken Shawarma Bowl with Hummus | 257 | 0 |
| MI010 | M010 | Grilled Cheese Sandwich with Tomato Soup | 508 | 1 |
| MI011 | M011 | Classic Breakfast Combo(Omelette and Toast) | 301 | 0 |
| MI012 | M012 | Shrimp Stir-Fry with Rice | 112 | 0 |
| MI013 | M013 | Classic Breakfast Combo(Omelette and Toast) | 301 | 1 |
| MI014 | M014 | Vegetarian Pasta Primavera | 324 | 1 |
| MI015 | M015 | Fruit and Yogurt Parfait | 112 | 1 |
| MI016 | M016 | Grilled Chicken Caesar Salad | 450 | 0 |
| MI017 | M017 | Grilled Chicken Caesar Salad | 450 | 0 |
| MI018 | M018 | Grilled Chicken Caesar Salad | 450 | 0 |
| MI019 | M019 | Vegetarian Pasta Primavera | 324 | 1 |
| MI020 | M020 | Shrimp Stir-Fry with Rice | 112 | 0 |
| MI021 | M021 | Fruit and Yogurt Parfait | 112 | 1 |
| MI022 | M022 | Vegetarian Pasta Primavera | 324 | 1 |
| MI023 | M023 | Classic Burger with Sweet Potato Fries | 458 | 1 |
| MI024 | M024 | Shrimp Stir-Fry with Rice | 112 | 0 |
| MI025 | M025 | Chicken Alfredo | 211 | 0 |
| MI026 | M026 | Pancakes with Syrup | 208 | 1 |
| MI027 | M027 | Shrimp Stir-Fry with Rice | 112 | 0 |
| MI028 | M028 | Classic Breakfast Combo(Omelette and Toast) | 301 | 0 |
| MI029 | M029 | Vegetarian Pasta Primavera | 324 | 1 |
| NULL | NULL | NULL | NULL | NULL |

## Table 7. Special Event Menu Items

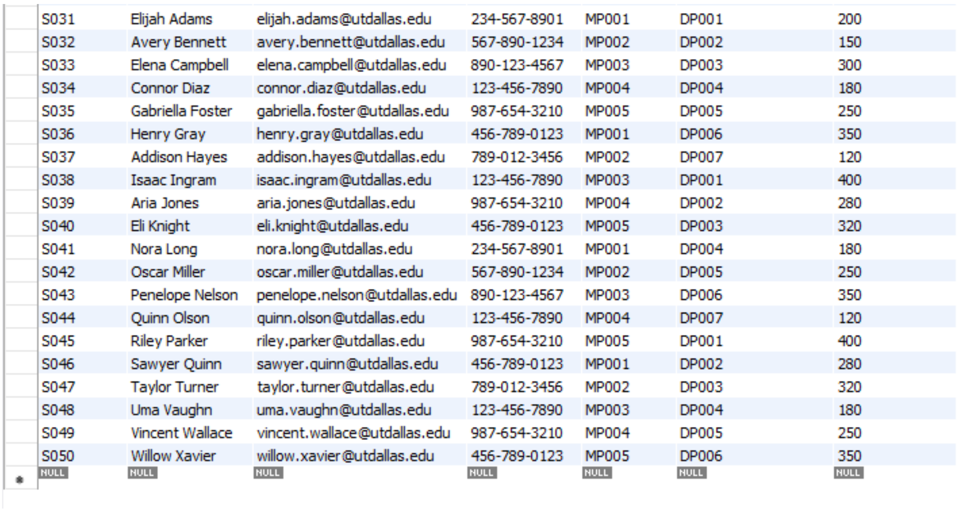


| SpecialMenuItemID | EventID | DishName | Calories | IsVegetarian |
|---|---|---|---|---|
| SMI001 | EVT001 | Fancy Appetizer Platter | 301 | 1 |
| SMI002 | EVT002 | Chocolate Cake | 700 | 0 |
| SMI003 | EVT003 | Tiramisu | 892 | 0 |
| SMI004 | EVT004 | New York Cheesecake | 623 | 0 |
| SMI005 | EVT005 | Breakfast Burrito | 598 | 0 |
| SMI006 | EVT006 | Fancy Appetizer Platter | 301 | 1 |
| SMI007 | EVT007 | Chocolate Cake | 700 | 0 |
| SMI008 | EVT008 | Tiramisu | 892 | 0 |
| SMI009 | EVT009 | New York Cheesecake | 623 | 0 |
| SMI010 | EVT010 | Breakfast Burrito | 598 | 0 |
| SMI011 | EVT011 | Fancy Appetizer Platter | 700 | 1 |
| NULL | NULL | NULL | NULL | NULL |

## Table 8. Students

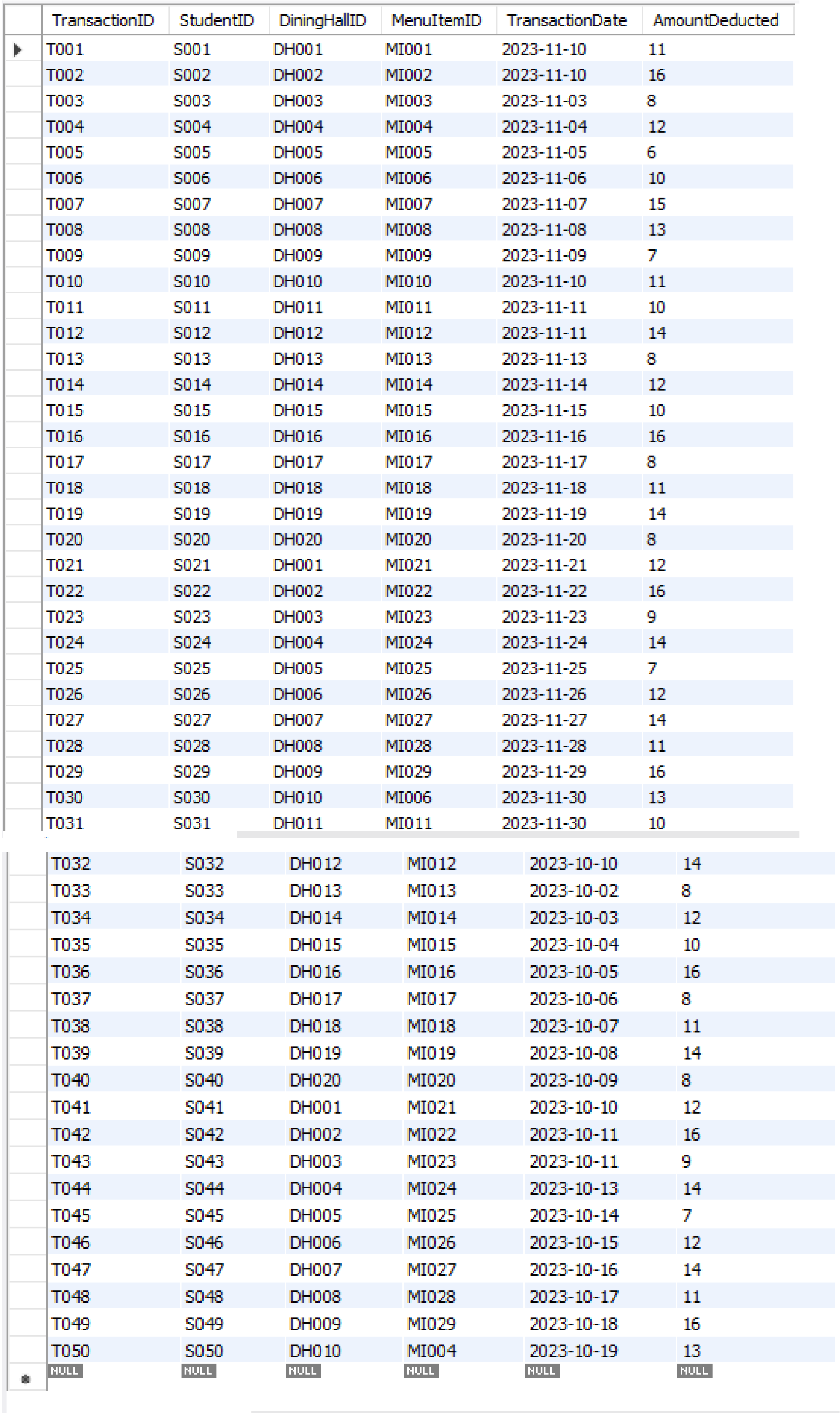


| StudentID | FullName | Email | Phone | MealPlanID | DietaryPreferenceID | BalanceAmount |
|---|---|---|---|---|---|---|
| S001 | John Doe | john.doe@utdallas.edu | 123-456-7890 | MP001 | DP001 | 200 |
| S002 | Jane Smith | jane.smith@utdallas.edu | 987-654-3210 | MP002 | DP002 | 150 |
| S003 | Robert Johnson | robert.johnson@utdallas.edu | 456-789-0123 | MP003 | DP003 | 300 |
| S004 | Emily White | emily.white@utdallas.edu | 789-012-3456 | MP004 | DP004 | 180 |
| S005 | Daniel Adams | daniel.adams@utdallas.edu | 234-567-8901 | MP005 | DP005 | 250 |
| S006 | Sophia Brown | sophia.brown@utdallas.edu | 567-890-1234 | MP001 | DP006 | 350 |
| S007 | William Clark | william.clark@utdallas.edu | 890-123-4567 | MP002 | DP007 | 120 |
| S008 | Olivia Davis | olivia.davis@utdallas.edu | 123-456-7890 | MP003 | DP001 | 400 |
| S009 | James Evans | james.evans@utdallas.edu | 987-654-3210 | MP004 | DP002 | 280 |
| S010 | Emma Fisher | emma.fisher@utdallas.edu | 456-789-0123 | MP005 | DP003 | 320 |
| S011 | Aiden Garcia | aiden.garcia@utdallas.edu | 234-567-8901 | MP001 | DP006 | 280 |
| S012 | Mia Hall | mia.hall@utdallas.edu | 567-890-1234 | MP002 | DP007 | 150 |
| S013 | Lucas Hill | lucas.hill@utdallas.edu | 890-123-4567 | MP003 | DP001 | 200 |
| S014 | Ava Jenkins | ava.jenkins@utdallas.edu | 123-456-7890 | MP004 | DP002 | 350 |
| S015 | Liam King | liam.king@utdallas.edu | 987-654-3210 | MP005 | DP003 | 180 |
| S016 | Isabella Lewis | isabella.lewis@utdallas.edu | 456-789-0123 | MP001 | DP004 | 300 |
| S017 | Jackson Martin | jackson.martin@utdallas.edu | 789-012-3456 | MP002 | DP005 | 250 |
| S018 | Sophie Mitchell | sophie.mitchell@utdallas.edu | 123-456-7890 | MP003 | DP006 | 400 |
| S019 | Oliver Nelson | oliver.nelson@utdallas.edu | 987-654-3210 | MP004 | DP007 | 320 |
| S020 | Emma Owen | emma.owen@utdallas.edu | 456-789-0123 | MP005 | DP001 | 150 |
| S021 | Carter Parker | carter.parker@utdallas.edu | 789-012-3456 | MP001 | DP002 | 280 |
| S022 | Chloe Perry | chloe.perry@utdallas.edu | 123-456-7890 | MP002 | DP003 | 350 |
| S023 | Ethan Reed | ethan.reed@utdallas.edu | 987-654-3210 | MP003 | DP004 | 200 |
| S024 | Aria Richardson | aria.richardson@utdallas.edu | 234-567-8901 | MP004 | DP005 | 180 |
| S025 | Lucas Ross | lucas.ross@utdallas.edu | 567-890-1234 | MP005 | DP006 | 320 |
| S026 | Zoe Russell | zoe.russell@utdallas.edu | 890-123-4567 | MP001 | DP007 | 150 |
| S027 | Leo Simmons | leo.simmons@utdallas.edu | 123-456-7890 | MP002 | DP001 | 250 |
| S028 | Ava Stewart | ava.stewart@utdallas.edu | 987-654-3210 | MP003 | DP002 | 300 |
| S029 | Logan Taylor | logan.taylor@utdallas.edu | 234-567-8901 | MP004 | DP003 | 400 |
| S030 | Lily Turner | lily.turner@utdallas.edu | 567-890-1234 | MP005 | DP004 | 280 |
| S031 | Elijah Adams | elijah.adams@utdallas.edu | 234-567-8901 | MP001 | DP001 | 200 |



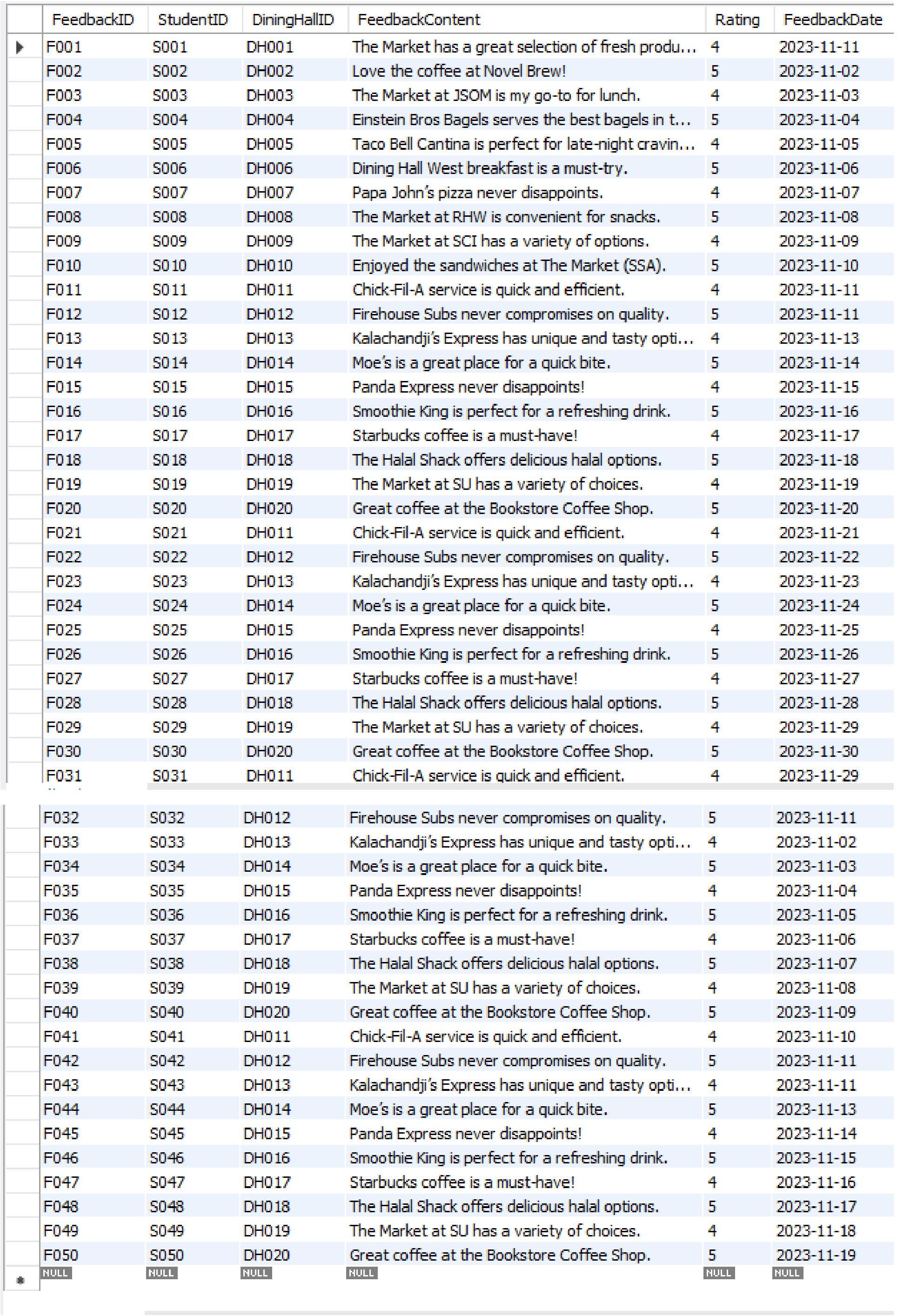
| | | | | | | |
|---|---|---|---|---|---|---|
| S031 | Elijah Adams | elijah.adams@utdallas.edu | 234-567-8901 | MP001 | DP001 | 200 |
| S032 | Avery Bennett | avery.bennett@utdallas.edu | 567-890-1234 | MP002 | DP002 | 150 |
| S033 | Elena Campbell | elena.campbell@utdallas.edu | 890-123-4567 | MP003 | DP003 | 300 |
| S034 | Connor Diaz | connor.diaz@utdallas.edu | 123-456-7890 | MP004 | DP004 | 180 |
| S035 | Gabriella Foster | gabriella.foster@utdallas.edu | 987-654-3210 | MP005 | DP005 | 250 |
| S036 | Henry Gray | henry.gray@utdallas.edu | 456-789-0123 | MP001 | DP006 | 350 |
| S037 | Addison Hayes | addison.hayes@utdallas.edu | 789-012-3456 | MP002 | DP007 | 120 |
| S038 | Isaac Ingram | isaac.ingram@utdallas.edu | 123-456-7890 | MP003 | DP001 | 400 |
| S039 | Aria Jones | aria.jones@utdallas.edu | 987-654-3210 | MP004 | DP002 | 280 |
| S040 | Eli Knight | eli.knight@utdallas.edu | 456-789-0123 | MP005 | DP003 | 320 |
| S041 | Nora Long | nora.long@utdallas.edu | 234-567-8901 | MP001 | DP004 | 180 |
| S042 | Oscar Miller | oscar.miller@utdallas.edu | 567-890-1234 | MP002 | DP005 | 250 |
| S043 | Penelope Nelson | penelope.nelson@utdallas.edu | 890-123-4567 | MP003 | DP006 | 350 |
| S044 | Quinn Olson | quinn.olson@utdallas.edu | 123-456-7890 | MP004 | DP007 | 120 |
| S045 | Riley Parker | riley.parker@utdallas.edu | 987-654-3210 | MP005 | DP001 | 400 |
| S046 | Sawyer Quinn | sawyer.quinn@utdallas.edu | 456-789-0123 | MP001 | DP002 | 280 |
| S047 | Taylor Turner | taylor.turner@utdallas.edu | 789-012-3456 | MP002 | DP003 | 320 |
| S048 | Uma Vaughn | uma.vaughn@utdallas.edu | 123-456-7890 | MP003 | DP004 | 180 |
| S049 | Vincent Wallace | vincent.wallace@utdallas.edu | 987-654-3210 | MP004 | DP005 | 250 |
| S050 | Willow Xavier | willow.xavier@utdallas.edu | 456-789-0123 | MP005 | DP006 | 350 |
| NULL | NULL | NULL | NULL | NULL | NULL | NULL |

**Table 9. Student Transactions**



| TransactionID | StudentID | DiningHallID | MenuItemID | TransactionDate | AmountDeducted |
|---|---|---|---|---|---|
| T001 | S001 | DH001 | MI001 | 2023-11-10 | 11 |
| T002 | S002 | DH002 | MI002 | 2023-11-10 | 16 |
| T003 | S003 | DH003 | MI003 | 2023-11-03 | 8 |
| T004 | S004 | DH004 | MI004 | 2023-11-04 | 12 |
| T005 | S005 | DH005 | MI005 | 2023-11-05 | 6 |
| T006 | S006 | DH006 | MI006 | 2023-11-06 | 10 |
| T007 | S007 | DH007 | MI007 | 2023-11-07 | 15 |
| T008 | S008 | DH008 | MI008 | 2023-11-08 | 13 |
| T009 | S009 | DH009 | MI009 | 2023-11-09 | 7 |
| T010 | S010 | DH010 | MI010 | 2023-11-10 | 11 |
| T011 | S011 | DH011 | MI011 | 2023-11-11 | 10 |
| T012 | S012 | DH012 | MI012 | 2023-11-11 | 14 |
| T013 | S013 | DH013 | MI013 | 2023-11-13 | 8 |
| T014 | S014 | DH014 | MI014 | 2023-11-14 | 12 |
| T015 | S015 | DH015 | MI015 | 2023-11-15 | 10 |
| T016 | S016 | DH016 | MI016 | 2023-11-16 | 16 |
| T017 | S017 | DH017 | MI017 | 2023-11-17 | 8 |
| T018 | S018 | DH018 | MI018 | 2023-11-18 | 11 |
| T019 | S019 | DH019 | MI019 | 2023-11-19 | 14 |
| T020 | S020 | DH020 | MI020 | 2023-11-20 | 8 |
| T021 | S021 | DH001 | MI021 | 2023-11-21 | 12 |
| T022 | S022 | DH002 | MI022 | 2023-11-22 | 16 |
| T023 | S023 | DH003 | MI023 | 2023-11-23 | 9 |
| T024 | S024 | DH004 | MI024 | 2023-11-24 | 14 |
| T025 | S025 | DH005 | MI025 | 2023-11-25 | 7 |
| T026 | S026 | DH006 | MI026 | 2023-11-26 | 12 |
| T027 | S027 | DH007 | MI027 | 2023-11-27 | 14 |
| T028 | S028 | DH008 | MI028 | 2023-11-28 | 11 |
| T029 | S029 | DH009 | MI029 | 2023-11-29 | 16 |
| T030 | S030 | DH010 | MI006 | 2023-11-30 | 13 |
| T031 | S031 | DH011 | MI011 | 2023-11-30 | 10 |
| T032 | S032 | DH012 | MI012 | 2023-10-10 | 14 |
| T033 | S033 | DH013 | MI013 | 2023-10-02 | 8 |
| T034 | S034 | DH014 | MI014 | 2023-10-03 | 12 |
| T035 | S035 | DH015 | MI015 | 2023-10-04 | 10 |
| T036 | S036 | DH016 | MI016 | 2023-10-05 | 16 |
| T037 | S037 | DH017 | MI017 | 2023-10-06 | 8 |
| T038 | S038 | DH018 | MI018 | 2023-10-07 | 11 |
| T039 | S039 | DH019 | MI019 | 2023-10-08 | 14 |
| T040 | S040 | DH020 | MI020 | 2023-10-09 | 8 |
| T041 | S041 | DH001 | MI021 | 2023-10-10 | 12 |
| T042 | S042 | DH002 | MI022 | 2023-10-11 | 16 |
| T043 | S043 | DH003 | MI023 | 2023-10-11 | 9 |
| T044 | S044 | DH004 | MI024 | 2023-10-13 | 14 |
| T045 | S045 | DH005 | MI025 | 2023-10-14 | 7 |
| T046 | S046 | DH006 | MI026 | 2023-10-15 | 12 |
| T047 | S047 | DH007 | MI027 | 2023-10-16 | 14 |
| T048 | S048 | DH008 | MI028 | 2023-10-17 | 11 |
| T049 | S049 | DH009 | MI029 | 2023-10-18 | 16 |
| T050 | S050 | DH010 | MI004 | 2023-10-19 | 13 |
| NULL | NULL | NULL | NULL | NULL | NULL |

Table 10. Feedback



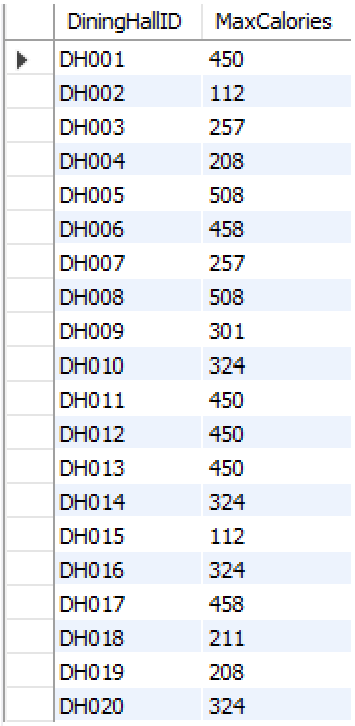
| FeedbackID | StudentID | DiningHallID | FeedbackContent | Rating | FeedbackDate |
|---|---|---|---|---|---|
| F001 | S001 | DH001 | The Market has a great selection of fresh produ… | 4 | 2023-11-11 |
| F002 | S002 | DH002 | Love the coffee at Novel Brew! | 5 | 2023-11-02 |
| F003 | S003 | DH003 | The Market at JSOM is my go-to for lunch. | 4 | 2023-11-03 |
| F004 | S004 | DH004 | Einstein Bros Bagels serves the best bagels in t… | 5 | 2023-11-04 |
| F005 | S005 | DH005 | Taco Bell Cantina is perfect for late-night cravin… | 4 | 2023-11-05 |
| F006 | S006 | DH006 | Dining Hall West breakfast is a must-try. | 5 | 2023-11-06 |
| F007 | S007 | DH007 | Papa John’s pizza never disappoints. | 4 | 2023-11-07 |
| F008 | S008 | DH008 | The Market at RHW is convenient for snacks. | 5 | 2023-11-08 |
| F009 | S009 | DH009 | The Market at SCI has a variety of options. | 4 | 2023-11-09 |
| F010 | S010 | DH010 | Enjoyed the sandwiches at The Market (SSA). | 5 | 2023-11-10 |
| F011 | S011 | DH011 | Chick-Fil-A service is quick and efficient. | 4 | 2023-11-11 |
| F012 | S012 | DH012 | Firehouse Subs never compromises on quality. | 5 | 2023-11-11 |
| F013 | S013 | DH013 | Kalachandji’s Express has unique and tasty opti… | 4 | 2023-11-13 |
| F014 | S014 | DH014 | Moe’s is a great place for a quick bite. | 5 | 2023-11-14 |
| F015 | S015 | DH015 | Panda Express never disappoints! | 4 | 2023-11-15 |
| F016 | S016 | DH016 | Smoothie King is perfect for a refreshing drink. | 5 | 2023-11-16 |
| F017 | S017 | DH017 | Starbucks coffee is a must-have! | 4 | 2023-11-17 |
| F018 | S018 | DH018 | The Halal Shack offers delicious halal options. | 5 | 2023-11-18 |
| F019 | S019 | DH019 | The Market at SU has a variety of choices. | 4 | 2023-11-19 |
| F020 | S020 | DH020 | Great coffee at the Bookstore Coffee Shop. | 5 | 2023-11-20 |
| F021 | S021 | DH011 | Chick-Fil-A service is quick and efficient. | 4 | 2023-11-21 |
| F022 | S022 | DH012 | Firehouse Subs never compromises on quality. | 5 | 2023-11-22 |
| F023 | S023 | DH013 | Kalachandji’s Express has unique and tasty opti… | 4 | 2023-11-23 |
| F024 | S024 | DH014 | Moe’s is a great place for a quick bite. | 5 | 2023-11-24 |
| F025 | S025 | DH015 | Panda Express never disappoints! | 4 | 2023-11-25 |
| F026 | S026 | DH016 | Smoothie King is perfect for a refreshing drink. | 5 | 2023-11-26 |
| F027 | S027 | DH017 | Starbucks coffee is a must-have! | 4 | 2023-11-27 |
| F028 | S028 | DH018 | The Halal Shack offers delicious halal options. | 5 | 2023-11-28 |
| F029 | S029 | DH019 | The Market at SU has a variety of choices. | 4 | 2023-11-29 |
| F030 | S030 | DH020 | Great coffee at the Bookstore Coffee Shop. | 5 | 2023-11-30 |
| F031 | S031 | DH011 | Chick-Fil-A service is quick and efficient. | 4 | 2023-11-29 |
| F032 | S032 | DH012 | Firehouse Subs never compromises on quality. | 5 | 2023-11-11 |
| F033 | S033 | DH013 | Kalachandji’s Express has unique and tasty opti… | 4 | 2023-11-02 |
| F034 | S034 | DH014 | Moe’s is a great place for a quick bite. | 5 | 2023-11-03 |
| F035 | S035 | DH015 | Panda Express never disappoints! | 4 | 2023-11-04 |
| F036 | S036 | DH016 | Smoothie King is perfect for a refreshing drink. | 5 | 2023-11-05 |
| F037 | S037 | DH017 | Starbucks coffee is a must-have! | 4 | 2023-11-06 |
| F038 | S038 | DH018 | The Halal Shack offers delicious halal options. | 5 | 2023-11-07 |
| F039 | S039 | DH019 | The Market at SU has a variety of choices. | 4 | 2023-11-08 |
| F040 | S040 | DH020 | Great coffee at the Bookstore Coffee Shop. | 5 | 2023-11-09 |
| F041 | S041 | DH011 | Chick-Fil-A service is quick and efficient. | 4 | 2023-11-10 |
| F042 | S042 | DH012 | Firehouse Subs never compromises on quality. | 5 | 2023-11-11 |
| F043 | S043 | DH013 | Kalachandji’s Express has unique and tasty opti… | 4 | 2023-11-11 |
| F044 | S044 | DH014 | Moe’s is a great place for a quick bite. | 5 | 2023-11-13 |
| F045 | S045 | DH015 | Panda Express never disappoints! | 4 | 2023-11-14 |
| F046 | S046 | DH016 | Smoothie King is perfect for a refreshing drink. | 5 | 2023-11-15 |
| F047 | S047 | DH017 | Starbucks coffee is a must-have! | 4 | 2023-11-16 |
| F048 | S048 | DH018 | The Halal Shack offers delicious halal options. | 5 | 2023-11-17 |
| F049 | S049 | DH019 | The Market at SU has a variety of choices. | 4 | 2023-11-18 |
| F050 | S050 | DH020 | Great coffee at the Bookstore Coffee Shop. | 5 | 2023-11-19 |
| NULL | NULL | NULL | NULL | NULL | NULL |

# 6. Query Scenario Design

## Query 01: Retrieve Students with the most recent Transaction Date and their Total amount deducted

```
SELECT s.StudentID, s.FullName, st.TransactionDate AS RecentTransactionDate, st.AmountDeducted AS
TotalAmountDeducted
FROM Students s
JOIN StudentTransactions st ON s.StudentID = st.StudentID
WHERE st.TransactionDate = (
   SELECT MAX(TransactionDate)
   FROM StudentTransactions
   WHERE StudentID = s.StudentID
);
```

## Result of Query 1

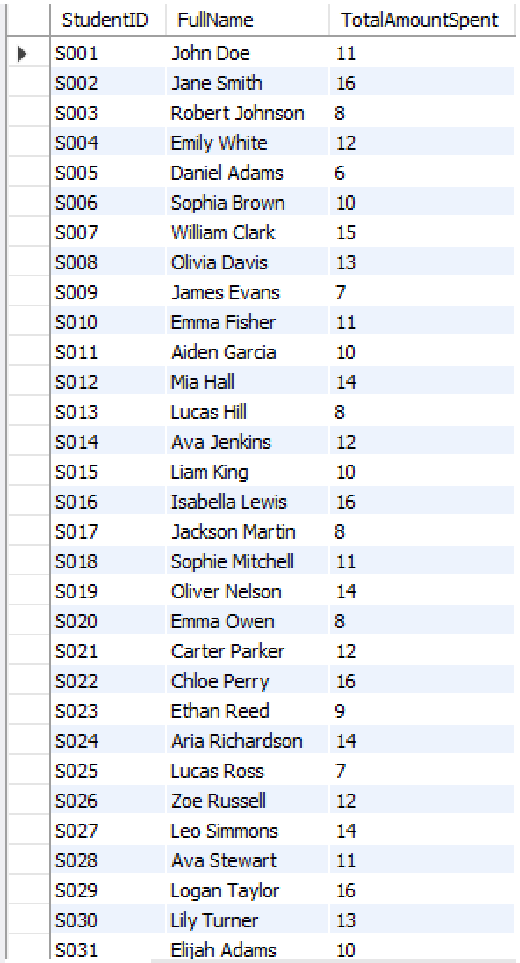


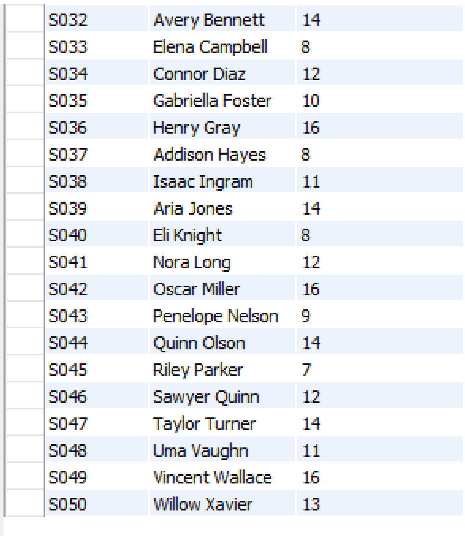
| DiningHallID | MaxCalories |
|---|---|
| DH001 | 450 |
| DH002 | 112 |
| DH003 | 257 |
| DH004 | 208 |
| DH005 | 508 |
| DH006 | 458 |
| DH007 | 257 |
| DH008 | 508 |
| DH009 | 301 |
| DH010 | 324 |
| DH011 | 450 |
| DH012 | 450 |
| DH013 | 450 |
| DH014 | 324 |
| DH015 | 112 |
| DH016 | 324 |
| DH017 | 458 |
| DH018 | 211 |
| DH019 | 208 |
| DH020 | 324 |

## Query 02: Find the total amount spent by each student on transactions.

```
SELECT s.StudentID, s.FullName, SUM(st.AmountDeducted) AS TotalAmountSpent
FROM Students s
LEFT JOIN StudentTransactions st ON s.StudentID = st.StudentID
GROUP BY s.StudentID, s.FullName;
```

## Result of Query 2





| StudentID | FullName | TotalAmountSpent |
|---|---|---|
| S001 | John Doe | 11 |
| S002 | Jane Smith | 16 |
| S003 | Robert Johnson | 8 |
| S004 | Emily White | 12 |
| S005 | Daniel Adams | 6 |
| S006 | Sophia Brown | 10 |
| S007 | William Clark | 15 |
| S008 | Olivia Davis | 13 |
| S009 | James Evans | 7 |
| S010 | Emma Fisher | 11 |
| S011 | Aiden Garcia | 10 |
| S012 | Mia Hall | 14 |
| S013 | Lucas Hill | 8 |
| S014 | Ava Jenkins | 12 |
| S015 | Liam King | 10 |
| S016 | Isabella Lewis | 16 |
| S017 | Jackson Martin | 8 |
| S018 | Sophie Mitchell | 11 |
| S019 | Oliver Nelson | 14 |
| S020 | Emma Owen | 8 |
| S021 | Carter Parker | 12 |
| S022 | Chloe Perry | 16 |
| S023 | Ethan Reed | 9 |
| S024 | Aria Richardson | 14 |
| S025 | Lucas Ross | 7 |
| S026 | Zoe Russell | 12 |
| S027 | Leo Simmons | 14 |
| S028 | Ava Stewart | 11 |
| S029 | Logan Taylor | 16 |
| S030 | Lily Turner | 13 |
| S031 | Elijah Adams | 10 |
| S032 | Avery Bennett | 14 |
| S033 | Elena Campbell | 8 |
| S034 | Connor Diaz | 12 |
| S035 | Gabriella Foster | 10 |
| S036 | Henry Gray | 16 |
| S037 | Addison Hayes | 8 |
| S038 | Isaac Ingram | 11 |
| S039 | Aria Jones | 14 |
| S040 | Eli Knight | 8 |
| S041 | Nora Long | 12 |
| S042 | Oscar Miller | 16 |
| S043 | Penelope Nelson | 9 |
| S044 | Quinn Olson | 14 |
| S045 | Riley Parker | 7 |
| S046 | Sawyer Quinn | 12 |
| S047 | Taylor Turner | 14 |
| S048 | Uma Vaughn | 11 |
| S049 | Vincent Wallace | 16 |
| S050 | Willow Xavier | 13 |

## Query 03: Retrieve the special events along with the number of menu items associated with each event.

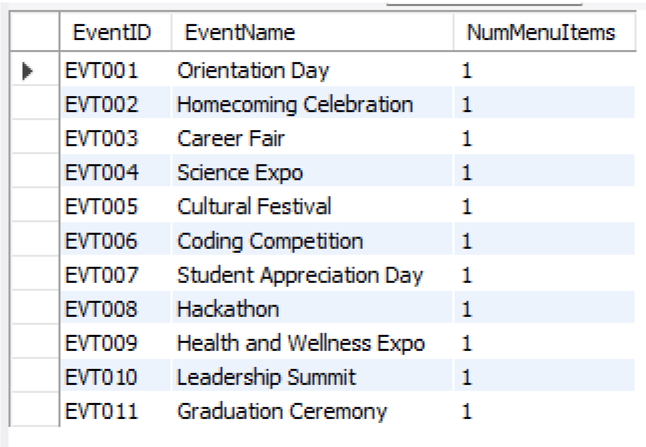
SELECT se.EventID, se.EventName, COUNT(semi.SpecialMenuItemID) AS NumMenuItems

FROM SpecialEvents se

LEFT JOIN SpecialEventMenuItems semi ON se.EventID = semi.EventID

GROUP BY se.EventID, se.EventName;

## Result of Query 3

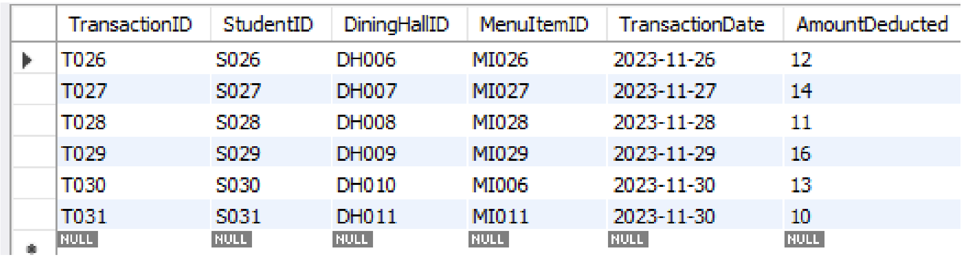


| EventID | EventName | NumMenuItems |
|---|---|---|
| EVT001 | Orientation Day | 1 |
| EVT002 | Homecoming Celebration | 1 |
| EVT003 | Career Fair | 1 |
| EVT004 | Science Expo | 1 |
| EVT005 | Cultural Festival | 1 |
| EVT006 | Coding Competition | 1 |
| EVT007 | Student Appreciation Day | 1 |
| EVT008 | Hackathon | 1 |
| EVT009 | Health and Wellness Expo | 1 |
| EVT010 | Leadership Summit | 1 |
| EVT011 | Graduation Ceremony | 1 |

## Query 04: Find transactions made in the last 7 days:

SELECT * FROM StudentTransactions WHERE TransactionDate >= CURDATE() - INTERVAL 7 DAY;

## Result of Query 4

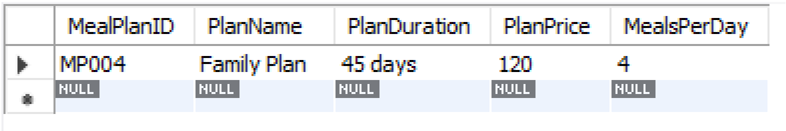


| TransactionID | StudentID | DiningHallID | MenuItemID | TransactionDate | AmountDeducted |
|---|---|---|---|---|---|
| T026 | S026 | DH006 | MI026 | 2023-11-26 | 12 |
| T027 | S027 | DH007 | MI027 | 2023-11-27 | 14 |
| T028 | S028 | DH008 | MI028 | 2023-11-28 | 11 |
| T029 | S029 | DH009 | MI029 | 2023-11-29 | 16 |
| T030 | S030 | DH010 | MI006 | 2023-11-30 | 13 |
| T031 | S031 | DH011 | MI011 | 2023-11-30 | 10 |
| NULL | NULL | NULL | NULL | NULL | NULL |

## Query 05: List meal plans with more than 3 meals per day:

SELECT * FROM MealPlans WHERE MealsPerDay > 3;

## Result of Query 5

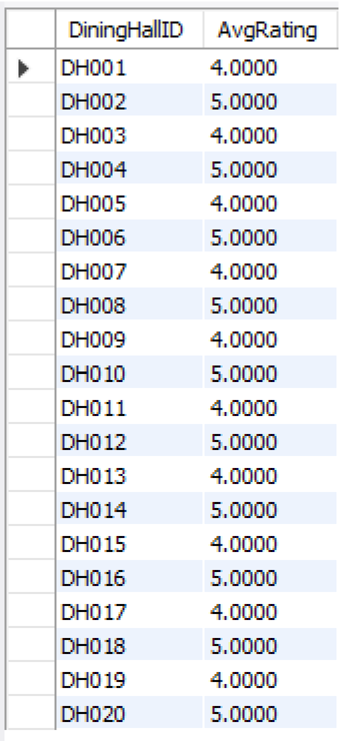


| MealPlanID | PlanName | PlanDuration | PlanPrice | MealsPerDay |
|---|---|---|---|---|
| MP004 | Family Plan | 45 days | 120 | 4 |
| NULL | NULL | NULL | NULL | NULL |

## Query 06: Retrieve the average rating of feedback for each dining hall.

```
SELECT f.DiningHallID, AVG(f.Rating) AS AvgRating
FROM Feedback f
GROUP BY f.DiningHallID;
```

## Result of Query 6

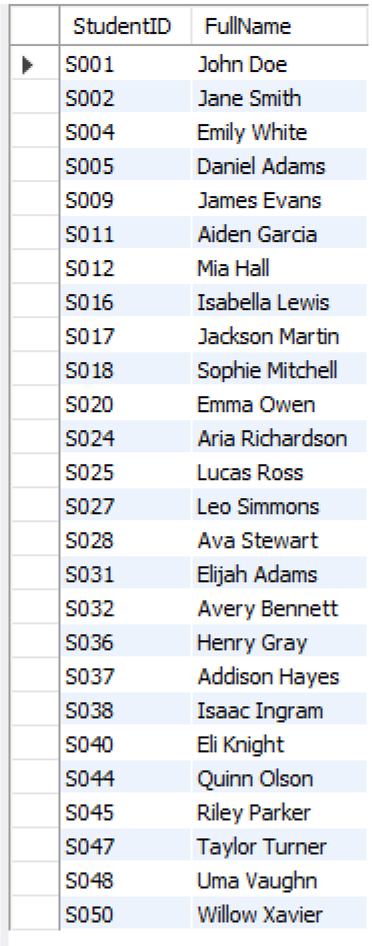


| DiningHallID | AvgRating |
|---|---|
| DH001 | 4.0000 |
| DH002 | 5.0000 |
| DH003 | 4.0000 |
| DH004 | 5.0000 |
| DH005 | 4.0000 |
| DH006 | 5.0000 |
| DH007 | 4.0000 |
| DH008 | 5.0000 |
| DH009 | 4.0000 |
| DH010 | 5.0000 |
| DH011 | 4.0000 |
| DH012 | 5.0000 |
| DH013 | 4.0000 |
| DH014 | 5.0000 |
| DH015 | 4.0000 |
| DH016 | 5.0000 |
| DH017 | 4.0000 |
| DH018 | 5.0000 |
| DH019 | 4.0000 |
| DH020 | 5.0000 |

## Query 07: Find the students who have tried all vegetarian menu items.

```
SELECT s.StudentID, s.FullName
FROM Students s
JOIN MenuItems mi ON s.DietaryPreferenceID = mi.IsVegetarian
WHERE NOT EXISTS (
    SELECT m.MenuID
    FROM Menus m
    WHERE NOT EXISTS (
        SELECT st.MenuItemID
        FROM StudentTransactions st
        WHERE st.StudentID = s.StudentID AND st.MenuItemID = mi.MenuItemID
    )
);
```

## Result of Query 7

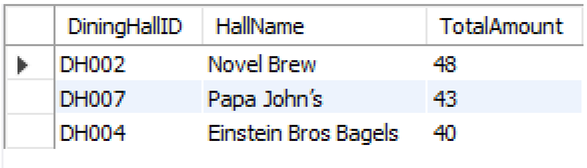


| StudentID | FullName |
|---|---|
| S001 | John Doe |
| S002 | Jane Smith |
| S004 | Emily White |
| S005 | Daniel Adams |
| S009 | James Evans |
| S011 | Aiden Garcia |
| S012 | Mia Hall |
| S016 | Isabella Lewis |
| S017 | Jackson Martin |
| S018 | Sophie Mitchell |
| S020 | Emma Owen |
| S024 | Aria Richardson |
| S025 | Lucas Ross |
| S027 | Leo Simmons |
| S028 | Ava Stewart |
| S031 | Elijah Adams |
| S032 | Avery Bennett |
| S036 | Henry Gray |
| S037 | Addison Hayes |
| S038 | Isaac Ingram |
| S040 | Eli Knight |
| S044 | Quinn Olson |
| S045 | Riley Parker |
| S047 | Taylor Turner |
| S048 | Uma Vaughn |
| S050 | Willow Xavier |

## Query 08: Retrieve the top 3 dining halls with the highest total transaction amounts.

```
SELECT dh.DiningHallID, dh.HallName, SUM(st.AmountDeducted) AS TotalAmount
FROM DiningHalls dh
LEFT JOIN StudentTransactions st ON dh.DiningHallID = st.DiningHallID
GROUP BY dh.DiningHallID, dh.HallName
ORDER BY TotalAmount DESC
LIMIT 3;
```

## Result of Query 8

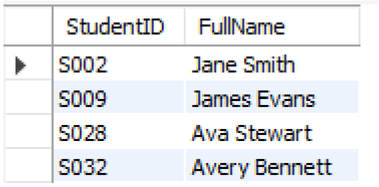


| DiningHallID | HallName | TotalAmount |
|---|---|---|
| DH002 | Novel Brew | 48 |
| DH007 | Papa John's | 43 |
| DH004 | Einstein Bros Bagels | 40 |

### Query 09: Find students who have a dietary preference for vegan but have made transactions for non-vegetarian menu items.

```
SELECT s.StudentID, s.FullName
FROM Students s
JOIN DietaryPreferences dp ON s.DietaryPreferenceID = dp.DietaryPreferenceID
JOIN StudentTransactions st ON s.StudentID = st.StudentID
JOIN MenuItems mi ON st.MenuItemID = mi.MenuItemID
WHERE dp.PreferenceName = 'Vegan' AND mi.IsVegetarian = FALSE;
```

### Result of Query 9

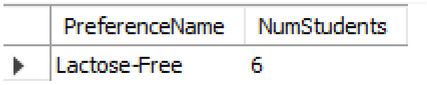


| | StudentID | FullName |
|---|---|---|
| ▶ | S002 | Jane Smith |
| | S009 | James Evans |
| | S028 | Ava Stewart |
| | S032 | Avery Bennett |

### Query 10: Retrieve the least popular dietary preference (the one with the fewest students).

```
SELECT dp.PreferenceName, COUNT(s.StudentID) AS NumStudents
FROM DietaryPreferences dp
LEFT JOIN Students s ON dp.DietaryPreferenceID = s.DietaryPreferenceID
GROUP BY dp.PreferenceName
ORDER BY NumStudents ASC
LIMIT 1;
```

### Result of Query 10

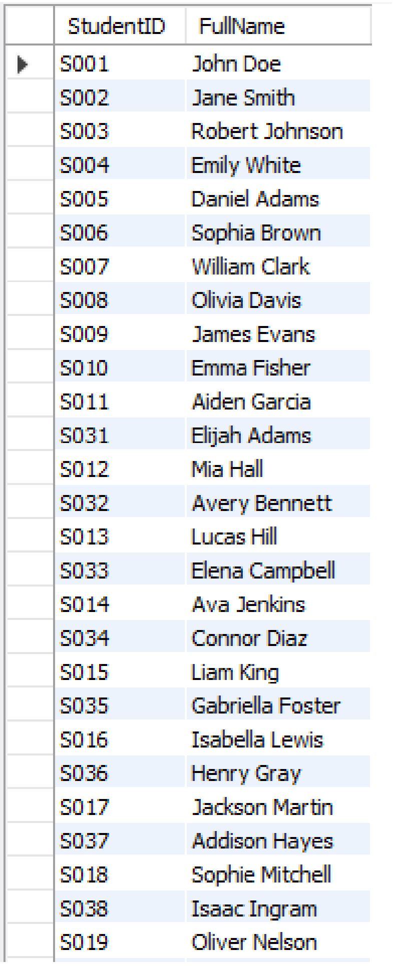
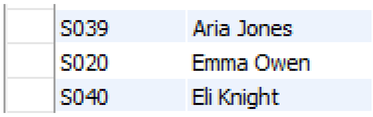


| | PreferenceName | NumStudents |
|---|---|---|
| ▶ | Lactose-Free | 6 |

### Query 11: Find the students who have given feedback for all dining halls they visited.

```
SELECT s.StudentID, s.FullName
FROM Students s
JOIN StudentTransactions st ON s.StudentID = st.StudentID
JOIN DiningHalls dh ON st.DiningHallID = dh.DiningHallID
WHERE NOT EXISTS (
    SELECT DISTINCT f.DiningHallID
    FROM Feedback f
    WHERE f.StudentID = s.StudentID AND f.DiningHallID != dh.DiningHallID
);
```

### Result of Query 11

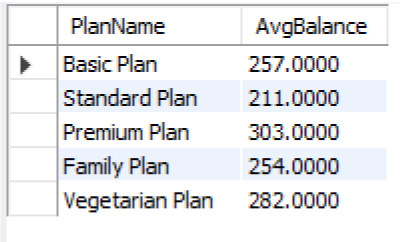



| StudentID | FullName |
|---|---|
| S001 | John Doe |
| S002 | Jane Smith |
| S003 | Robert Johnson |
| S004 | Emily White |
| S005 | Daniel Adams |
| S006 | Sophia Brown |
| S007 | William Clark |
| S008 | Olivia Davis |
| S009 | James Evans |
| S010 | Emma Fisher |
| S011 | Aiden Garcia |
| S031 | Elijah Adams |
| S012 | Mia Hall |
| S032 | Avery Bennett |
| S013 | Lucas Hill |
| S033 | Elena Campbell |
| S014 | Ava Jenkins |
| S034 | Connor Diaz |
| S015 | Liam King |
| S035 | Gabriella Foster |
| S016 | Isabella Lewis |
| S036 | Henry Gray |
| S017 | Jackson Martin |
| S037 | Addison Hayes |
| S018 | Sophie Mitchell |
| S038 | Isaac Ingram |
| S019 | Oliver Nelson |
| S039 | Aria Jones |
| S020 | Emma Owen |
| S040 | Eli Knight |

### Query 12: Retrieve the average balance amount for students with a specific meal plan.

```
SELECT m.PlanName, AVG(s.BalanceAmount) AS AvgBalance
FROM MealPlans m
JOIN Students s ON m.MealPlanID = s.MealPlanID
GROUP BY m.PlanName;
```

### Result of Query 12

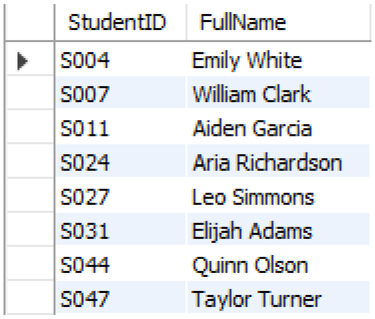


| PlanName | AvgBalance |
|---|---|
| Basic Plan | 257.0000 |
| Standard Plan | 211.0000 |
| Premium Plan | 303.0000 |
| Family Plan | 254.0000 |
| Vegetarian Plan | 282.0000 |

## Query 13: Find the students who have attended at least two special events in different dining halls.

```
SELECT s.StudentID, s.FullName
FROM Students s
JOIN StudentTransactions st ON s.StudentID = st.StudentID
JOIN SpecialEvents se ON st.DiningHallID = se.DiningHallID
GROUP BY s.StudentID, se.DiningHallID
HAVING COUNT(DISTINCT se.EventID) >= 2;
```

## Result of Query 13

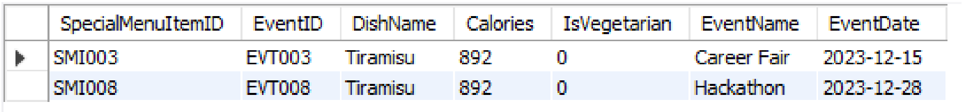


| StudentID | FullName |
|---|---|
| S004 | Emily White |
| S007 | William Clark |
| S011 | Aiden Garcia |
| S024 | Aria Richardson |
| S027 | Leo Simmons |
| S031 | Elijah Adams |
| S044 | Quinn Olson |
| S047 | Taylor Turner |

## Query 14: Find the special event menu items with the highest number of calories and list the corresponding event details:

```
SELECT semi.*, se.EventName, se.EventDate
FROM SpecialEventMenuItems semi
JOIN SpecialEvents se ON semi.EventID = se.EventID
WHERE semi.Calories = (SELECT MAX(Calories) FROM SpecialEventMenuItems);
```

## Result of Query 14

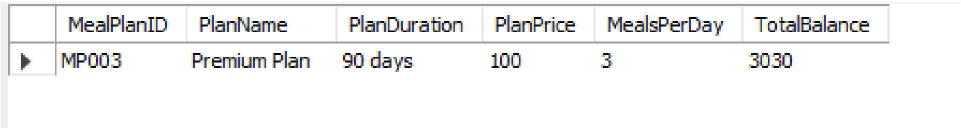


| SpecialMenuItemID | EventID | DishName | Calories | IsVegetarian | EventName | EventDate |
|---|---|---|---|---|---|---|
| SMI003 | EVT003 | Tiramisu | 892 | 0 | Career Fair | 2023-12-15 |
| SMI008 | EVT008 | Tiramisu | 892 | 0 | Hackathon | 2023-12-28 |

### Query 15: Retrieve the meal plan with the highest total balance amount of students subscribed to it:

```
SELECT mp.*, SUM(s.BalanceAmount) AS TotalBalance
FROM MealPlans mp
JOIN Students s ON mp.MealPlanID = s.MealPlanID
GROUP BY mp.MealPlanID
ORDER BY TotalBalance DESC
LIMIT 1;
```

### Result of Query 15



| MealPlanID | PlanName | PlanDuration | PlanPrice | MealsPerDay | TotalBalance |
|---|---|---|---|---|---|
| MP003 | Premium Plan | 90 days | 100 | 3 | 3030 |

## 7. MongoDB Implementation

We created two collections in MongoDB – Employee and Department
We then migrated these two collections in MySQL which can be further used to query more insightful information.

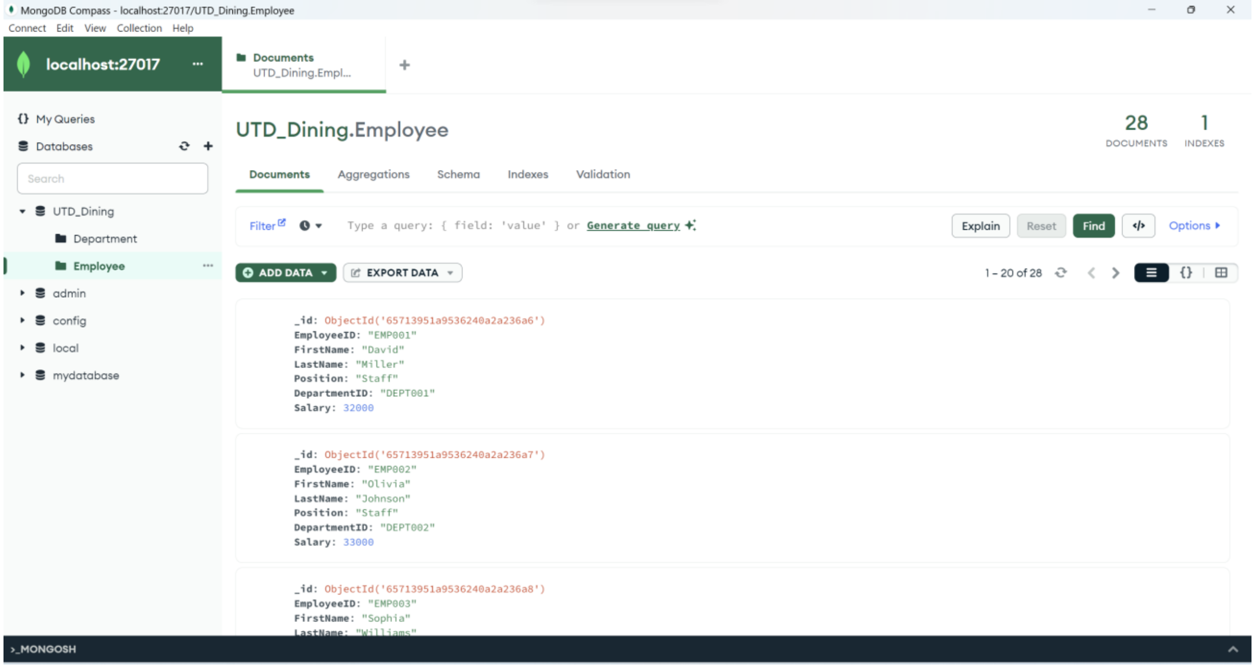
Process for implementing NoSQL

1. Created the database and collection in MongoDB
2. Created document type data of each entity required
3. Adding data into collection using ‘INSERT_MANY’
4. Implemented python code to generate SQL Script reading JSON file
5. Created table for each entity in MySQL workbench to add data from NoSQL using ‘CREATE TABLE’
6. Inserted values in the table using the SQL script generated by python code

```
import pymongo
client = pymongo.MongoClient("mongodb://localhost:27017")
db = client["UTD_Dining"]
Employee_Collection = db["Employee"]
Department_collection = db["Department"]

departments_data = [
    {
        "DepartmentID": "DEPT001",
        "DepartmentName": "Dining Services",
    },
        {
        "DepartmentID": "DEPT002",
        "DepartmentName": "Kitchen Operations",
    },
        {
        "DepartmentID": "DEPT003",
        "DepartmentName": "Customer Service",
    },
        {
        "DepartmentID": "DEPT004",
        "DepartmentName": "Facility Management",
    }

]

Department_collection.insert_many(departments_data)
```

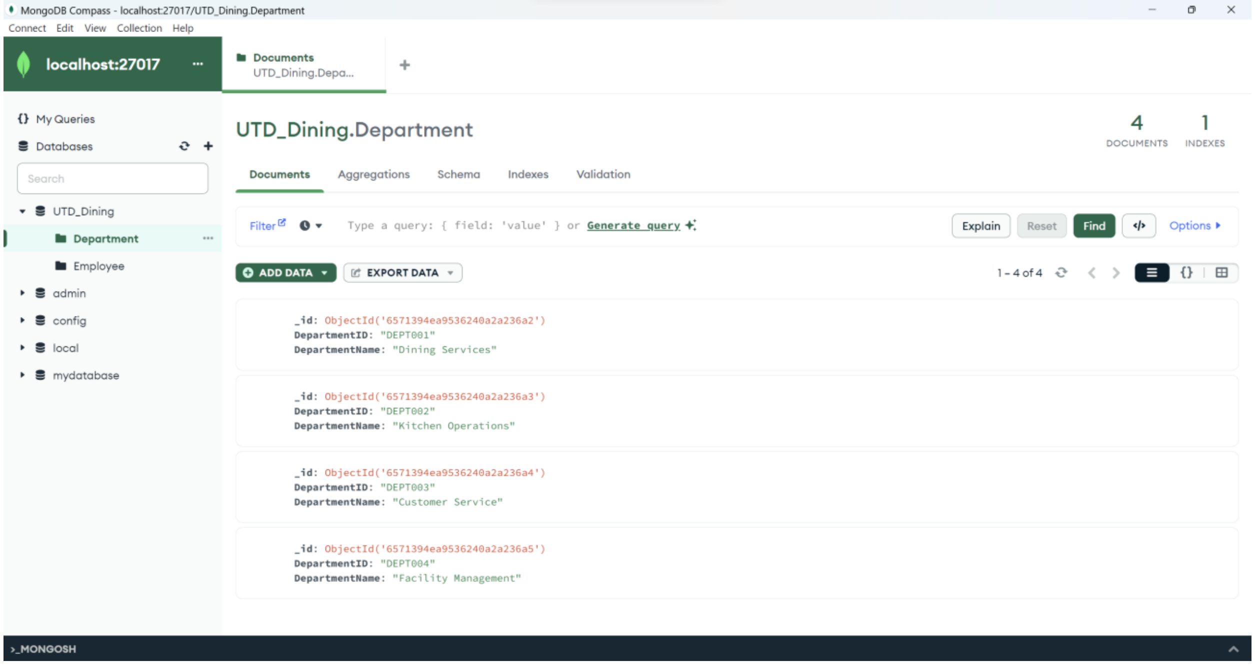
```
employees_data = [
    {
        "EmployeeID": f"EMP001",
        "FirstName": "David",
        "LastName": "Miller",
        "Position": "Staff",
        "DepartmentID": "DEPT001",
        "Salary": 32000
    },
    {
        "EmployeeID": f"EMP002",
        "FirstName": "Olivia",
        "LastName": "Johnson",
        "Position": "Staff",
        "DepartmentID": "DEPT002",
        "Salary": 33000
    },
    {
        "EmployeeID": f"EMP003",
        "FirstName": "Sophia",
        "LastName": "Williams",
        "Position": "Staff",
        "DepartmentID": "DEPT003",
        "Salary": 34000
    },
    {
        "EmployeeID": f"EMP004",
        "FirstName": "Michael",
        "LastName": "Brown",
        "Position": "Staff",
        "DepartmentID": "DEPT004",
        "Salary": 35000
    },
    {
        "EmployeeID": f"EMP005",
        "FirstName": "Emma",
        "LastName": "Jones",
        "Position": "Staff",
        "DepartmentID": "DEPT001",
        "Salary": 36000
    },
    {
        "EmployeeID": f"EMP006",
        "FirstName": "William",
        "LastName": "Garcia",
        "Position": "Staff",
        "DepartmentID": "DEPT002",
        "Salary": 37000
    },
    {
        "EmployeeID": f"EMP007",
        "FirstName": "Amelia",
        "LastName": "Smith",
        "Position": "Staff",
        "DepartmentID": "DEPT003",
        "Salary": 38000
    },
    {
        "EmployeeID": f"EMP008",
        "FirstName": "Daniel",
        "LastName": "Davis",
        "Position": "Manager",
        "DepartmentID": "DEPT004",
        "Salary": 50000
    },
    {
        "EmployeeID": f"EMP009",
        "FirstName": "Ava",
        "LastName": "Moore",
        "Position": "Staff",
        "DepartmentID": "DEPT001",
        "Salary": 48000
    },
    {
        "EmployeeID": f"EMP010",
        "FirstName": "Logan",
        "LastName": "Martin",
        "Position": "Staff",
        "DepartmentID": "DEPT002",
        "Salary": 30000
    },
    {
        "EmployeeID": f"EMP011",
        "FirstName": "Ella",
        "LastName": "Anderson",
        "Position": "Staff",
        "DepartmentID": "DEPT003",
        "Salary": 42000
    },
    {
        "EmployeeID": f"EMP012",
        "FirstName": "Caleb",
        "LastName": "Thompson",
        "Position": "Staff",
        "DepartmentID": "DEPT004",
        "Salary": 28000
    },
    {
        "EmployeeID": f"EMP013",
        "FirstName": "Mia",
        "LastName": "White",
        "Position": "Staff",
        "DepartmentID": "DEPT001",
        "Salary": 31000
    },
    {
        "EmployeeID": f"EMP014",
        "FirstName": "James",
        "LastName": "Harris",
        "Position": "Staff",
        "DepartmentID": "DEPT002",
        "Salary": 52000
    },
    {
        "EmployeeID": f"EMP015",
        "FirstName": "Evelyn",
        "LastName": "Martinez",
        "Position": "Staff",
        "DepartmentID": "DEPT003",
        "Salary": 45000
    },
    {
        "EmployeeID": f"EMP016",
        "FirstName": "Benjamin",
        "LastName": "Jackson",
        "Position": "Staff",
        "DepartmentID": "DEPT004",
        "Salary": 48000
    },
    {
        "EmployeeID": f"EMP017",
        "FirstName": "Lily",
        "LastName": "Clark",
        "Position": "Staff",
        "DepartmentID": "DEPT001",
        "Salary": 47000
    },
    {
        "EmployeeID": f"EMP018",
        "FirstName": "Aiden",
        "LastName": "Taylor",
        "Position": "Staff",
        "DepartmentID": "DEPT002",
        "Salary": 49000
    },
    {
        "EmployeeID": f"EMP019",
        "FirstName": "Zoe",
        "LastName": "Baker",
        "Position": "Staff",
        "DepartmentID": "DEPT003",
        "Salary": 51000
    },
    {
        "EmployeeID": f"EMP020",
        "FirstName": "Jackson",
        "LastName": "Cooper",
        "Position": "Staff",
        "DepartmentID": "DEPT004",
        "Salary": 32000
    },
    {
        "EmployeeID": f"EMP021",
        "FirstName": "Chloe",
        "LastName": "Hill",
        "Position": "Staff",
        "DepartmentID": "DEPT001",
        "Salary": 31000
    },
    {
        "EmployeeID": f"EMP022",
        "FirstName": "Daniel",
        "LastName": "Barnes",
        "Position": "Staff",
        "DepartmentID": "DEPT002",
        "Salary": 49000
    },
    {
        "EmployeeID": f"EMP023",
        "FirstName": "Grace",
        "LastName": "Ward",
        "Position": "Staff",
        "DepartmentID": "DEPT003",
        "Salary": 52000
    },
    {
        "EmployeeID": f"EMP024",
        "FirstName": "Mason",
        "LastName": "Carter",
        "Position": "Staff",
        "DepartmentID": "DEPT004",
        "Salary": 30000
    },
    {
        "EmployeeID": f"EMP025",
        "FirstName": "Layla",
        "LastName": "Fisher",
        "Position": "Staff",
        "DepartmentID": "DEPT001",
        "Salary": 28000
    },
    {
        "EmployeeID": f"EMP026",
        "FirstName": "Ethan",
        "LastName": "Perez",
        "Position": "Staff",
        "DepartmentID": "DEPT002",
        "Salary": 31000
    },
    {
        "EmployeeID": f"EMP027",
        "FirstName": "Madison",
        "LastName": "Foster",
        "Position": "Staff",
        "DepartmentID": "DEPT003",
        "Salary": 53000
    },
    {
        "EmployeeID": f"EMP028",
        "FirstName": "Liam",
        "LastName": "Hernandez",
        "Position": "Staff",
        "DepartmentID": "DEPT004",
        "Salary": 46000
    }
]

Employee_collection.insert_many(employees_data)
```

# Employee Collection created in MongoDB



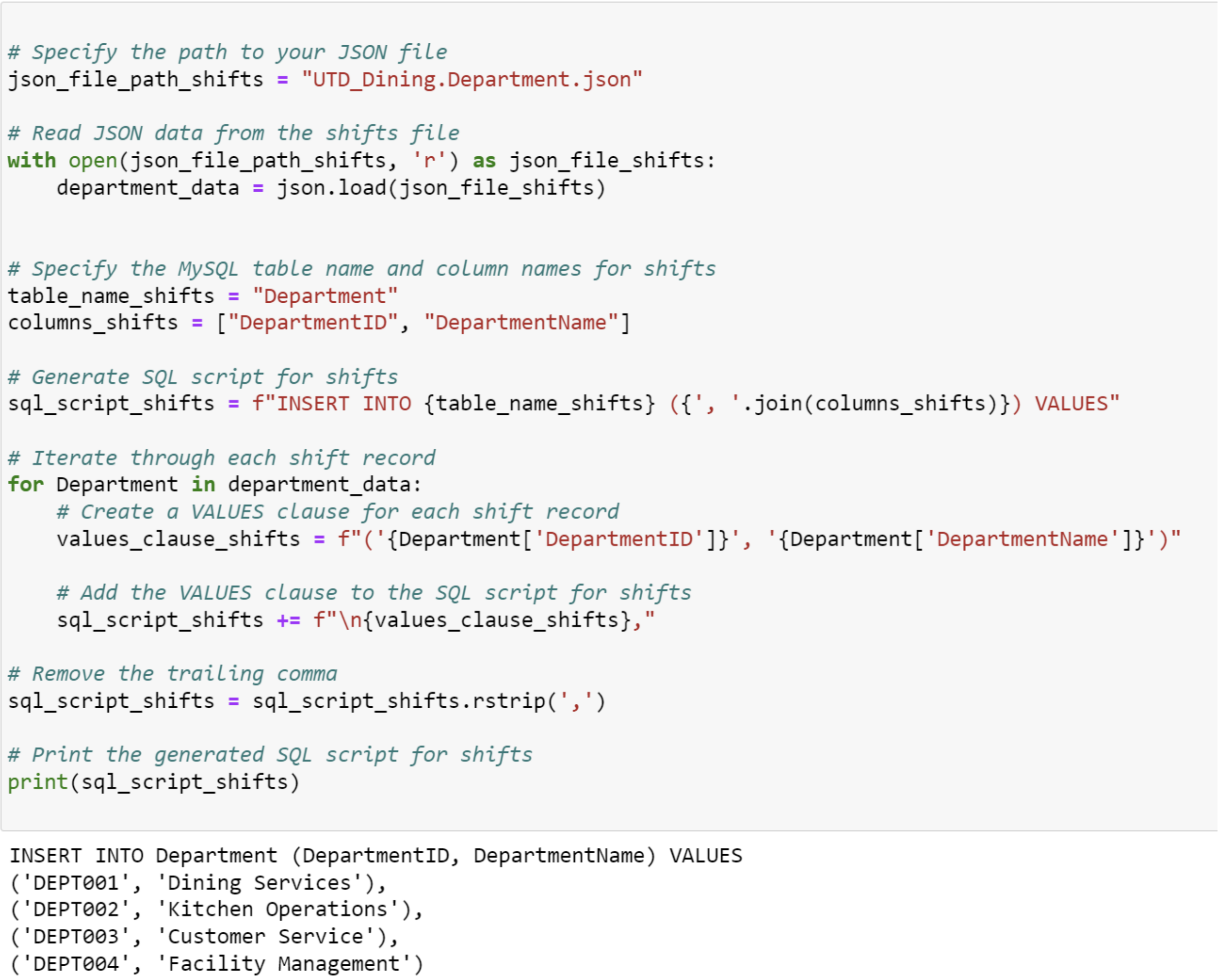
```
_id: ObjectId('65713951a9536240a2a236a6')
EmployeeID: "EMP001"
FirstName: "David"
LastName: "Miller"
Position: "Staff"
DepartmentID: "DEPT001"
Salary: 32000

_id: ObjectId('65713951a9536240a2a236a7')
EmployeeID: "EMP002"
FirstName: "Olivia"
LastName: "Johnson"
Position: "Staff"
DepartmentID: "DEPT002"
Salary: 33000

_id: ObjectId('65713951a9536240a2a236a8')
EmployeeID: "EMP003"
FirstName: "Sophia"
LastName: "Williams"
```

# Department Collection created in Mongodb



```
_id: ObjectId('6571394ea9536240a2a236a2')
DepartmentID: "DEPT001"
DepartmentName: "Dining Services"

_id: ObjectId('6571394ea9536240a2a236a3')
DepartmentID: "DEPT002"
DepartmentName: "Kitchen Operations"

_id: ObjectId('6571394ea9536240a2a236a4')
DepartmentID: "DEPT003"
DepartmentName: "Customer Service"

_id: ObjectId('6571394ea9536240a2a236a5')
DepartmentID: "DEPT004"
DepartmentName: "Facility Management"
```

# Implemented python code to generate SQL Script



```python
# Specify the path to your JSON file
json_file_path_shifts = "UTD_Dining.Department.json"

# Read JSON data from the shifts file
with open(json_file_path_shifts, 'r') as json_file_shifts:
    department_data = json.load(json_file_shifts)


# Specify the MySQL table name and column names for shifts
table_name_shifts = "Department"
columns_shifts = ["DepartmentID", "DepartmentName"]

# Generate SQL script for shifts
sql_script_shifts = f"INSERT INTO {table_name_shifts} ({', '.join(columns_shifts)}) VALUES"

# Iterate through each shift record
for Department in department_data:
    # Create a VALUES clause for each shift record
    values_clause_shifts = f"('{Department['DepartmentID']}', '{Department['DepartmentName']}')"

    # Add the VALUES clause to the SQL script for shifts
    sql_script_shifts += f"\n{values_clause_shifts},"

# Remove the trailing comma
sql_script_shifts = sql_script_shifts.rstrip(',')

# Print the generated SQL script for shifts
print(sql_script_shifts)
```

```
INSERT INTO Department (DepartmentID, DepartmentName) VALUES
('DEPT001', 'Dining Services'),
('DEPT002', 'Kitchen Operations'),
('DEPT003', 'Customer Service'),
('DEPT004', 'Facility Management')
```

# Inserted values in the table using the SQL script generated by python code as shown above.

## 8. Conclusion

In conclusion, the UTD Dining System database project successfully integrated MySQL Workbench and MongoDB, incorporating two tables created in MongoDB using Python. The seamless interaction between relational and NoSQL databases was achieved through a user-defined function and SQL script. This hybrid approach not only demonstrated flexibility in handling different data types but also showcased adaptability and readiness for future scalability. The project highlights the effective use of diverse database technologies to meet specific application needs, providing a solid foundation for efficient data management and future development.