**UTD Dining and Meal Plan System**

**Phase III. Implementation**

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

**Semester Project Group- 8**

**Group Members:**

Jagdish Umashankar Vishwakarma ([juv230000@utdallas.edu](mailto:juv230000@utdallas.edu))

Uchaansh Lodhi ([uxl230001@utdallas.edu](mailto:uxl230001@utdallas.edu))

Sonakshi Handoo ([sxh230044@utdallas.edu](mailto:sxh230044@utdallas.edu))

Aditya Chaganti ([vxc230016@utdallas.edu](mailto:vxc230016@utdallas.edu))

Utsav Mukesh Maisheri ([uxm230000@utdallas.edu](mailto: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**

1. List of Figures
2. List of Tables
3. Pre-Illumination
4. Modified Relational Schema
5. Creation of Database with SQL Statements
6. Query Scenario Design
7. MongoDb Implementation
8. Conclusion

**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)**

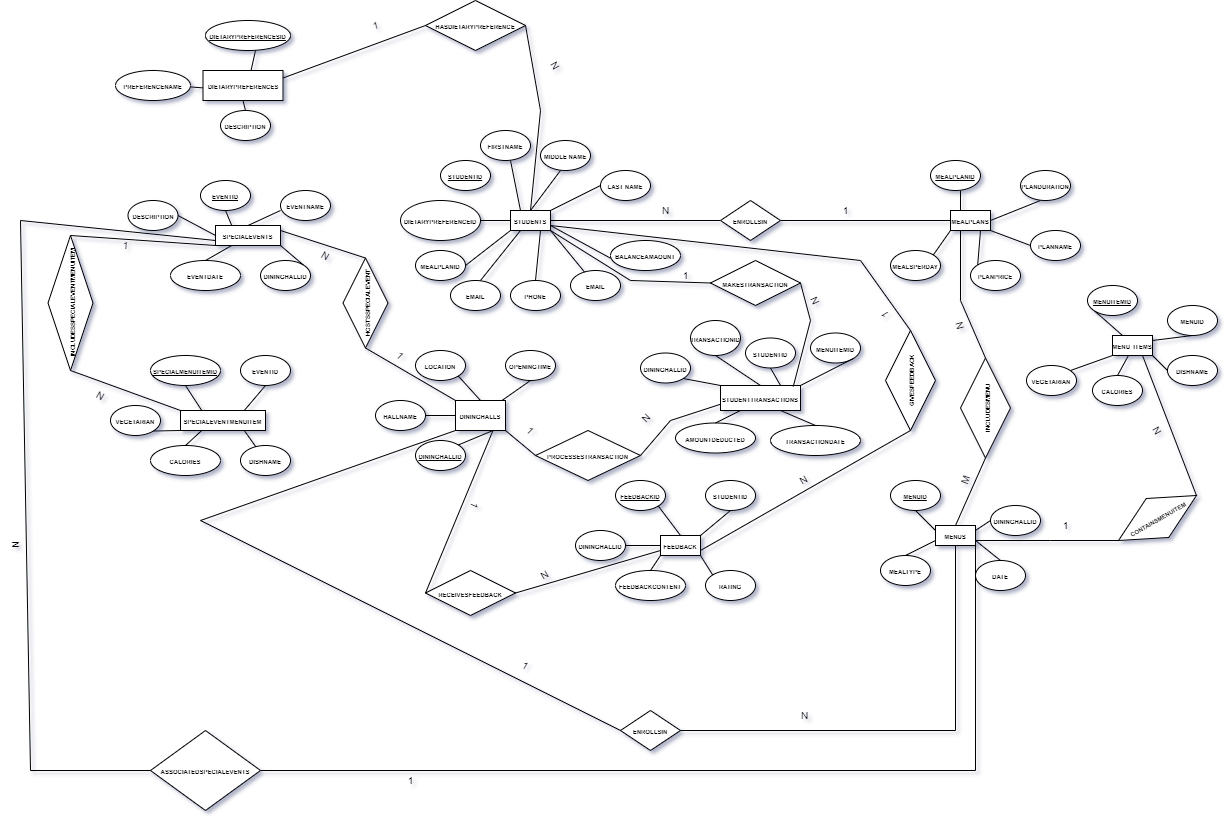


Figure 1. EER Diagram

**1.2 Relational Schema**

Our Relational Schema has no weak entities

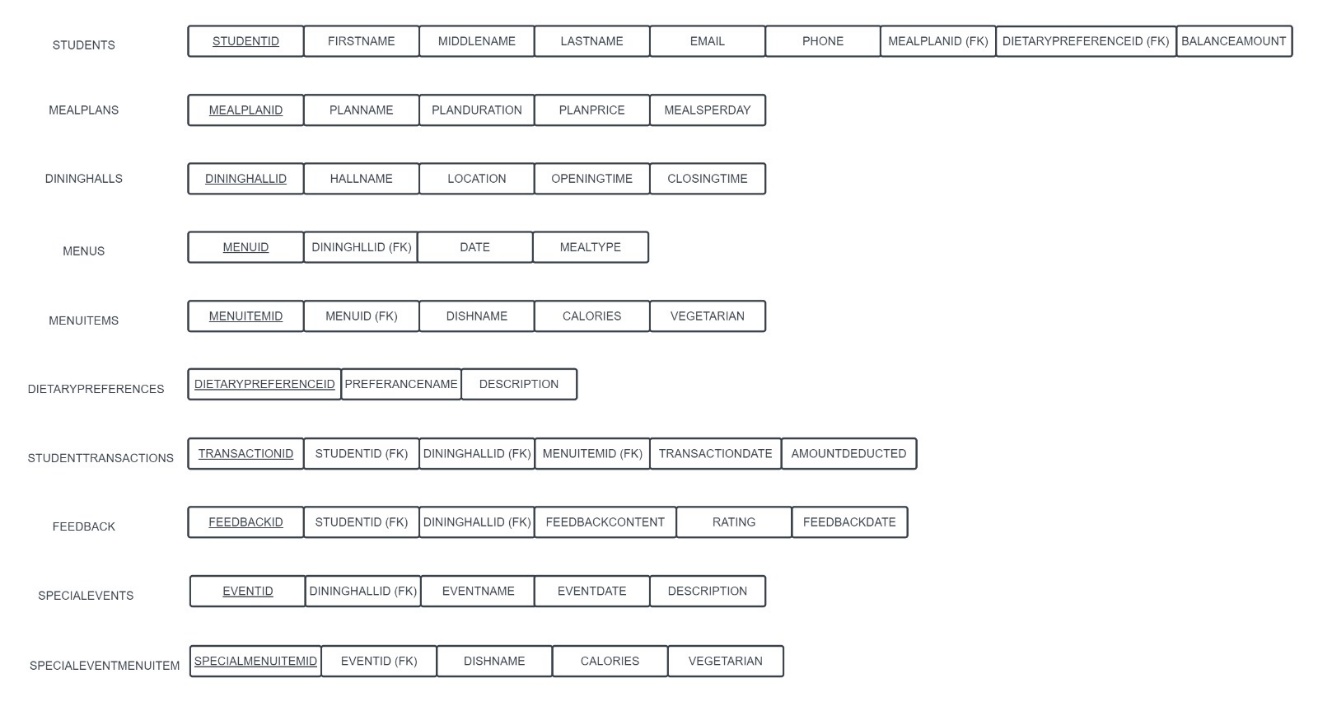


Figure 2: Relational Schema

**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.

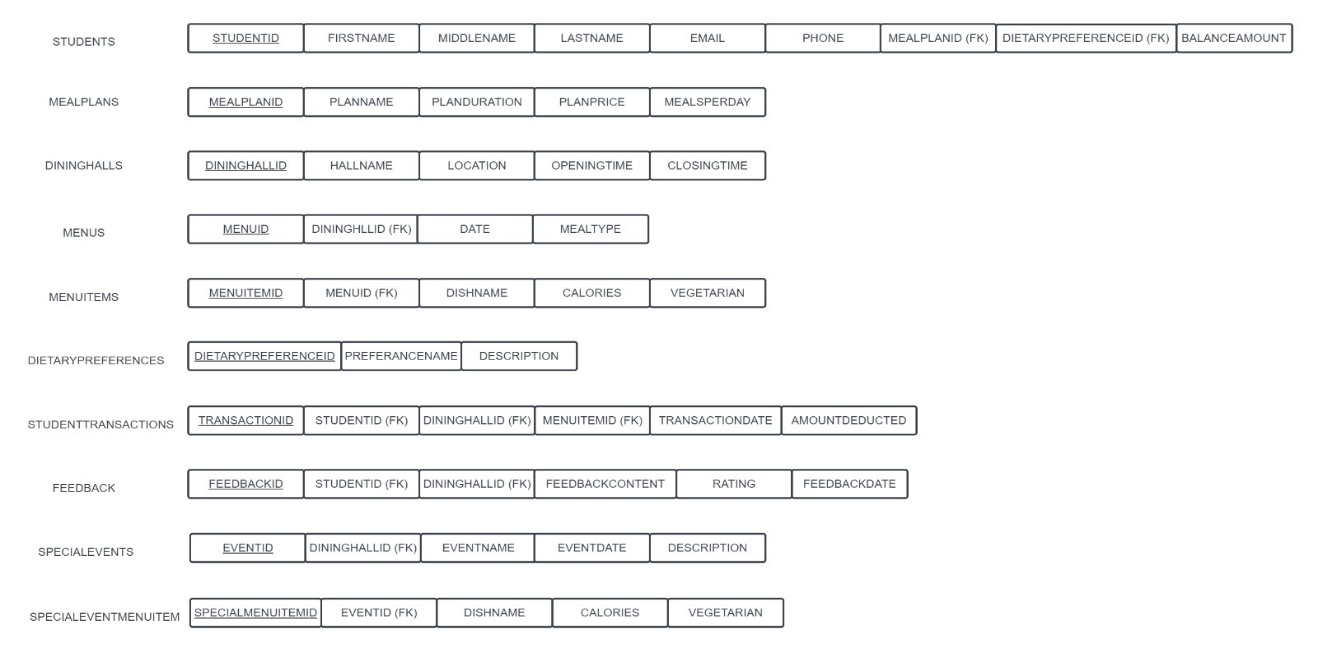


Figure 3: Normalized Relational Schema

**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.  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.  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.  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

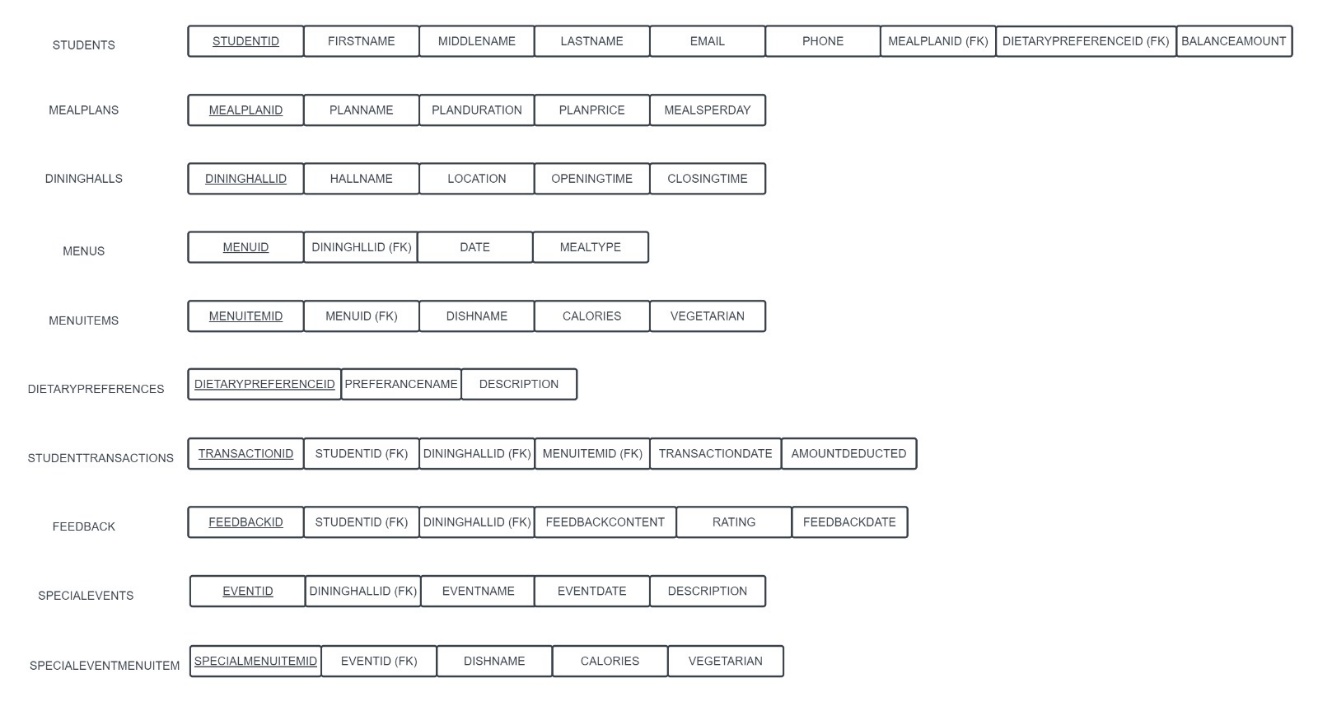
|  |  |  |  |
| --- | --- | --- | --- |
| **Relation Names** | **Attributes** | **Data Type** | **Length (Max) or Size** |
| Students | StudentID (PK) | String (Primary Key) | 10 |
|  | FullName | String | 50 |
|  | Email | String | 255 |
|  | Phone | String | 12 |
|  | MealPlanID (FK) | String (Foreign 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.



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'),

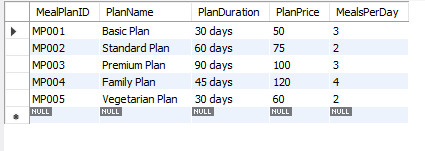
('F048', 'S048', 'DH018', 'The Halal Shack offers delicious halal options.', 5, '2023-11-17'),

('F049', 'S049', 'DH019', 'The Market at SU has a variety of choices.', 4, '2023-11-18'),

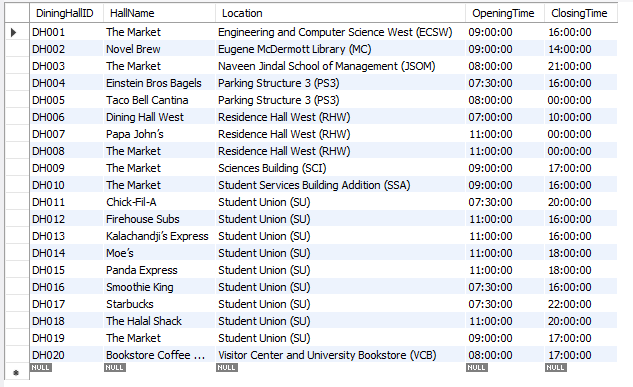
('F050', 'S050', 'DH020', 'Great coffee at the Bookstore Coffee Shop.', 5, '2023-11-19');

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

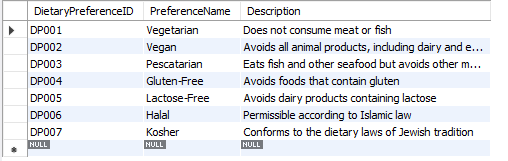
**Table 1**. **Meal Plans**



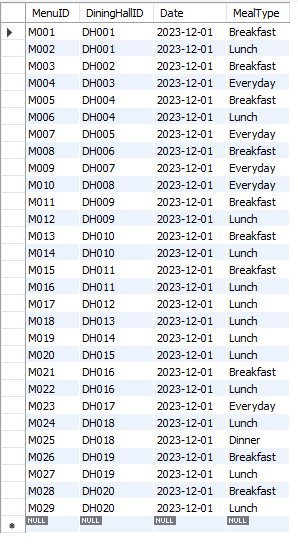
**Table 2**. **Dining Halls**



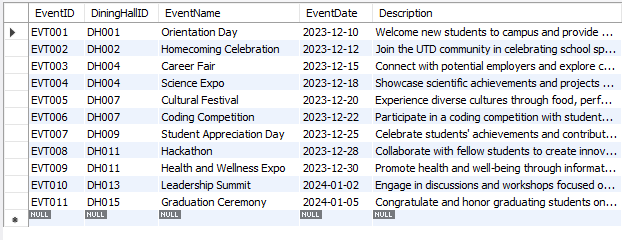
**Table 3. Dietary Preferences**



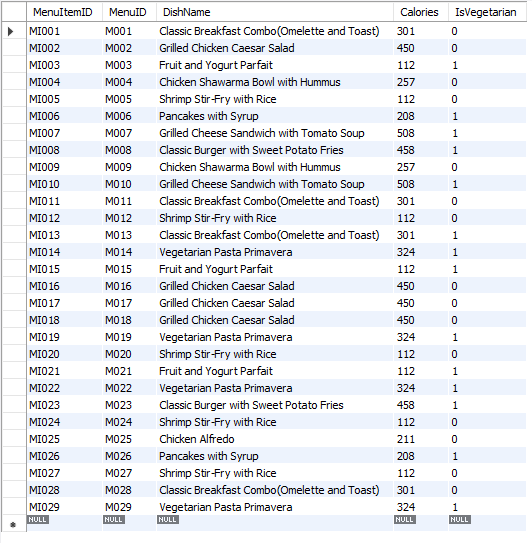
**Table 4**. **Menus**



**Table 5. Special Events**



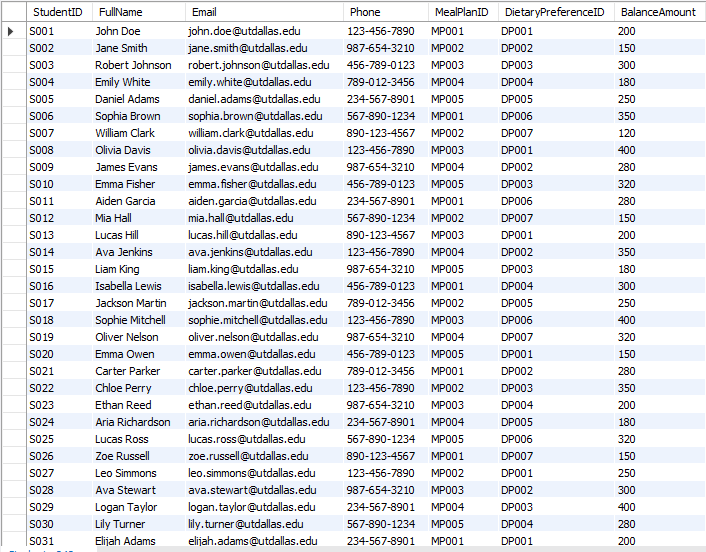
**Table 6. Menu Items**

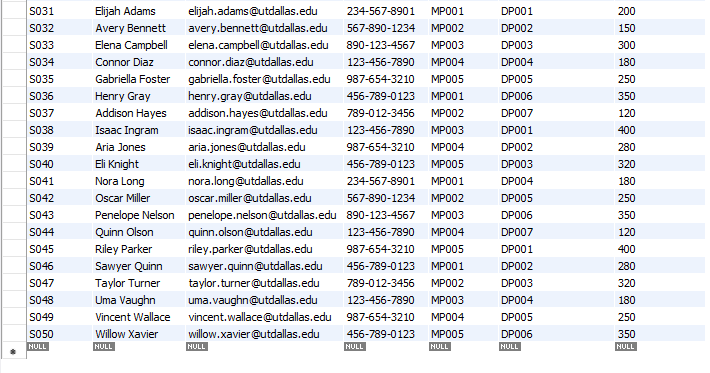


**Table 7. Special Event Menu Items**

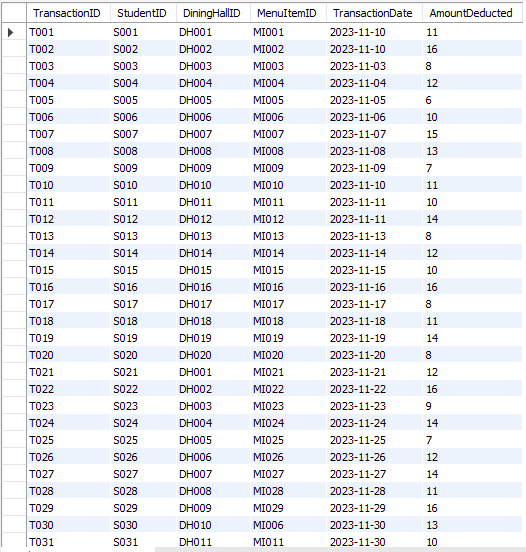


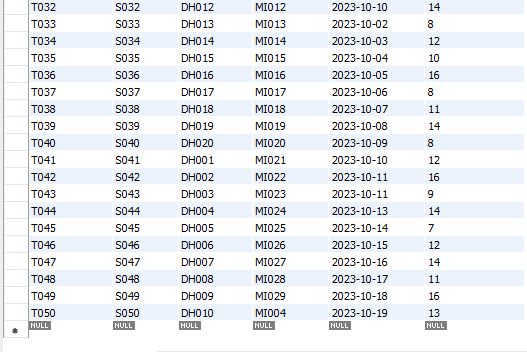
**Table 8. Students**



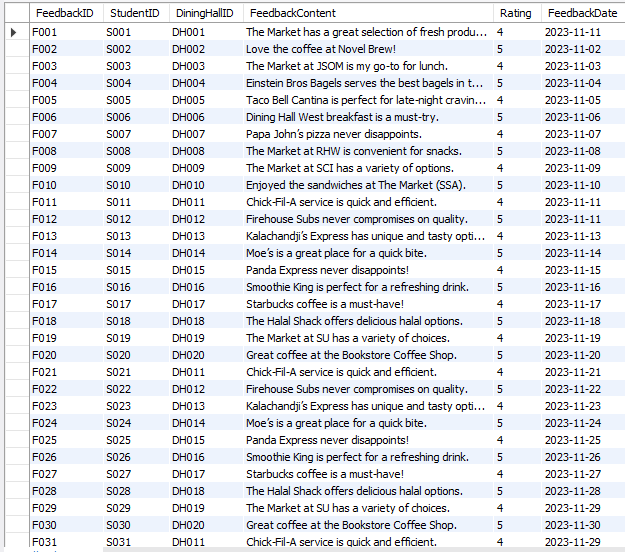


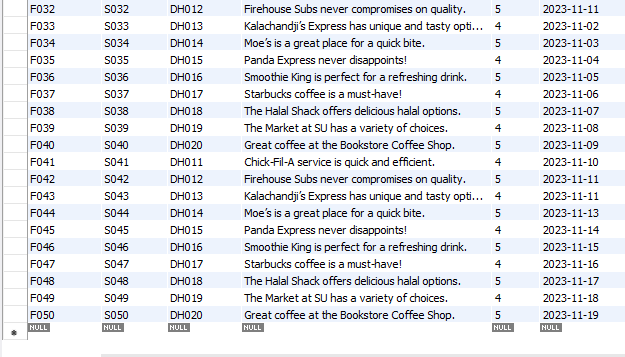
**Table 9. Student Transactions**





**Table 10. Feedback**





**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 = (

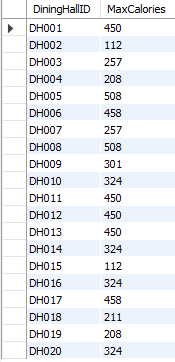
SELECT MAX(TransactionDate)

FROM StudentTransactions

WHERE StudentID = s.StudentID

);

**Result of Query 1**



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

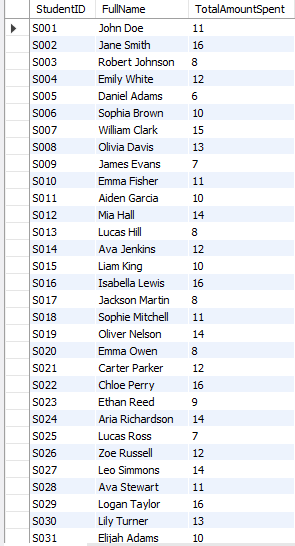
SELECT s.StudentID, s.FullName, SUM(st.AmountDeducted) AS TotalAmountSpent

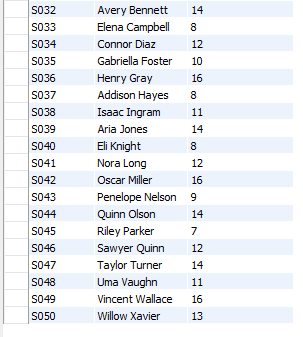
FROM Students s

LEFT JOIN StudentTransactions st ON s.StudentID = st.StudentID

GROUP BY s.StudentID, s.FullName;

**Result of Query 2**





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

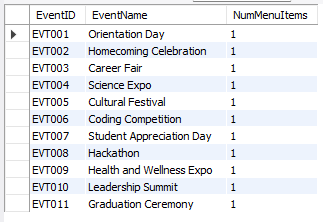
SELECT se.EventID, se.EventName, COUNT(semi.SpecialMenuItemID) AS NumMenuItems

FROM SpecialEvents se

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

GROUP BY se.EventID, se.EventName;

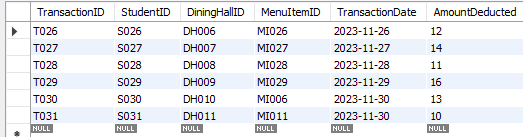
**Result of Query 3**



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

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

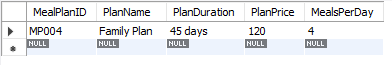
**Result of Query 4**



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

SELECT \* FROM MealPlans WHERE MealsPerDay > 3;

**Result of Query 5**



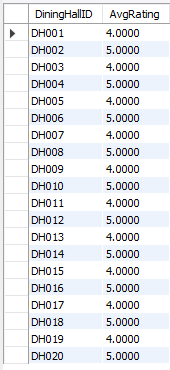
**Query 06: Retrieve the average rating of feedback for each dining hall.**

SELECT f.DiningHallID, AVG(f.Rating) AS AvgRating

FROM Feedback f

GROUP BY f.DiningHallID;

**Result of Query 6**



**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

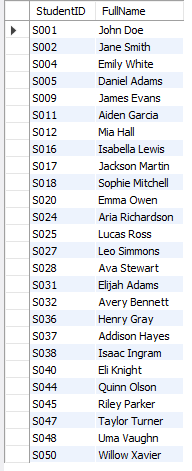
FROM StudentTransactions st

WHERE st.StudentID = s.StudentID AND st.MenuItemID = mi.MenuItemID

)

);

**Result of Query 7**



**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

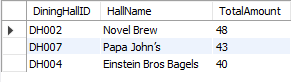
LEFT JOIN StudentTransactions st ON dh.DiningHallID = st.DiningHallID

GROUP BY dh.DiningHallID, dh.HallName

ORDER BY TotalAmount DESC

LIMIT 3;

**Result of Query 8**



**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

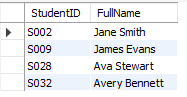
JOIN DietaryPreferences dp ON s.DietaryPreferenceID = dp.DietaryPreferenceID

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**



**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**



**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**





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

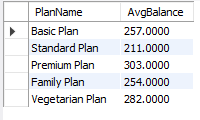
SELECT m.PlanName, AVG(s.BalanceAmount) AS AvgBalance

FROM MealPlans m

JOIN Students s ON m.MealPlanID = s.MealPlanID

GROUP BY m.PlanName;

**Result of Query 12**



**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

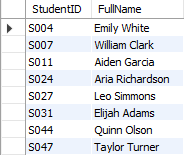
JOIN StudentTransactions st ON s.StudentID = st.StudentID

JOIN SpecialEvents se ON st.DiningHallID = se.DiningHallID

GROUP BY s.StudentID, se.DiningHallID

HAVING COUNT(DISTINCT se.EventID) >= 2;

**Result of Query 13**



**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**



**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

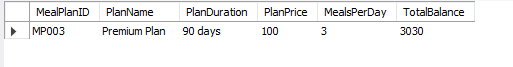
JOIN Students s ON mp.MealPlanID = s.MealPlanID

GROUP BY mp.MealPlanID

ORDER BY TotalBalance DESC

LIMIT 1;

**Result of Query 15**



**7. MongoDB Implementation**

We created two collections in MongoDB – Employee and Shifts

We then migrated these two collections in MySQL which can be further used to query more insightful information.

import pymongo

client = pymongo.MongoClient("mongodb://localhost:27017")

db = client["UTD\_Dining"]

Employee\_Collection = db["Employee"]

shifts\_collection = db["Shifts"]

employees\_data = [

{

"EmployeeID": f"EMP006",

"FirstName": "David",

"LastName": "Miller",

"Position": "Staff",

"Department": "Service",

"Salary": 32000

},

{

"EmployeeID": f"EMP007",

"FirstName": "Olivia",

"LastName": "Johnson",

"Position": "Staff",

"Department": "Service",

"Salary": 33000

},

{

"EmployeeID": f"EMP008",

"FirstName": "Sophia",

"LastName": "Williams",

"Position": "Staff",

"Department": "Service",

"Salary": 34000

},

{

"EmployeeID": f"EMP009",

"FirstName": "Michael",

"LastName": "Brown",

"Position": "Staff",

"Department": "Service",

"Salary": 35000

},

{

"EmployeeID": f"EMP010",

"FirstName": "Emma",

"LastName": "Jones",

"Position": "Staff",

"Department": "Service",

"Salary": 36000

},

{

"EmployeeID": f"EMP011",

"FirstName": "William",

"LastName": "Garcia",

"Position": "Staff",

"Department": "Service",

"Salary": 37000

},

{

"EmployeeID": f"EMP012",

"FirstName": "Amelia",

"LastName": "Smith",

"Position": "Staff",

"Department": "Service",

"Salary": 38000

},

{

"EmployeeID": f"EMP013",

"FirstName": "Henry",

"LastName": "Wilson",

"Position": "Staff",

"Department": "Service",

"Salary": 39000

},

{

"EmployeeID": f"EMP014",

"FirstName": "Lily",

"LastName": "Martinez",

"Position": "Staff",

"Department": "Service",

"Salary": 40000

},

{

"EmployeeID": f"EMP015",

"FirstName": "James",

"LastName": "Anderson",

"Position": "Staff",

"Department": "Service",

"Salary": 41000

}

]

employee\_collection.insert\_many(employees\_data)

shifts\_data = [

{

"ShiftID": f"SHIFT006",

"EmployeeID": f"EMP006",

"Date": f"2023-07-21",

"StartTime": "09:00",

"EndTime": "17:00"

},

{

"ShiftID": f"SHIFT007",

"EmployeeID": f"EMP007",

"Date": f"2023-07-22",

"StartTime": "09:00",

"EndTime": "17:00"

},

{

"ShiftID": f"SHIFT008",

"EmployeeID": f"EMP008",

"Date": f"2023-07-23",

"StartTime": "09:00",

"EndTime": "17:00"

},

{

"ShiftID": f"SHIFT009",

"EmployeeID": f"EMP009",

"Date": f"2023-07-24",

"StartTime": "09:00",

"EndTime": "17:00"

},

{

"ShiftID": f"SHIFT010",

"EmployeeID": f"EMP010",

"Date": f"2023-07-25",

"StartTime": "09:00",

"EndTime": "17:00"

},

{

"ShiftID": f"SHIFT011",

"EmployeeID": f"EMP011",

"Date": f"2023-07-26",

"StartTime": "09:00",

"EndTime": "17:00"

},

{

"ShiftID": f"SHIFT012",

"EmployeeID": f"EMP012",

"Date": f"2023-07-27",

"StartTime": "09:00",

"EndTime": "17:00"

},

{

"ShiftID": f"SHIFT013",

"EmployeeID": f"EMP013",

"Date": f"2023-07-28",

"StartTime": "09:00",

"EndTime": "17:00"

},

{

"ShiftID": f"SHIFT014",

"EmployeeID": f"EMP014",

"Date": f"2023-07-29",

"StartTime": "09:00",

"EndTime": "17:00"

},

{

"ShiftID": f"SHIFT015",

"EmployeeID": f"EMP015",

"Date": f"2023-07-30",

"StartTime": "09:00",

"EndTime": "17:00"

}

]

shifts\_collection.insert\_many(shifts\_data)

**Employee Collection created in MongoDB**

**A screenshot of a computer

Description automatically generated**

**Shifts Collection created in Mongodb**

**A screenshot of a computer

Description automatically generated**

**8.** **Conclusion**

In this report we describe the implementation process. This includes steps for setting up the database, populating it with data, and, if applicable, developing the software component. Challenges faced during implementation are outlined along with their resolutions.