CS4347 Database Systems Final Project Deliverable 1

Healthy Recipes Generation/Meal Planning/Nutrition System

Group Members: Muhammad Siddique, Mudit Upadhyay, Anirudh Nemmani, Syed Kabir, Vaishnavi Karaguppi, Senthen Velmurugan, Arvindh Kumar Kalainathan

0. Description:

Project Title

Healthy Recipes Generation/Meal Planning/Nutrition System

GitHub Repo

https://github.com/msiddique12/Meal-Planning-Nutrition-System

The repository currently contains the implementation of our design including SQL statements for database and table creation, along with the setup of primary and foreign keys. It also houses insert, update, delete, and query operations on the database.

Team members

- Muhammad Siddique
- Mudit Upadhyay
- Anirudh Nemmani
- Syed Kabir
- Vaishnavi Karaguppi
- Senthen Velmurugan
- Arvindh Kumar Kalainathan

Delegation of Tasks for Deliverable 1

- **Muhammad Siddique** - Involved in creating the conceptual design of the database and implementing the design using SQL, creating a functional schema to manage users, dietary preferences, meal plans, and cuisine options.
- **Mudit Upadhyay -** Involved in researching background and finding related work as well as database design.
- **Anirudh Nemmani -** Involved in creating the conceptual design of the database and implementing the design using SQL
- **Syed Kabir -** Involved in creating the conceptual design of the database and the Enhanced Entity Relational model designs of the project
- **Vaishnavi Karaguppi -** Involved in creating the conceptual design of the database and creating a functional schema of the project.
- **Senthen Velmurugan-** Involved in implementing the design using SQL, and Test all execution types
- **Arvindh Kumar Kalainathan -** Involved in creating the relational database schema and the Enhanced Entity Relational model designs of the project

Project Motivation

The motivation for this project stems from our team's collective interest in leading a healthier lifestyle without sacrificing the enjoyment of food. We all face challenges in balancing our busy lives with proper nutrition, and finding the right recipes that fit both our dietary restrictions and taste preferences can be difficult. We believe that a tool like this can simplify meal planning, not only for ourselves but for anyone looking to eat healthier while maintaining variety and cultural tastes in their meals.

Additionally, This tool could be useful for fitness enthusiasts who want to keep track of their exact calories and macronutrients, people with specific dietary restrictions, or even healthcare providers looking to offer personalized meal plans for patients. It could also be extended for use in meal prep services, fitness apps, or nutrition platforms that want to incorporate more personalized dietary options.

Project Timeline

1. Conceptual Design
    - Held extensive team discussion on how we wanted the design to look and what exact use cases our system could serve.
    - Created an EER (Enhanced Entity-Relationship) diagram to illustrate the design we came up with.
    - The EER diagram was continuously updated throughout the design time of this deliverable to reflect our most current plans for the system.
2. Logical Design
    - Performed ER/EER to relational mapping to create the relational schema of our system. This involved mapping our entities to relational tables, converting some of the relationships to relational tables, ETC.
3. Physical Design (SQL Implementation)
    - Created the database schema using SQL based on the relational model with referential integrity constraints enforced
    - Populated the database with initial data and performed query, update, and delete operations on the database state.

1. Introduction:

Our project, a meal planning app, focuses on simplifying daily meal preparation by generating personalized meal plans. We chose this topic to address the growing need for individuals to maintain a healthy diet despite their busy schedules. With the rise of diverse dietary requirements and preferences, people often struggle to plan meals that cater to their unique needs. By developing this app, we aim to contribute to the field of security, particularly regarding data privacy and dietary data protection, ensuring that users' personal information, such as their calorie intake, dietary restrictions, and meal preferences, is kept secure.

Our app's database is designed to store user profiles, meal plans, recipes, and nutritional information securely. The reason we chose this particular database design is to create a comprehensive, scalable solution capable of handling personalized meal data while maintaining strict data privacy. Each user profile includes details such as daily calorie goals, dietary preferences, and restrictions, allowing the app to generate non-repetitive, customized meal plans.

The real-world problem this design helps solve is the increasing demand for personalized, accessible, and secure nutritional planning. Many apps store data but do not use it effectively to cater to individual preferences or to protect it sufficiently from breaches. By focusing on data protection, we ensure that users can trust the app with their sensitive health-related information while enjoying tailored meal plans that help them meet their nutritional goals.

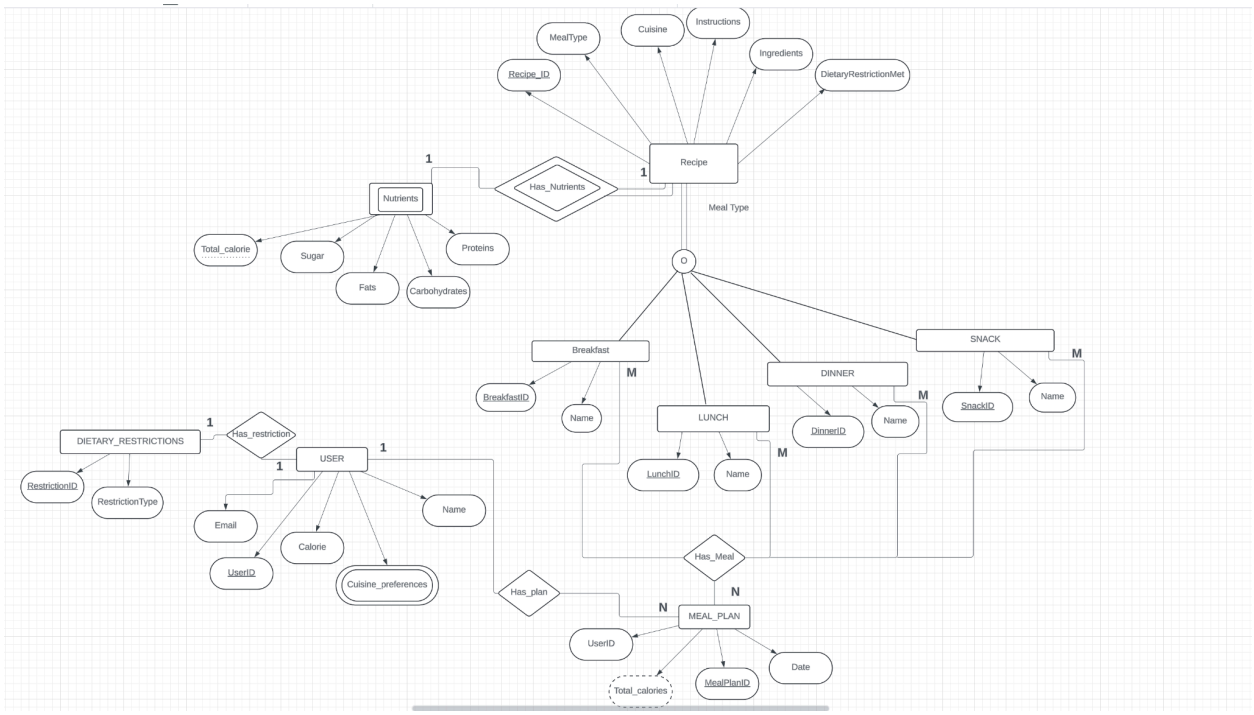
2. Background and Related Work:

After looking over several other sources, we found 2 other applications that are working towards the same problem statement. **HealthifyMe** is a comprehensive health and fitness app that empowers users to opt for healthy food choices. It offers a vast database of healthy recipes catering to various dietary restrictions, allowing users to customize their meal schedules by day or week. This all-in-one approach emphasizes holistic wellness but may lack the depth of automated meal planning. **MealBoard** is focusing on streamlining meal prep by combining recipe storage, meal scheduling, and shopping management in one app. Unlike HealthifyMe, which leans heavily on nutritional guidance, MealBoard provides a straightforward meal planning experience. Our app, **MealPlanning** differentiates itself by automatically generating personalized meal plans for breakfast, lunch, dinner, and snacks based on user-inputted calorie goals and dietary preferences, ensuring a balanced and varied diet while minimizing repetitive meals.

3. Design & Implementation (Phase I):

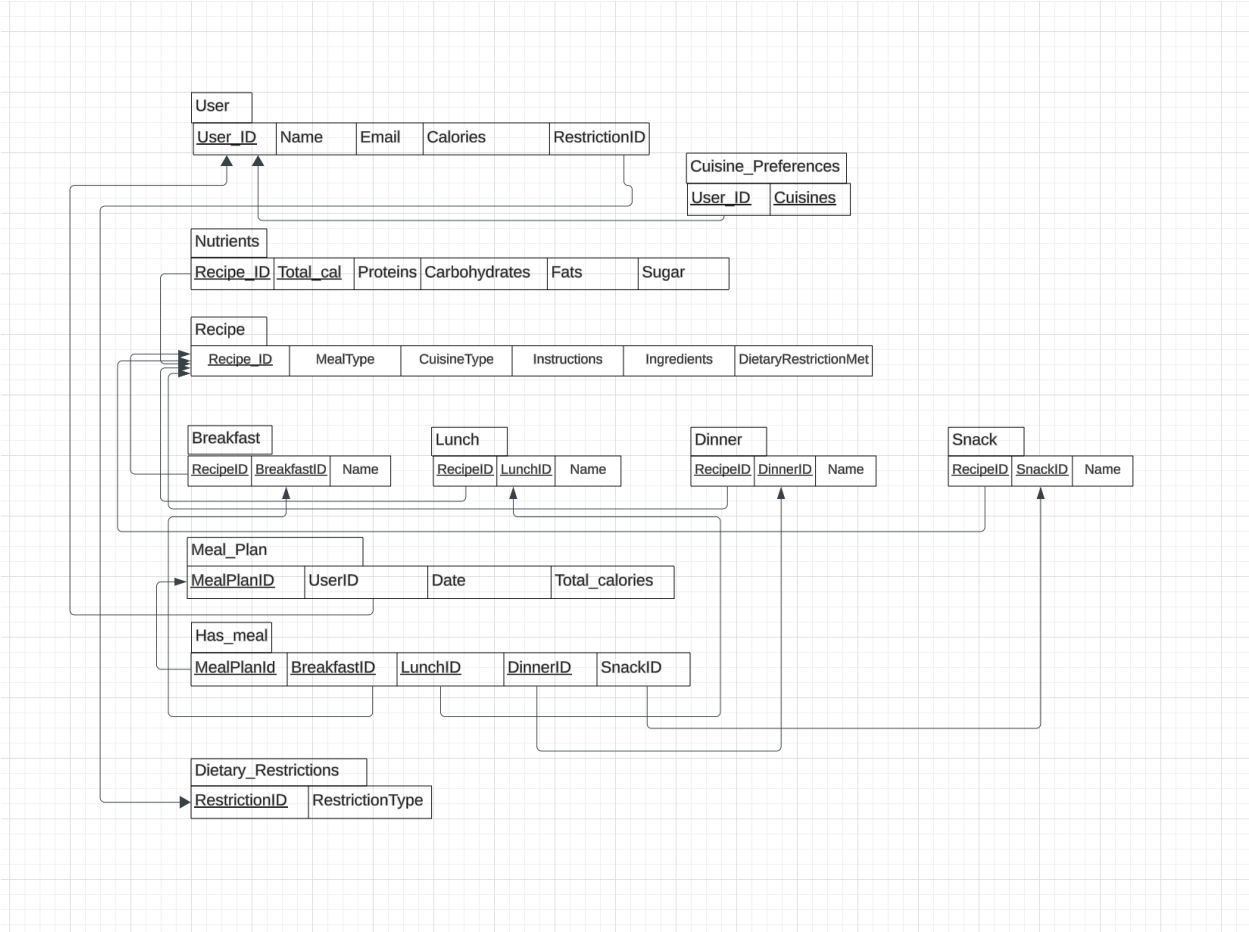
3.1. EER Conceptual Data Model Design:

Draw the EER diagram of your
design USING A TOOL, rather than drawing manually. Provide the conceptual data model design for your project in the form of an EER diagram. Your design should have a minimum of 5 different tables, and minimum 10 tuples per table. The number of attributes in each entity and relation should be a reasonable value to describe your entity/relationship depending on your design, so no enforcement on the number of attributes per table. There should be minimum one component in your design that makes it an Enhanced ER (EER) diagram. Also make sure to include a variety of parameters, such as different types of entities (e.g. composite, multi-valued, etc.), different types of cardinalities (e.g 1:1, 1:M, M:N, etc.), and relationships (e.g. unary, binary, etc.). Also include minimum one weak entity in your design.



- 3.2. Relational Data Model Design:

Perform EER to relational model mapping to create the following: Draw the schema diagram of your design USING A TOOL, rather than drawing manually. Specify Primary Keys, Foreign Keys and referential integrity constraints in this schema



- 3.3. Create your Database and Populate:

Use an SQL platform to create
your database and its tables as you described in sections 3.1 and 3.2. Then, populate each table with corresponding data. Remember that each table should have a minimum 10 tuples. Provide a screenshot of each database creation statement, and each database table population, as well as each resulting table in your report exactly in this section.

DIETARY RESTRICTIONS:

```sql
-- Create table for Dietary Restrictions
CREATE TABLE IF NOT EXISTS Dietary_Restrictions (
    RestrictionID INT PRIMARY KEY,
    RestrictionType VARCHAR(255) NOT NULL
);
```

```sql
-- Populating Dietary_Restrictions table
INSERT INTO Dietary_Restrictions (RestrictionID, RestrictionType) VALUES
(1, 'Vegetarian'),
(2, 'Vegan'),
(3, 'Halal'),
(4, 'Kosher'),
(5, 'Gluten-Free'),
(6, 'Dairy-Free'),
(7, 'Nut-Free'),
(8, 'Pescatarian'),
(9, 'Low-Carb'),
(10, 'Paleo');
```

RestrictionID	RestrictionType
1	Vegetarian
2	Vegan
3	Halal
4	Kosher
5	Gluten-Free
6	Dairy-Free
7	Nut-Free
8	Pescatarian
9	Low-Carb
10	Paleo
NULL	NULL

USERS:

```sql
-- Create table for Users
CREATE TABLE IF NOT EXISTS User (
    User_ID INT PRIMARY KEY,
    Name VARCHAR(255) NOT NULL,
    Email VARCHAR(255) NOT NULL UNIQUE,
    Calories INT,
    RestrictionID INT,
    FOREIGN KEY (RestrictionID) REFERENCES Dietary_Restrictions(RestrictionID)
);
```

```sql
-- Populating User table
INSERT INTO User (User_ID, Name, Email, Calories, RestrictionID) VALUES
(1, 'John Doe', 'john.doe@example.com', 2000, 1),
(2, 'Jane Smith', 'jane.smith@example.com', 1500, 2),
(3, 'Ahmed Ali', 'ahmed.ali@example.com', 1800, 3),
(4, 'Sarah Cohen', 'sarah.cohen@example.com', 1700, 4),
(5, 'Emily Johnson', 'emily.johnson@example.com', 1600, 5),
(6, 'David Lee', 'david.lee@example.com', 2200, 6),
(7, 'Anna Gupta', 'anna.gupta@example.com', 1900, 7),
(8, 'Paul Martin', 'paul.martin@example.com', 2100, 8),
(9, 'Olivia Taylor', 'olivia.taylor@example.com', 1400, 9),
(10, 'Carlos Ramirez', 'carlos.ramirez@example.com', 2000, 10);
```

User_ID	Name	Email	Calories	RestrictionID
1	John Doe	john.doe@example.com	2000	1
2	Jane Smith	jane.smith@example.com	1500	2
3	Ahmed Ali	ahmed.ali@example.com	1800	3
4	Sarah Cohen	sarah.cohen@example.com	1700	4
5	Emily Johnson	emily.johnson@example.com	1600	5
6	David Lee	david.lee@example.com	2200	6
7	Anna Gupta	anna.gupta@example.com	1900	7
8	Paul Martin	paul.martin@example.com	2100	8
9	Olivia Taylor	olivia.taylor@example.com	1400	9
10	Carlos Ramirez	carlos.ramirez@example.com	2000	10
NULL	NULL	NULL	NULL	NULL

Cuisine Preferences:

```sql
-- Create table for Cuisine Preferences
CREATE TABLE IF NOT EXISTS Cuisine_Preferences(
    User_ID INT,
    Cuisine VARCHAR(255),
    PRIMARY KEY (User_ID, Cuisine),
    FOREIGN KEY (User_ID) REFERENCES User(User_ID)
);
```

```sql
-- Populating Cuisine_Preferences table
INSERT INTO Cuisine_Preferences (User_ID, Cuisine) VALUES
(1, 'Indian'),
(1, 'Italian'),
(2, 'Mediterranean'),
(2, 'Mexican'),
(3, 'Middle Eastern'),
(3, 'South Asian'),
(4, 'Jewish'),
(5, 'French'),
(6, 'Chinese'),
(7, 'Thai'),
(8, 'Japanese'),
(9, 'Keto'),
(10, 'Paleo');
```

User_ID	Cuisine
1	Indian
1	Italian
2	Mediterranean
2	Mexican
3	Middle Eastern
3	South Asian
4	Jewish
5	French
6	Chinese
7	Thai
8	Japanese
9	Keto
10	Paleo
NULL	NULL

Recipe:

```sql
-- Create table for Recipe
CREATE TABLE IF NOT EXISTS Recipe (
    Recipe_ID INT PRIMARY KEY,
    MealType VARCHAR(255) NOT NULL,
    CuisineType VARCHAR(255),
    Instructions TEXT,
    Ingredients TEXT,
    DietaryRestrictionMet BOOLEAN
);
```

```sql
-- Populating Recipe table
INSERT INTO Recipe (Recipe_ID, MealType, CuisineType, Instructions, Ingredients, DietaryRestrictionMet) VALUES
(1, 'Lunch', 'Indian', 'Cook roti and curry', 'wheat flour, vegetables, spices', TRUE),
(2, 'Dinner', 'Italian', 'Bake pizza with cheese and sauce', 'wheat flour, cheese, tomato sauce', FALSE),
(3, 'Lunch', 'Indian', 'Cook biryani', 'rice, chicken, spices', FALSE),
(4, 'Breakfast', 'American', 'Prepare scrambled eggs', 'eggs, butter, salt', FALSE),
(5, 'Snack', 'American', 'Prepare french fries', 'potatoes, oil, salt', FALSE),
(6, 'Dinner', 'Indian', 'Cook dal with spices', 'lentils, turmeric, cumin', TRUE),
(7, 'Lunch', 'Mexican', 'Make tacos', 'corn tortillas, meat, cheese, lettuce', FALSE),
(8, 'Snack', 'Italian', 'Bake garlic bread', 'bread, garlic, butter', FALSE),
(9, 'Breakfast', 'French', 'Make croissants', 'flour, butter, sugar', FALSE),
(10, 'Dinner', 'Mediterranean', 'Grill fish with veggies', 'fish, olive oil, vegetables', TRUE);
```

Recipe_ID	MealType	CuisineType	Instructions	Ingredients	DietaryRestrictionMet
1	Lunch	Indian	Cook roti and curry	wheat flour, vegetables, spices	1
2	Dinner	Italian	Bake pizza with cheese and sauce	wheat flour, cheese, tomato sauce	0
3	Lunch	Indian	Cook biryani	rice, chicken, spices	0
4	Breakfast	American	Prepare scrambled eggs	eggs, butter, salt	0
5	Snack	American	Prepare french fries	potatoes, oil, salt	0
6	Dinner	Indian	Cook dal with spices	lentils, turmeric, cumin	1
7	Lunch	Mexican	Make tacos	corn tortillas, meat, cheese, lettuce	0
8	Snack	Italian	Bake garlic bread	bread, garlic, butter	0
9	Breakfast	French	Make croissants	flour, butter, sugar	0
10	Dinner	Mediterranean	Grill fish with veggies	fish, olive oil, vegetables	1
NULL	NULL	NULL	NULL	NULL	NULL

Nutrients:

```sql
-- Create table for Nutrients
CREATE TABLE IF NOT EXISTS Nutrients (
    Recipe_ID INT,
    Total_cal INT,
    Proteins DECIMAL(5,2),
    Carbohydrates DECIMAL(5,2),
    Fats DECIMAL(5,2),
    Sugar DECIMAL(5,2),
    PRIMARY KEY (Recipe_ID, Total_cal),
    FOREIGN KEY (Recipe_ID) REFERENCES Recipe(Recipe_ID)
);
```

```sql
-- Populating Nutrients table
INSERT INTO Nutrients (Recipe_ID, Total_cal, Proteins, Carbohydrates, Fats, Sugar) VALUES
(1, 400, 10.5, 60.3, 8.2, 2.1),
(2, 800, 20.3, 90.1, 30.5, 5.2),
(3, 650, 25.2, 70.5, 15.8, 3.4),
(4, 300, 20.0, 2.0, 25.0, 1.0),
(5, 500, 5.2, 50.5, 25.3, 0.5),
(6, 450, 12.5, 60.0, 10.0, 2.5),
(7, 550, 15.8, 65.3, 20.2, 3.0),
(8, 350, 8.5, 40.0, 10.5, 2.5),
(9, 600, 12.3, 55.5, 35.0, 6.0),
(10, 480, 25.0, 20.0, 18.0, 0.5);
```

Recipe_ID	Total_cal	Proteins	Carbohydrates	Fats	Sugar
1	400	10.50	60.30	8.20	2.10
2	800	20.30	90.10	30.50	5.20
3	650	25.20	70.50	15.80	3.40
4	300	20.00	2.00	25.00	1.00
5	500	5.20	50.50	25.30	0.50
6	450	12.50	60.00	10.00	2.50
7	550	15.80	65.30	20.20	3.00
8	350	8.50	40.00	10.50	2.50
9	600	12.30	55.50	35.00	6.00
10	480	25.00	20.00	18.00	0.50
NULL	NULL	NULL	NULL	NULL	NULL

Breakfast:

```sql
-- Create table for Breakfast
CREATE TABLE IF NOT EXISTS Breakfast (
    RecipeID INT,
    BreakfastID INT PRIMARY KEY,  -- Make BreakfastID a primary key
    Name VARCHAR(255),
    FOREIGN KEY (RecipeID) REFERENCES Recipe(Recipe_ID)
);
```

```sql
-- Populating Breakfast table
INSERT INTO Breakfast (RecipeID, BreakfastID, Name) VALUES
(4, 1, 'Scrambled Eggs'),
(9, 2, 'Croissants'),
(10, 3, 'Fish with veggies'),
(6, 4, 'Dal with spices'),
(1, 5, 'Roti and curry'),
(2, 6, 'Pizza'),
(5, 7, 'French fries'),
(8, 8, 'Garlic Bread'),
(7, 9, 'Tacos'),
(3, 10, 'Biryani');
```

RecipeID	BreakfastID	Name
4	1	Scrambled Eggs
9	2	Croissants
10	3	Fish with veggies
6	4	Dal with spices
1	5	Roti and curry
2	6	Pizza
5	7	French fries
8	8	Garlic Bread
7	9	Tacos
3	10	Biryani
NULL	NULL	NULL

Lunch:

```sql
-- Create table for Lunch
CREATE TABLE IF NOT EXISTS Lunch (
    RecipeID INT,
    LunchID INT PRIMARY KEY,  -- Make LunchID a primary key
    Name VARCHAR(255),
    FOREIGN KEY (RecipeID) REFERENCES Recipe(Recipe_ID)
);
```

```sql
-- Populating Lunch table
INSERT INTO Lunch (RecipeID, LunchID, Name) VALUES
(1, 1, 'Roti with Curry'),
(3, 2, 'Chicken Biryani'),
(7, 3, 'Tacos'),
(5, 4, 'French Fries'),
(6, 5, 'Dal with rice'),
(9, 6, 'Croissants with jam'),
(8, 7, 'Garlic bread with dipping'),
(10, 8, 'Grilled fish'),
(2, 9, 'Cheese Pizza'),
(4, 10, 'Scrambled eggs');
```

RecipeID	LunchID	Name
1	1	Roti with Curry
3	2	Chicken Biryani
7	3	Tacos
5	4	French Fries
6	5	Dal with rice
9	6	Croissants with jam
8	7	Garlic bread with dipping
10	8	Grilled fish
2	9	Cheese Pizza
4	10	Scrambled eggs
NULL	NULL	NULL

Dinner:

```sql
-- Create table for Dinner
CREATE TABLE IF NOT EXISTS Dinner (
    RecipeID INT,
    DinnerID INT PRIMARY KEY,  -- Make DinnerID a primary key
    Name VARCHAR(255),
    FOREIGN KEY (RecipeID) REFERENCES Recipe(Recipe_ID)
);
```

```sql
-- Populating Dinner table
INSERT INTO Dinner (RecipeID, DinnerID, Name) VALUES
(3, 1, 'Mutton Biryani'),
(1, 2, 'Roti with Curry'),
(4, 3, 'Scrambled Eggs'),
(2, 4, 'Cheese Pizza'),
(6, 5, 'Dal with spices'),
(7, 6, 'Tacos'),
(5, 7, 'French fries'),
(9, 8, 'Croissants'),
(8, 9, 'Garlic bread'),
(10, 10, 'Fish with veggies');
```

RecipeID	DinnerID	Name
3	1	Mutton Biryani
1	2	Roti with Curry
4	3	Scrambled Eggs
2	4	Cheese Pizza
6	5	Dal with spices
7	6	Tacos
5	7	French fries
9	8	Croissants
8	9	Garlic bread
10	10	Fish with veggies
NULL	NULL	NULL

Snack:

```sql
-- Create table for Snack
CREATE TABLE IF NOT EXISTS Snack (
    RecipeID INT,
    SnackID INT PRIMARY KEY,  -- Make SnackID a primary key
    Name VARCHAR(255),
    FOREIGN KEY (RecipeID) REFERENCES Recipe(Recipe_ID)
);
```

```sql
-- Populating Snack table
INSERT INTO Snack (RecipeID, SnackID, Name) VALUES
(5, 1, 'French Fries'),
(8, 2, 'Garlic Bread'),
(7, 3, 'Tacos'),
(9, 4, 'Croissants'),
(1, 5, 'Roti and Curry'),
(3, 6, 'Biryani'),
(10, 7, 'Grilled Fish'),
(2, 8, 'Cheese Pizza'),
(6, 9, 'Dal with spices'),
(4, 10, 'Scrambled Eggs');
```

RecipeID	SnackID	Name
5	1	French Fries
8	2	Garlic Bread
7	3	Tacos
9	4	Croissants
1	5	Roti and Curry
3	6	Biryani
10	7	Grilled Fish
2	8	Cheese Pizza
6	9	Dal with spices
4	10	Scrambled Eggs
NULL	NULL	NULL

Meal_Plan:

```sql
-- Create table for Meal_Plan
CREATE TABLE IF NOT EXISTS Meal_Plan (
    MealPlanID INT PRIMARY KEY,
    Date DATE NOT NULL,
    UserID INT,
    Total_calories INT,
    FOREIGN KEY (UserID) REFERENCES User(User_ID)
);
```

```sql
-- Populating Meal_Plan table
INSERT INTO Meal_Plan (MealPlanID, Date, UserID, Total_calories) VALUES
(1, '2024-10-15', 1, 2000),
(2, '2024-10-16', 2, 1500),
(3, '2024-10-17', 3, 1800),
(4, '2024-10-18', 4, 1700),
(5, '2024-10-19', 5, 1600),
(6, '2024-10-20', 6, 2200),
(7, '2024-10-21', 7, 1900),
(8, '2024-10-22', 8, 2100),
(9, '2024-10-23', 9, 1400),
(10, '2024-10-24', 10, 2000);
```

MealPlanID	Date	UserID	Total_calories
1	2024-10-15	1	2000
2	2024-10-16	2	1500
3	2024-10-17	3	1800
4	2024-10-18	4	1700
5	2024-10-19	5	1600
6	2024-10-20	6	2200
7	2024-10-21	7	1900
8	2024-10-22	8	2100
9	2024-10-23	9	1400
10	2024-10-24	10	2000
NULL	NULL	NULL	NULL

Has_Meal:

```sql
-- Create table for Has_Meal (to associate Meal_Plan with meals)
CREATE TABLE IF NOT EXISTS Has_meal (
    MealPlanId INT,
    BreakfastID INT,
    LunchID INT,
    DinnerID INT,
    SnackID INT,
    PRIMARY KEY (MealPlanId, BreakfastID, LunchID, DinnerID, SnackID),
    FOREIGN KEY (MealPlanId) REFERENCES Meal_Plan(MealPlanID),
    FOREIGN KEY (BreakfastID) REFERENCES Breakfast(BreakfastID),
    FOREIGN KEY (LunchID) REFERENCES Lunch(LunchID),
    FOREIGN KEY (DinnerID) REFERENCES Dinner(DinnerID),
    FOREIGN KEY (SnackID) REFERENCES Snack(SnackID)
);
```

```sql
-- Populating Has_Meal table
INSERT INTO Has_meal (MealPlanId, BreakfastID, LunchID, DinnerID, SnackID) VALUES
(1, 1, 1, 1, 1),
(2, 2, 2, 2, 2),
(3, 3, 3, 3, 3),
(4, 4, 4, 4, 4),
(5, 5, 5, 5, 5),
(6, 6, 6, 6, 6),
(7, 7, 7, 7, 7),
(8, 8, 8, 8, 8),
(9, 9, 9, 9, 9),
(10, 10, 10, 10, 10);
```

MealPlanId	BreakfastID	LunchID	DinnerID	SnackID
1	1	1	1	1
2	2	2	2	2
3	3	3	3	3
4	4	4	4	4
5	5	5	5	5
6	6	6	6	6
7	7	7	7	7
8	8	8	8	8
9	9	9	9	9
10	10	10	10	10
NULL	NULL	NULL	NULL	NULL

- 3.4. Database Query Execution (from inside your SQL client):

Use an SQL platform to provide sample executions for each of the following operations on each table of your database:

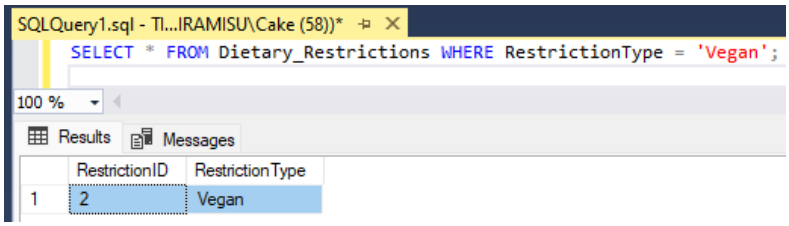
- Query – to perform operations such as list employees earning more than 150K per
- year
- Insert – to add new tuple(s), and/or field(s) to your table(s)
- Delete - to remove tuple(s), and/or field(s) from your table(s)
- Update - to modify tuple(s), and/or field(s) from your table(s)

Provide a screenshot of each operation and each resulting table in your report exactly in this Section.

1. **Dietary_Restrictions**

Query:

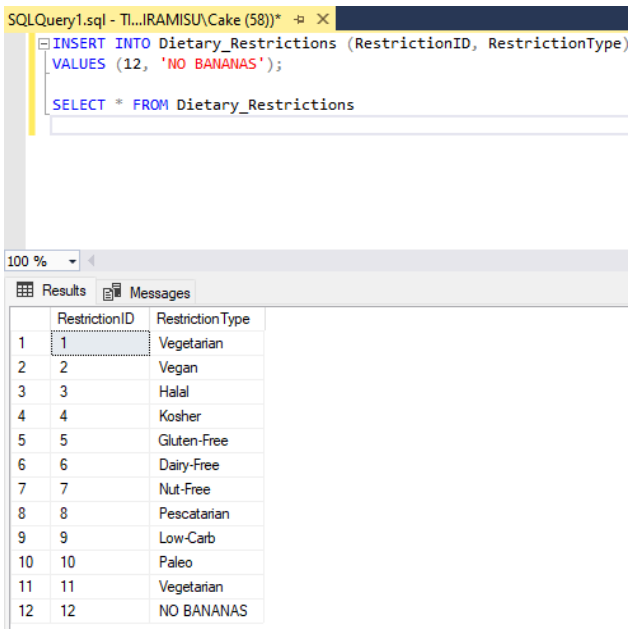
SELECT * FROM Dietary_Restrictions WHERE RestrictionType = 'Vegan';



Insert:

INSERT INTO Dietary_Restrictions (RestrictionID, RestrictionType)
VALUES (12, 'NO BANANAS');

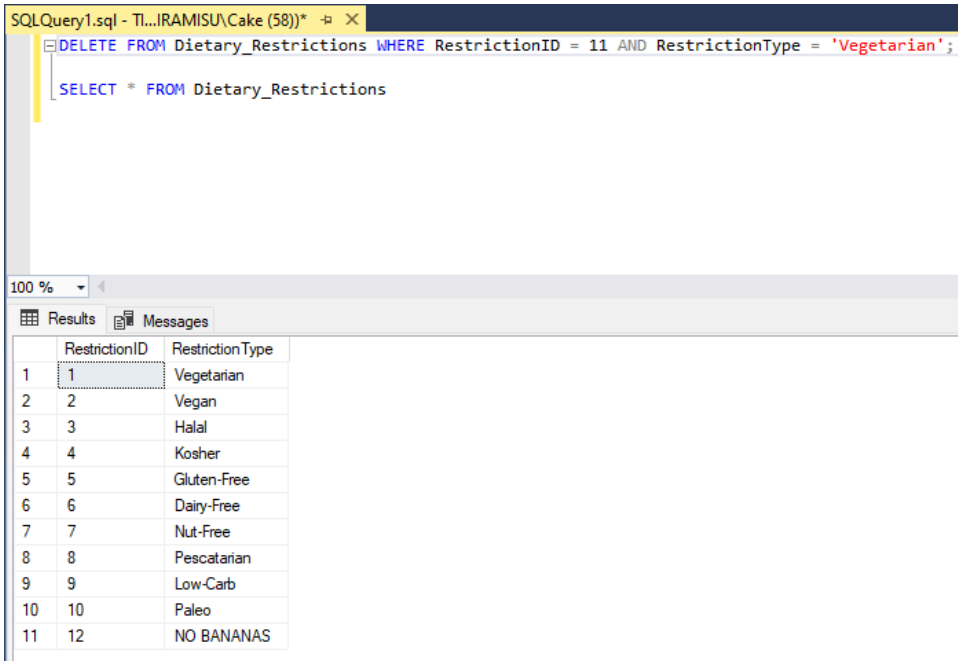
SELECT * FROM Dietary_Restrictions



	RestrictionID	RestrictionType
1	1	Vegetarian
2	2	Vegan
3	3	Halal
4	4	Kosher
5	5	Gluten-Free
6	6	Dairy-Free
7	7	Nut-Free
8	8	Pescatarian
9	9	Low-Carb
10	10	Paleo
11	11	Vegetarian
12	12	NO BANANAS

Delete:

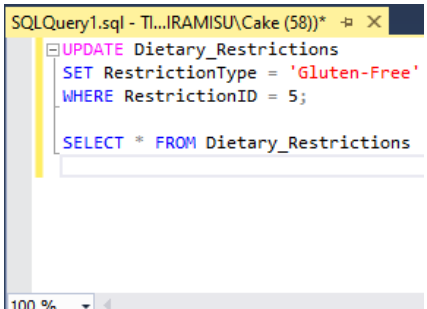
DELETE FROM Dietary_Restrictions WHERE RestrictionID = 11 AND RestrictionType = 'Vegetarian';



	RestrictionID	RestrictionType
1	1	Vegetarian
2	2	Vegan
3	3	Halal
4	4	Kosher
5	5	Gluten-Free
6	6	Dairy-Free
7	7	Nut-Free
8	8	Pescatarian
9	9	Low-Carb
10	10	Paleo
11	12	NO BANANAS

Update:

UPDATE Dietary_Restrictions
SET RestrictionType = 'Gluten-Free'
WHERE RestrictionID = 5;

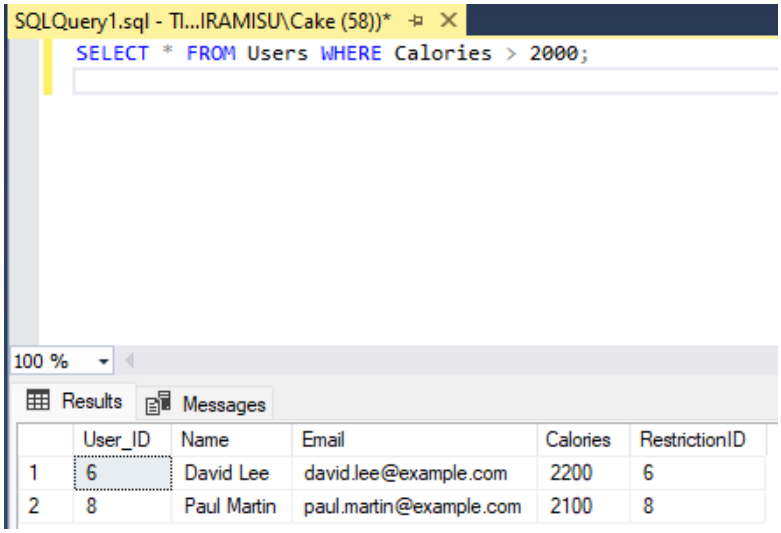


	RestrictionID	RestrictionType
1	1	Vegetarian
2	2	Vegan
3	3	Halal
4	4	Kosher
5	5	Gluten-Free
6	6	Dairy-Free
7	7	Nut-Free
8	8	Pescatarian
9	9	Low-Carb
10	10	Paleo
11	12	NO BANANAS

2. Users

Query:

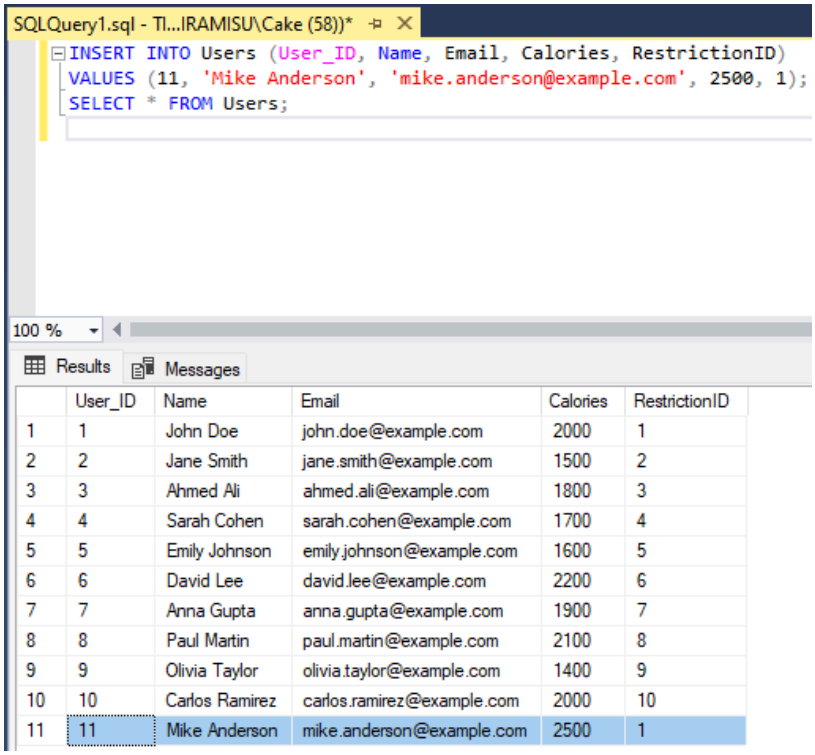
SELECT * FROM Users WHERE Calories > 2000;



	User_ID	Name	Email	Calories	RestrictionID
1	6	David Lee	david.lee@example.com	2200	6
2	8	Paul Martin	paul.martin@example.com	2100	8

Insert:

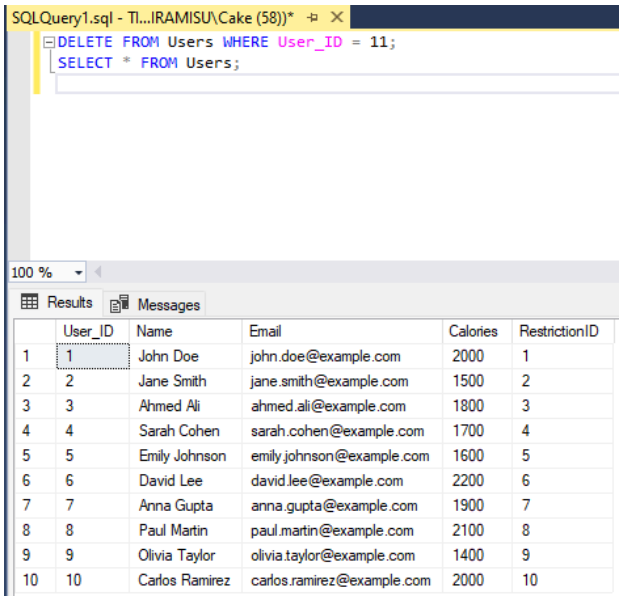
INSERT INTO Users (User_ID, Name, Email, Calories, RestrictionID)
VALUES (11, 'Mike Anderson', 'mike.anderson@example.com', 2500, 1);
SELECT * FROM Users;



	User_ID	Name	Email	Calories	RestrictionID
1	1	John Doe	john.doe@example.com	2000	1
2	2	Jane Smith	jane.smith@example.com	1500	2
3	3	Ahmed Ali	ahmed.ali@example.com	1800	3
4	4	Sarah Cohen	sarah.cohen@example.com	1700	4
5	5	Emily Johnson	emily.johnson@example.com	1600	5
6	6	David Lee	david.lee@example.com	2200	6
7	7	Anna Gupta	anna.gupta@example.com	1900	7
8	8	Paul Martin	paul.martin@example.com	2100	8
9	9	Olivia Taylor	olivia.taylor@example.com	1400	9
10	10	Carlos Ramirez	carlos.ramirez@example.com	2000	10
11	11	Mike Anderson	mike.anderson@example.com	2500	1

Delete:

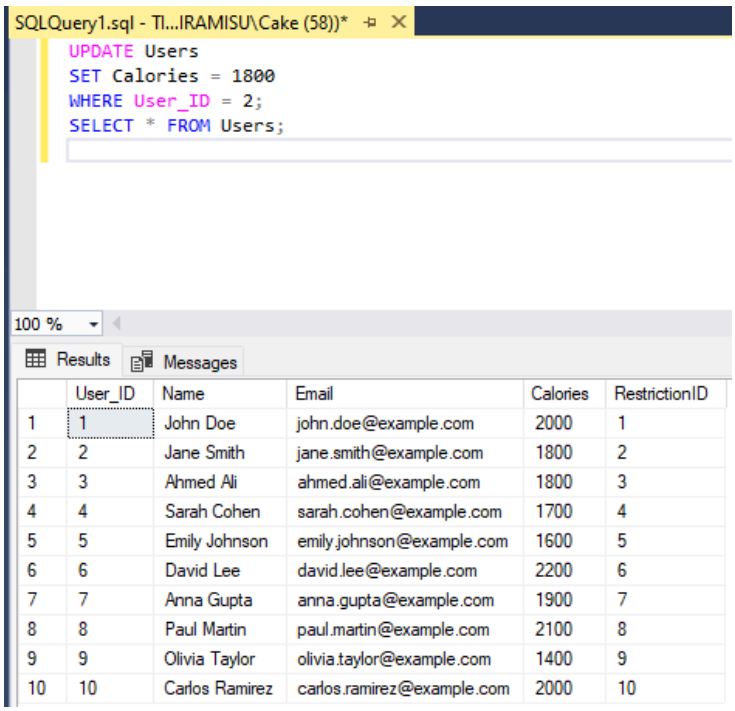
DELETE FROM Users WHERE User_ID = 11;
SELECT * FROM Users;



	User_ID	Name	Email	Calories	RestrictionID
1	1	John Doe	john.doe@example.com	2000	1
2	2	Jane Smith	jane.smith@example.com	1500	2
3	3	Ahmed Ali	ahmed.ali@example.com	1800	3
4	4	Sarah Cohen	sarah.cohen@example.com	1700	4
5	5	Emily Johnson	emily.johnson@example.com	1600	5
6	6	David Lee	david.lee@example.com	2200	6
7	7	Anna Gupta	anna.gupta@example.com	1900	7
8	8	Paul Martin	paul.martin@example.com	2100	8
9	9	Olivia Taylor	olivia.taylor@example.com	1400	9
10	10	Carlos Ramirez	carlos.ramirez@example.com	2000	10

Update:

UPDATE Users
SET Calories = 1800
WHERE User_ID = 2;

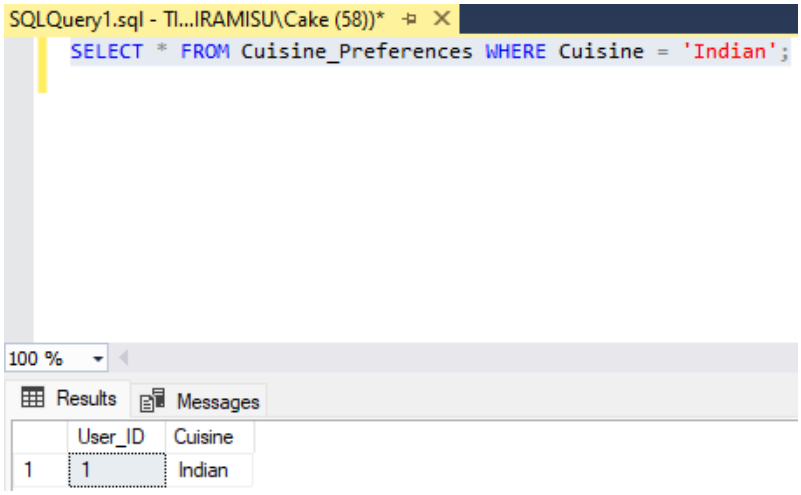


	User_ID	Name	Email	Calories	RestrictionID
1	1	John Doe	john.doe@example.com	2000	1
2	2	Jane Smith	jane.smith@example.com	1800	2
3	3	Ahmed Ali	ahmed.ali@example.com	1800	3
4	4	Sarah Cohen	sarah.cohen@example.com	1700	4
5	5	Emily Johnson	emily.johnson@example.com	1600	5
6	6	David Lee	david.lee@example.com	2200	6
7	7	Anna Gupta	anna.gupta@example.com	1900	7
8	8	Paul Martin	paul.martin@example.com	2100	8
9	9	Olivia Taylor	olivia.taylor@example.com	1400	9
10	10	Carlos Ramirez	carlos.ramirez@example.com	2000	10

3. Cuisine_Preferences

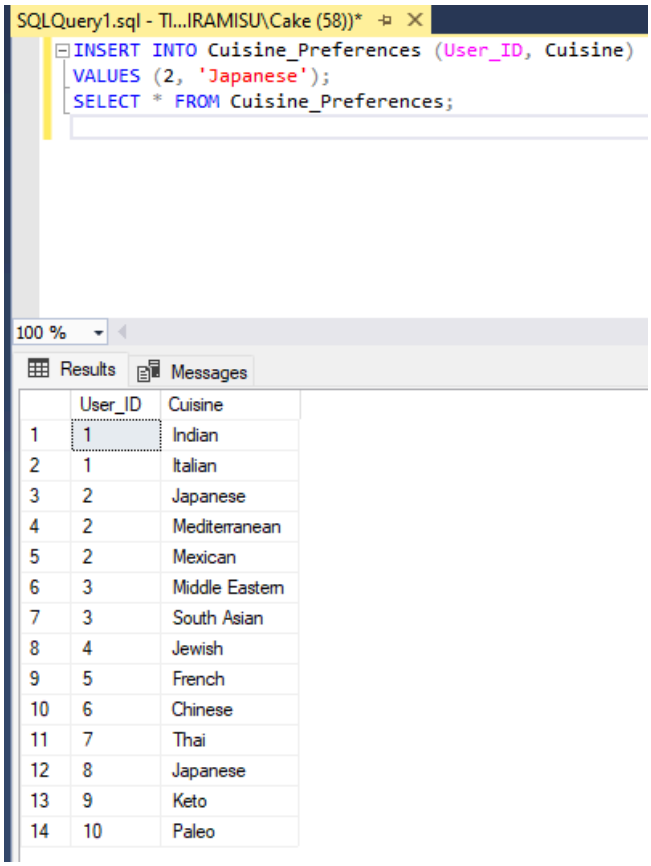
Query:

SELECT * FROM Cuisine_Preferences WHERE Cuisine = 'Indian';



Insert:

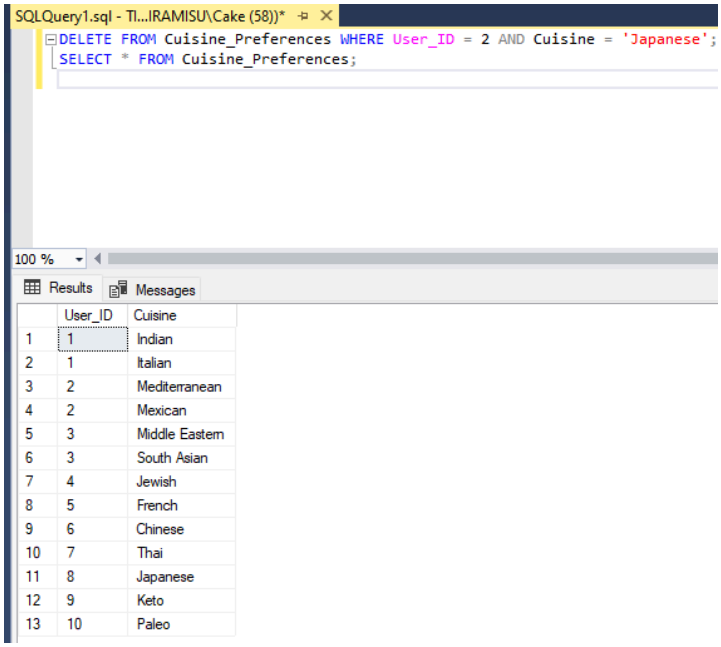
INSERT INTO Cuisine_Preferences (User_ID, Cuisine)
VALUES (2, 'Japanese');
SELECT * FROM Cuisine_Preferences;



	User_ID	Cuisine
1	1	Indian
2	1	Italian
3	2	Japanese
4	2	Mediterranean
5	2	Mexican
6	3	Middle Eastern
7	3	South Asian
8	4	Jewish
9	5	French
10	6	Chinese
11	7	Thai
12	8	Japanese
13	9	Keto
14	10	Paleo

Delete:

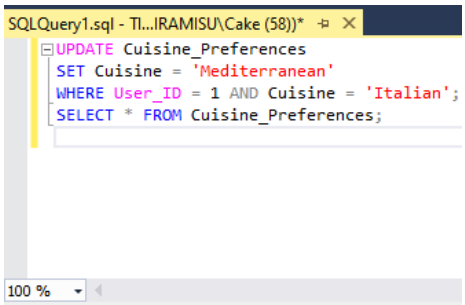
DELETE FROM Cuisine_Preferences WHERE User_ID = 2 AND Cuisine = 'Japanese';
SELECT * FROM Cuisine_Preferences;



	User_ID	Cuisine
1	1	Indian
2	1	Italian
3	2	Mediterranean
4	2	Mexican
5	3	Middle Eastern
6	3	South Asian
7	4	Jewish
8	5	French
9	6	Chinese
10	7	Thai
11	8	Japanese
12	9	Keto
13	10	Paleo

Update:

UPDATE Cuisine_Preferences
SET Cuisine = 'Mediterranean'
WHERE User_ID = 1 AND Cuisine = 'Italian';
SELECT * FROM Cuisine_Preferences;

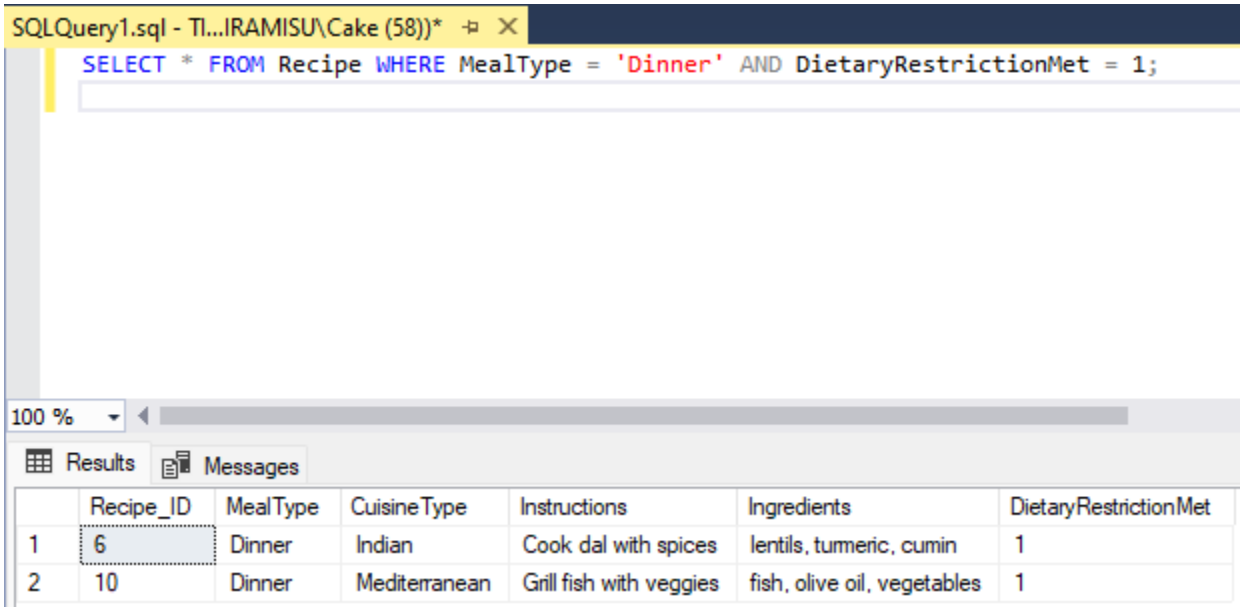


	User_ID	Cuisine
1	1	Indian
2	1	Mediterranean
3	2	Mediterranean
4	2	Mexican
5	3	Middle Eastern
6	3	South Asian
7	4	Jewish
8	5	French
9	6	Chinese
10	7	Thai
11	8	Japanese
12	9	Keto
13	10	Paleo

4. **Recipe**

Query:

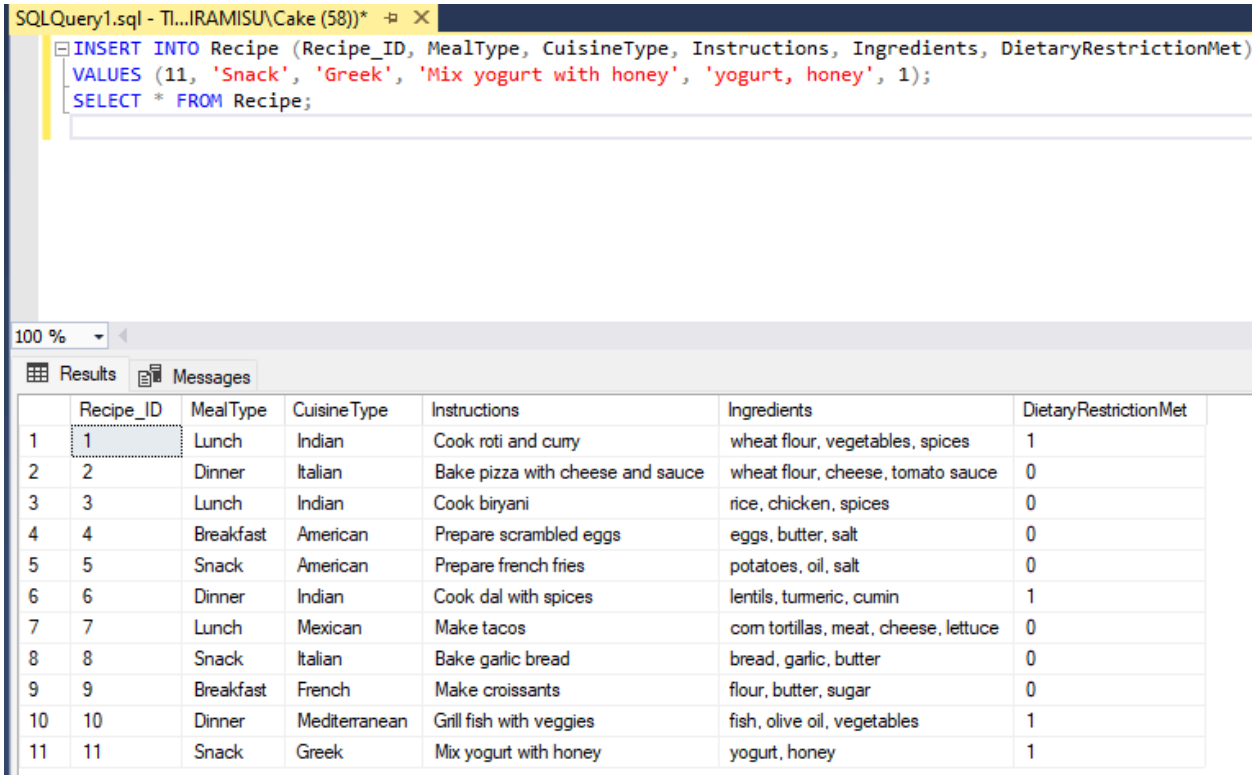
SELECT * FROM Cuisine_Preferences WHERE Cuisine = 'Indian';



	Recipe_ID	MealType	CuisineType	Instructions	Ingredients	DietaryRestrictionMet
1	6	Dinner	Indian	Cook dal with spices	lentils, turmeric, cumin	1
2	10	Dinner	Mediterranean	Grill fish with veggies	fish, olive oil, vegetables	1

Insert:

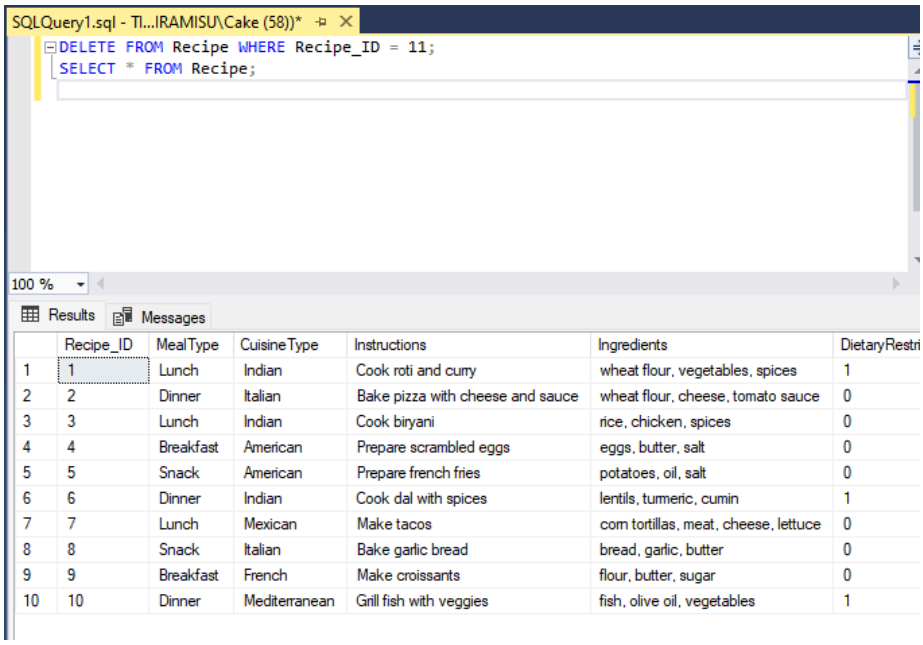
INSERT INTO Recipe (Recipe_ID, MealType, CuisineType, Instructions, Ingredients, DietaryRestrictionMet)
VALUES (11, 'Snack', 'Greek', 'Mix yogurt with honey', 'yogurt, honey', 1);
SELECT * FROM Recipe;



	Recipe_ID	MealType	CuisineType	Instructions	Ingredients	DietaryRestrictionMet
1	1	Lunch	Indian	Cook roti and curry	wheat flour, vegetables, spices	1
2	2	Dinner	Italian	Bake pizza with cheese and sauce	wheat flour, cheese, tomato sauce	0
3	3	Lunch	Indian	Cook biryani	rice, chicken, spices	0
4	4	Breakfast	American	Prepare scrambled eggs	eggs, butter, salt	0
5	5	Snack	American	Prepare french fries	potatoes, oil, salt	0
6	6	Dinner	Indian	Cook dal with spices	lentils, turmeric, cumin	1
7	7	Lunch	Mexican	Make tacos	corn tortillas, meat, cheese, lettuce	0
8	8	Snack	Italian	Bake garlic bread	bread, garlic, butter	0
9	9	Breakfast	French	Make croissants	flour, butter, sugar	0
10	10	Dinner	Mediterranean	Grill fish with veggies	fish, olive oil, vegetables	1
11	11	Snack	Greek	Mix yogurt with honey	yogurt, honey	1

Delete:

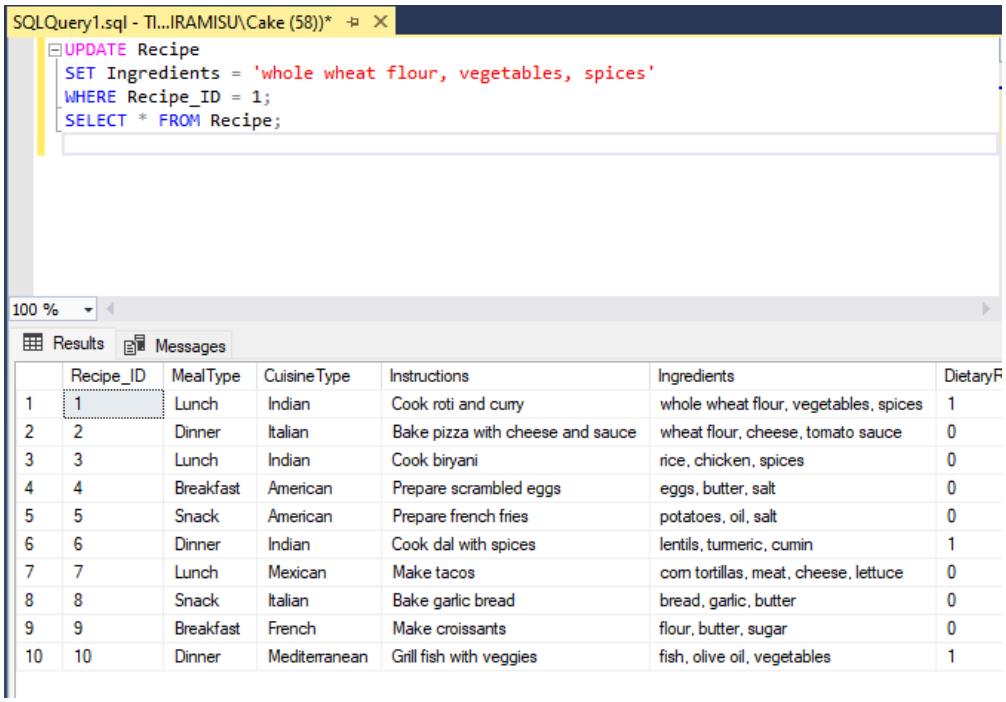
DELETE FROM Recipe WHERE Recipe_ID = 11;
SELECT * FROM Recipe;



	Recipe_ID	MealType	CuisineType	Instructions	Ingredients	DietaryRestri
1	1	Lunch	Indian	Cook roti and curry	wheat flour, vegetables, spices	1
2	2	Dinner	Italian	Bake pizza with cheese and sauce	wheat flour, cheese, tomato sauce	0
3	3	Lunch	Indian	Cook biryani	rice, chicken, spices	0
4	4	Breakfast	American	Prepare scrambled eggs	eggs, butter, salt	0
5	5	Snack	American	Prepare french fries	potatoes, oil, salt	0
6	6	Dinner	Indian	Cook dal with spices	lentils, turmeric, cumin	1
7	7	Lunch	Mexican	Make tacos	corn tortillas, meat, cheese, lettuce	0
8	8	Snack	Italian	Bake garlic bread	bread, garlic, butter	0
9	9	Breakfast	French	Make croissants	flour, butter, sugar	0
10	10	Dinner	Mediterranean	Grill fish with veggies	fish, olive oil, vegetables	1

Update:

UPDATE Recipe
SET Ingredients = 'whole wheat flour, vegetables, spices'
WHERE Recipe_ID = 1;
SELECT * FROM Recipe;

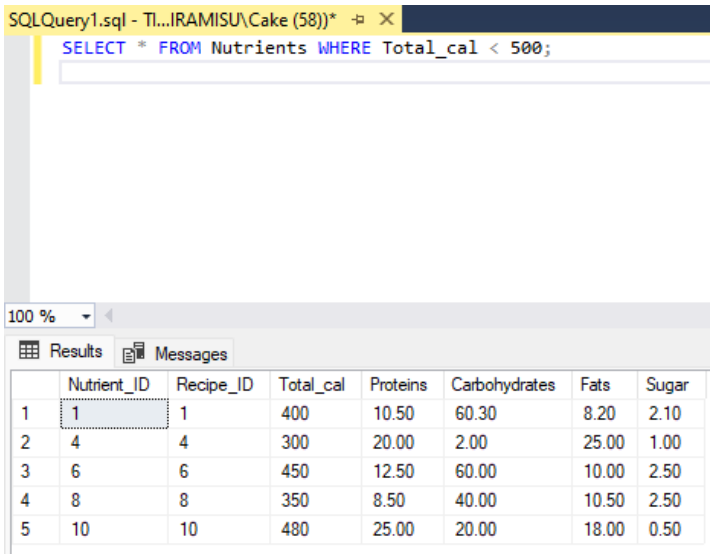


	Recipe_ID	MealType	CuisineType	Instructions	Ingredients	DietaryF
1	1	Lunch	Indian	Cook roti and curry	whole wheat flour, vegetables, spices	1
2	2	Dinner	Italian	Bake pizza with cheese and sauce	wheat flour, cheese, tomato sauce	0
3	3	Lunch	Indian	Cook biryani	rice, chicken, spices	0
4	4	Breakfast	American	Prepare scrambled eggs	eggs, butter, salt	0
5	5	Snack	American	Prepare french fries	potatoes, oil, salt	0
6	6	Dinner	Indian	Cook dal with spices	lentils, turmeric, cumin	1
7	7	Lunch	Mexican	Make tacos	corn tortillas, meat, cheese, lettuce	0
8	8	Snack	Italian	Bake garlic bread	bread, garlic, butter	0
9	9	Breakfast	French	Make croissants	flour, butter, sugar	0
10	10	Dinner	Mediterranean	Grill fish with veggies	fish, olive oil, vegetables	1

5. **Nutrients**

Query:

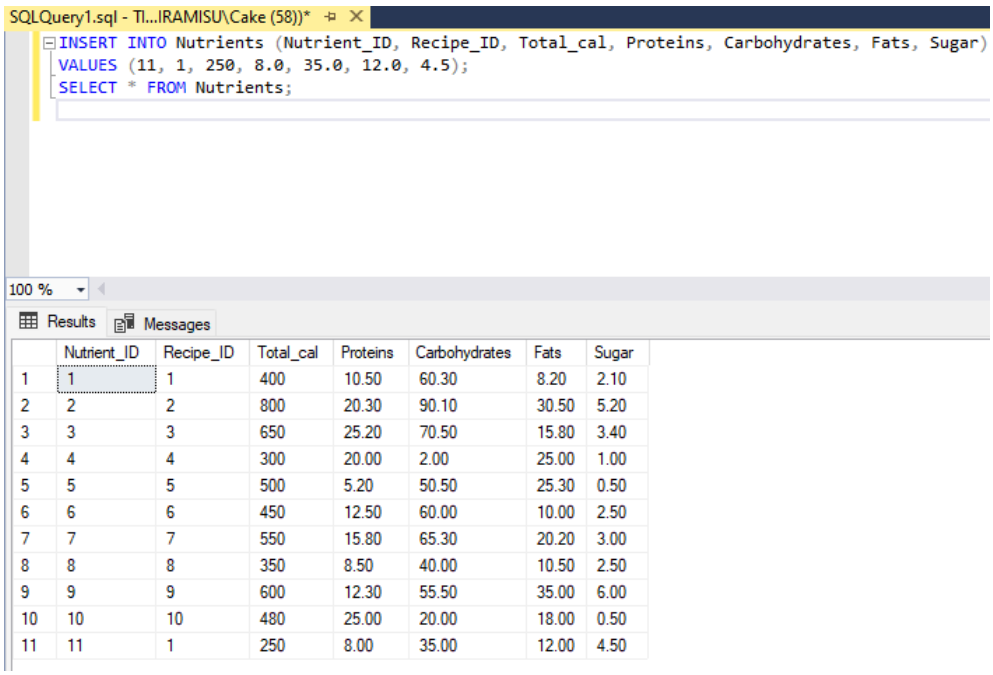
SELECT * FROM Nutrients WHERE Total_cal < 500;



	Nutrient_ID	Recipe_ID	Total_cal	Proteins	Carbohydrates	Fats	Sugar
1	1	1	400	10.50	60.30	8.20	2.10
2	4	4	300	20.00	2.00	25.00	1.00
3	6	6	450	12.50	60.00	10.00	2.50
4	8	8	350	8.50	40.00	10.50	2.50
5	10	10	480	25.00	20.00	18.00	0.50

Insert:

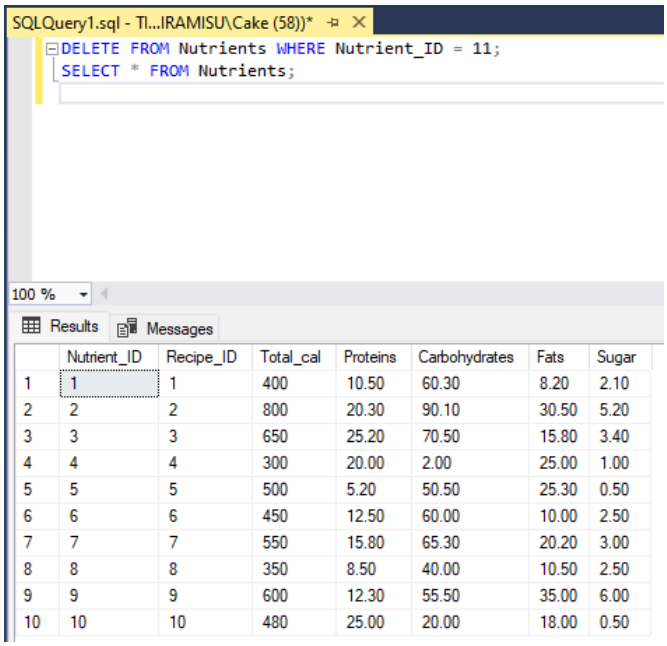
INSERT INTO Nutrients (Nutrient_ID, Recipe_ID, Total_cal, Proteins, Carbohydrates, Fats, Sugar)
VALUES (11, 1, 250, 8.0, 35.0, 12.0, 4.5);
SELECT * FROM Nutrients;



	Nutrient_ID	Recipe_ID	Total_cal	Proteins	Carbohydrates	Fats	Sugar
1	1	1	400	10.50	60.30	8.20	2.10
2	2	2	800	20.30	90.10	30.50	5.20
3	3	3	650	25.20	70.50	15.80	3.40
4	4	4	300	20.00	2.00	25.00	1.00
5	5	5	500	5.20	50.50	25.30	0.50
6	6	6	450	12.50	60.00	10.00	2.50
7	7	7	550	15.80	65.30	20.20	3.00
8	8	8	350	8.50	40.00	10.50	2.50
9	9	9	600	12.30	55.50	35.00	6.00
10	10	10	480	25.00	20.00	18.00	0.50
11	11	1	250	8.00	35.00	12.00	4.50

Delete:

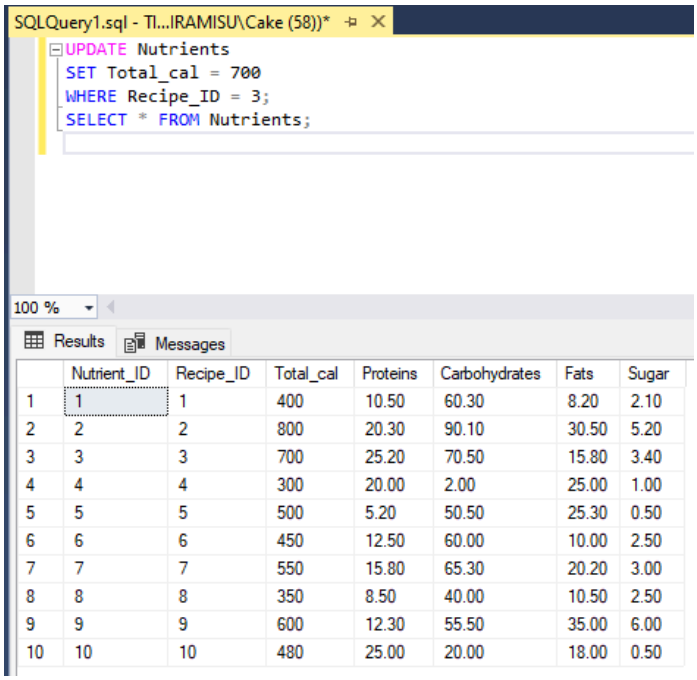
DELETE FROM Nutrients WHERE Nutrient_ID = 11;
SELECT * FROM Nutrients;



	Nutrient_ID	Recipe_ID	Total_cal	Proteins	Carbohydrates	Fats	Sugar
1	1	1	400	10.50	60.30	8.20	2.10
2	2	2	800	20.30	90.10	30.50	5.20
3	3	3	650	25.20	70.50	15.80	3.40
4	4	4	300	20.00	2.00	25.00	1.00
5	5	5	500	5.20	50.50	25.30	0.50
6	6	6	450	12.50	60.00	10.00	2.50
7	7	7	550	15.80	65.30	20.20	3.00
8	8	8	350	8.50	40.00	10.50	2.50
9	9	9	600	12.30	55.50	35.00	6.00
10	10	10	480	25.00	20.00	18.00	0.50

Update:

UPDATE Nutrients
SET Total_cal = 700
WHERE Recipe_ID = 3;
SELECT * FROM Nutrients;

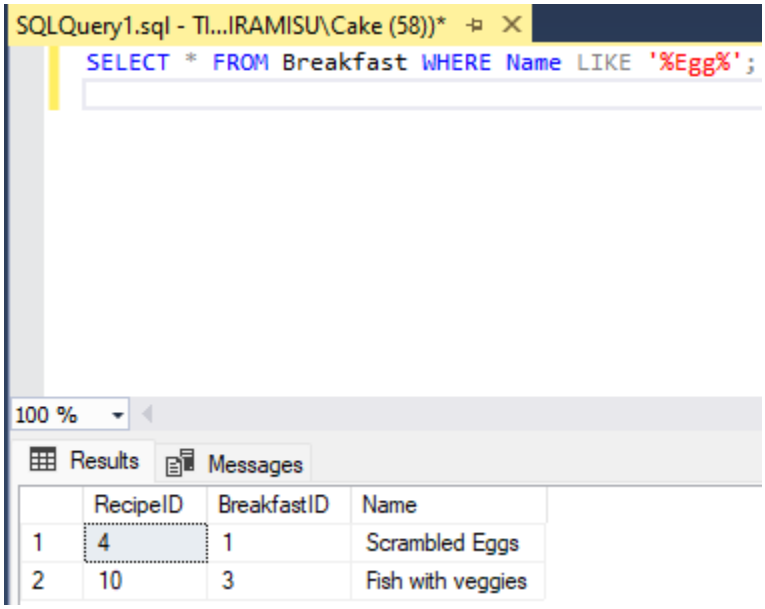


	Nutrient_ID	Recipe_ID	Total_cal	Proteins	Carbohydrates	Fats	Sugar
1	1	1	400	10.50	60.30	8.20	2.10
2	2	2	800	20.30	90.10	30.50	5.20
3	3	3	700	25.20	70.50	15.80	3.40
4	4	4	300	20.00	2.00	25.00	1.00
5	5	5	500	5.20	50.50	25.30	0.50
6	6	6	450	12.50	60.00	10.00	2.50
7	7	7	550	15.80	65.30	20.20	3.00
8	8	8	350	8.50	40.00	10.50	2.50
9	9	9	600	12.30	55.50	35.00	6.00
10	10	10	480	25.00	20.00	18.00	0.50

6. **Breakfast**

Query:

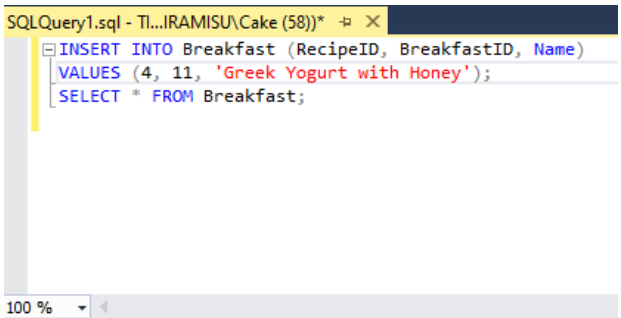
SELECT * FROM Breakfast WHERE Name LIKE '%Egg%';



	RecipeID	BreakfastID	Name
1	4	1	Scrambled Eggs
2	10	3	Fish with veggies

Insert:

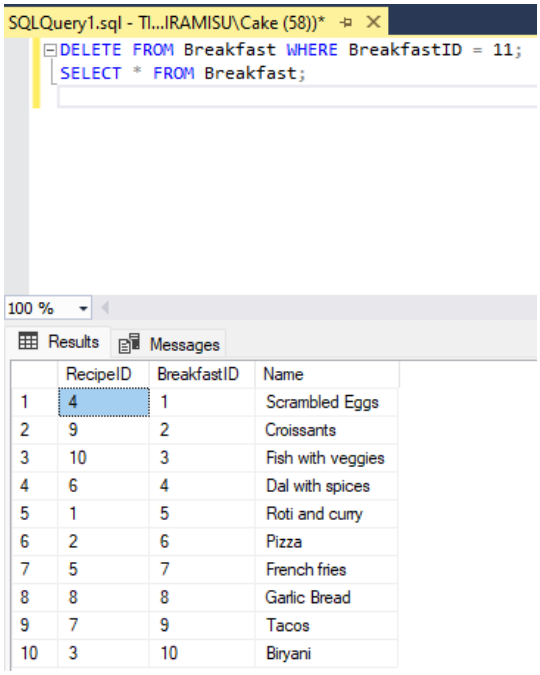
INSERT INTO Breakfast (RecipeID, BreakfastID, Name)
VALUES (4, 11, 'Greek Yogurt with Honey');
SELECT * FROM Breakfast;



	RecipeID	BreakfastID	Name
1	4	1	Scrambled Eggs
2	9	2	Croissants
3	10	3	Fish with veggies
4	6	4	Dal with spices
5	1	5	Roti and curry
6	2	6	Pizza
7	5	7	French fries
8	8	8	Garlic Bread
9	7	9	Tacos
10	3	10	Biryani
11	4	11	Greek Yogurt with Honey

Delete:

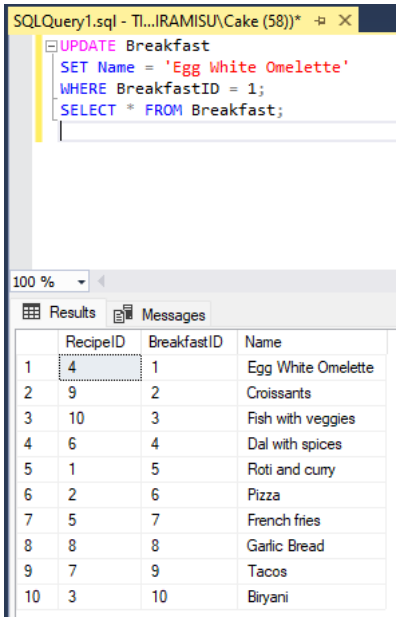
DELETE FROM Breakfast WHERE BreakfastID = 5;
SELECT * FROM Breakfast;



	RecipeID	BreakfastID	Name
1	4	1	Scrambled Eggs
2	9	2	Croissants
3	10	3	Fish with veggies
4	6	4	Dal with spices
5	1	5	Roti and curry
6	2	6	Pizza
7	5	7	French fries
8	8	8	Garlic Bread
9	7	9	Tacos
10	3	10	Biryani

Update:

UPDATE Breakfast
SET Name = 'Egg White Omelette'
WHERE BreakfastID = 1;
SELECT * FROM Breakfast;

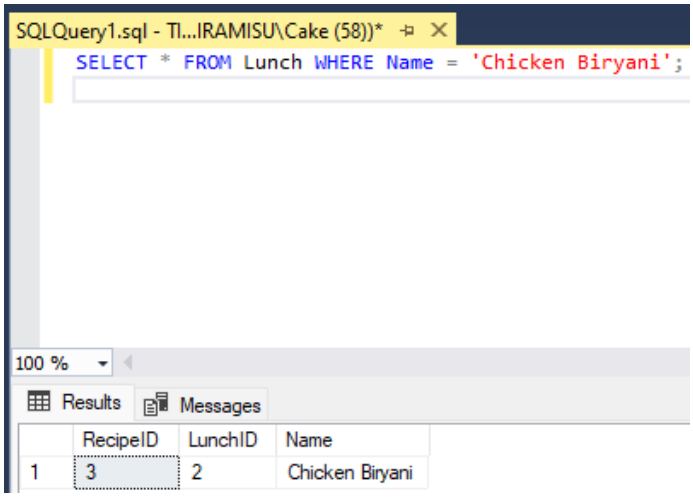


	RecipeID	BreakfastID	Name
1	4	1	Egg White Omelette
2	9	2	Croissants
3	10	3	Fish with veggies
4	6	4	Dal with spices
5	1	5	Roti and curry
6	2	6	Pizza
7	5	7	French fries
8	8	8	Garlic Bread
9	7	9	Tacos
10	3	10	Biryani

7. **Lunch**

Query:

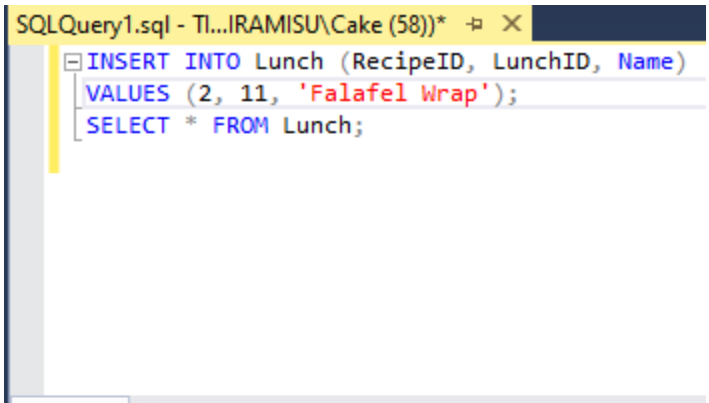
SELECT * FROM Lunch WHERE Name = 'Chicken Biryani';



	RecipeID	LunchID	Name
1	3	2	Chicken Biryani

Insert:

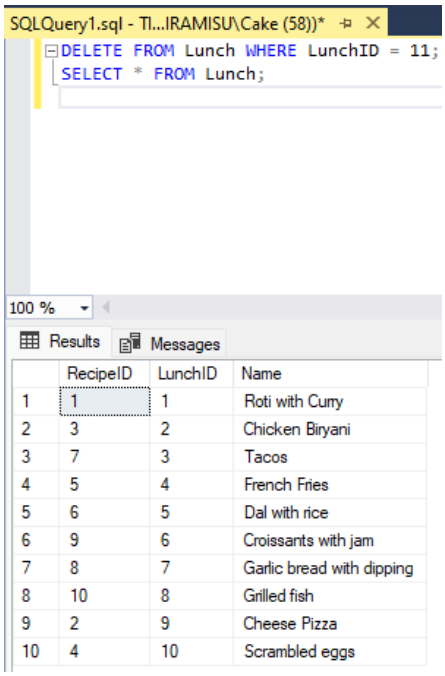
INSERT INTO Lunch (RecipeID, LunchID, Name)
VALUES (2,11, 'Falafel Wrap');
SELECT * FROM Lunch;



	RecipeID	LunchID	Name
1	1	1	Roti with Curry
2	3	2	Chicken Biryani
3	7	3	Tacos
4	5	4	French Fries
5	6	5	Dal with rice
6	9	6	Croissants with jam
7	8	7	Garlic bread with dipping
8	10	8	Grilled fish
9	2	9	Cheese Pizza
10	4	10	Scrambled eggs
11	2	11	Falafel Wrap

Delete:

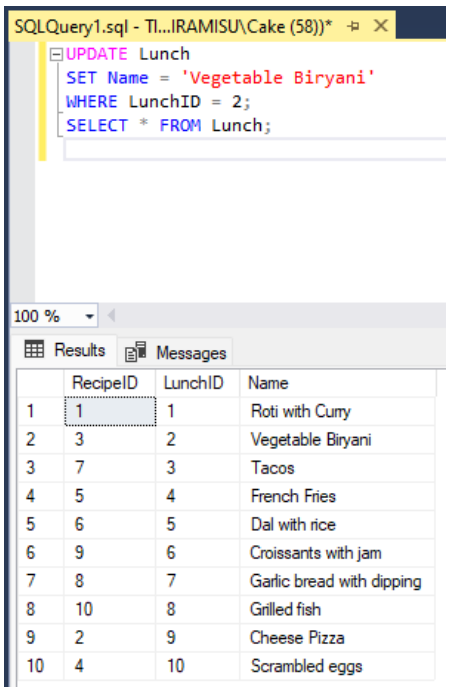
```
DELETE FROM Lunch WHERE LunchID = 11;
SELECT * FROM Lunch;
```



	RecipeID	LunchID	Name
1	1	1	Roti with Curry
2	3	2	Chicken Biryani
3	7	3	Tacos
4	5	4	French Fries
5	6	5	Dal with rice
6	9	6	Croissants with jam
7	8	7	Garlic bread with dipping
8	10	8	Grilled fish
9	2	9	Cheese Pizza
10	4	10	Scrambled eggs

Update:

```
UPDATE Lunch
SET Name = 'Vegetable Biryani'
WHERE LunchID = 2;
SELECT * FROM Lunch;
```

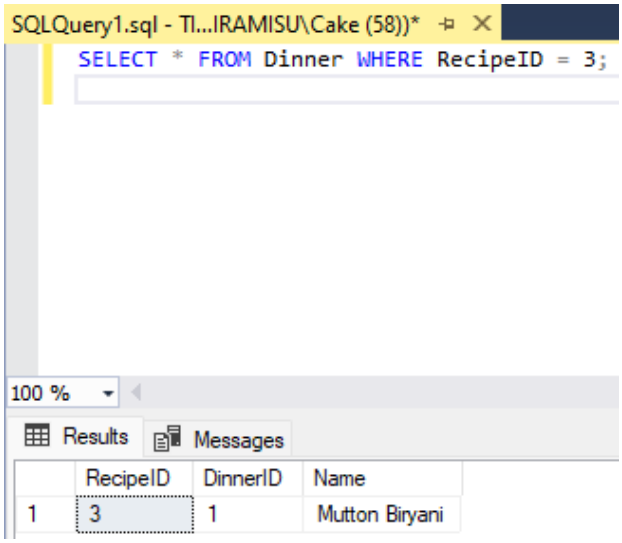


	RecipeID	LunchID	Name
1	1	1	Roti with Curry
2	3	2	Vegetable Biryani
3	7	3	Tacos
4	5	4	French Fries
5	6	5	Dal with rice
6	9	6	Croissants with jam
7	8	7	Garlic bread with dipping
8	10	8	Grilled fish
9	2	9	Cheese Pizza
10	4	10	Scrambled eggs

8. **Dinner**

Query:

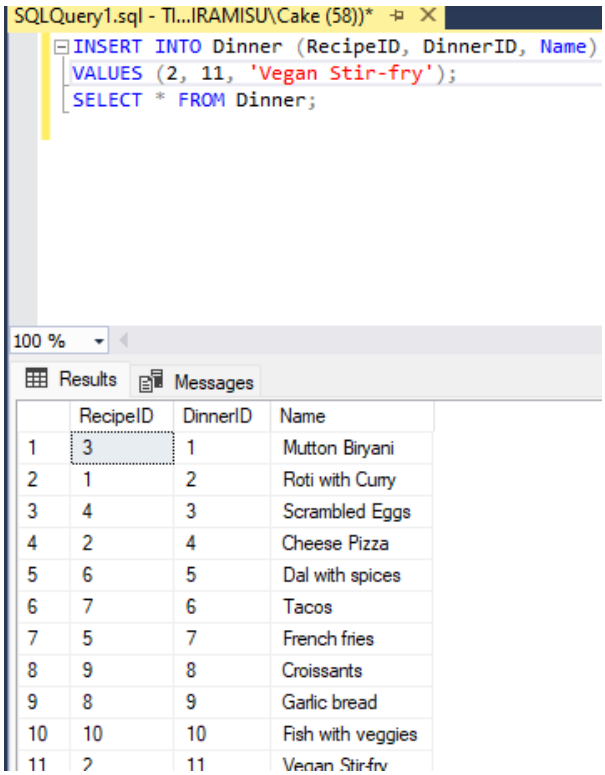
SELECT * FROM Dinner WHERE RecipeID = 3;



	RecipeID	DinnerID	Name
1	3	1	Mutton Biryani

Insert:

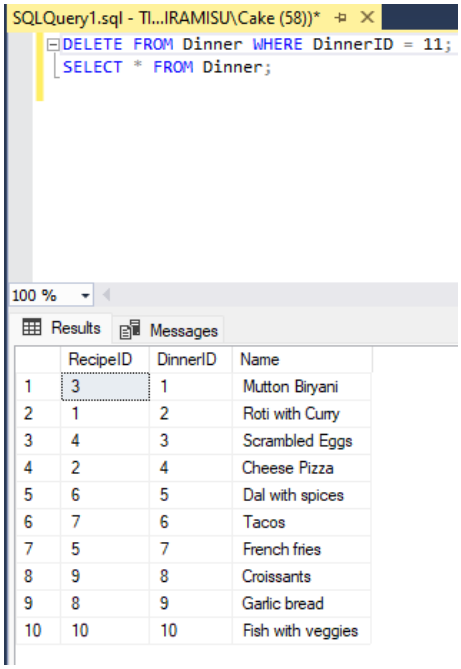
INSERT INTO Dinner (RecipeID, DinnerID, Name)
VALUES (2, 11, 'Vegan Stir-fry');
SELECT * FROM Dinner;



	RecipeID	DinnerID	Name
1	3	1	Mutton Biryani
2	1	2	Roti with Curry
3	4	3	Scrambled Eggs
4	2	4	Cheese Pizza
5	6	5	Dal with spices
6	7	6	Tacos
7	5	7	French fries
8	9	8	Croissants
9	8	9	Garlic bread
10	10	10	Fish with veggies
11	2	11	Vegan Stir-fry

Delete:

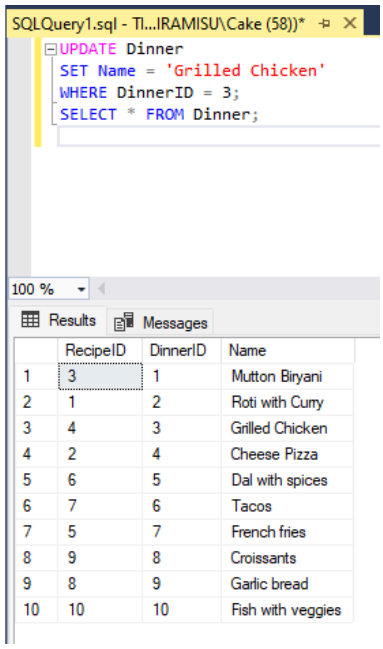
```
DELETE FROM Dinner WHERE DinnerID = 11;
SELECT * FROM Dinner;
```



	RecipeID	DinnerID	Name
1	3	1	Mutton Biryani
2	1	2	Roti with Curry
3	4	3	Scrambled Eggs
4	2	4	Cheese Pizza
5	6	5	Dal with spices
6	7	6	Tacos
7	5	7	French fries
8	9	8	Croissants
9	8	9	Garlic bread
10	10	10	Fish with veggies

Update:

```
UPDATE Dinner
SET Name = 'Grilled Chicken'
WHERE DinnerID = 3;
SELECT * FROM Dinner;
```

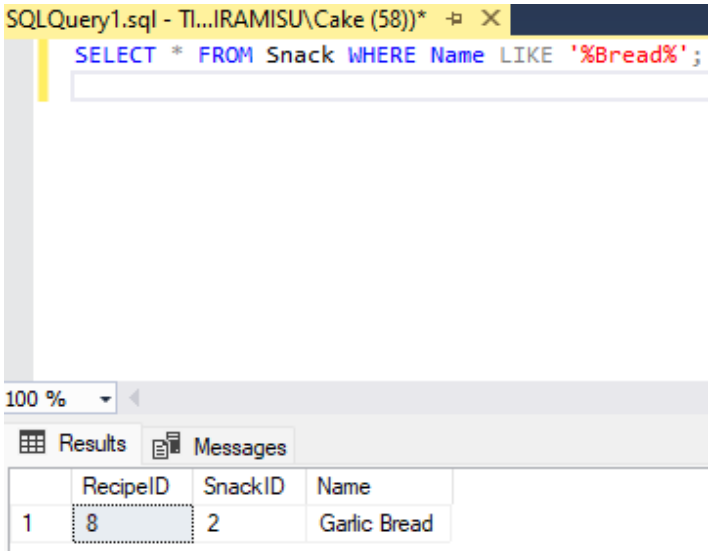


	RecipeID	DinnerID	Name
1	3	1	Mutton Biryani
2	1	2	Roti with Curry
3	4	3	Grilled Chicken
4	2	4	Cheese Pizza
5	6	5	Dal with spices
6	7	6	Tacos
7	5	7	French fries
8	9	8	Croissants
9	8	9	Garlic bread
10	10	10	Fish with veggies

9. **Snack**

Query:

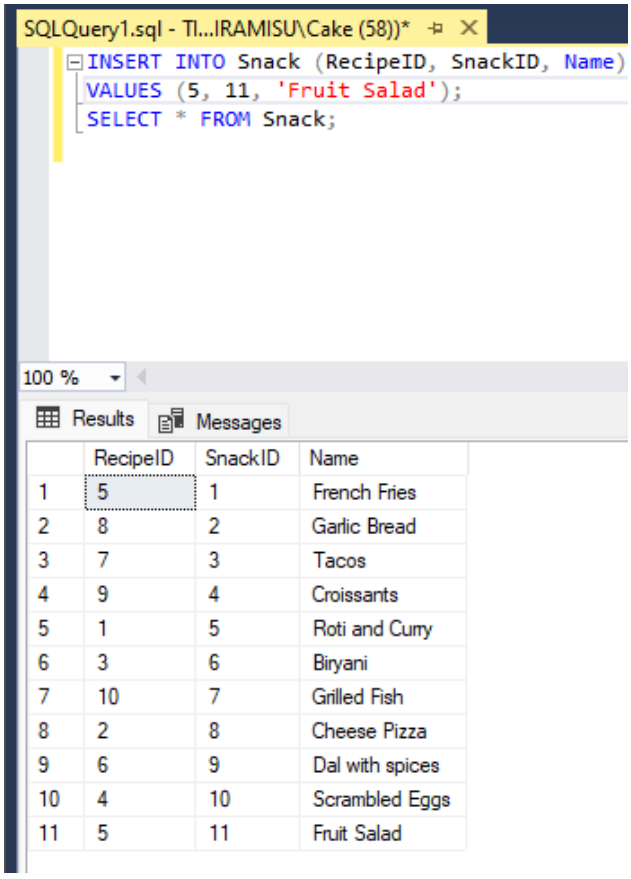
SELECT * FROM Snack WHERE Name LIKE '%Bread%';



	RecipeID	SnackID	Name
1	8	2	Garlic Bread

Insert:

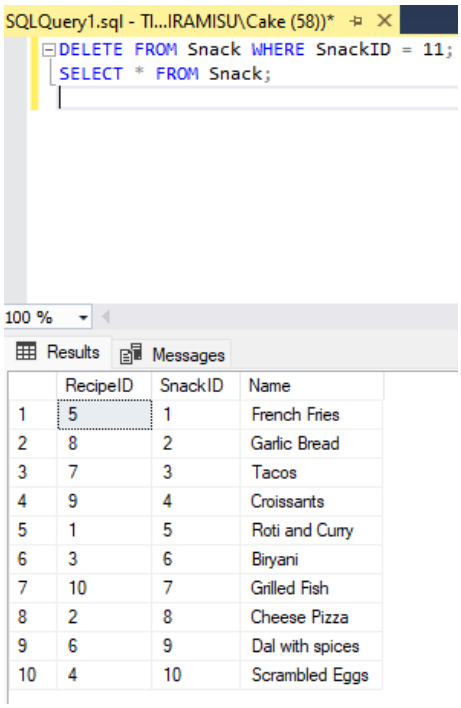
INSERT INTO Snack (RecipeID, SnackID, Name)
VALUES (5, 11, 'Fruit Salad');
SELECT * FROM Snack;



	RecipeID	SnackID	Name
1	5	1	French Fries
2	8	2	Garlic Bread
3	7	3	Tacos
4	9	4	Croissants
5	1	5	Roti and Curry
6	3	6	Biryani
7	10	7	Grilled Fish
8	2	8	Cheese Pizza
9	6	9	Dal with spices
10	4	10	Scrambled Eggs
11	5	11	Fruit Salad

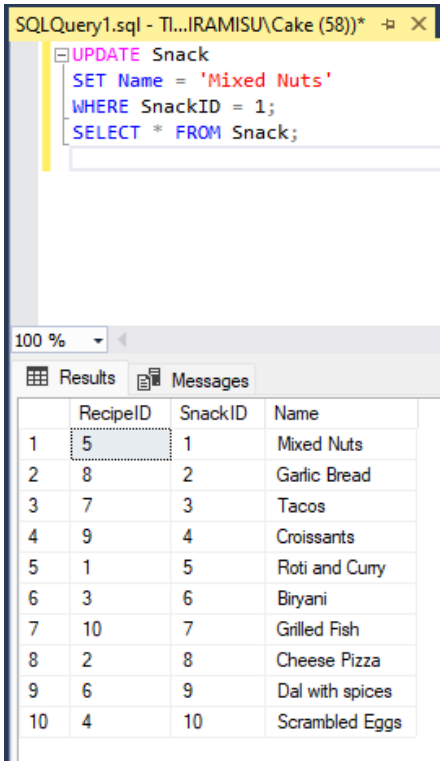
Delete:

DELETE FROM Snack WHERE SnackID = 11;
SELECT * FROM Snack;



	RecipeID	SnackID	Name
1	5	1	French Fries
2	8	2	Garlic Bread
3	7	3	Tacos
4	9	4	Croissants
5	1	5	Roti and Curry
6	3	6	Biryani
7	10	7	Grilled Fish
8	2	8	Cheese Pizza
9	6	9	Dal with spices
10	4	10	Scrambled Eggs

Update:

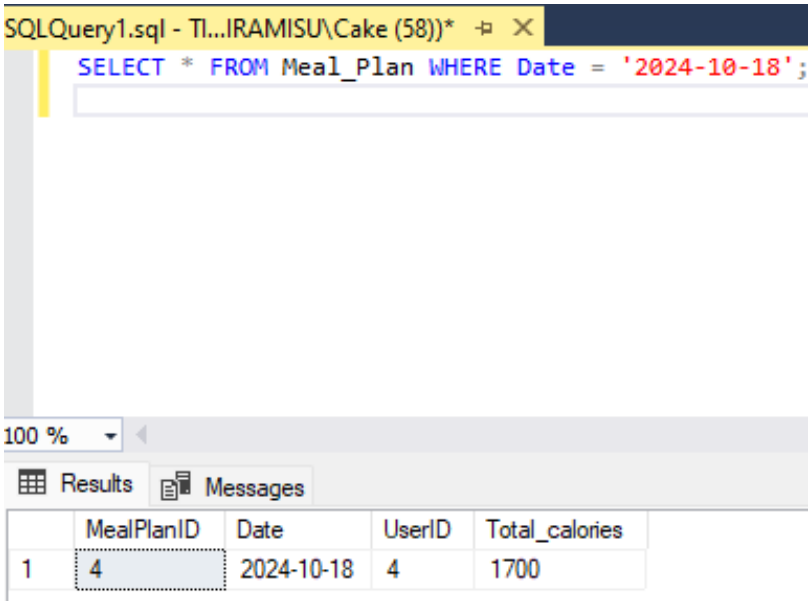


	RecipeID	SnackID	Name
1	5	1	Mixed Nuts
2	8	2	Garlic Bread
3	7	3	Tacos
4	9	4	Croissants
5	1	5	Roti and Curry
6	3	6	Biryani
7	10	7	Grilled Fish
8	2	8	Cheese Pizza
9	6	9	Dal with spices
10	4	10	Scrambled Eggs

10. Meal_Plan

Query:

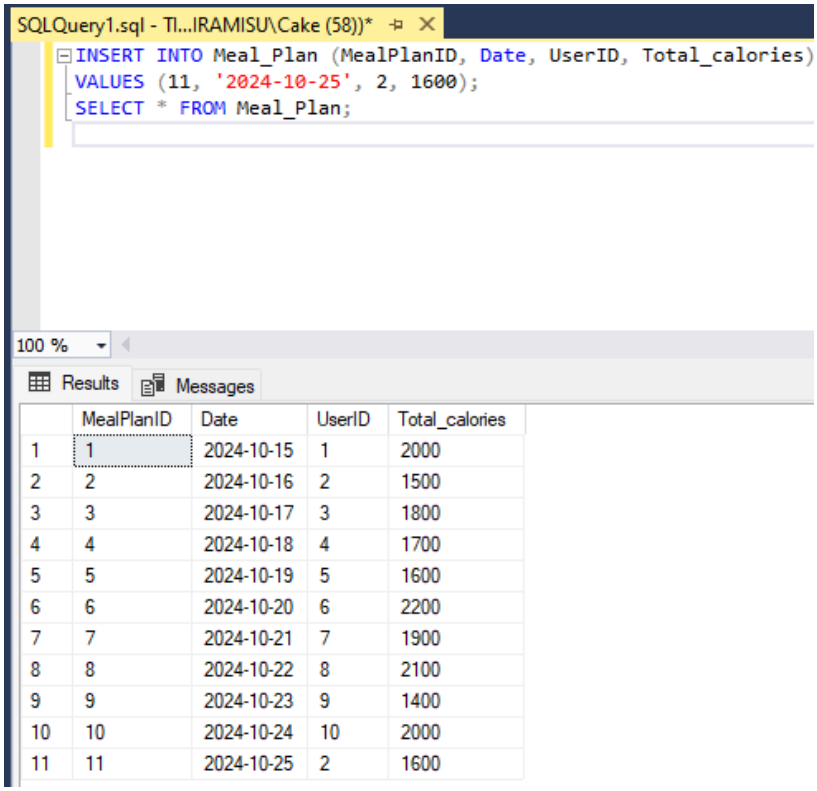
SELECT * FROM Meal_Plan WHERE Date = '2024-10-18';



	MealPlanID	Date	UserID	Total_calories
1	4	2024-10-18	4	1700

Insert:

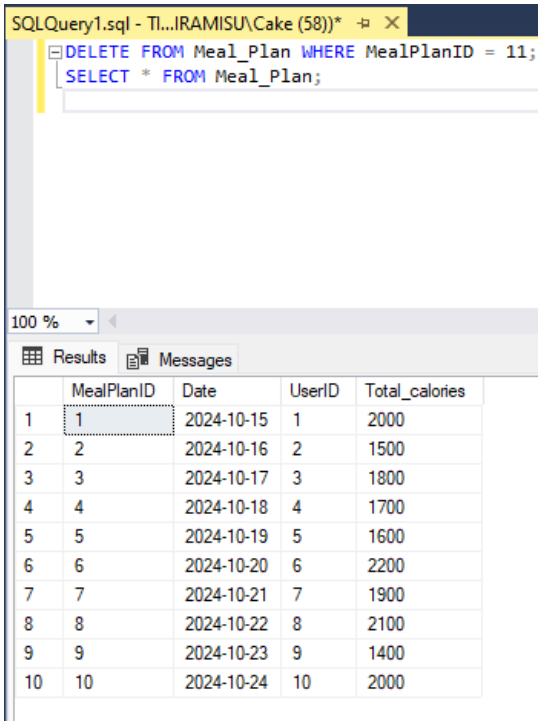
INSERT INTO Meal_Plan (MealPlanID, Date, UserID, Total_calories)
VALUES (11, '2024-10-25', 2, 1600);
SELECT * FROM Meal_Plan;



	MealPlanID	Date	UserID	Total_calories
1	1	2024-10-15	1	2000
2	2	2024-10-16	2	1500
3	3	2024-10-17	3	1800
4	4	2024-10-18	4	1700
5	5	2024-10-19	5	1600
6	6	2024-10-20	6	2200
7	7	2024-10-21	7	1900
8	8	2024-10-22	8	2100
9	9	2024-10-23	9	1400
10	10	2024-10-24	10	2000
11	11	2024-10-25	2	1600

Delete:

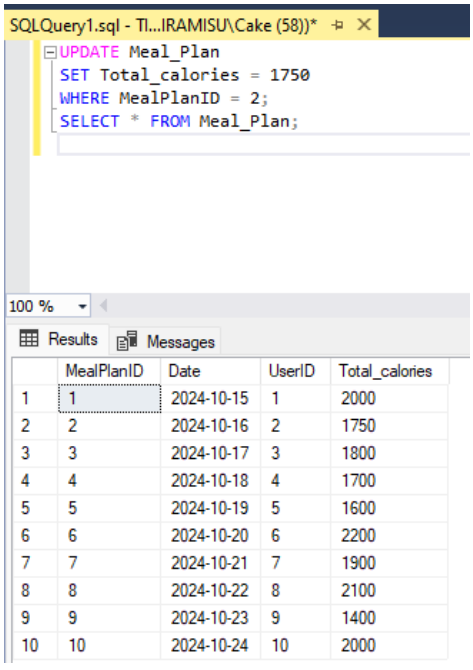
DELETE FROM Meal_Plan WHERE MealPlanID = 11;
SELECT * FROM Meal_Plan;



	MealPlanID	Date	UserID	Total_calories
1	1	2024-10-15	1	2000
2	2	2024-10-16	2	1500
3	3	2024-10-17	3	1800
4	4	2024-10-18	4	1700
5	5	2024-10-19	5	1600
6	6	2024-10-20	6	2200
7	7	2024-10-21	7	1900
8	8	2024-10-22	8	2100
9	9	2024-10-23	9	1400
10	10	2024-10-24	10	2000

Update:

UPDATE Meal_Plan
SET Total_calories = 1750
WHERE MealPlanID = 2;
SELECT * FROM Meal_Plan;

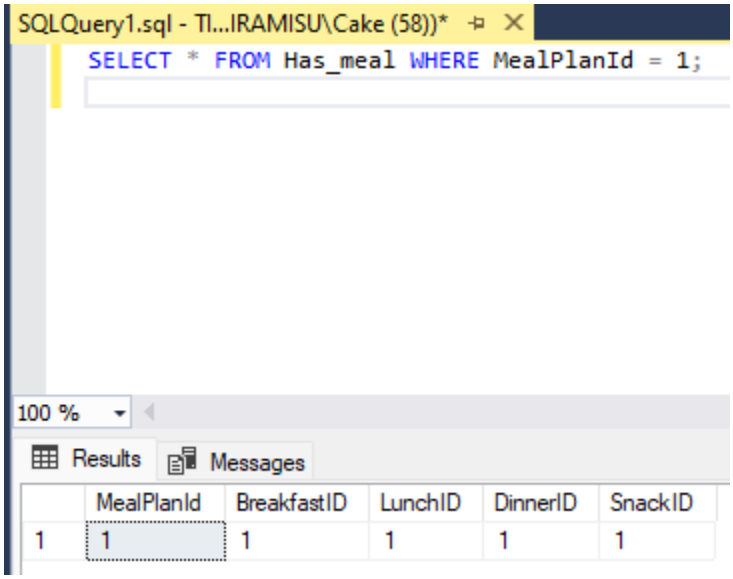


	MealPlanID	Date	UserID	Total_calories
1	1	2024-10-15	1	2000
2	2	2024-10-16	2	1750
3	3	2024-10-17	3	1800
4	4	2024-10-18	4	1700
5	5	2024-10-19	5	1600
6	6	2024-10-20	6	2200
7	7	2024-10-21	7	1900
8	8	2024-10-22	8	2100
9	9	2024-10-23	9	1400
10	10	2024-10-24	10	2000

11. Has_Meal

Query:

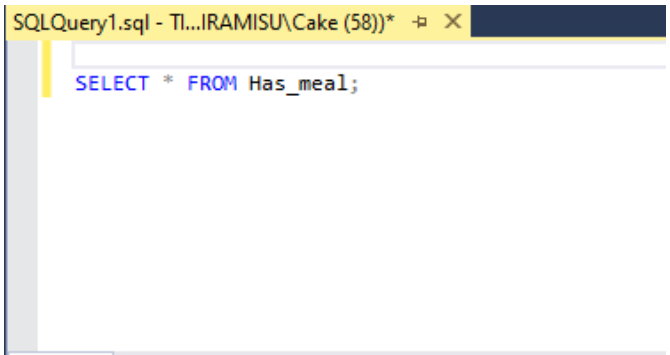
SELECT * FROM Has_meal WHERE MealPlanId = 1;



	MealPlanId	BreakfastID	LunchID	DinnerID	SnackID
1	1	1	1	1	1

Insert:

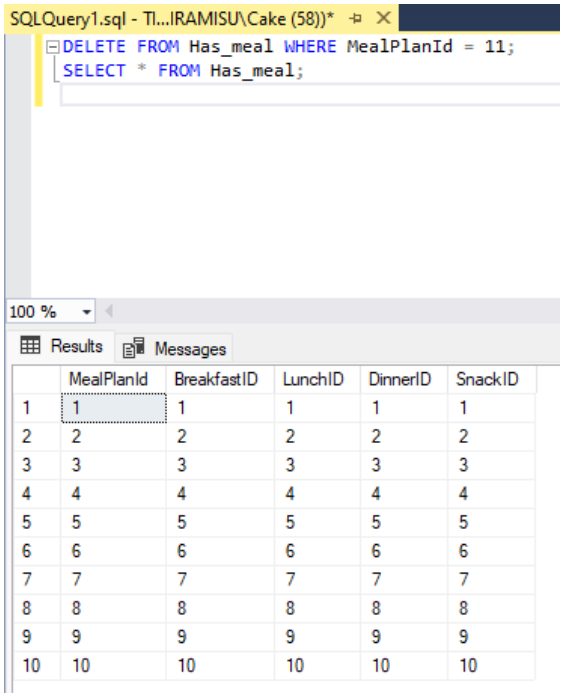
INSERT INTO Has_meal (MealPlanId, BreakfastID, LunchID, DinnerID, SnackID)
VALUES (11, 2, 3, 1, 4);



	MealPlanId	BreakfastID	LunchID	DinnerID	SnackID
1	1	1	1	1	1
2	2	2	2	2	2
3	3	3	3	3	3
4	4	4	4	4	4
5	5	5	5	5	5
6	6	6	6	6	6
7	7	7	7	7	7
8	8	8	8	8	8
9	9	9	9	9	9
10	10	10	10	10	10
11	11	2	3	1	4

Delete:

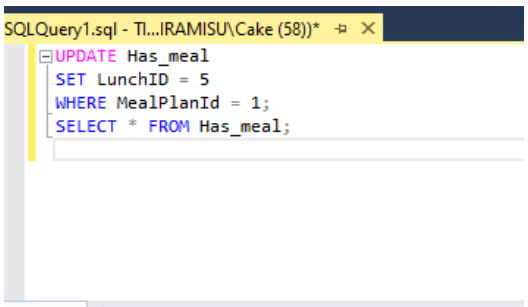
DELETE FROM Has_meal WHERE MealPlanId = 11;
SELECT * FROM Has_meal;



	MealPlanId	BreakfastID	LunchID	DinnerID	SnackID
1	1	1	1	1	1
2	2	2	2	2	2
3	3	3	3	3	3
4	4	4	4	4	4
5	5	5	5	5	5
6	6	6	6	6	6
7	7	7	7	7	7
8	8	8	8	8	8
9	9	9	9	9	9
10	10	10	10	10	10

Update:

UPDATE Has_meal
SET LunchID = 5
WHERE MealPlanId = 1;
SELECT * FROM Has_meal;



	MealPlanId	BreakfastID	LunchID	DinnerID	SnackID
1	1	1	5	1	1
2	2	2	2	2	2
3	3	3	3	3	3
4	4	4	4	4	4
5	5	5	5	5	5
6	6	6	6	6	6
7	7	7	7	7	7
8	8	8	8	8	8
9	9	9	9	9	9
10	10	10	10	10	10

4. References:

[**1**] HealthifyMe, “HealthifyMe Model- How it Works and Makes Money?,” 2024 [Online]. Available: //oyelabs.com/healthifyme-business-model/age (oyelabs.com) [Accessed: Oct 2024]

[**2**] MealBoard, “MealBoard app review: plan in advance-2020,” 2020 [Online]. Available: MealBoard - Meal and Grocery Planner app review - appPicker [Accessed: Oct 2024]