CIST-3222 Database Systems

Group 1 Final Report

Stockton Dining DB

Cachary Tolentino, Tyler Mong, Melvin Vazquez, Dylan Calixtro

Project Overview

1. Executive Summary

The business, Stockton Dining, was created to foster more community and shared experiences among students, staff, and guests in Stockton while providing a wide variety of foods and cuisines to try. The primary goal of Stockton Dining is providing the entire student body an opportunity to have a meticulously crafted meal that is widely available throughout the day, week, and month of each school year. But to do so, the business is in dire need of tracking customer needs and interests. Many times, the business is found lacking in traffic and general interest in the food being served.

To change this, a new database to track customer interest and general food inventory was needed to be tracked. Thus, the new database will track both customer behavior through transactions and inventory stock as well as each meal provided. By tracking this information, it will enable the business to examine customer interest, trends, and dislikes regarding which meals they prefer to purchase. More so, by also tracking inventory, it ensures the business does not keep an abundance of stock but will only keep a stable amount of inventory needed.

With the newly established database, the business started to see an increase in customer traffic and a steady flow of customers while maintaining a manageable inventory. It has garnered more attention from guests as customers, and not just students or staff. The business foresees a future that may expand the establishment beyond its current estate and may provide even more services and cuisine for all to try.

2. Team Contribution

Cachary Tolentino

- Set up Group GDrive with all Report templates
- Created the Tables and Inserted the Data into SQL Oracle (including screenshots and debugging)
- Wrote most of the business description
- Wrote the executive summary, cardinalities/relationships, database design model description, and functional dependencies

Tyler Mong

- Helped with the initial draft of the Entity Relationship Diagram
- Helped define and finalize table attributes and data types
- Verified functional dependencies were correctly mapped to the database model
- Double-checked all documentation for formatting, clarity, etc.
- Contributed to the presentation by writing speaker notes
- Reviewed and ensured consistency between ERD, schema, and report

Melvin Vazquez:

- Filled out all Team Meeting Minutes
- Created the SQL code that made the tables
- Created insertion code for 3 of the tables
- Verified and checked the insertion data to make sure it meets requirements
- Created the Entity Relationship Diagram and Database design model
- Filled the database design model with PK and FK
- Helped Cachary input the SQL code into Oracle and fix any errors
- Got the required screenshots from SQL development
- Organized the Screenshots inside the final report
- Did my part of the presentation slides

Dylan Calixtro

- Compared data model to align with Pearson slides logic
- Found some missing entities in the data model and put the screenshot in part 1 report
- Ensured the creation code matched the design model diagram
- Pointed out consitency and redundancy improvements in the creation code
- Final Report grammar fixed and added a functional dependency
- Google Slides design

3. Table Of Contents

Project Overview..... 2
Business Description..... 5
1. Name of Business:..... 5
2. Purpose of Business:..... 5
3. Summary of Business Activities:..... 5
4. Problems, Opportunities, and Objectives:..... 5
5. Business Case:..... 6
6. Information and Data Requirements:..... 6
7. List of Entities:..... 7
Conceptual Data Model..... 8
1. Entity Relationship Diagram..... 8
2. Cardinalities and Relationships..... 8
Database Design Model..... 11
1. Database Design Diagram..... 11
2. Functional Dependencies..... 12
Database Implementation..... 13
1. SQL Code for Creating all Tables..... 13
2. SQL Code for Inserting Data..... 16

Business Description

1. Name of Business:

Stockton Dining

2. Purpose of Business:

The purpose of this business is to provide Stockton students, employees, and more with delicious and affordable meals on a day-to-day basis. The business operates 7 days a week, with rotating menus.

3. Summary of Business Activities:

Stockton Dining's goal is to provide for everyone within the Stockton institution area. It operates 7 days a week, with each day having a variety of food and items to choose from. Stockton Dining ensures not only a fair and balanced meal but also the freedom for anyone to choose what they eat. To continue doing so, Stockton Dining ensures that each meal is worth one's money; therefore, a very affordable price is used to make sure that everyone is able to afford a meal.

4. Problems, Opportunities, and Objectives:

Problems: Currently, Stockton Dining does not have an interactive and fluent database. The creation of a centralized database would lead to further developments in sales and organization within each location. Additionally, Stockton Dining has seen far less traffic in customers and general food consumption within the institution. Furthermore, keeping track of food stock has been a hassle due to old-fashioned usage, which leads to mismanagement and a lack of ingredients for certain dishes. Due to the lack of management, many ingredients are misused and sometimes lead to allergic reactions in students due to mishandling of ingredients. Lastly, students are confused about what is available due to the lack of a cohesive menu.

Opportunities: With a newly established database, management will be a much smoother operation in which tracking stock will be up to date at all times. This would lead to better management, as information will be available at a glance. Hence, improvement in safety precautions. Furthermore, the opportunity to research and analyze buyer patterns can help in providing far more engaging menus. This would lead to higher customer retention and new customers. Lastly, with an established database, a concrete and concise menu would be available for students to view.

Objectives: The main objective of this business is to maximize Stockton dining efficiency in keeping track of their stock when it comes to food ingredients. Doing that will also help make their business flow more efficiently with a well-organized menu that points out the ingredients used for those food items and nutritional information on what the food contains. From that, we can also track popular food items and the popular times that people come, so we can find relationships between the food and consumer demands.

5. Business Case:

This database will help ensure Stockton Dining is more organized and efficient in keeping track of its stock and information about consumers buying from their dining. Consumers will gain insight into what exactly they are eating, along with potential allergens. With the use of an SQL database, we can keep track of important information to ensure that Stockton Dining does not run out of inventory and meets consumer needs when it comes to knowing what the menu will look like daily.

6. Information and Data Requirements:

The data that will be required are the ‘Menu Item Name’, ‘Menu ID’, ‘Menu Category’, ‘Ingredient Name’, ‘Ingredient Quantity’, ‘Menu Item Allergens’, and ‘Transaction Time’.

The information that will be generated will include ‘Most Popular’, ‘Ingredient Usage’, ‘Gluten Trends’, and ‘Busy Hours’.

The data obtained from Menu Item Name, Menu Category, and Ingredient Quantity would generate information for the Most Popular. The data obtained from the Menu Item Name and Quantity would generate information for Ingredient Usage and Gluten Trends. The data obtained from Transaction Time would generate information for Busy Hours.

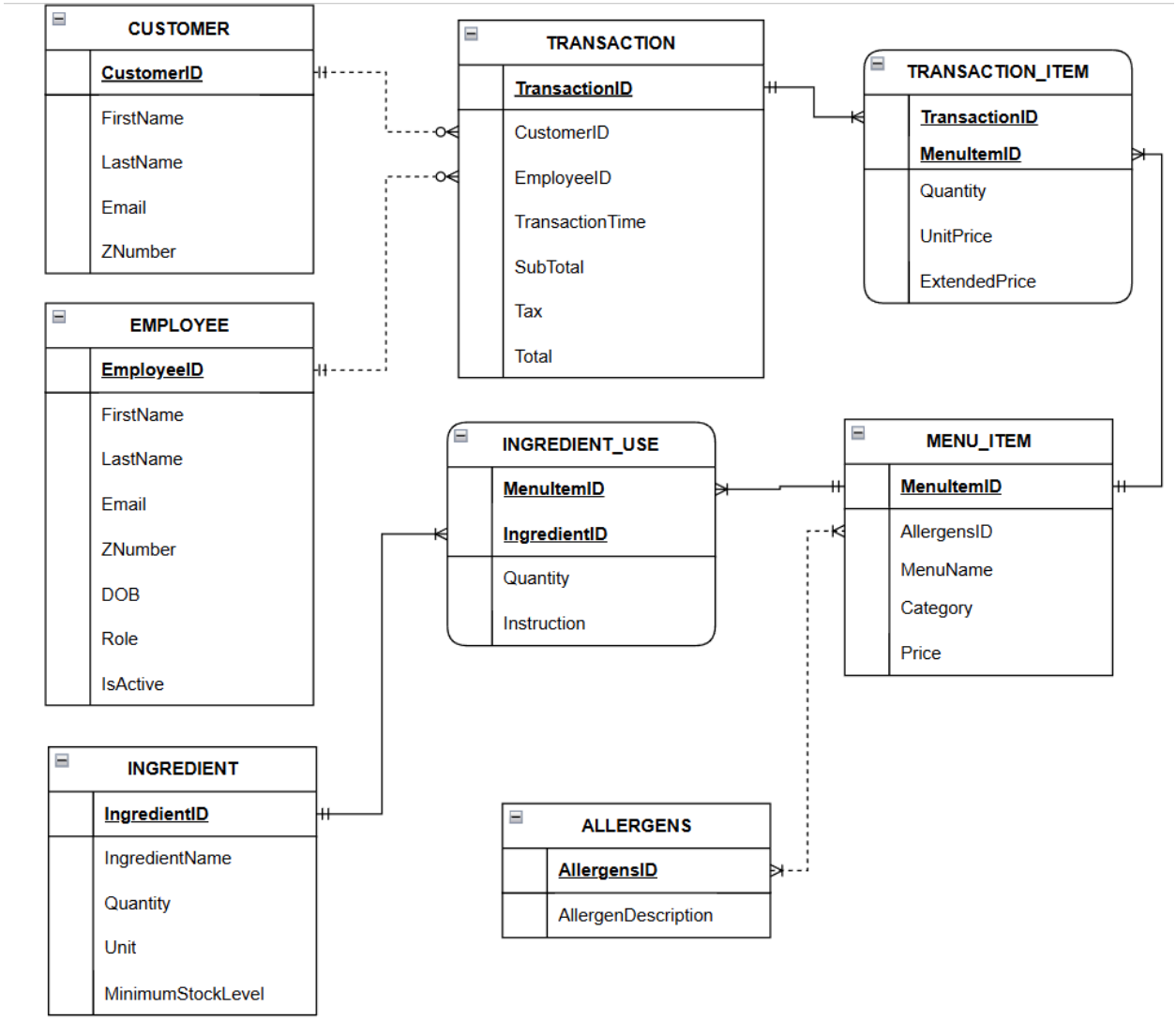
Important data that needs to be tracked are buyers (students and/or others), transactions, ingredient inventory, etc. From this data, we can learn which meals are most popular (to optimize menus), track ingredient usage (to avoid shortages or waste), identify customer trends (like demand for gluten-free options), predict busy times (to adjust staffing), and ensure food safety by flagging allergens automatically.

7. List of Entities:

EMPLOYEE(): (EmployeeID, FirstName, LastName, Email, ZNumber, DOB, Role, IsActive)
MENU_ITEM (MenuItemID, MenuName, Category, Price)
INGREDIENT(IngredientID, IngredientName, Quantity, Unit, MinimumStockLevel)
INGREDIENT_USE (MenuItemID, IngredientID, Quantity, Instruction)
ALLERGENS(): (AllergensID, AllergenDescription)
CUSTOMER(CustomerID, FirstName, LastName, Email, ZNumber)
TRANSACTION_ITEM(TransactionID, MenuItemID, Quantity, UnitPrice, ExtendedPrice)
TRANSACTION(TransactionID, *CustomerID*, *EmployeeID*, TransactionTime, SubTotal, Tax, Total)

Conceptual Data Model

1. Entity Relationship Diagram



2. Cardinalities and Relationships

The conceptual data model contains eight entities with seven relationships between entities.

In the relationship between the EMPLOYEE entity and the TRANSACTION entity, there exists a **minimum cardinality of mandatory to optional** and a **maximum**

cardinality of one to many. This defines the relationship in which an employee can be responsible for many transactions, while each transaction can only belong to one employee. Any employee can have none or more transactions for which they are responsible, while each transaction must have an employee.

In the relationship between the MENU_ITEM entity and INGREDIENT_USE entity, there exists a **minimum cardinality of mandatory to mandatory**; in addition, it also has a **maximum cardinality of one to many**. This defines the relationship in which a menu item can have many usages for a certain ingredient. And that each usage only belongs to a single menu item. This also means that every menu item must have at least one ingredient. Any ingredient used must be related to a menu item.

In the relationship between the INGREDIENT entity and INGREDIENT_USE entity, there exists a **minimum cardinality of mandatory to optional**; in addition, it also has a **maximum cardinality of one to many**. This defines the relationship in which an ingredient has multiple ingredient usages. And that each ingredient usage belongs to a single ingredient. This also means that each ingredient can have zero or more usages. While each ingredient usage must be related to one ingredient.

In the relationship between the MENU_ITEM entity and the ALLERGENS entity, there exists a **minimum cardinality of optional to optional**; in addition, it also has a **maximum cardinality of many to many**. This defines the relationship in which a menu item can have multiple allergens, while each allergen can also belong to many menu items. This also means that each menu item can have zero or more allergens, while each allergen can be related to zero or more menu items.

In the relationship between the CUSTOMER entity and the TRANSACTION entity, there exists a **minimum cardinality of mandatory to optional**; in addition, it also has a **maximum cardinality of one to many**. This defines the relationship in which a customer can have multiple transactions, but each transaction only belongs to a single customer. This also means that a customer can have zero or more transactions, while each transaction has to belong to a customer.

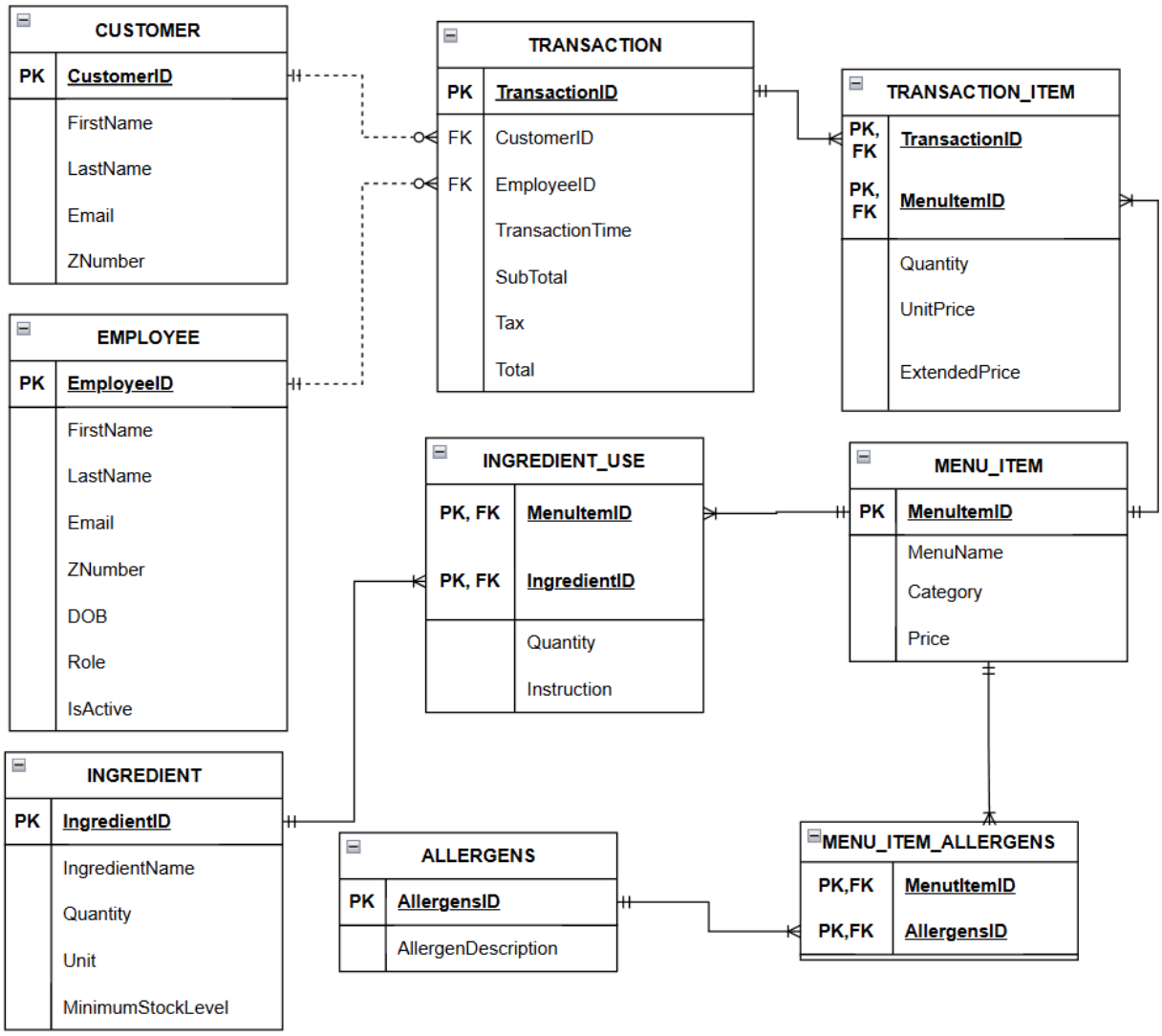
In the relationship between the TRANSACTION entity and the TRANSACTION_ITEM entity, there exists a **minimum cardinality of mandatory to**

mandatory; in addition, it also has a **maximum cardinality of one to many**. This defines the relationship in which a transaction can contain multiple items, and each item bought belongs to a single transaction. This also means that each transaction must have an item and that each item must belong to a transaction.

In the relationship between the MENU_ITEM entity and the TRANSACTION_ITEM entity, there exists a **minimum cardinality of mandatory to optional**; in addition, it also has a **maximum cardinality of one to many**. This defines the relationship in which a menu item can belong to many transactions, and each item bought belongs to a single transaction. This also means that a menu item can be related to a transaction item, while each transaction item has to be related to a menu item.

Database Design Model

1. Database Design Diagram



The database diagram contains nine tables with eight relationships. Two of the relationships are non-identifying, while the other six are identifying relationships. Each entity from the ER model will become a table, and each one will also have its primary keys as identifiers.

For the EMPLOYEE to TRANSACTION table, the parent table will be the EMPLOYEE, and EmployeeID will be its primary key, which is a surrogate key. The TRANSACTION table will be the child table and have TransactionID as its primary key, but will also have EmployeeID as a foreign key.

For the CUSTOMER table to the TRANSACTION table, the parent table will be the CUSTOMER table. The CUSTOMER table will have CustomerID as its primary key, which is a surrogate key. While the TRANSACTION table will still have the TransactionID as its primary key, but will also have CustomerID as a foreign key.

From the TRANSACTION table to the TRANSACTION_ITEM table, the TRANSACTION table will be the parent. It will have TransactionID as its primary key, which is a surrogate key. While TRANSACTION_ITEM will be the child table, which uses TransactionID as a primary key, a foreign key, and part of the composite key.

From the MENU_ITEM table to the TRANSACTION_ITEM table, the MENU_ITEM table will be the parent table with MenuItemID as its primary key, which is a surrogate key. While the TRANSACTION_ITEM table will be the child that uses MenuItemID as a primary key, foreign key, and part of a composite key.

From the MENU_ITEM table to the MENU_ITEM_ALLERGENS table, the MENU_ITEM table will be the parent table with MenuItemID as its primary key, which is a surrogate key. MENU_ITEM_ALERGENS will be the child table with MenuItemID as a primary key, foreign key, and part of a composite key.

For the ALLERGENS table and MENU_ITEM_ALLERGENS table, the ALLERGENS table will be the parent table with AllergensID as a primary key, which is a surrogate key. While the MENU_ITEMALLERGENs table will be the child table with AllergensID as a primary key, foreign key, and part of a composite key.

From the MENU_ITEM table to the INGREDIENT_USE table, the MENU_ITEM table will be the parent table with MenuItemID as its primary key, which is a surrogate key. While the INGREDIENT_USE table will be the child table with MenuItemID as its primary key, foreign key, and part of a composite key.

Finally, from the INGREDIENT table to the INGREDIENT_USE table, the INGREDIENT table will be the parent table with IngredientID as its primary key, which is a surrogate key. While the INGREDIENT_USE table will be the child table with IngredientID as a primary key, a foreign key, and part of a composite key.

2. Functional Dependencies

Functional Dependency of CUSTOMER Table

CustomerID → (FirstName, LastName, Email, ZNumber)

Functional Dependency of EMPLOYEE Table

EmployeeID → (FirstName, LastName, Email, ZNumber, DOB, Role IsActive)

Functional Dependency of TRANSACTION Table

TransactionID → (CustomerID, EmployeeID, TransactionTime, Subtotal, Tax, Total)

Subtotal, Tax → (Total)

Functional Dependency of TRANSACTION_ITEM Table

(TransactionID, MenuItemID) → (Quantity, UnitPrice, ExtendedPrice)

Functional Dependency of MENU_ITEM Table

MenuItemID → (MenuName, Category, Price)

Functional Dependency of INGREDIENT Table

IngredientID → (IngredientName, Quantity, Unit, MinimumStockLevel)

Functional Dependency of INGREDIENT_USE Table

(MenuItemID, IngredientID) → (Quantity, Instruction)

Functional Dependency of ALLERGENS Table

AllergensID → AllergensDescription

Functional Dependency of MENU_ITEM_ALLERGENS Table

(MenuItemID, AllergensID) → ()

Database Implementation

1. SQL Code for Creating all Tables

```
CREATE TABLE CUSTOMER (
CustomerID              Int             NOT NULL,
FirstName               Char(25)        NULL,
LastName                Char(25)        NULL,
Email                   VarChar(100)    NULL,
ZNumber                 Char(9)         NULL,
CONSTRAINT      CustomerPK          Primary Key(CustomerID),
CONSTRAINT EmailAK1             Unique(Email),
CONSTRAINT ZNumberAK1               Unique(Znumber),
CONSTRAINT ZNumberCheck     CHECK(REGEXP_LIKE(ZNumber, '^Z\d{8}$'))
);

CREATE TABLE EMPLOYEE (
EmployeeID          Int             NOT NULL,
FirstName           CHAR(25)        NOT NULL,
LastName            CHAR(25)        NOT NULL,
Email               VARCHAR(100)    NOT NULL,
ZNumber             CHAR(9)         NULL,
DOB                 DATE            NOT NULL,
Role                CHAR(25)        NOT NULL,
IsActive            NUMBER(1)           DEFAULT 1 NOT NULL,
CONSTRAINT EmployeePK           PRIMARY KEY(EmployeeID),
CONSTRAINT EmailAK2             UNIQUE(Email),
CONSTRAINT ZNumberAK2               UNIQUE(ZNumber)
);

CREATE TABLE INGREDIENT(
IngredientID            Int             NOT NULL,
IngredientName          VARCHAR2(40)    NOT NULL,
Quantity                NUMBER          NOT NULL,
Unit                    VARCHAR2(30)    NOT NULL,
MinimumStockLevel           NUMBER          NOT NULL,
CONSTRAINT IngredientPK         PRIMARY KEY(IngredientID),
```

```
CONSTRAINT QuantityCheck        CHECK(Quantity >= 0),
CONSTRAINT Stock                CHECK(MinimumStockLevel >= 0),
CONSTRAINT IngredientNameAK1                UNIQUE(IngredientName)
);

CREATE TABLE TRANSACTION(
TransactionID                   Int                     NOT NULL,
CustomerID                      Int                     NULL,
EmployeeID                      Int                     NULL,
TransactionTime                 TIMESTAMP           NULL,
SubTotal                        NUMBER(6,2)                 NOT NULL,
Tax                             NUMBER(6,2)                 NOT NULL,
Total                           NUMBER(6,2)                 NOT NULL,
CONSTRAINT TransactionPK            PRIMARY KEY(TransactionID),
CONSTRAINT CustomerFK1                  FOREIGN KEY(CustomerID)
REFERENCES              CUSTOMER(CustomerID),
CONSTRAINT EmployeeFK1                  FOREIGN KEY(EmployeeID)
REFERENCES              EMPLOYEE(EmployeeID),
CONSTRAINT SubtotalCheck        CHECK(SubTotal > 0),
CONSTRAINT TaxCheck             CHECK(Tax > 0),
CONSTRAINT TotalCheck           CHECK(Total > 0)
);

CREATE TABLE MENU_ITEM(
MenuItemID                      Int                 NOT NULL,
MenuName                        VarChar(25)         NOT NULL,
Category                        Char(25)            NULL,
Price                           Number(4,2)         NOT NULL,
CONSTRAINT MenuPK                   Primary Key(MenuItemID),
CONSTRAINT MenuNameAK               Unique(MenuName),
CONSTRAINT PriceCheck           CHECK( Price >= 0)
);

CREATE TABLE TRANSACTION_ITEM(
TransactionID                   Int                 NOT NULL,
MenuItemID                      Int                 NOT NULL,
Quantity                        Number(2)           NOT NULL,
UnitPrice                       Number(5,2)         NOT NULL,
ExtendedPrice                   Number(5,2)         NOT NULL,
```

```
CONSTRAINT TransactionItemPK        Primary Key(TransactionID, MenuItemID),
CONSTRAINT TransactionFK2           Foreign Key(TransactionID) REFERENCES
TRANSACTION(TransactionID),
CONSTRAINT MenuItemFK1                  Foreign Key(MenuItemID) REFERENCES
        MENU_ITEM(MenuItemID),
CONSTRAINT QuantityCheck1           CHECK((Quantity >=0) AND (Quantity <= 99))
);

CREATE TABLE INGREDIENT_USE(
MenuItemID                  Int             NOT NULL,
IngredientID            Int                 NOT NULL,
Quantity                    Int             NOT NULL,
Instruction                 VarChar(100)    NULL,
CONSTRAINT IngredientUsePK          Primary Key(MenuItemID, IngredientID),
CONSTRAINT MenuItemFK2                  Foreign Key(MenuItemID) REFERENCES
MENU_ITEM(MenuItemID),
CONSTRAINT IngredientFK1            Foreign Key(IngredientID) REFERENCES
INGREDIENT(IngredientID),
CONSTRAINT QuantityCheck2           CHECK(Quantity > 0)
);

CREATE TABLE ALLERGENS(
AllergensID                 Int             NOT NULL,
AllergenDescription         VarChar(100)    NOT NULL,
CONSTRAINT AllergensPK              Primary Key(AllergensID),
CONSTRAINT AllergenDescriptionAK    UNIQUE(AllergenDescription)
);

CREATE TABLE MENU_ITEM_ALLERGENS(
MenuItemID                  Int             NOT NULL,
AllergensID                 Int             NOT NULL,
CONSTRAINT MenuItemAllergenPK       Primary Key(MenuItemID, AllergensID),
CONSTRAINT MenuItemFK3              Foreign Key(MenuItemID) REFERENCES
        MENU_ITEM(MenuItemID),
CONSTRAINT AllergensFK1             Foreign Key(AllergensID) REFERENCES
ALLERGENS(AllergensID)
);
```

2. SQL Code for Inserting Data

```
INSERT INTO CUSTOMER VALUES(
1, 'John', 'Doe', 'doeJ@go.stockton.edu', 'Z01234567'
);
INSERT INTO CUSTOMER VALUES(
2, 'Billy', NULL, 'joelB@go.stockton.edu', 'Z01234568'
);
INSERT INTO CUSTOMER VALUES(
3, NULL, 'Kane', 'kaneH@go.stockton.edu', 'Z01234569'
);
INSERT INTO CUSTOMER VALUES(
4, NULL, NULL, NULL, NULL
);
INSERT INTO CUSTOMER VALUES(
5, 'Lilly', 'Shurtz', NULL,  'Z01234572'
);
INSERT INTO CUSTOMER VALUES(
6, 'Tiffany', 'Hughes', 'hughesT@go.stockton.edu', NULL
);
INSERT INTO CUSTOMER VALUES(
7, 'Mikael', 'Rigby', NULL, NULL
);
INSERT INTO CUSTOMER VALUES(
8, NULL, NULL, NULL, 'Z01234575'
);
INSERT INTO CUSTOMER VALUES(
9, NULL, NULL,  'bryanZ@go.stockton.edu', NULL
);
INSERT INTO CUSTOMER VALUES(
10, 'Billy', 'Joel', 'billyJ@go.stockton.edu', 'Z01234577'
);
CREATE sequence seqCID Increment by 1 start with 11;
```

```
INSERT INTO EMPLOYEE VALUES(
1, 'Tyler', 'Vans', 'vansT@go.stockton.edu', NULL, TO_DATE('02/12/1990',
'MM/DD/YYYY'), 'Floor Manager', 1
);
INSERT INTO EMPLOYEE VALUES(
2, 'Deep', 'Patel', 'patelD@go.stockton.edu', NULL, TO_DATE('05/22/2000',
'MM/DD/YYYY'), 'Chef', 1
);
INSERT INTO EMPLOYEE VALUES(
3, 'Jack', 'Brown', 'brownJ@go.stockton.edu', NULL, TO_DATE('01/08/1995',
'MM/DD/YYYY'), 'Janitor', 1
);
INSERT INTO EMPLOYEE VALUES(
4, 'Mike', 'Power', 'powerM@go.stockton.edu', 'Z09876521', TO_DATE('08/20/1999',
'MM/DD/YYYY'), 'Cashier', 0
);
INSERT INTO EMPLOYEE VALUES(
5, 'John', 'Hancock', 'hancockJ@go.stockton.edu', 'Z09876551', TO_DATE('03/25/1988',
'MM/DD/YYYY'), 'Chef Manager', 1
);
INSERT INTO EMPLOYEE VALUES(
6, 'Ben', 'Simmons', 'simmonsB@go.stockton.edu', 'Z09886551', TO_DATE('07/01/1998',
'MM/DD/YYYY'), 'Director', 1
);
INSERT INTO EMPLOYEE VALUES(
7, 'Zoe', 'Powell', 'powellZ@go.stockton.edu', NULL, TO_DATE('11/22/2001',
'MM/DD/YYYY'), 'Line Cook', 0
);
INSERT INTO EMPLOYEE VALUES(
8, 'Mya', 'Cruz', 'cruzM@go.stockton.edu', NULL, TO_DATE('04/05/1973', 'MM/DD/YYYY'),
'Apprentice', 0
);
INSERT INTO EMPLOYEE VALUES(
9, 'John', 'Doe', 'doeJ@go.stockton.edu', NULL, TO_DATE('04/05/1973', 'MM/DD/YYYY'),
'Apprentice', 1
);
INSERT INTO EMPLOYEE VALUES(
10, 'John', 'Doe', 'doeJ@go.stockton.edu', 'Z09986551',  TO_DATE('04/05/1973',
'MM/DD/YYYY'), 'Apprentice', 1
```

```
);
CREATE sequence seqEID Increment by 1 start with 11;
```

```
INSERT INTO INGREDIENT VALUES(
1, 'Tomato', 3, 'kg', 3
);
INSERT INTO INGREDIENT VALUES(
2, 'Beef', 15, 'kg', 5
);
INSERT INTO INGREDIENT VALUES(
3, 'Eggs', 48, 'pieces', 40
);
INSERT INTO INGREDIENT VALUES(
4, 'Salt', 5, 'kg', 2
);
INSERT INTO INGREDIENT VALUES(
5, 'Onion', 5, 'kg', 3
);
INSERT INTO INGREDIENT VALUES(
6, 'Tuna', 30, 'cans', 25
);
INSERT INTO INGREDIENT VALUES(
7, 'Potatoes', 30, 'kg', 20
);
INSERT INTO INGREDIENT VALUES(
8, 'Butter', 10, 'kg',  3
);
INSERT INTO INGREDIENT VALUES(
9, 'Peppers', 30, 'pieces', 10
);
INSERT INTO INGREDIENT VALUES(
10, 'Rice', 10, 'kg', 5
);
CREATE sequence seqIID Increment by 1 start with 11;
```

```
INSERT INTO TRANSACTION VALUES(
1, 1, 2, NULL, 10.00, 0.80, 10.80
);
INSERT INTO TRANSACTION VALUES(
2, 2, 3, TO_TIMESTAMP('2025-01-01 08:00:00', 'YYYY-MM-DD HH24:MI:SS'), 22.50, 1.80,
24.30
);
INSERT INTO TRANSACTION VALUES(
3, 1, 4, TO_TIMESTAMP('2025-02-14 12:30:45', 'YYYY-MM-DD HH24:MI:SS'), 15.25, 1.22,
16.47
);
INSERT INTO TRANSACTION VALUES(
4, 7, 9, TO_TIMESTAMP('2025-03-01 00:00:00', 'YYYY-MM-DD HH24:MI:SS'), 5.00, 0.40,
5.40
);
INSERT INTO TRANSACTION VALUES(
5, 2, 3, TO_TIMESTAMP('2025-04-23 15:45:10', 'YYYY-MM-DD HH24:MI:SS'), 20.10, 1.61,
21.71
);
INSERT INTO TRANSACTION VALUES(
6, 8, 4, NULL, 12.99, 1.04, 14.03
);
INSERT INTO TRANSACTION VALUES(
7, 10, 1, TO_TIMESTAMP('2025-05-05 22:15:55', 'YYYY-MM-DD HH24:MI:SS'), 6.00, 0.48,
6.48
);
INSERT INTO TRANSACTION VALUES(
8, 1, 1, TO_TIMESTAMP('2025-06-30 18:00:00', 'YYYY-MM-DD HH24:MI:SS'), 18.75, 1.50,
20.25
);
INSERT INTO TRANSACTION VALUES(
9, 6, 4, TO_TIMESTAMP('2025-07-04 23:59:59', 'YYYY-MM-DD HH24:MI:SS'), 9.90, 0.79,
10.69
);
INSERT INTO TRANSACTION VALUES(
10, 5, 7, NULL, 24.00, 1.92, 25.92
);
CREATE sequence seqTID Increment by 1 start with 11;
```

```
INSERT INTO MENU_ITEM VALUES(
1, 'Roast beef', 'Main Course',  14.99
);
INSERT INTO MENU_ITEM VALUES(
2, 'Fried rice', 'Main Course',  9.50
);
INSERT INTO MENU_ITEM VALUES(
3, 'Caesar salad', 'Salad',  6.25
);
INSERT INTO MENU_ITEM VALUES(
4, 'Baked salmon', 'Main Course',  13.75
);
INSERT INTO MENU_ITEM VALUES(
5, 'Chicken wings', 'Appetizer',  7.95
);
INSERT INTO MENU_ITEM VALUES(
6, 'Mashed potatoes', 'Side Dish',  4.50
);
INSERT INTO MENU_ITEM VALUES(
7, 'Steamed vegetables', NULL,  3.99
);
INSERT INTO MENU_ITEM VALUES(
8, 'Spaghetti with marinara', 'Main Course',  8.95
);
INSERT INTO MENU_ITEM VALUES(
9, 'Garlic bread', 'Side Dish',  2.75
);
INSERT INTO MENU_ITEM VALUES(
10, 'Chocolate cake', 'Dessert',  5.80
);
CREATE sequence seqMID Increment by 1 start with 11;
```

```
INSERT INTO TRANSACTION_ITEM VALUES(
1, 2, 2, 9.50, 19.00
);
INSERT INTO TRANSACTION_ITEM VALUES(
2, 5, 4, 7.95, 31.80
);
INSERT INTO TRANSACTION_ITEM VALUES(
3, 1, 2, 13.99, 27.98
);
INSERT INTO TRANSACTION_ITEM VALUES(
4, 2, 1, 6.25, 6.25
);
INSERT INTO TRANSACTION_ITEM VALUES(
5, 6, 4, 2.50, 10.00
);
INSERT INTO TRANSACTION_ITEM VALUES(
6, 9, 5, 2.75, 13.75
);
INSERT INTO TRANSACTION_ITEM VALUES(
7, 10, 1, 5.80, 5.80
);
INSERT INTO TRANSACTION_ITEM VALUES(
8, 2, 3, 7.50, 22.50
);
INSERT INTO TRANSACTION_ITEM VALUES(
9, 8, 2, 8.95, 17.90
);
INSERT INTO TRANSACTION_ITEM VALUES(
10, 10, 2, 5.80, 11.60
);
CREATE sequence seqTIID Increment by 1 start with 11;
```

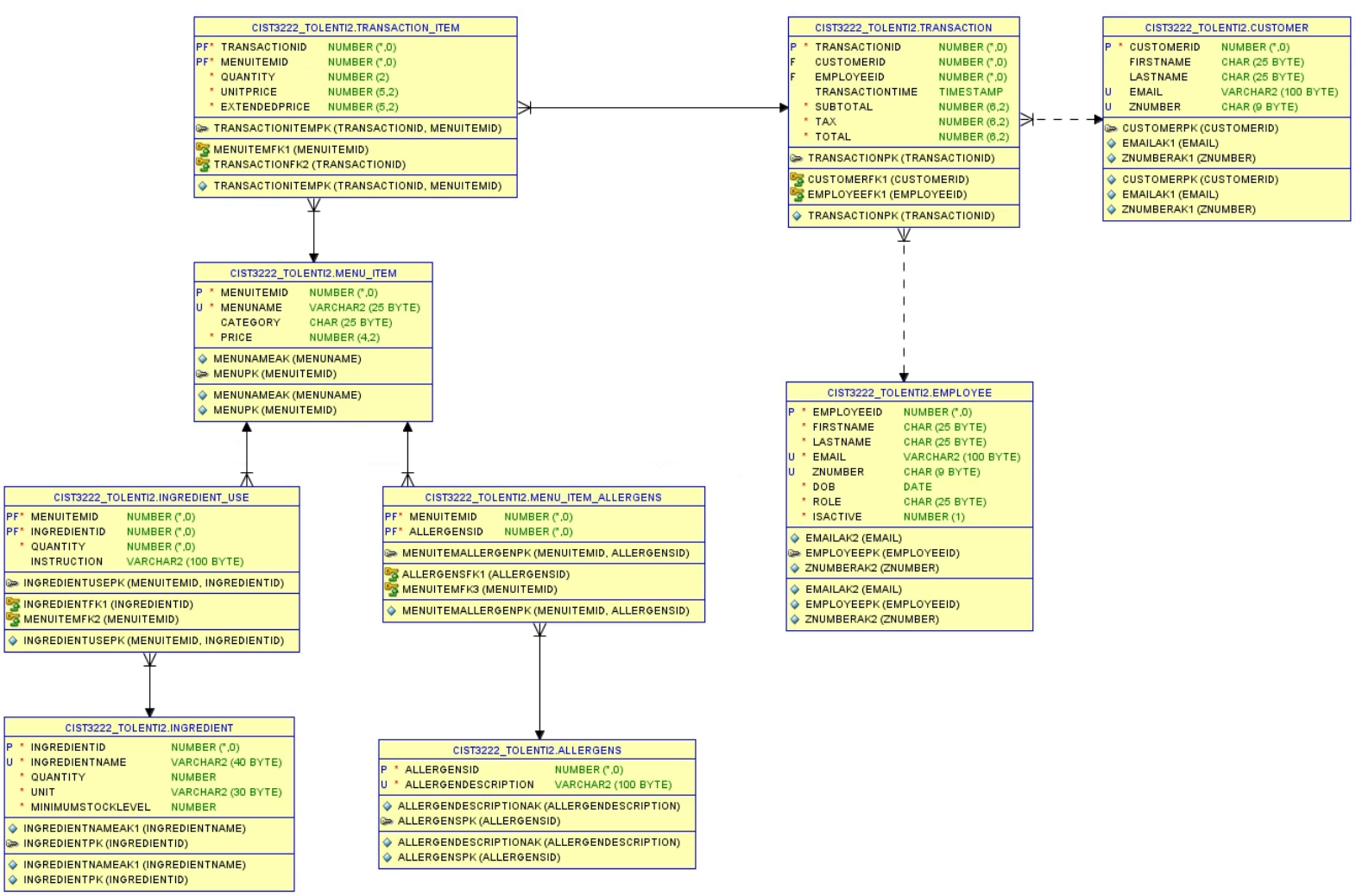
```
INSERT INTO INGREDIENT_USE VALUES(
1, 1, 2, 'Blend two tomatoes into a puree'
);
INSERT INTO INGREDIENT_USE VALUES(
2, 2, 3, 'slice into cube chunks'
);
INSERT INTO INGREDIENT_USE VALUES(
3, 3, 6, 'beat six eggs into a bowl'
);
INSERT INTO INGREDIENT_USE VALUES(
4, 4, 1, 'sprinkle any dish with a pinch of salt'
);
INSERT INTO INGREDIENT_USE VALUES(
5, 5, 2, 'chop and mince until fine texture'
);
INSERT INTO INGREDIENT_USE VALUES(
6, 6, 10, 'slice into three pieces: head, body, and tail'
);
INSERT INTO INGREDIENT_USE VALUES(
7, 7, 5, 'slice into four equal chunks'
);
INSERT INTO INGREDIENT_USE VALUES(
8, 8, 3, 'melt into a pan until slightly brown'
);
INSERT INTO INGREDIENT_USE VALUES(
9, 9, 5, NULL
);
INSERT INTO INGREDIENT_USE VALUES(
10, 10, 2, 'rinse in a bowl, cook for 10 minutes'
);
INSERT INTO INGREDIENT_USE VALUES(
2, 5, 2, 'Cut onions to cook the fried rice'
);
CREATE sequence seqIUID Increment by 1 start with 12;
```

```
INSERT INTO ALLERGENS VALUES(
1, 'Peanuts'
);
INSERT INTO ALLERGENS VALUES(
2, 'Tree Nuts'
);
INSERT INTO ALLERGENS VALUES(
3, 'Milk'
);
INSERT INTO ALLERGENS VALUES(
4, 'Eggs'
);
INSERT INTO ALLERGENS VALUES(
5, 'Fish'
);
INSERT INTO ALLERGENS VALUES(
6, 'Shell Fish'
);
INSERT INTO ALLERGENS VALUES(
7, 'Soy'
);
INSERT INTO ALLERGENS VALUES(
8, 'Wheat'
);
INSERT INTO ALLERGENS VALUES(
9, 'Sesame'
);
INSERT INTO ALLERGENS VALUES(
10, 'Gluten'
);
CREATE sequence seqAID Increment by 1 start with 11;
```

```
INSERT INTO MENU_ITEM_ALLERGENS VALUES(
1, 4
);
INSERT INTO MENU_ITEM_ALLERGENS VALUES(
2, 10
);
INSERT INTO MENU_ITEM_ALLERGENS VALUES(
3, 4
);
INSERT INTO MENU_ITEM_ALLERGENS VALUES(
4, 5
);
INSERT INTO MENU_ITEM_ALLERGENS VALUES(
5, 7
);
INSERT INTO MENU_ITEM_ALLERGENS VALUES(
6, 3
);
INSERT INTO MENU_ITEM_ALLERGENS VALUES(
7, 9
);
INSERT INTO MENU_ITEM_ALLERGENS VALUES(
8, 8
);
INSERT INTO MENU_ITEM_ALLERGENS VALUES(
9, 8
);
INSERT INTO MENU_ITEM_ALLERGENS VALUES(
10, 4
);

CREATE sequence seqMIAID Increment by 1 start with 11;
```

3. ER Diagram in SQL Developer



4. Tables with Data

Customer Table with Data:

	CUSTOMERID	FIRSTNAME	LASTNAME	EMAIL	ZNUMBER
1	1	John	Doe	doeJ@go.stockton.edu	Z01234567
2	2	Billy	(null)	joelB@go.stockton.edu	Z01234568
3	3	(null)	Kane	kaneH@go.stockton.edu	Z01234569
4	4	(null)	(null)	(null)	(null)
5	5	Lilly	Shurtz	(null)	Z01234572
6	6	Tiffany	Hughes	hughesT@go.stockton.edu	(null)
7	7	Mikael	Rigby	(null)	(null)
8	8	(null)	(null)	(null)	Z01234575
9	9	(null)	(null)	bryanZ@go.stockton.edu	(null)
10	10	Billy	Joel	billyJ@go.stockton.edu	Z01234577

Employee Table with Data:

	EMPLOYEEID	FIRSTNAME	LASTNAME	EMAIL	ZNUMBER	DOB	ROLE	ISACTIVE
1	1	Tyler	Vans	vansT@go.stockton.edu	(null)	12-FEB-90	Floor Manager	1
2	2	Deep	Patel	patelD@go.stockton.edu	(null)	22-MAY-00	Chef	1
3	3	Jack	Brown	brownJ@go.stockton.edu	(null)	08-JAN-95	Janitor	1
4	4	Mike	Power	powerM@go.stockton.edu	Z09876521	20-AUG-99	Cashier	0
5	5	John	Hancock	hancockJ@go.stockton.edu	Z09876551	25-MAR-88	Chef Manager	1
6	6	Ben	Simmons	simmonsB@go.stockton.edu	Z09886551	01-JUL-98	Director	1
7	7	Zoe	Powell	powellZ@go.stockton.edu	(null)	22-NOV-01	Line Cook	0
8	8	Mya	Cruz	cruzM@go.stockton.edu	(null)	05-APR-73	Apprentice	0
9	9	John	Doe	doelJ@go.stockton.edu	(null)	05-APR-73	Apprentice	1
10	10	John	Doe	doeJ@go.stockton.edu	Z09986551	05-APR-73	Apprentice	1

Ingredient Table with Data:

	INGREDIENTID	INGREDIENTNAME	QUANTITY	UNIT	MINIMUMSTOCKLEVEL
1	1	Tomato	3	kg	3
2	2	Beef	15	kg	5
3	3	Eggs	48	pieces	40
4	4	Salt	5	kg	2
5	5	Onion	5	kg	3
6	6	Tuna	30	cans	25
7	7	Potatoes	30	kg	20
8	8	Butter	10	kg	3
9	9	Peppers	30	pieces	10
10	10	Rice	10	kg	5

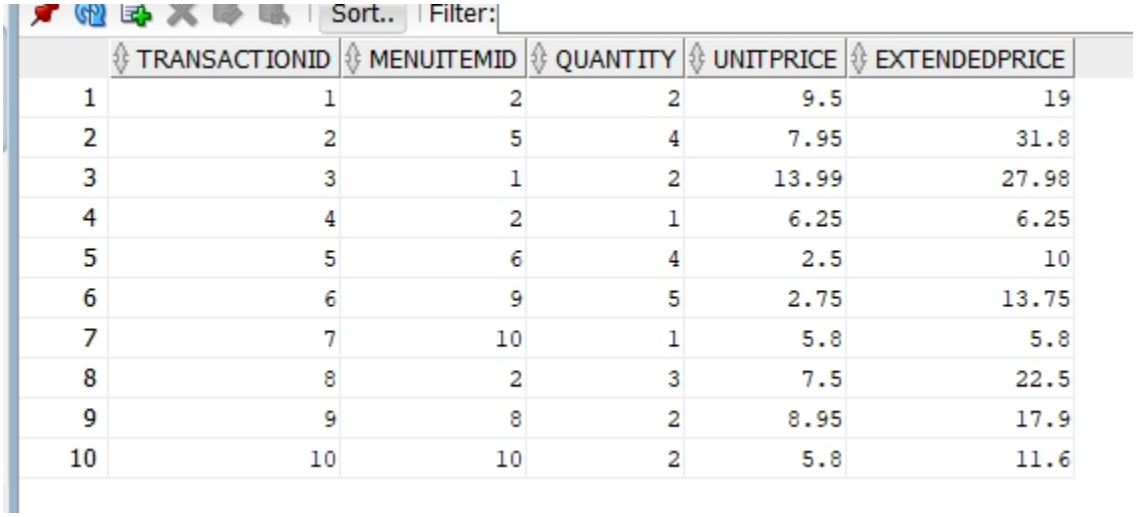
Transaction Table with Data:

	TRANSACTIONID	CUSTOMERID	EMPLOYEEID	TRANSACTIONTIME	SUBTOTAL	TAX	TOTAL
1	1	1	2	(null)	10	0.8	10.8
2	2	2	3	01-JAN-25 08.00.00.000000000 AM	22.5	1.8	24.3
3	3	1	4	14-FEB-25 12.30.45.000000000 PM	15.25	1.22	16.47
4	4	7	9	01-MAR-25 12.00.00.000000000 AM	5	0.4	5.4
5	5	2	3	23-APR-25 03.45.10.000000000 PM	20.1	1.61	21.71
6	6	8	4	(null)	12.99	1.04	14.03
7	7	10	1	05-MAY-25 10.15.55.000000000 PM	6	0.48	6.48
8	8	1	1	30-JUN-25 06.00.00.000000000 PM	18.75	1.5	20.25
9	9	6	4	04-JUL-25 11.59.59.000000000 PM	9.9	0.79	10.69
10	10	5	7	(null)	24	1.92	25.92

Menu_Item Table with Data:

	MENUITEMID	MENUNAME	CATEGORY	PRICE
1	1	Roast beef	Main Course	14.99
2	2	Fried rice	Main Course	9.5
3	3	Caesar salad	Salad	6.25
4	4	Baked salmon	Main Course	13.75
5	5	Chicken wings	Appetizer	7.95
6	6	Mashed potatoes	Side Dish	4.5
7	7	Steamed vegetables	(null)	3.99
8	8	Spaghetti with marinara	Main Course	8.95
9	9	Garlic bread	Side Dish	2.75
10	10	Chocolate cake	Dessert	5.8

Transaction_Item Table with Data:



	TRANSACTIONID	MENUITEMID	QUANTITY	UNITPRICE	EXTENDEDPRICE
1	1	2	2	9.5	19
2	2	5	4	7.95	31.8
3	3	1	2	13.99	27.98
4	4	2	1	6.25	6.25
5	5	6	4	2.5	10
6	6	9	5	2.75	13.75
7	7	10	1	5.8	5.8
8	8	2	3	7.5	22.5
9	9	8	2	8.95	17.9
10	10	10	2	5.8	11.6

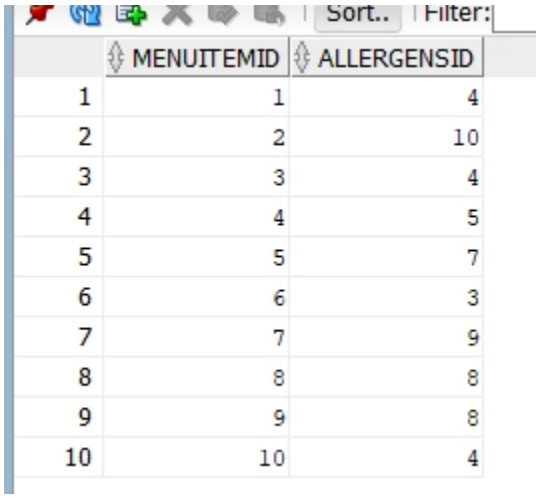
Ingredient_Use Table with Data:

	MENUITEMID	INGREDIENTID	QUANTITY	INSTRUCTION
1	1	1	2	Blend two tomatoes into a puree
2	2	2	3	slice into cube chunks
3	3	3	6	beat six eggs into a bowl
4	4	4	1	sprinkle any dish with a pinch of salt
5	5	5	2	chop and mince until fine texture
6	6	6	10	slice into three pieces: head, body, and tail
7	7	7	5	slice into four equal chunks
8	8	8	3	melt into a pan until slightly brown
9	9	9	5	(null)
10	10	10	2	rinse in a bowl, cook for 10 minutes
11	2	5	2	Cut onions to cook the fried rice

Allergens Table with Data:

	ALLERGENSID	ALLERGENDESCRIPTION
1	1	Peanuts
2	2	Tree Nuts
3	3	Milk
4	4	Eggs
5	5	Fish
6	6	Shell Fish
7	7	Soy
8	8	Wheat
9	9	Sesame
10	10	Gluten

Menu_Item_Allergens Table with Data:



Sort.. Filter:

	MENUITEMID	ALLERGENSID
1	1	4
2	2	10
3	3	4
4	4	5
5	5	7
6	6	3
7	7	9
8	8	8
9	9	8
10	10	4