# CSC261 Database Project Design And Diagramming

Name of Project – University of Rochester Food Delivery Service Database

Team ID – 26

Member IDs – Simranjit Singh: 57 & Alan Kuo: 5 & Edmund Doyle: 18

# 

# ER-to-Relational Mapping Algorithm

## Step1: Mapping of Regular Entity Types

User

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| UserID | Username | Password | FirstName | LastName | Email | Street Address | Residence Hall | City |

Order

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| OrderID | TimePlaced | Quantity | Instructions | TotalPrice |

Item

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| ItemID | ItemName | ServingSize | QuantityInStock | Calories | TotalFat | SaturatedFat | TransFat | Cholesterol | Sodium |
| TotalCarbs | **Fiber** | **Sugar** | **Protein** | **Vegan** | **Vegetarian** | **Halal** | **Kosher** | **International** | **Image** |
| Price |  |

## Step2: Mapping of Weak Entity Types

Payment

|  |  |  |  |
| --- | --- | --- | --- |
| OrderID | CardNum | CVV | ExpirationDate |

# Step3: Mapping of Binary 1:1 Relationship Types

Merged relation approach: Merge Order and Payment entity types into one relation

Order

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| OrderID | TimePlaced | Quantity | Instructions | TotalPrice |
| CardNum | **CVV** | **ExpirationDate** |  |  |

# Step4: Mapping of Binary 1:N Relationship Types

In our schema, we have two 1:N relationships. Below are the actions we took for this step:

Foreign key approach:

1) Include Primary key- UserID from User as a foreign key in Order.

2) Include Primary Key- ItemID from Item as a foreign key in Order.

Order

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| OrderID | TimePlaced | Quantity | Instructions | TotalPrice |
| CardNum | **CVV** | **ExpirationDate** | **UserID (FK)** | **ItemID (FK)** |

# Step5: Mapping of Binary M:N Relationship Types

Relationship relation (cross-reference) option: Create a new relation -Preferences, including Primary key UserID from User and Primary key ItemID as foreign keys in this relation. Together they are the Super key of Preferences.

Preferences

|  |  |
| --- | --- |
| UserID | ItemID |

## Step 6: Mapping of Multivalued Attributes

We have one multivalued attribute- Location for item entity. A new table – ItemLocation will be created with below schema. ItemID and Location are both primary keys in this relation.

ItemLocation

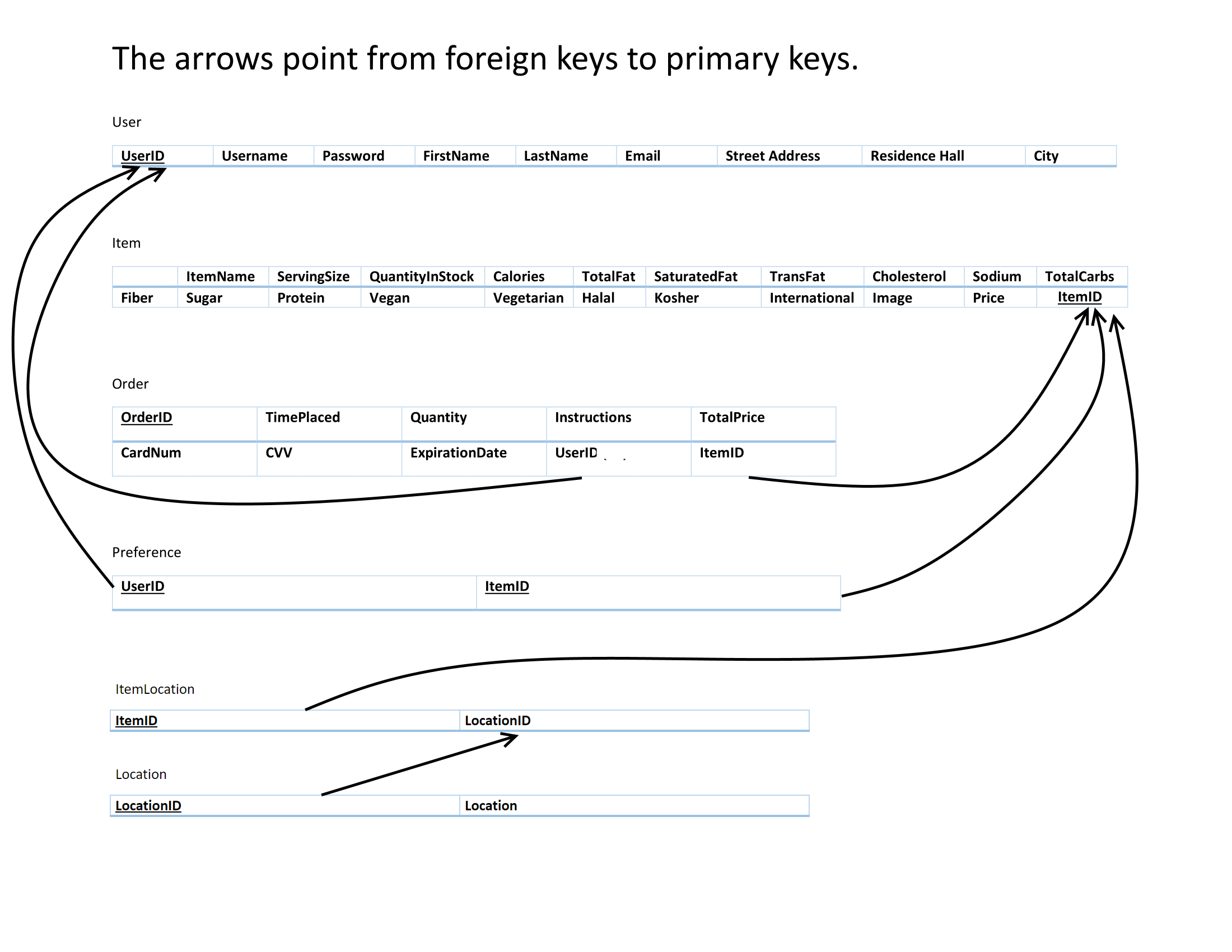
|  |  |
| --- | --- |
| ItemID | Location |

# Step 7:Mapping of N-ary Relationship Types

Not applicable

## Mapping Diagram

|  |  |
| --- | --- |
| Relation Name | ER Diagram Components |
| User | E(User) |
| Item | E(Item) |
| Order | 1. (E)User + R(Places)+(R)Order+ R(OR(Has)+R(Payment) 2. E)User + R(Places)+(R)Order+ R(contains) +E(item) |
| Preference | (E)User +R(Prefers)+R(Item) |
| ItemLocation | E(Item)+R(Location) |
| Location | A(Location) |



# Schema

Table - User

|  |  |  |  |
| --- | --- | --- | --- |
| Attribute | Data type | Constraint | Description |
| UserID | int | Primary Key(not null) | Specifies the unique ID assigned to a given user |
| Username | VARCHAR(255) | Not null | The display name of a user in the database |
| Password | VARCHAR(255) | Not null | An individual security key chosen by a user |
| FirstName | VARCHAR(255) | Not null | first name |
| LastName | VARCHAR(255) | Not null | last name |
| Email | VARCHAR(255) | not null | Email address |
| Street Address | VARCHAR(255) | null | street address |
| Residence Hall | VARCHAR(255) | null | Residence hall |
| City | VARCHAR(255) | null | City of residence |

Table -Item

|  |  |  |  |
| --- | --- | --- | --- |
| Attribute | Data type | Constraint | Description |
| ItemID | int | Primary Key ,Not null | Specifies the unique ID assigned to a given item |
| ItemName | VARCHAR(255) | NOT NULL | Gives the full name of the item in stock |
| ServingSize | VARCHAR(255) | NOT NULL | size of portion |
| QuantityInStock | int | NOT NULL | quantity of stock |
| Calories | int | null | calories per serving size |
| TotalFat (g) | int | null | g of fat per serving size |
| SaturatedFat (g) | int | null | g of saturated fat per serving size |
| TransFat (g) | int | null | g of trans fat per serving size |
| Cholesterol (mg) | int | null | mg of cholesterol per serving size |
| Sodium (mg) | int | null | mg of sodium per serving size |
| TotalCarbs (g) | int | null | g of carbs per serving size |
| Fiber (g) | int | null | g of fiber per serving size |
| Sugar (g) | int | null | g of sugar per serving size |
| Protein (g) | int | null | g of protein per serving size |
| Vegan | BOOLEAN | default false | Flag indicating whether a food conforms to a vegan diet |
| Vegetarian | BOOLEAN | default false | Flag indicating whether a food conforms to a vegetarian diet |
| Halal | BOOLEAN | default false | Flag indicating whether a food conforms to a Halal diet |
| Kosher | BOOLEAN | default false | Flag indicating whether a food conforms to a kosher diet |
| International | BOOLEAN | NOT NULL | Flag indicating whether a food originates from a non-American culture |
| Image | Binary | null | Image of item |
| Price | float | NOT NULL | Price of one serving size |

Table -Order

|  |  |  |  |
| --- | --- | --- | --- |
| Attribute | Data type | Constraint | Description |
| OrderID | int | Primary Key, NOT NULL | Specifies the unique ID assigned to a given order |
| TimePlaced | DATETIME | NOT NULL | Time the order was placed |
| Quantity | int | NOT NULL | Quantity of item that was ordered |
| Instructions | TEXT |  | Special instructions for preparation/delivery |
| TotalPrice | float | NOT NULL | Total price of the order |
| CardNum | int | NOT NULL | Credit card number (16 digits) |
| CVV | int | NOT NULL | Credit card security code(3 digits) |
| ExpirationDate | int | NOT NULL | Month and year of credit card expiration |
| UserID(FK) | int | NOT NULL | Foreign Key references User(UserID)  On delete: no action |
| ItemID(FK) | int | NOT NULL | Foreign Key references item(ItemID)  On delete: no action |

Table -Preference

|  |  |  |  |
| --- | --- | --- | --- |
| Attribute | Data type | Constraint | Description |
| UserID (FK) | int | NOT NULL | Foreign Key references User(UserID)  On delete: Delete cascade |
| ItemID (FK) | int | NOT NULL | Foreign Key references item(ItemID)  On delete: Delete cascade |

Table -ItemLocation

|  |  |  |  |
| --- | --- | --- | --- |
| Attribute | Data type | Constraint | Description |
| ItemID (FK) | int | NOT NULL | Foreign Key References Item(itemID) On delete: delete cascade |
| LocationID | int | NOT NULL | unique ID for location |

Table -Location

(created to avoid anomalies)

|  |  |  |  |
| --- | --- | --- | --- |
| Attribute | Data type | Constraint | Description |
| LocationID(FK) | int | Primary Key,NOT NULL | Foreign Key References ItemLocation(LocationID) On delete: delete cascade |
| Location | TEXT | NOT NULL | Location where the item is stored. An item can be stored in more than one location. |