Ing-Ready-nts

10.06.2022

—

CS 411 PT-1 Team 083
Aaditya Dhruv
Nidhish Kamath
Vaasay Farid
Vishal Dayalan

Assumptions for Entities & their Relationships -

The database should store information about the **Users, Recipes, Inventory, Orders and DiscountCoupons.**

Users are uniquely identified by their UserID. Other attributes are Name, Address, CardNum, LoginPassword and IsAdmin indicating whether they are admins or not.

Recipes are uniquely identified by their RecipeID. Other attributes are RecipeName, Serves (serves how many people), Description and Process.

Inventory contains a list of all products available in the store ready to be ordered. Tuples are uniquely identified by their ProductID. Other attributes are ProductName, MOQ (minimum order quantity), QtyAvailable, Price and other attributes that represent tags such as IsVegetarian, IsVegan, IsLactoseFree, IsHalal, IsGlutenFree, IsOnionGarlicFree.

Orders are uniquely identified by OrderID, TotalPrice & CouponID (that is applied on the order).

DiscountCoupons are uniquely identified by their CouponID and another attribute is DiscPercent that tells the percentage of discount to be applied.

Users can place multiple **Orders**. In the other direction of relation, one **Order** can only be placed by a single **User**.

An **Order** can contain at least one but upto multiple **Recipes,** and a **Recipe** can be contained in multiple **Recipes.**

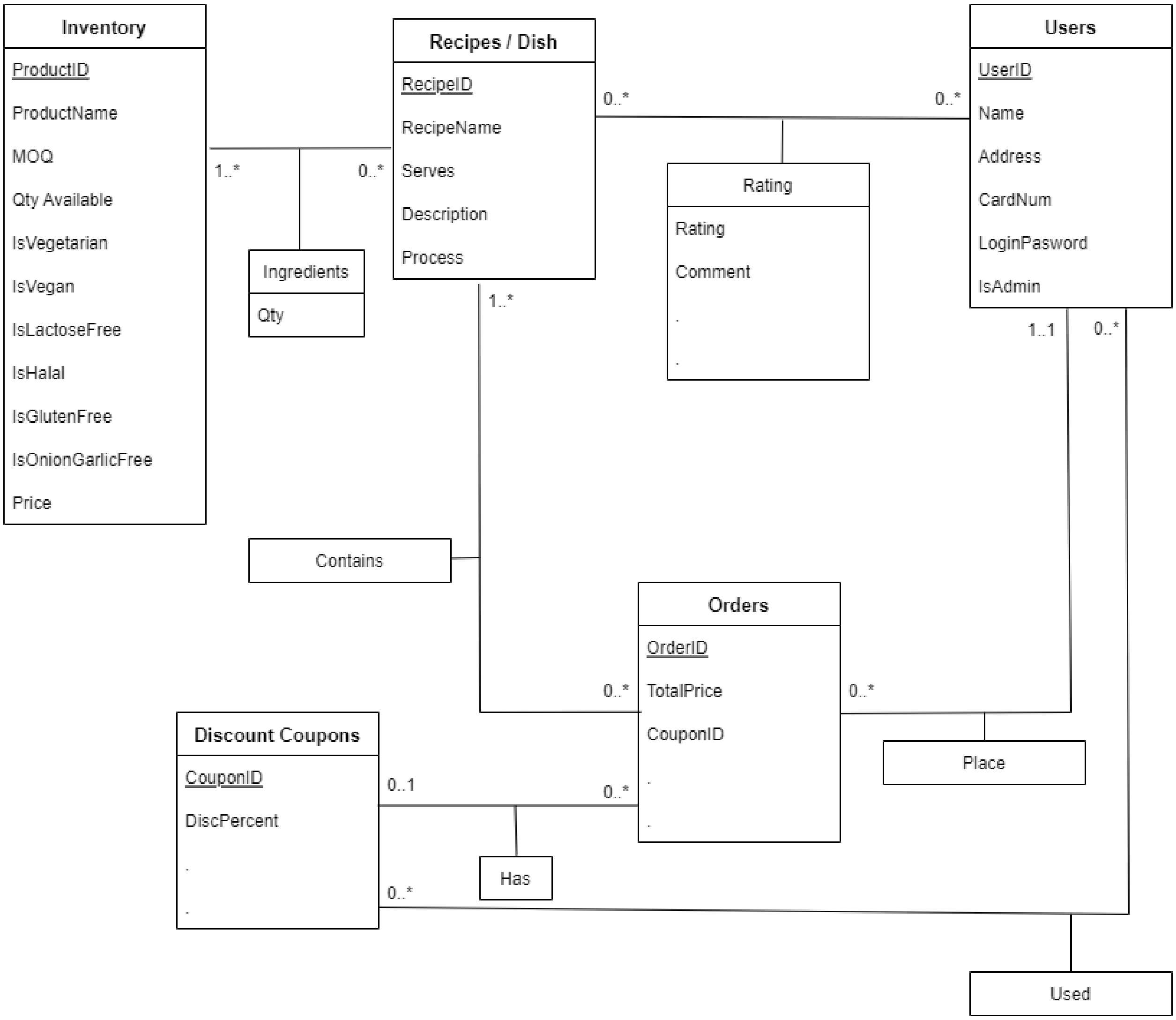
A **Recipe** needs to have at least one ingredient from the **Inventory**. We assume for convenience, that an ingredient in the Inventory need not be referenced by a **Recipe.** The quantity of the product needed in a **Recipe** will also be stored in the relational entity.

DiscountCoupons uniquely identified by their CouponID can be optionally applied to any **Order** but an **Order** can only contain one **DiscountCoupon.**

A **User** can use any number of **DiscountCoupons** that they like, or not at all. Likewise, a **DiscountCoupon** may or may not be used by any number of **Users**.

A **User** leaves a rating on any number of **Recipes** that he chooses too (and has ordered), while a **Recipe** can be rated by none or many **Users**.

UML Diagram


Inventory
ProductID
ProductName
MOQ
Qty Available
IsVegetarian
IsVegan
IsLactoseFree
IsHalal
IsGlutenFree
IsOnionGarlicFree
Price
1..*
0..*
Ingredients
Qty
Recipes / Dish
RecipeID
RecipeName
Serves
Description
Process
0..*
0..*
Rating
Rating
Comment
.
.
Users
UserID
Name
Address
CardNum
LoginPasword
IsAdmin
1..*
1..1
0..*
Contains
Orders
OrderID
TotalPrice
CouponID
.
.
0..*
0..*
Place
Discount Coupons
CouponID
DiscPercent
.
.
0..1
0..*
Has
0..*
Used

Relational Schema

Users(UserID: INT [PK], Name: VARCHAR(30), Address: VARCHAR(200), CardNum: CHAR(16), LoginPassword: VARCHAR(64), isAdmin: INT);

Rating(UserID: INT FK REFERENCES Users.UserID, RecipeID: INT REFERENCES Recipes.RecipeID, Rating: REAL, Comment : VARCHAR(300));

Recipes(RecipeID: INT [PK], RecipeName: VARCHAR(30), Serves: INT, Description: VARCHAR(500), Process: VARCHAR(500));

Orders(OrderID: INT [PK], TotalPrice: REAL, CouponID: INT FK DiscountCoupons);

DiscountCoupons(CouponID: INT [PK], DiscPercent: REAL);

Inventory(ProductID: INT [PK], ProductName: VARCHAR(200), MOQ: REAL, Qty: REAL, isVegetarian: INT, isVegan: INT, isLactoseFree: INT, isHalal: INT, isGlutenFree: INT, isOnionGarlicFree: INT, Price: REAL);

Ingredients(RecipeID: INT FK REFERENCES Recipes.RecipeID, ProductID: INT REFERENCES Inventory.ProductID, Qty: INT, (RecipeID, ProductID) [PK]);

Contains (RecipeID: INT FK REFERENCES RECIPE.RecipeID , OrderID: INT REFERENCES Orders.OrderID, (RecipeID, OrderID) [PK]);

Place(OrderID: INT FK REFERENCES Orders.OrderID, UserID: INT REFERENCES Users.UserID, (OrderID, USERID) [PK]);

Has(CouponID: INT FK REFERENCES DiscountCoupons.CouponID, OrderID: INT REFERENCES Orders.OrderID, (CouponID, OrderID) [PK]);

Used(CouponID:INT FK REFERENCES DiscountCoupons.CouponID, UserID:INT REFERENCES Users.UserID, (CouponID, UserID) [PK]);