

Final Project

Shannon Davis, Nicholas Tippner, Talanda Williams

SCOPE:

The database design that we implemented stands to solve an issue faced by restaurants by tracking information from the menu, to the inventory, all the way to the supplier.

In the restaurant business, there is a delicate balance between having too little or too many ingredients necessary for the menu items in a given timeframe. You don't want to run out of things or have so many that they spoil.

ORGANIZATION:

The database consists of 4 tables: menu, recipe, inventory, and supplier.

Menu contains:

- ItemID INT(11) - PK - Also FK
- ItemName VARCHAR(45)
- ItemPriceAfterTax DECIMAL(9,2)
- ItemPriceBeforeTax DECIMAL(9,2)
- ItemCalories INT(11)
- ItemType VARCHAR(45)
- ItemDescription VARCHAR(140)

Recipe contains:

- ItemID INT(11) - PK - Also FK
- InventoryID INT(11) - PK - Also FK
- AmountNeeded INT(11)

Inventory contains:

- InventoryID INT(11) - PK
- InventoryName VARCHAR(45)

- InventoryPrice DECIMAL(9,2)
- InventoryCount INT(11)
- InventoryType VARCHAR(45)
- InventoryCalories INT(11)
- SupplierID INT(11) - FK

Supplier contains:

- SupplierID INT(11) - PK - Also FK
- SupplierName VARCHAR(45)
- SupplierAddress VARCHAR(45)
- SupplierCity VARCHAR(45)
- SupplierState VARCHAR(45)

Menu and recipe are linked via foreign key by the ItemID, which links with a foreign key of InventoryID to the inventory table. The menu items refers to things like soup, lasagna, and chocolate cake. The inventory table keeps track of the quantity of the ingredients necessary for the menu items. These would be things such as flour, tomatoes, and chicken stock. Finally, the SupplierID acts as a foreign key to the supplier table.

NEEDS/QUESTIONS:

The main needs that the database addresses:

What supplier sends in a particular menu item and how much does it cost to procure? How many of the item is in inventory at a given moment and what does it sell for as a menu item?

Some of the questions that were faced when normalizing the table:

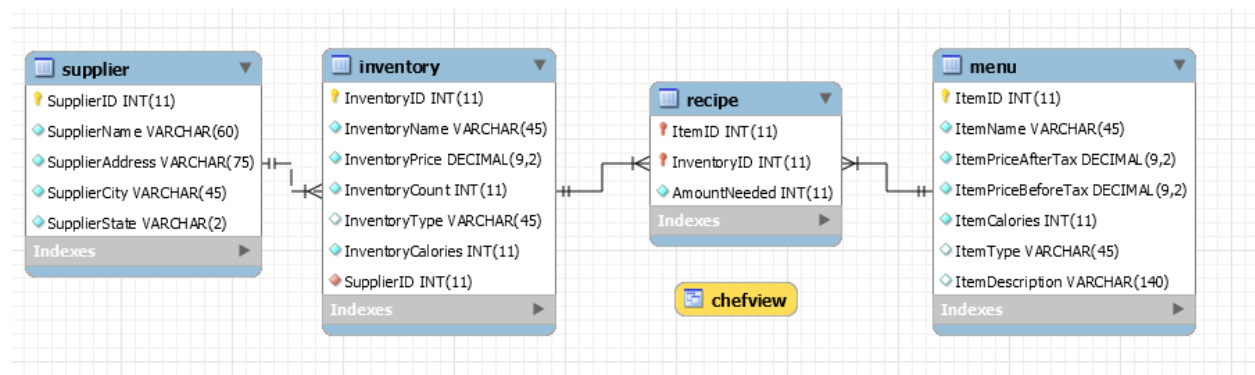
Do we allow multiple suppliers to ship a particular menu item? Does inventory refer to a particular storage location or does it refer to the combination of all?

These questions had to be answered in order to structure the 1:N or 1:1 nature of the FK links.

PURPOSE:

The overall purpose is to provide a high-level inventory tracker for items on a menu and a method of finding the procurement/reordering supplier information.

ER DIAGRAM:



NORMALIZATION:

To normalize our database we separated each leaf, the menu, recipe, inventory and supplier. For inventory (ingredients) the supplier is separated. This allows us to add a supplier, or remove them. Also if we stop carrying a particular inventory item, we don't lose out on the fact that the supplier that sent that particular item exists.

If we remove a particular menu item, we don't inadvertently delete the inventory (ingredients) that were involved in making it. If we change the recipe to make a menu item, it won't lose information and it would easily update (via trigger or otherwise) the caloric content of the menu item.

Menu price is dictated in part but the ingredient cost. In this way, an update to an ingredient cost could easily cascade the change to reflect on the menu.

By this explanation we found this to be a normalized table.

EXTRA CREDIT:

There would be two useful functions that an individual using this table might rely on. One would be the total cost of the inventory in case of a catastrophic failure, such as long-term power outage causing spoilage or a natural disaster. The other would be the easy access of a quantity of a particular item. While the latter could be done through a select statement, this is only if the individual has a basic understanding of SQL, which a function could be used for generally everyone.

```
CREATE FUNCTION restaurant.InventoryTotal ()
```

```
returns decimal deterministic
```

```
return
```

```
    (SELECT SUM(InventoryPrice * InventoryCount) AS TotalInventoryValue FROM  
restaurant.inventory);
```

```
CREATE FUNCTION restaurant.CountOf (InvName CHAR(45))
```

```
returns int deterministic
```

```
return
```

```
    (select InventoryCount from restaurant.inventory where InventoryName = InvName);
```