# Database Design –Assignment 4

Robert Daigle

You have inherited a small catering company that is losing money. You suspect you are not charging enough for the dinners so you decide to put the components of the dinners in a database so you can use your new SQL skills to analyze your financial status. Fortunatly, each food item is used in only one dinner. (use the data in the table below)   The Fish dinner includes Salmon, a house salad, and beans; the chicken dinner includes cordon bleu, corn , a roll and ice cream; the steak dinner includes NY strip steak, a ceasar salad, asparagas, and pie.

ItemID	ItemName	ItemCost	ItemPrice	ItemDinner
	DinnerEntree			
1000	Fish Dinner		6.77	
1001	Chicken Dinner		8.05	
1002	Steak Dinner		10.85	
100	Salmon Filets	3.95	4.50	1000
201	House Salad	0.75	.97	1000
301	String Beans	1.00	1.30	1000
101	Cordon Bleu	4.25	4.75	1001
303	Corn	.65	0.90	1001
401	Ice Cream	1.60	2.10	1001
501	Dinner Rolls	.25	0.30	1001
103	NY Strip	4.95	5.55	1002
202	Caesar Salad	1.20	1.45	1002
302	Asparagus	1.10	1.40	1002
402	Pie	1.85	2.45	1002

1. Use Visio to draw the ERD for this table. See "How to Create a Recursive Relationship in Visio" for help. Paste the ERD   here.
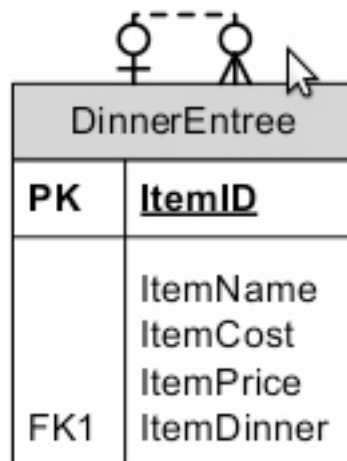
2. Use Create and Insert statements, to create and populate the table. Name it DinnerEntree.

Set ansi_nulls on
Go
Set quoted_identifier off
Go

/* 2. Create Table*/

Create table DinnerEntree (
ItemID INTEGER,
ItemName VARCHAR(20),
ItemCost MONEY,
ItemPrice MONEY,
ItemDinner SMALLINT,
PRIMARY KEY(ItemID));

/* 2. Populate DinnerEntree table*/

Insert Into DinnerEntree
  values (1000,"Fish Dinner",NULL,6.77,Null)
Insert Into DinnerEntree
  values (1001,"Chicken Dinner",NULL,8.05,Null)
Insert Into DinnerEntree
  values (1002,"Steak Dinner",NULL,10.85,Null)
Insert Into DinnerEntree
  values (100,"Salmon Filets",3.95,4.50,1000)
Insert Into DinnerEntree
  values (201,"House Salad",0.75,0.97,1000)
Insert Into DinnerEntree
  values (301,"String Beans",1.00,1.30,1000)
Insert Into DinnerEntree
  values (101,"Cordon Bleu",4.25,4.75,1001)
Insert Into DinnerEntree
  values (303,"Corn",0.65,0.90,1001)
Insert Into DinnerEntree
  values (401,"Ice Cream",1.60,2.10,1001)
Insert Into DinnerEntree
  values (501,"Dinner Rolls",0.25,0.30,1001)
Insert Into DinnerEntree
  values (103,"NY Strip",4.95,5.55,1002)
Insert Into DinnerEntree
  values (202,"Caesar Salad",1.20,1.45,1002)
Insert Into DinnerEntree
  values (302,"Asparagus",1.10,1.40,1002)
Insert Into DinnerEntree
  values (402,"Pie",1.85,2.45,1002)

Write the SQL queries to answer the following requirements. Remember to sort all lists.
-3. List the names of the three dinners.

/*3.List Dinners*/

Select ItemName, ItemDinner From DinnerEntree
Where ItemDinner is NULL
Order By ItemName ASC;

-4. List the names of the dinners and names of the component items of each dinner.

/*4. List Dinners and Components*/

Select Dinner.ItemName, Component.ItemName From DinnerEntree Dinner, DinnerEntree Component
Where Dinner.ItemID = Component.ItemDinner
Order By Dinner.ItemName,Component.ItemName ASC;

-5.List each dinner name and current price, the total cost of each dinner, and the price each dinner would be with a 30% markup over cost.

/*5. List dinner, current price, cost and cost plus 30% */

Select Dinner.ItemName AS "Name", Dinner.ItemPrice AS "Price", Sum(Component.ItemCost) As Cost,
  (SUM(Component.ItemCost))*1.30 AS "MarkUp Price"

From DinnerEntree Dinner, DinnerEntree Component
Where Dinner.ItemID = Component.ItemDinner
Group By Dinner.ItemName, Dinner.ItemPrice
Order By Dinner.ItemName;

6. Use the Update command to enter the total cost and the new price of the dinners using the 30% markup amount from #5 rounded up to the nearest 5 cents.

/*6. Update price*/

Update DinnerEntree

        Set ItemPrice = 8.80
        Where ItemID = 1001;

Update DinnerEntree

        Set ItemPrice = 7.45
        Where ItemID = 1000;

Update DinnerEntree

        Set ItemPrice = 11.85
        Where ItemID = 1002;