# Part A Normalized Model

## One Normal F

### Table Design

CREATE TABLE [dbo].[Sales]

(

[Id] INT NOT NULL PRIMARY KEY,

[SalesOrderId] INT NULL,

[DonutId] INT NULL,

[Name] NVARCHAR(50) NULL,

[Description] NVARCHAR(250) NULL,

[UnitPrice] MONEY NULL,

[Quantity] INT NULL,

[SaleDate] DATE NULL,

[SpecialHandlingNotes] NVARCHAR(500) NULL,

[CustomerId] INT NULL,

[CustomerFirstName] NVARCHAR(50) NULL,

[CustomerLastName] NVARCHAR(50) NULL,

[CustomerStreetAddress1] NVARCHAR(50) NULL,

[CustomerStreetAddress2] NVARCHAR(50) NULL,

[CustomerCity] NVARCHAR(50) NULL,

[CustomerState] NCHAR(2) NULL,

[CustomerZip] NCHAR(6) NULL,

[CustomerHomePhone] NCHAR(10) NULL,

[CustomerMobilePhone] NCHAR(10) NULL,

[CustomerOtherPhone] NCHAR(10) NULL

);

### Reasoning

I took the Sales form sheet and reviewed the data to break out each individual artifact. The table has been broken up based on the requirements and the unique data points found within the form. From there I used a standard naming convention to give each data point a self describing name like, CustomerFirstName, to make a clear designation on the type of value one could find in the column. Each data point was also examined to determine what type of data it best represented. A whole number such as id or count column was assigned as an integer, any short text string stored as nchar, longer text strings stored as nvarchar, and then money for the unit price.

## Two Normal F

### Table Design

### Reasoning