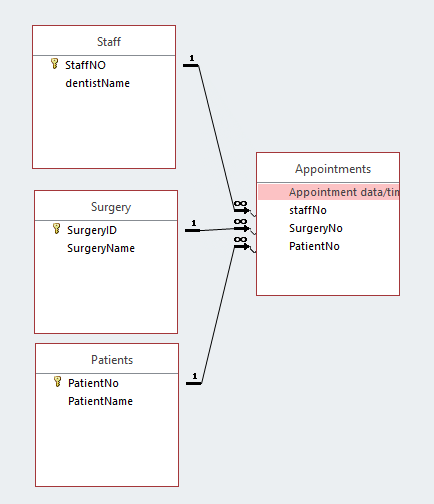
# Dental clinic data normalization

1. Anomaly Examples
   1. Insert Anomaly: We would run into an insert anomaly when a new dentist would join the practice. Not all of the field would be fill especially if he or she does not have any patients yet.
   2. Delete Anomaly: Similarly, if a dentist was to quit and we were to delete all the rows related to them, then the appointments that their patients had set up for the future would be lost.
   3. Update Anomaly: An update anomaly would happen in this data set if either a dentist or a patient were to change their name, multiple rows would need to be updated.
2. Normalize the Table
   1. 
   2. To avoid the duplication of both dentist and patient names and numbers I created 2 tables one labeled the Staff table and the other labeled Patients. I also created an appointments table that represents the time and time of surgery. A Surgery table was create to give additional information to what surgery was taking place.

# Sales Data in SQL

1. SELECT [ProductID],[SpecialOfferID],AVG([LineTotal]) as Average\_Price,SUM([LineTotal]) as Sub\_Total  
   FROM [AdventureWorks2014].[Sales].[SalesOrderDetail]  
   GROUP BY [ProductID],[SpecialOfferID];

# CAST and CONVERT

1. SELECT [ProductID]  
    ,CAST([LastReceiptCost] AS decimal(18,2)) AS CAST\_LAST\_RECEIPT   
    ,CONVERT(VARCHAR,[LastReceiptDate],101) AS CONVERT\_DATE\_MMDDYYYY  
   FROM [AdventureWorks2014].[Purchasing].[ProductVendor];
2. Anything you can do with CAST can be done with CONVERT. The Difference is that CONVERT will allow for a style parameter when formatting, an example whould be changing the format of a date.

# Shipping methods

1. SELCET [ShipMethod].[Name]  
    ,DATEPART(yyyy,[PurchaseOrderHeader].[OrderDate]) AS OrderYear  
    ,SUM([PurchaseOrderHeader].[TotalDue]) AS Total\_Due  
    ,AVG([PurchaseOrderHeader].[TotalDue]) AS AVG\_Total\_Due  
    , COUNT([PurchaseOrderHeader].[EmployeeID]) AS Number\_Of\_Employees  
    , COUNT(DISTINCT [PurchaseOrderHeader].[EmployeeID]) AS Distinct\_Number\_Of\_Emp  
   FROM [AdventureWorks2014].[Purchasing].[PurchaseOrderHeader]  
   INNER JOIN [AdventureWorks2014].[Purchasing].[ShipMethod]  
   ON [PurchaseOrderHeader].[ShipMethodID] = [ShipMethod].[ShipMethodID]  
   GROUP BY [ShipMethod].[Name],DATEPART(yyyy,[PurchaseOrderHeader].[OrderDate])  
   HAVING SUM([PurchaseOrderHeader].[TotalDue]) > 5000000  
   ORDER BY [ShipMethod].[Name] DESC,OrderYear;

# Stored procedure

1. USE [AdventureWorks2014]  
   GO  
   CREATE PROCEDURE AddProc  
   AS  
   SELECT \* FROM [AdventureWorks2014].[Person].[Address];  
   GO

# Left and right join

1. SELECT [Person].[BusinessEntityID]  
    ,[Person].[FirstName]  
    ,[Person].[LastName]  
    ,[EmailAddress].[EmailAddress]  
    ,[Password].[PasswordHash]  
    ,[Password].[PasswordSalt]  
   FROM [AdventureWorks2014].[Person].[Password]  
   LEFT JOIN [AdventureWorks2014].[Person].[EmailAddress]  
   ON [Password].[BusinessEntityID] = [EmailAddress].[BusinessEntityID]  
   RIGHT JOIN [AdventureWorks2014].[Person].[Person]  
   ON [Password].[BusinessEntityID] = [Person].[BusinessEntityID];

# PIVOT

1. SELECT VendorID, [250] AS EMP1, [251] AS EMP2, [256] AS EMP3, [257] AS EMP4, [260] AS EMP5  
   FROM(  
   SELECT [PurchaseOrderHeader].[PurchaseOrderID]   
    ,[PurchaseOrderHeader].[EmployeeID]  
    , [PurchaseOrderHeader].[VendorID]  
   FROM [AdventureWorks2014].[Purchasing].[PurchaseOrderHeader]  
   WHERE EmployeeID=250  
   OR EmployeeID=251  
   OR EmployeeID=256  
   OR EmployeeID=257  
   OR EmployeeID=260  
   ) as s  
   PIVOT  
   (  
   COUNT([PurchaseOrderID])  
   FOR [EmployeeID] IN ([250],[251],[256],[257],[260])  
   )AS pvt  
   ORDER BY pvt.VendorID;