Creating Entity Relationship Diagram Assignment

Neha Tarey

Grand Canyon University: MIS-605

February 28, 2019

Assignment

**Creating Entity Relationship Diagram (ERD)**

1. Create a temporary table for each of the Topic 4 assignment data components.

**Solution:**

There can be two temp tables from the previous assignment:

1. SalesOrderIDs with their OrderQuantities for Product LL Road Frame – Black, 60. And this table has 52 rows.
2. ListPrice history with the StandardCost and dates when the List Price was applicable for Product LL Road Frame – Black, 60. And this table has 3 rows.

---------Temp table 1 on Sales Order and Quantity-----------------------------

---1. Create temp table

SELECT

SOD.SalesOrderID ,

PP.ProductID,

PP.Name,

SOD.OrderQty AS Quantity

INTO #SalesQuantityTemp

FROM Sales.SalesOrderDetail SOD

JOIN Production.Product PP

ON SOD.ProductID = PP.ProductID

WHERE PP.Name = 'LL Road Frame - Black, 60';

--2. Alter table to add new column

ALTER TABLE #SalesQuantityTemp

ADD DateRan DATETIME NOT NULL DEFAULT (GETDATE());

---------Temp table 2 on ListPrice History-----------------------------

---1. Create temp table-------

SELECT

LPH.ProductID,

PP.Name AS ProductName,

LPH.ListPrice,

LPH.StartDate AS ListPriceStartDate,

LPH.EndDate AS ListPriceEndDate

INTO #BlackFrameListPrice

FROM Production.ProductListPriceHistory LPH

JOIN Production.Product PP

ON LPH.ProductID = PP.ProductID

WHERE LPH.ProductID = '723';

--2. SELECT to get data---

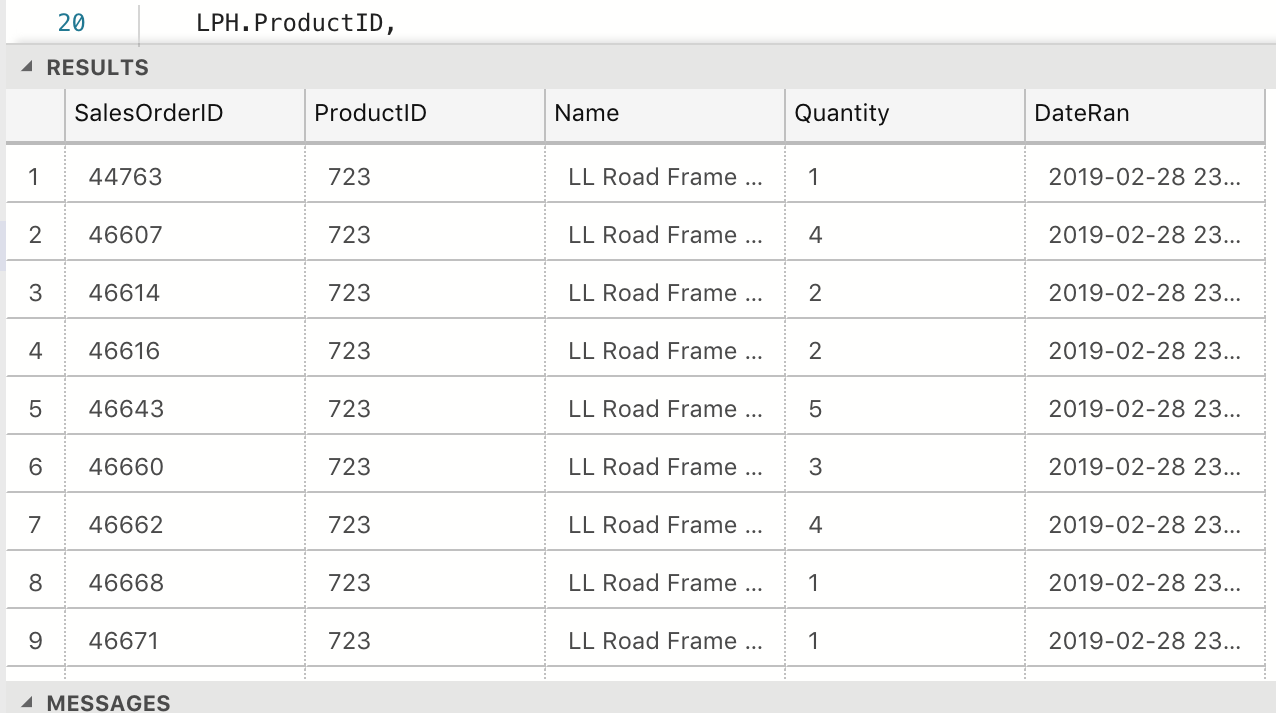
SELECT

\*

FROM #BlackFrameListPrice;

-------END OF TABLE CREATION----------------

**#SalesQuantityTemp –** With 52 rows



**#BlackFrameListPrice –** with 3 rows



1. Use the skills you have learned and practiced to write a query to determine the dates of each sales orders from a table you have not looked at yet. Create a temporary table for this information.

**Solution:**

Earlier, I had not looked at the sales order dates in the TransactionHistory and TransactionHistoryArchives table where each of the orders are separated by transaction type, for Sales orders the transaction type will be ‘S’. First, I will be fetching data from TransactionHistory (gives 9 rows) and using that to create the temp table, followed by appending old records from TransactionHistoryArchives (gives 43 rows). These numbers in total should match with the total sales orders from the sales table, that is earlier determined to be 52 rows fir the LL Road Frame – Black, 60.

---------Question 2 --Date of Sales Orders----------------------------

----1. Create the temp table with Transaction History----------------------

select

PP.ProductID,

PP.Name,

TH.TransactionType,

TH.ReferenceOrderID as SalesOrderID ,

TH.TransactionDate as SalesDate,

TH.ModifiedDate

into #SalesTranDates

from Production.Product PP

join Production.TransactionHistory TH

on PP.ProductID = TH.ProductID

where PP.Name = 'LL Road Frame - Black, 60'

and TH.TransactionType = 'S';

----2. Add the remaining records from Transaction History Archive----------------

INSERT into #SalesTranDates

select

PP.ProductID,

PP.Name,

THA.TransactionType,

THA.ReferenceOrderID as SalesOrderID ,

THA.TransactionDate as SalesDate,

THA.ModifiedDate

from Production.Product PP

join Production.TransactionHistoryArchive THA

on PP.ProductID = THA.ProductID

where PP.Name = 'LL Road Frame - Black, 60'

and THA.TransactionType = 'S';

-------END OF Table Creation and Data population--------------------------

----3. SELECT the data for viewing---------------------------------------

SELECT

ProductID,

Name,

TransactionType,

SalesOrderID,

SalesDate,

ModifiedDate

FROM #SalesTranDates

ORDER BY SalesOrderID DESC;

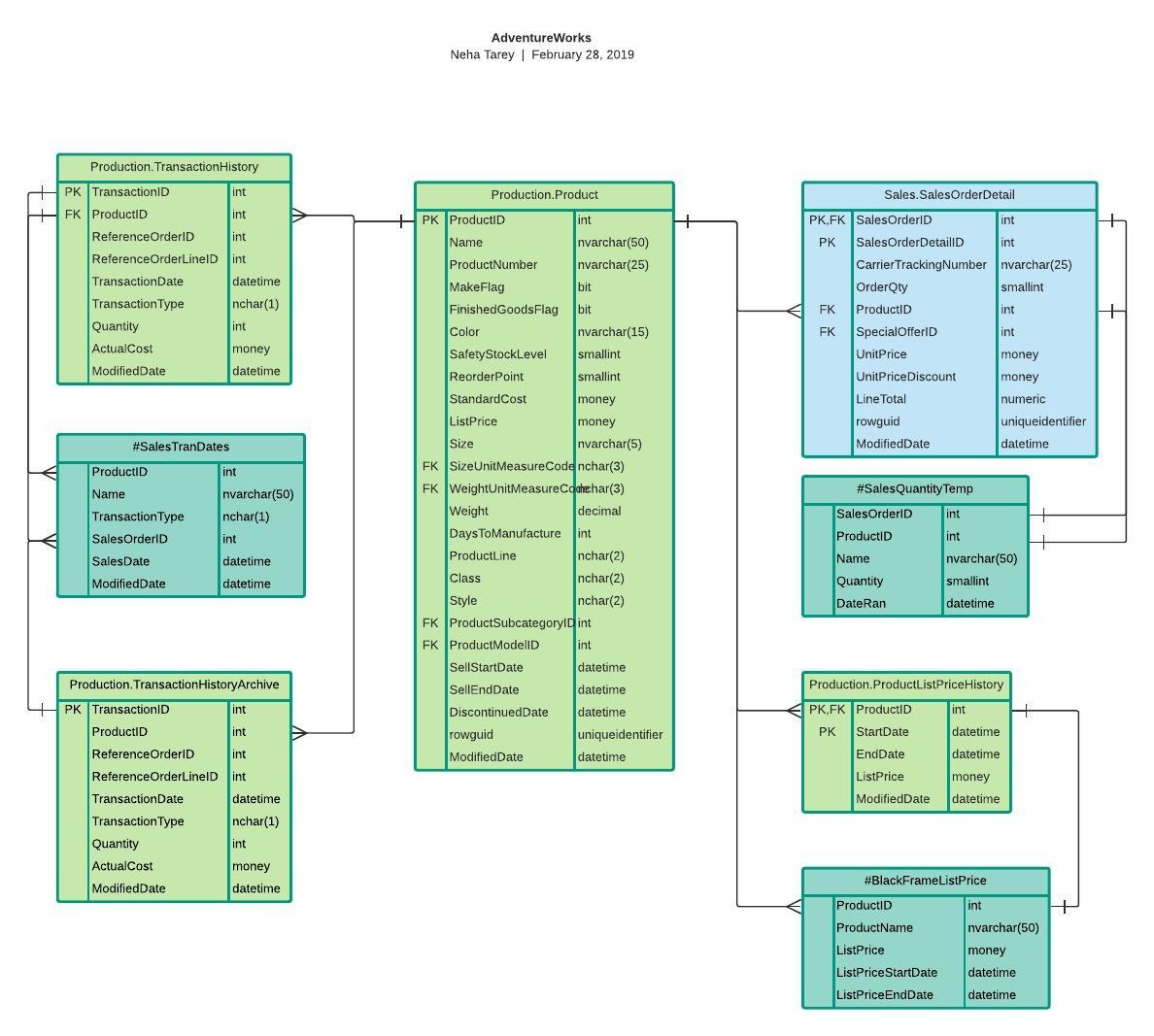


1. Use OpenOffice Draw to create a new entity relationship diagram (ERD) to document the addition of the temporary tables and the work stream. Show the relationship between the tables.

**Solution:**

1. We have so far created three temp tables: BlackFrameListPrice, SalesQuantityTemp and SalesTranDates.
2. Each of these tables contain the ProductID and hence are related to the Production.Product table.
3. SalesTranDates is related to TransactionHistory and TransactionHistoryArchive on the ProductID and SalesID field that contains the TransactionIDs.
4. SalesQuantityTemp is related to SalesOrderDetails table on the SalesOrderID and ProductID.
5. BlackFrameListPrice is related to ProductListPriceHistory on the ProductID.

The following ER diagram shows the three temp tables and the other associated tables.



**Result spreadsheet:**