Q30

Write a short essay talking about a scenario: Good news everyone! We (Wide World Importers) just brought out a small company called “Adventure works”! Now that bike shop is our sub-company. The first thing of all works pending would be to merge the user logon information, person information (including emails, phone numbers) and products (of course, add category, colors) to WWI database. Include screenshot, mapping and query.

1. We loaded the ***AdventureWorks*** sample databases from this website:

   https://docs.microsoft.com/en-us/sql/samples/adventureworks-install-configure?view=sql-server-ver15&tabs=ssms.

2. We inserted data into "***Application. People***" in WWI from "***Person. Person***" in ***AdventureWorks*** joined with multiple related tables shown in following fig:

```
INSERT INTO WideWorldImporters.Application.People
(FullName, PreferredName, IsPermittedToLogon, LogonName, IsExternalLogonProvider, HashedPassword,
IsSystemUser, IsEmployee, IsSalesperson, PhoneNumber, EmailAddress, CustomFields, LastEditedBy)
SELECT CONCAT(p.FirstName, p.MiddleName, p.LastName) AS FullName,
p.FirstName AS PreferredName,
CASE WHEN e.LoginID IS NOT NULL THEN 1 ELSE 0 END AS IsPermittedToLogon,
ISNULL(e.LoginID, 'NO LOGON') AS LogonName,
0 AS IsExternalLogonProvider,
CONVERT(varbinary(max), pw.PasswordHash) AS HashedPassword,
CASE WHEN pw.PasswordHash IS NOT NULL THEN 1 ELSE 0 END AS IsSystemUser,
CASE WHEN e.JobTitle IS NOT NULL THEN 1 ELSE 0 END AS IsEmployee,
CASE WHEN e.JobTitle LIKE '%Sales%' THEN 1 ELSE 0 END AS IsSalesperson,
pp.PhoneNumber AS PhoneNumber,
email.EmailAddress AS EmailAddress,
CONCAT('{ "OtherLanguages": [] ,"HireDate":"', e.HireDate, '","Title":"', e.JobTitle, '"}') AS CustomFields,
1 AS LastEditedBy
FROM AdventureWorks2019.Person.Person p
LEFT JOIN AdventureWorks2019.HumanResources.Employee e
ON p.BusinessEntityID = e.BusinessEntityID
LEFT JOIN AdventureWorks2019.Person.Password pw
ON p.BusinessEntityID = pw.BusinessEntityID
LEFT JOIN AdventureWorks2019.Person.PersonPhone pp
ON p.BusinessEntityID = pp.BusinessEntityID
LEFT JOIN AdventureWorks2019.Person.EmailAddress email
ON p.BusinessEntityID = email.BusinessEntityID;
```

3. we inserted data into "***Warehouse. Colors***", "***Purchasing. Suppliers***", and “***Warehouse.StockGroups***” in WWI from "***Production. Product***" and "***Production.ProductCategory***" in ***AdventureWorks*** shown in following fig.

```
INSERT INTO WideWorldImporters.Warehouse.Colors
(ColorName, LastEditedBy)
SELECT DISTINCT Color AS ColorName, 1 AS LastEditedBy
FROM AdventureWorks2019.Production.Product p
WHERE p.Color IS NOT NULL AND NOT EXISTS
(SELECT * FROM WideWorldImporters.Warehouse.Colors c
WHERE c.ColorName = p.Color COLLATE Latin1_General_100_CI_AS);
```

```
INSERT INTO WideWorldImporters.Purchasing.Suppliers
(SupplierName, SupplierCategoryID, PrimaryContactPersonID, AlternateContactPersonID, DeliveryCityID,
PostalCityID, PaymentDays, BankAccountNumber, PhoneNumber, FaxNumber, WebsiteURL, DeliveryAddressLine1, DeliveryPostalCode,
PostalAddressLine1, PostalPostalCode, LastEditedBy)
SELECT v.Name AS SupplierName,
1 AS SupplierCategoryID, 1 AS PrimaryContactPersonID, 1 AS AlternateContactPersonID, 1 AS DeliveryCityID, 1 AS PostalCityID,
0 AS PaymentDays, v.AccountNumber AS BankAccountNumber, '' AS PhoneNumber, '' [FaxNumber], '' [WebsiteURL], '' [DeliveryAddressLine1],
'' [DeliveryPostalCode], '' [PostalAddressLine1], '' [PostalPostalCode], 1 [LastEditedBy]
FROM AdventureWorks2019.Purchasing.Vendor v
WHERE NOT EXISTS
(SELECT * FROM WideWorldImporters.Purchasing.Suppliers s
WHERE s.SupplierName = v.Name COLLATE Latin1_General_100_CI_AS);
```

```
INSERT INTO WideWorldImporters.Warehouse.StockGroups
(StockGroupName, LastEditedBy)
SELECT pc.Name AS StockGroupName, 1 AS LastEditedBy
FROM AdventureWorks2019.Production.ProductCategory pc
WHERE NOT EXISTS
(SELECT * FROM WideWorldImporters.Warehouse.StockGroups
WHERE StockGroupName = pc.Name COLLATE Latin1_General_100_CI_AS);
```

4. we created a temporal table to save the query results from joined multiple tables related to product information in ***AdventureWorks*** and inserted them into "***Warehouse.StockItems***" in WWI shown in following fig.

```
SELECT DISTINCT p.Name AS StockItemName, s.SupplierID AS SupplierID, c.ColorID AS ColorID, 7 AS UnitPackageID,
7 AS OuterPackageID, p.Size AS Size, pv.AverageLeadTime AS LeadTimeDays, 1 As QuantityPerOuter, 0 AS IsChillerStock,
6.0 AS TaxRate, p.ListPrice AS UnitPrice, pv.StandardPrice AS RecommendedRetailPrice, ISNULL(p.Weight,0) AS TypicalWeightPerUnit,
pd.Description AS MarketingComments, pp.LargePhoto AS Photo, 1 AS LastEditedBy,
ROW_NUMBER() OVER(PARTITION BY p.ProductID ORDER BY p.Name) AS Row
INTO #Temp
FROM AdventureWorks2019.Production.Product p
INNER JOIN AdventureWorks2019.Purchasing.ProductVendor pv
ON p.ProductID = pv.ProductID
INNER JOIN AdventureWorks2019.Purchasing.Vendor v
ON pv.BusinessEntityID = v.BusinessEntityID
INNER JOIN WideWorldImporters.Purchasing.Suppliers s
ON v.Name = s.SupplierName COLLATE Latin1_General_100_CI_AS
INNER JOIN AdventureWorks2019.Production.ProductModel pm
ON p.ProductModelID = pm.ProductModelID
INNER JOIN AdventureWorks2019.Production.ProductModelProductDescriptionCulture pmpdc
ON pm.ProductModelID = pmpdc.ProductModelID
INNER JOIN AdventureWorks2019.Production.ProductDescription pd
ON pmpdc.ProductDescriptionID = pd.ProductDescriptionID
INNER JOIN AdventureWorks2019.Production.ProductProductPhoto ppp
ON p.ProductID = ppp.ProductID
INNER JOIN AdventureWorks2019.Production.ProductPhoto pp
ON ppp.ProductPhotoID = pp.ProductPhotoID
INNER JOIN WideWorldImporters.Warehouse.Colors c
ON p.Color = c.ColorName COLLATE Latin1_General_100_CI_AS
WHERE NOT EXISTS
(SELECT * FROM WideWorldImporters.Warehouse.StockItems si
WHERE si.StockItemName = p.Name COLLATE Latin1_General_100_CI_AS);

INSERT INTO WideWorldImporters.Warehouse.StockItems
(StockItemName, SupplierID, ColorID, UnitPackageID, OuterPackageID, [Size], LeadTimeDays, QuantityPerOuter, IsChillerStock,
TaxRate, UnitPrice, [RecommendedRetailPrice], TypicalWeightPerUnit, [MarketingComments], [Photo], LastEditedBy)
SELECT
CONCAT(StockItemName, Row) AS StockItemName, SupplierID, ColorID, UnitPackageID, OuterPackageID, Size, LeadTimeDays, QuantityPerOuter,
IsChillerStock, TaxRate, UnitPrice, RecommendedRetailPrice, TypicalWeightPerUnit, MarketingComments, Photo, LastEditedBy
FROM #Temp;
```

5. Last, we insert the information to the joined table, "***Warehouse.StockItemStockGroups***", in WWI from "***Warehouse.StockItems***" and "***Warehouse.StockGroups***" as following fig.

```
INSERT INTO WideWorldImporters.Warehouse.StockItemStockGroups
(StockItemID, StockGroupID, LastEditedBy)
SELECT si.StockItemID, ps.ProductCategoryID AS StockGroupID, 1 [LastEditedBy]
FROM AdventureWorks2019.Production.Product p
INNER JOIN AdventureWorks2019.Production.ProductSubcategory ps
ON p.ProductSubcategoryID = ps.ProductSubcategoryID
INNER JOIN #Temp ON p.Name = #Temp.StockItemName
INNER JOIN WideWorldImporters.Warehouse.StockItems si
ON CONCAT(#Temp.StockItemName, #Temp.Row) = si.StockItemName COLLATE Latin1_General_100_CI_AS;
```

We met some problems include the different server collations in the two databases, and the duplicated product names that we added the ROW_NUMBER() function to solve.