

1. Step 1:

Object Explorer

```

USE master
GO
IF EXISTS (
    SELECT name
    FROM sys.databases
    WHERE name = N'FashionKey'
)
    Drop DATABASE FashionKey
GO

Create Database FashionKey
GO

USE FashionKey
/* - CREATE TABLE statements */

CREATE TABLE Customer
(
    CustomerID INT IDENTITY(1,1) PRIMARY KEY,
    FirstName VARCHAR(50) NOT NULL,
    LastName VARCHAR(50) NOT NULL,
    Email VARCHAR(100) NOT NULL
)

```

Messages

Commands completed successfully.

Completion time: 2023-03-27T19:01:03.4880726-05:00

Query executed successfully.

BCIS4660-022 (15.0 RTM) | STUDENTS\sm1763 (56) | FashionKey | 00:00:00 | 0 rows

Ready

Object Explorer

```

USE master
GO
IF EXISTS (
    SELECT name
    FROM sys.databases
    WHERE name = N'FashionKey'
)
    Drop DATABASE FashionKey
GO

Create Database FashionKey
GO

USE FashionKey
/* - CREATE TABLE statements */

CREATE TABLE Customer
(
    CustomerID INT IDENTITY(1,1) PRIMARY KEY,
    FirstName VARCHAR(50) NOT NULL,
    LastName VARCHAR(50) NOT NULL,
    Email VARCHAR(100) NOT NULL
)

```

Messages

Commands completed successfully.

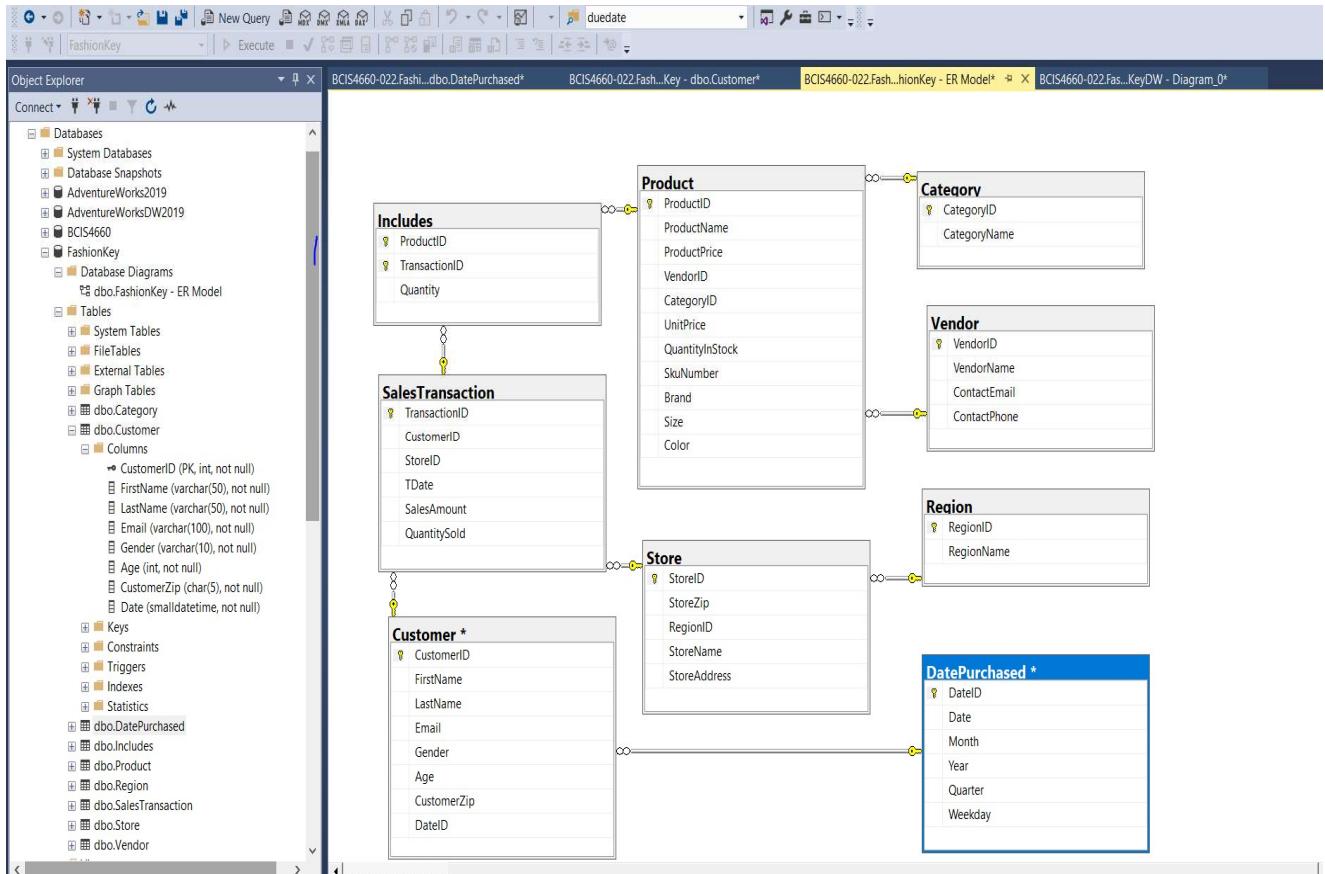
Completion time: 2023-03-27T19:01:03.4880726-05:00

Query executed successfully.

BCIS4660-022 (15.0 RTM) | STUDENTS\sm1763 (56) | FashionKey | 00:00:00 | 0 rows

SS 1: FashionKey operational database with the required tables

2. Step 2:



SS2: ER model for FashionKey

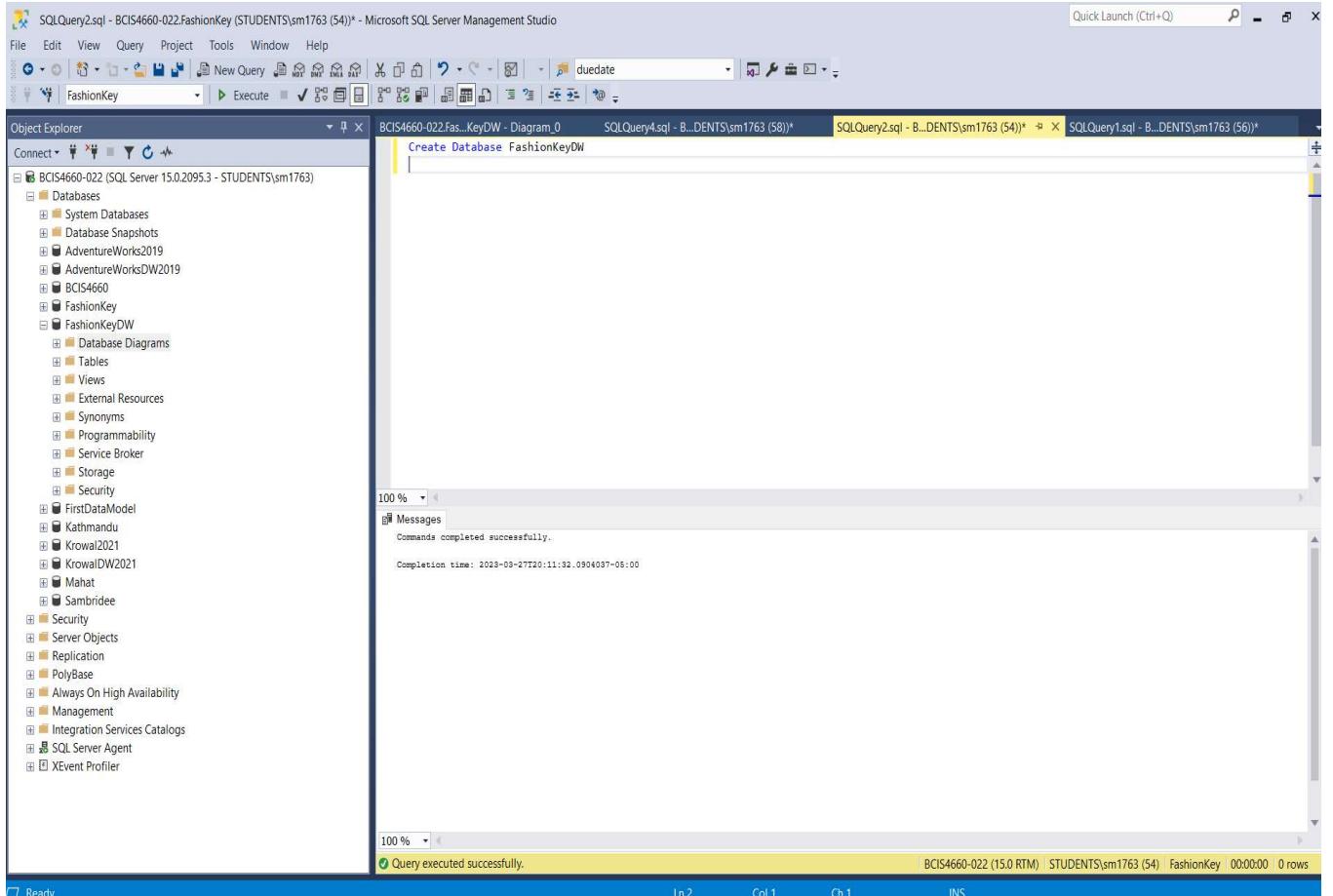
After the operational database was successfully created, first of all, SQL query to authorize for database diagrams was executed. Then under the FashionKey, right clicked on the database diagrams and selected the new database diagrams. Then selected all the tables and Ok.

The 4-step process for designing a dimensional model for FashionKey Inc is as follows:

- **Business Process:** the business process that will be the subject of the analysis is the first step in creating a dimensional model. The sales process, in which FashionKey Inc. sells clothing items to customers through its online and physical stores, could be considered the business process in this case.
- **Grain:** Grain is the lowest level of detail. The granularity, for instance, may be at the individual transaction level in the case of FashionKey Inc., where each row in the fact table represents a distinct sales transaction and the dimensions give extra information.
- **Dimensions:** Selecting the dimensions that will be utilized to set the context for the analysis is the third stage. Date, client, product, shop, vendor, and area are a few examples of conceivable aspects in the context of FashionKey Inc.

- Fact: The last stage is to define the fact table, which will include the measurements or metrics that will be examined. The fact table for FashionKey Inc. might include statistics like sales revenue, amount sold, and profit margin.

3. Step 3:



SS 3: Created FashionKeyDW Database

Executed t-sql to create a FashionKeyDW Database

4. Step 4:

Column Name	Data Type	Allow Nulls
CustomerKey	int	<input type="checkbox"/>
CustomerName	nvarchar(50)	<input checked="" type="checkbox"/>
Email	nvarchar(50)	<input checked="" type="checkbox"/>
Gender	nvarchar(50)	<input checked="" type="checkbox"/>
Age	int	<input checked="" type="checkbox"/>
CustomerZip	nchar(5)	<input checked="" type="checkbox"/>

Column Name	Data Type	Allow Nulls
VendorKey	int	<input type="checkbox"/>
VendorName	nvarchar(50)	<input checked="" type="checkbox"/>
ContactEmail	nvarchar(50)	<input checked="" type="checkbox"/>
ContactPhone	nvarchar(12)	<input checked="" type="checkbox"/>

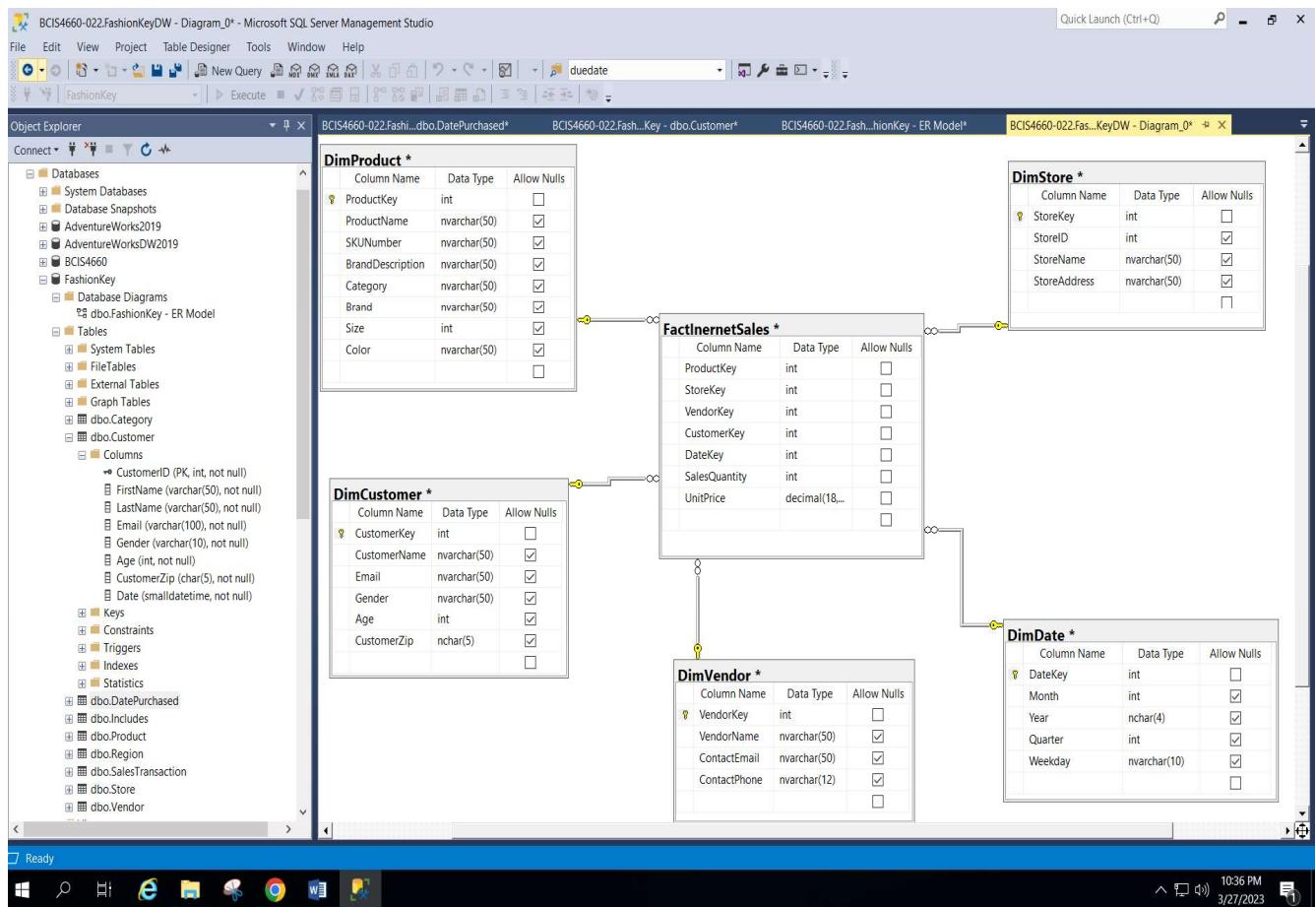
Column Name	Data Type	Allow Nulls
ProductKey	int	<input type="checkbox"/>
ProductName	nvarchar(50)	<input checked="" type="checkbox"/>
SKUNumber	nvarchar(50)	<input checked="" type="checkbox"/>
BrandDescription	nvarchar(50)	<input checked="" type="checkbox"/>
Category	nvarchar(50)	<input checked="" type="checkbox"/>
Brand	nvarchar(50)	<input checked="" type="checkbox"/>
Size	int	<input checked="" type="checkbox"/>
Color	nvarchar(50)	<input checked="" type="checkbox"/>

Column Name	Data Type	Allow Nulls
DateKey	int	<input type="checkbox"/>
Month	int	<input checked="" type="checkbox"/>
Year	nchar(4)	<input checked="" type="checkbox"/>
Quarter	int	<input checked="" type="checkbox"/>
Weekday	nvarchar(10)	<input checked="" type="checkbox"/>

SS 4: Created the dimension tables

Like in FashionKey, database diagram authorization was successfully executed on FashionKeyDW. Then looking at the ER Model, dimensions (who, what, where, when) were selected from Customer, Vendor, Product, Store and date. Dimension tables were created for all of them while assigning a Surrogate key to each of the dimension tables and the included columns with their respective datatypes.

5. Step 5:

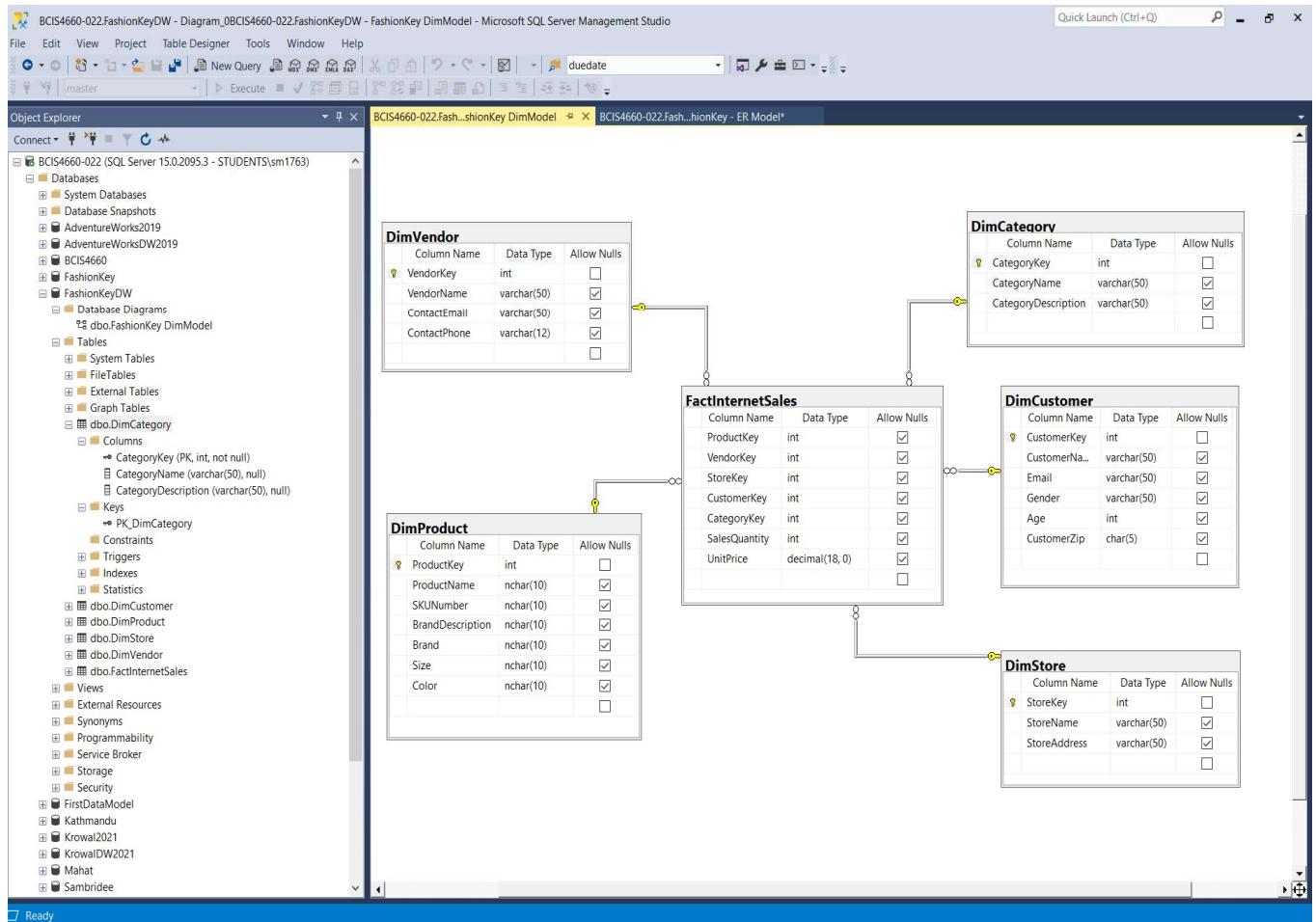


SS 5: Star schema with the fact table in the middle

The fact table is what is in the middle of the screenshot, that is connected to all other dimension tables. To create a fact table, just right click on the database diagram page and select create new table and include all the surrogate key from each of the dimension tables and the measurement metrics like the quantity and the price.

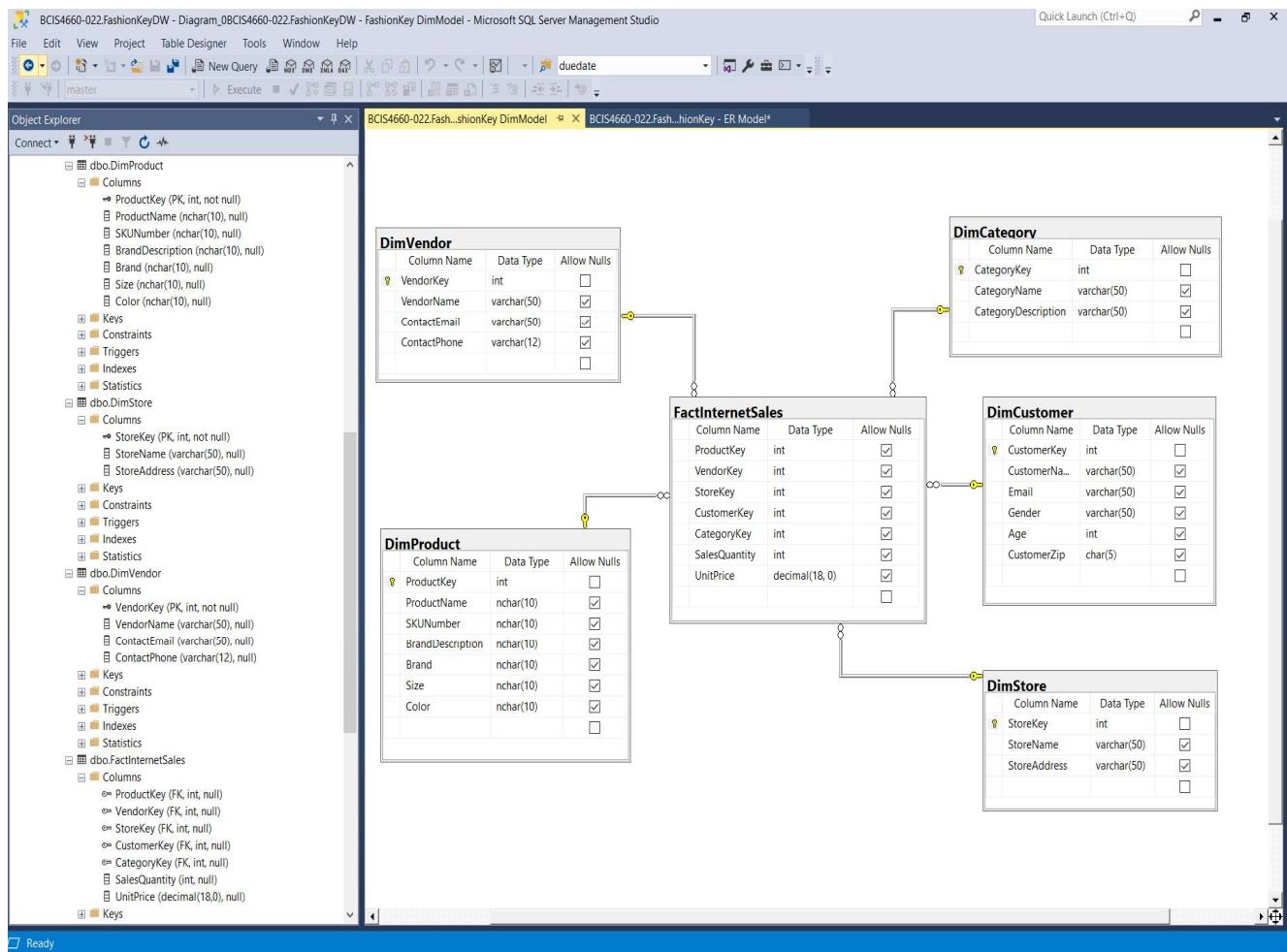
- A fact table contains numerical measurements like sales quantity and price while dimension tables contain descriptive information about the attributes like customer information, geographics, etc
- Fact tables are usually larger than dimension tables as fact tables contain dynamic information while dimension tables contain static information like description.
- The fact table's primary key is made up of foreign keys from each of the dimension tables, that joins the fact tables with dimension tables.

6. Step 6:

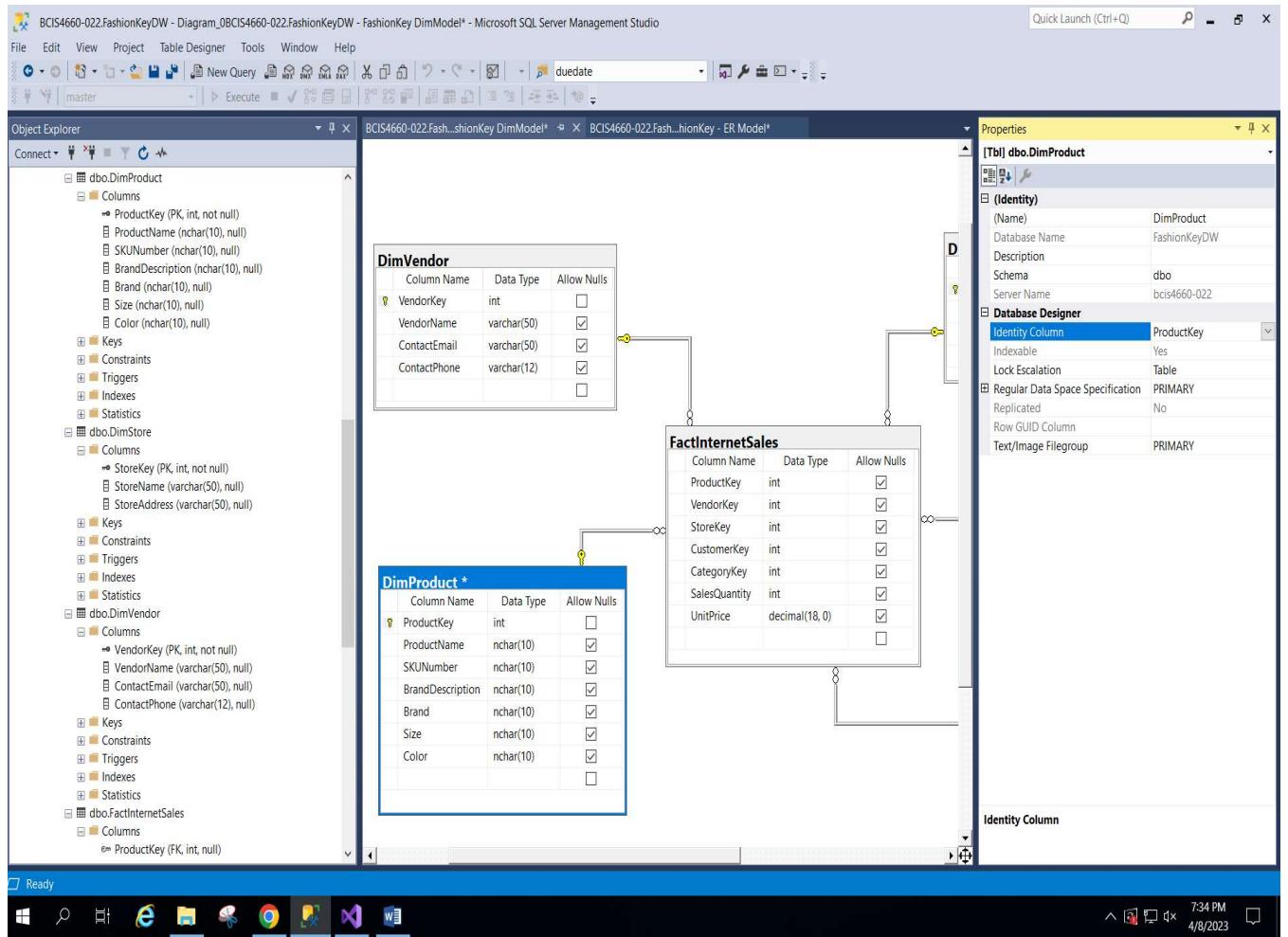


SS 1a: Dimensional Model

Database diagram showing the five dimension tables called DimVendor, DimProduct, DimCustomer, DimStore, and DimCategory and a fact table called FactInternetSales.

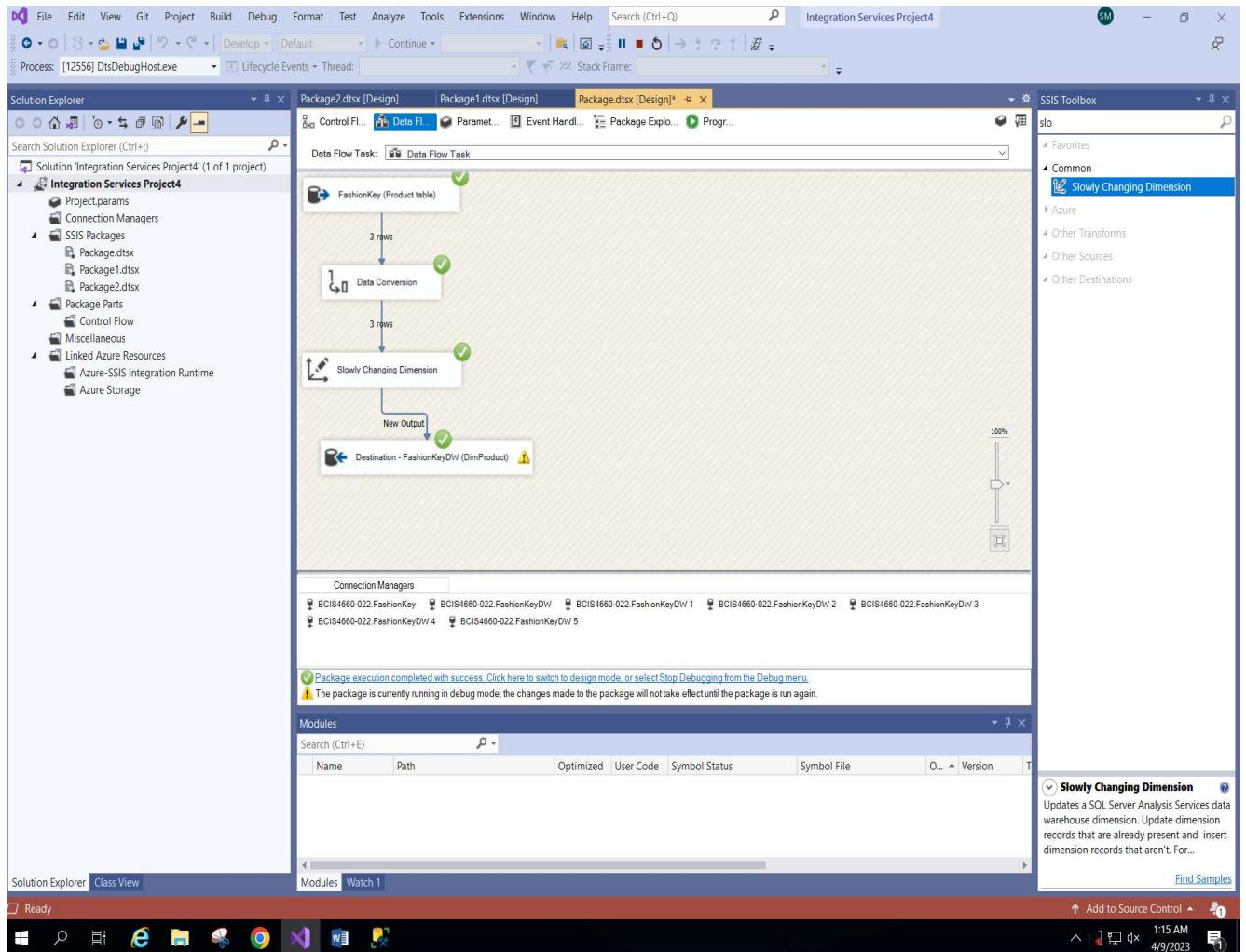


SS 1b. Object explorer showing the tables created



SS 1C : SSMS showing the Identity key property of the DimProduct table

7. Step 7:



SS 2A: ETL package to extract data from Product table to DimProduct table

The data from the operational database FashionKey table in the Product table was extracted using a ETL package which included tasks like data flow task. The source was connected, the data was converted to appropriate data types, slowly changing dimension was added and the destination was set to FashionKeyDW, DimProduct table.

SQLQuery5.sql - BCIS4660-022.FashionKeyDW (STUDENTS\sm1763 (60)) - Microsoft SQL Server Management Studio

File Edit View Query Project Tools Window Help

Quick Launch (Ctrl+Q)

FashionKeyDW

Object Explorer

SQLQuery5.sql - B...DENTS\sm1763 (60)

SQLQuery4.sql - B...DENTS\sm1763 (65)*

BCIS4660-022.Krow...2021 - Diagram_45*

BCIS4660-022.Adv...W2019 - Diagram_0*

```
***** Script for SelectTopNRows command from SSMS *****/  
SELECT TOP (1000) [ProductKey]  
    ,[ProductName]  
    ,[SKUNumber]  
    ,[Brand]  
    ,[Size]  
    ,[Color]  
    ,[ProductID]  
FROM [FashionKeyDW].[dbo].[DimProduct]
```

Results

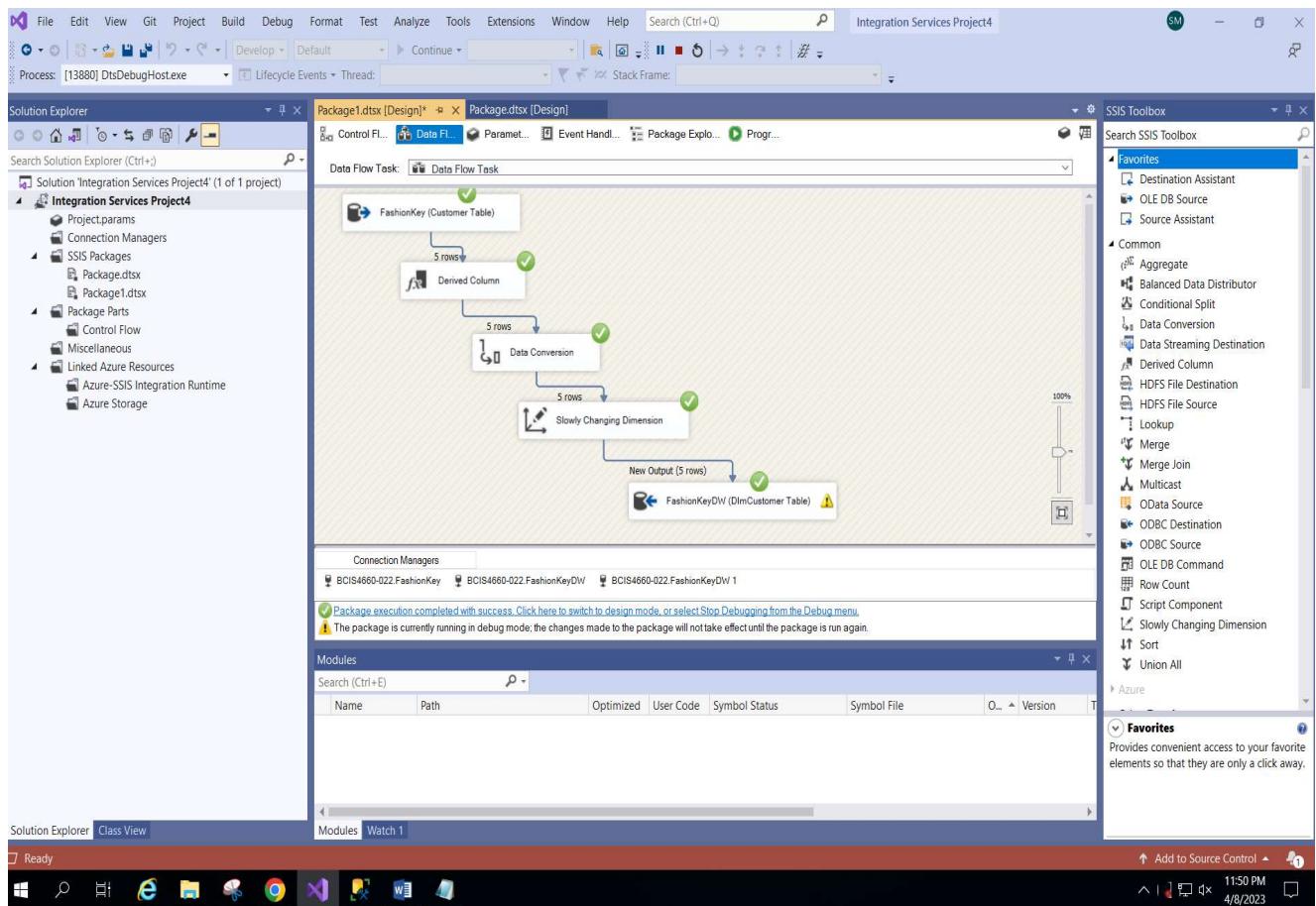
	ProductKey	ProductName	SKUNumber	Brand	Size	Color	ProductID
1	1	Product 1	SKU001	Brand 1	4	Red	10001
2	2	Product 2	SKU002	Brand 2	8	Blue	10002
3	3	Product 3	SKU003	Brand 3	12	Green	10003

Query executed successfully. BCIS4660-022 (15.0 RTM) | STUDENTS\sm1763 (60) | FashionKeyDW | 00:00:00 | 3 rows

Ln 1 Col 1 Ch 1 INS 10:49 PM 4/8/2023

SS 2B: SSMS showing the data loaded into the destination table

8. Step 8:



SS 3A: ETL package to extract data from Customer table to DimCustomer table

The data from the operational database FashionKey table in the Customer table was extracted using a ETL package which included tasks like data flow task. The source was connected, data was transformed (FirstName and LastName columns were merged into CustomerName column, the data was converted to appropriate data types, slowly changing dimension was added and the destination was set to FashionKeyDW, DiCustomer table.

SQLQuery6.sql - BCIS4660-022.FashionKeyDW (STUDENTS\sm1763 (54)) - Microsoft SQL Server Management Studio

File Edit View Query Project Tools Window Help

FashionKeyDW New Query Execute

SQLQuery6.sql - B...DENTS\sm1763 (54) SQLQuery5.sql - B...DENTS\sm1763 (60) SQLQuery4.sql - B...DENTS\sm1763 (65)* BCIS4660-022.Fash...hionKey DimModel*

Object Explorer

Connect AdventureWorksDW2019 BCIS4660 FashionKey Database Diagrams Tables Views External Resources Synonyms Programmability Service Broker Storage Security FashionKeyDW Database Diagrams Tables System Tables FileTables External Tables Graph Tables dbo.DimCategory dbo.DimCustomer dbo.DimProduct dbo.DimStore dbo.DimVendor dbo.FactInternetSales Views External Resources Synonyms Programmability Service Broker Storage Security FirstDataModel Kathmandu Krowal2021 Mahat Sambridee

SQLQuery6.sql - B...DENTS\sm1763 (54)

```
***** Script for SelectTopNRows command from SSMS *****/
SELECT TOP (1000) [CustomerKey]
    ,[CustomerName]
    ,[Email]
    ,[Gender]
    ,[Age]
    ,[CustomerZip]
FROM [FashionKeyDW].[dbo].[DimCustomer]
```

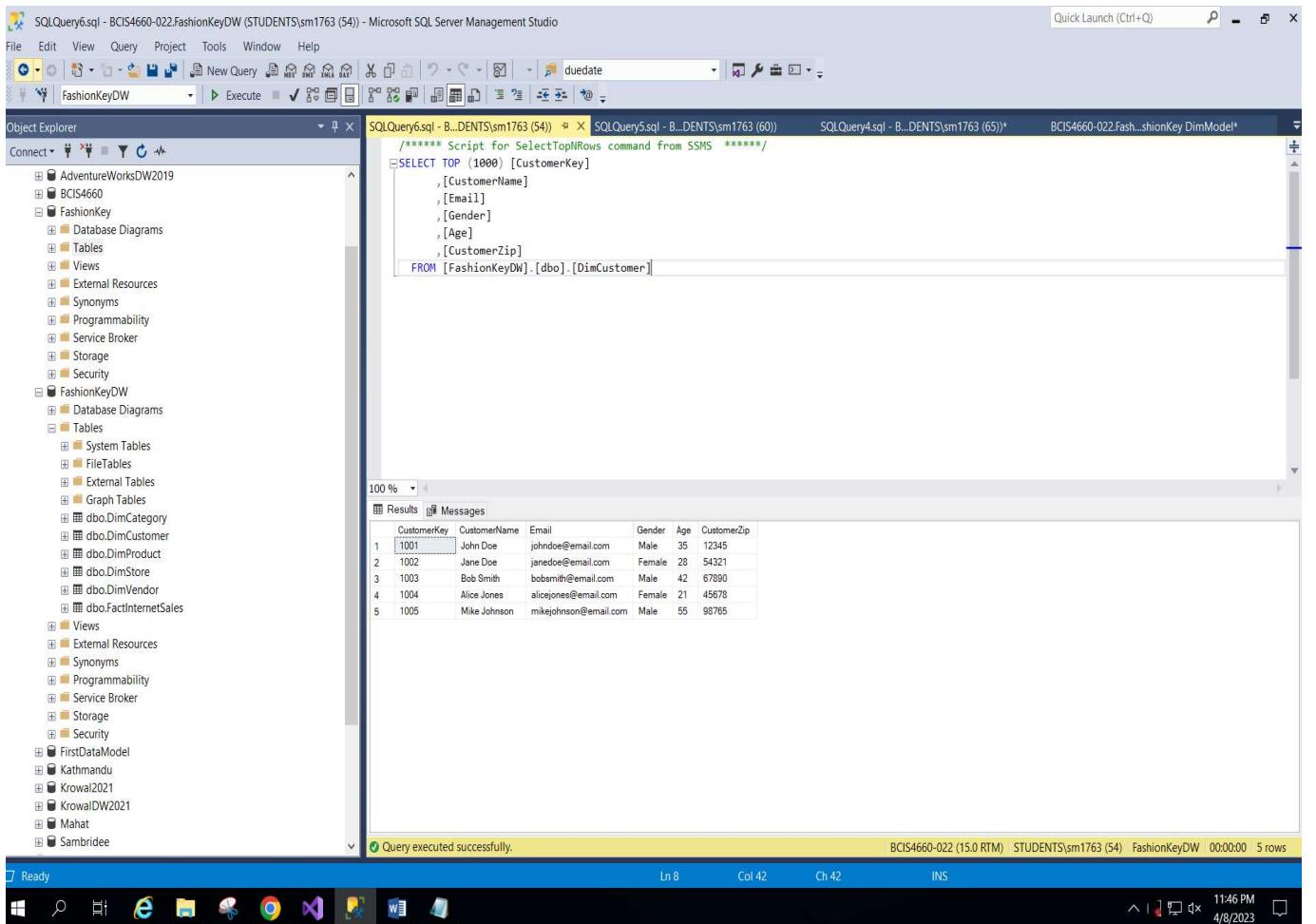
Results Messages

	CustomerKey	CustomerName	Email	Gender	Age	CustomerZip
1	1001	John Doe	johndoe@email.com	Male	35	12345
2	1002	Jane Doe	janedoe@email.com	Female	28	54321
3	1003	Bob Smith	bobsmith@email.com	Male	42	67890
4	1004	Alice Jones	alicejones@email.com	Female	21	45678
5	1005	Mike Johnson	mikejohnson@email.com	Male	55	98765

Query executed successfully.

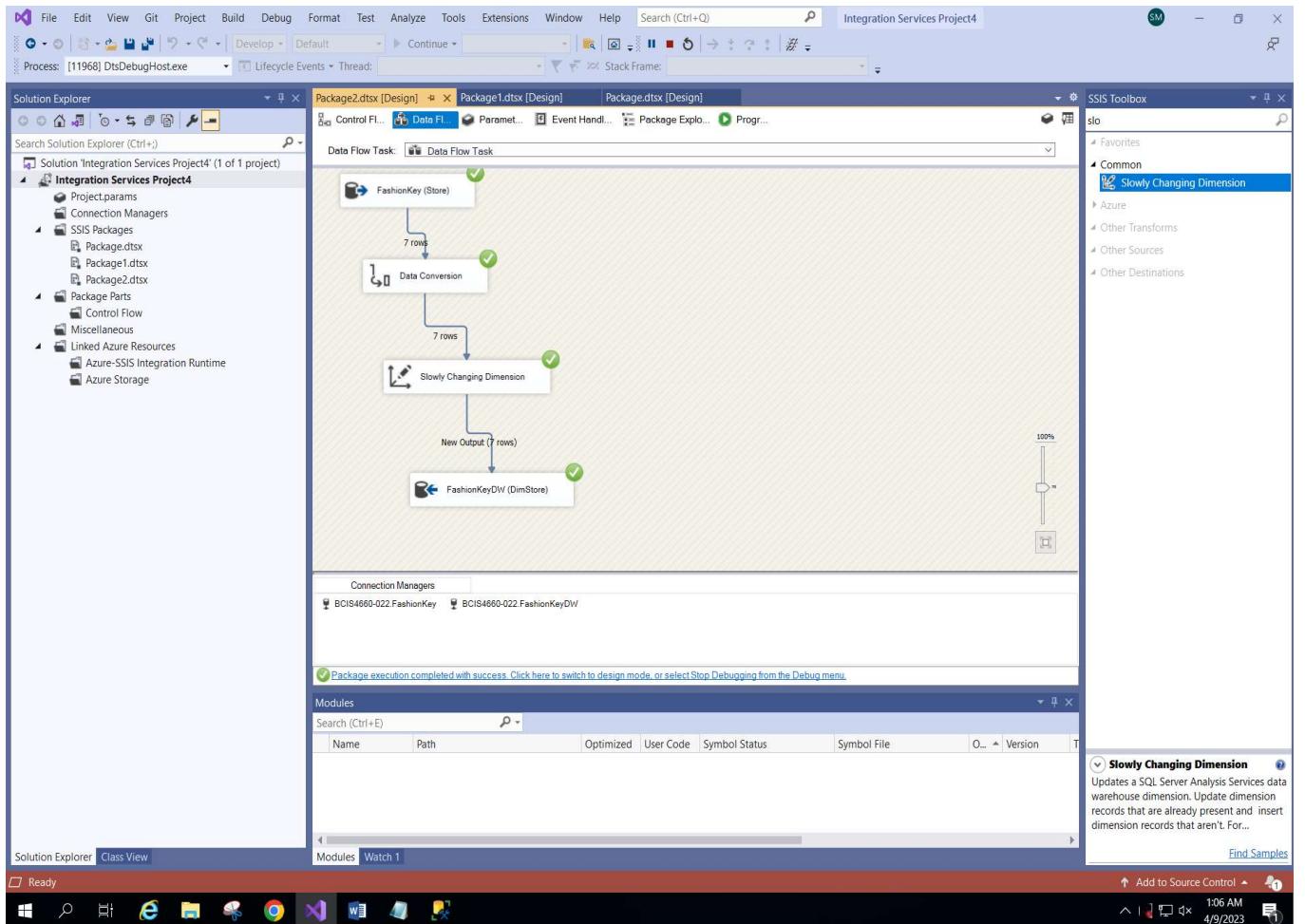
BCIS4660-022 (15.0 RTM) STUDENTS\sm1763 (54) FashionKeyDW 00:00:00 5 rows

Ready Ln 8 Col 42 Ch 42 INS 11:46 PM 4/8/2023



SS 3B: SSMS showing the data loaded from Customer table into the destination table (DimCustomer)

9. Step 9:



SS 4A: ETL package to extract data from Store table to DimStore table

The data from the operational database FashionKey table in the Store table was extracted using a ETL package which included tasks like data flow task. The source was connected, the data was converted to appropriate data types, slowly changing dimension was added and the destination was set to FashionKeyDW, DimStore table.

SQLQuery1.sql - BCIS4660-022.FashionKeyDW (STUDENTS\sm1763 (66)) - Microsoft SQL Server Management Studio

File Edit View Query Project Tools Window Help

New Query Execute

Object Explorer

FashionKeyDW

Script for SelectTopNRows command from SSMS

```
***** Script for SelectTopNRows command from SSMS *****/
SELECT TOP (1000) [StoreKey]
    ,[StoreName]
    ,[StoreAddress]
    ,[StoreID]
    ,[StoreZip]
FROM [FashionKeyDW].[dbo].[DimStore]
```

Results Messages

	StoreKey	StoreName	StoreAddress	StoreID	StoreZip
1	1	ABC Store	123 Main St	123	12345
2	2	DEF Store	456 Oak Ave	211	23456
3	3	GHI Store	789 Pine St	345	34567
4	4	JKL Store	321 Elm St	422	45678
5	5	MNO Store	654 Maple Ave	521	56789
6	6	PQR Store	987 Cedar St	634	67890
7	7	STU Store	246 Walnut Ave	756	78901

Query executed successfully.

BCIS4660-022 (15.0 RTM) STUDENTS\sm1763 (66) FashionKeyDW 00:00:00 7 rows

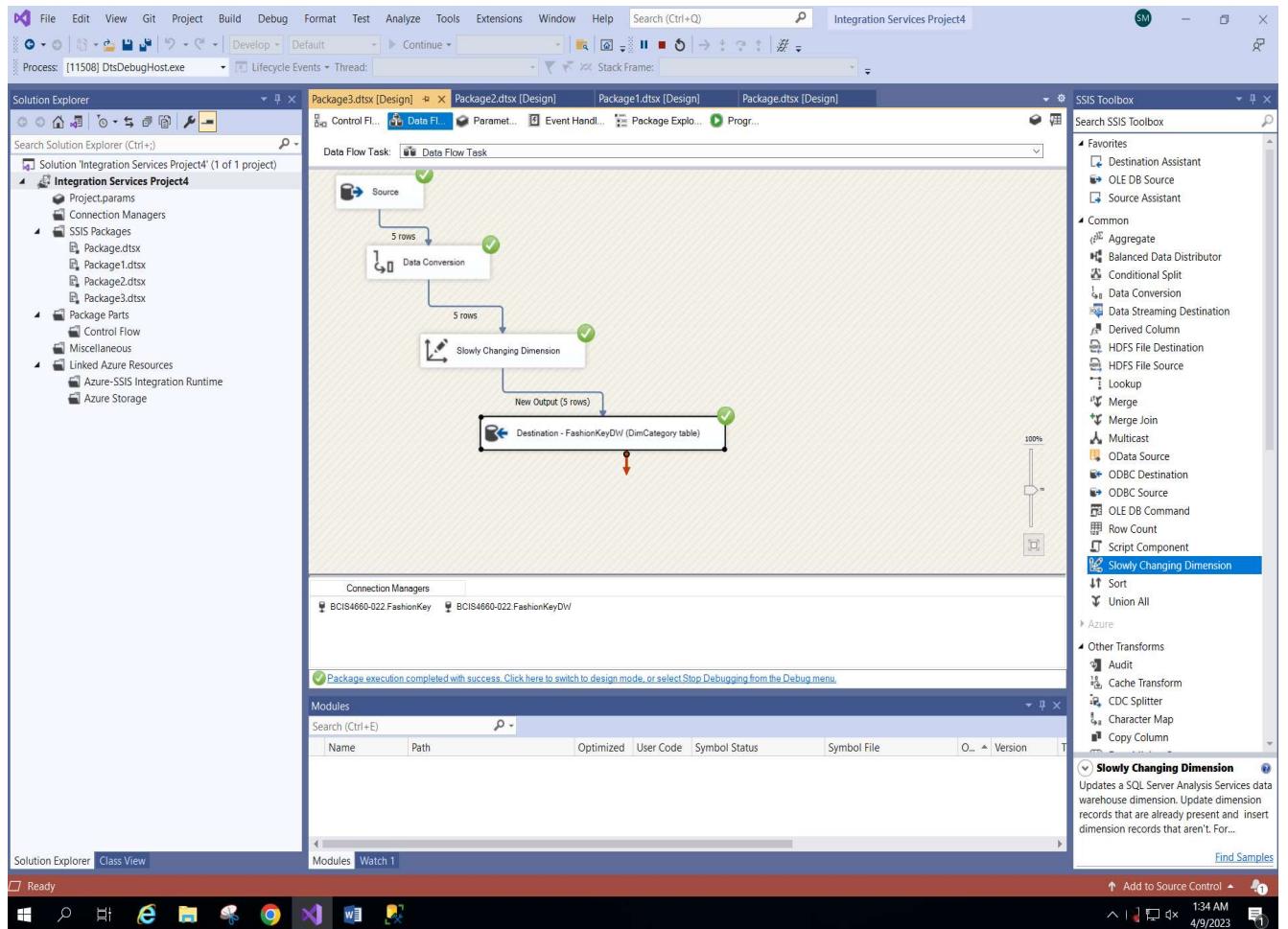
Ready

Ln 1 Col 1 Ch 1 INS

1:07 AM 4/9/2023

SS 4B: SSMS showing the data loaded from Store table into the destination table (DimStore)

10. Step 10:



SS 5A: ETL package to extract data from Category table to DimCategory table

The data from the operational database **FashionKey** table in the **Category** table was extracted using a ETL package which included tasks like data flow task. The source was connected, the data was converted to appropriate data types, slowly changing dimension was added and the destination was set to **FashionKeyDW**, **DimCategory** table.

SQLQuery2.sql - BCIS4660-022.FashionKeyDW (STUDENTS\sm1763 (58)) - Microsoft SQL Server Management Studio

File Edit View Query Project Tools Window Help

FashionKeyDW Quick Launch (Ctrl+Q) ↗

Object Explorer

SQLQuery2.sql - B...DENTS\sm1763 (58) BCIS4660-022.Fash...hionKey - ER Model ~vsED18.sql - BC...DENTS\sm1763 (56)*

```
***** Script for SelectTopNRows command from SSMS *****
SELECT TOP (1000) [CategoryKey]
      ,[CategoryName]
      ,[CategoryDescription]
      ,[CategoryID]
   FROM [FashionKeyDW].[dbo].[DimCategory]
```

Results Messages

	CategoryKey	CategoryName	CategoryDescription	CategoryID
1	1	Sweaters	Warm and cozy sweaters for men and women	1005
2	2	T-Shirts	Casual wear for men and women	3005
3	3	Jeans	Denim pants for men and women	4005
4	4	Dresses	Formal and casual dresses for women	5005
5	5	Suits	Formal suits for men	6005

Query executed successfully.

BCIS4660-022 (15.0 RTM) | STUDENTS\sm1763 (58) | FashionKeyDW | 00:00:00 | 5 rows

Ready Ln 1 Col 1 Ch 1 INS 135 AM 4/9/2023

SS 5B: SSMS showing the data loaded from Category table into the destination table (DimCategory)

Exercise T-SQL

The screenshot shows the Microsoft SQL Server Management Studio (SSMS) interface. In the Object Explorer on the left, the database 'AdventureWorks2019' is selected, displaying various tables and their columns. The 'Production' schema is expanded, showing tables like Product, ProductCategory, and ProductDescription. In the center, a query window titled 'SQLQuery3.sql' is open, containing the following T-SQL code:

```
USE [AdventureWorks2019]
GO
SELECT MAX(ListPrice) AS HighestListPrice
FROM Production.Product
```

The results pane shows a single row with the value '3578.27' under the column 'HighestListPrice'. At the bottom of the screen, the status bar indicates 'Query executed successfully.' and 'BCIS4660-022 (15.0 RTM) STUDENTS\sm1763 (54) AdventureWorks2019 00:00:00 1 rows'.

SS 7: Highest ListPrice in AdventureWorks2019

Executed in SSMS using t-sql

SQLQuery3.sql - BCIS4660-022.AdventureWorks2019 (STUDENTS\sm1763 (54))* - Microsoft SQL Server Management Studio

File Edit View Query Project Tools Window Help

AdventureWorks2019 Execute

Object Explorer

SQLQuery3.sql - B...DENTS\sm1763 (54)* SQLQuery2.sql - B...DENTS\sm1763 (58)

```
USE [AdventureWorks2019]
GO

SELECT * FROM Person.Person
WHERE LastName LIKE 'Co%'
```

Results Messages

BusinessEntityID	PersonTypeID	NameStyle	Title	FirstName	MiddleName	LastName	Suffix	EmailPromotion	AdditionalContactInfo	Demographics	rowguid
1	711	SC	0	Ms.	Tearria	M.	Cobb	NULL	2	NULL	<IndividualSurvey.xm...>
2	713	SC	0	Ms.	Connie	L.	Coffman	NULL	0	NULL	<IndividualSurvey.xm...>
3	715	SC	0	Ms.	Jeanette	R.	Cole	NULL	2	NULL	<IndividualSurvey.xm...>
4	5488	IN	0	NULL	Aaron		Coleman	NULL	0	NULL	<IndividualSurvey.xm...>
5	12798	IN	0	Ms.	Abigail	C	Coleman	NULL	0	NULL	<IndividualSurvey.xm...>
6	3961	IN	0	NULL	Abigail	E	Coleman	NULL	0	NULL	<IndividualSurvey.xm...>
7	12906	IN	0	NULL	Aidan	L	Coleman	NULL	1	NULL	<IndividualSurvey.xm...>
8	2805	IN	0	NULL	Alexandra	A	Coleman	NULL	0	NULL	<IndividualSurvey.xm...>
9	7608	IN	0	NULL	Alexandra	C	Coleman	NULL	0	NULL	<IndividualSurvey.xm...>
10	7102	IN	0	NULL	Alexia	E	Coleman	NULL	1	NULL	<IndividualSurvey.xm...>
11	12751	IN	0	NULL	Alexis	M	Coleman	NULL	0	NULL	<IndividualSurvey.xm...>
12	20189	IN	0	NULL	Alyssa	C	Coleman	NULL	0	NULL	<IndividualSurvey.xm...>
13	3034	IN	0	NULL	Amanda	NULL	Coleman	NULL	1	NULL	<IndividualSurvey.xm...>
14	7438	IN	0	NULL	Ana	L	Coleman	NULL	0	NULL	<IndividualSurvey.xm...>
15	7654	IN	0	NULL	Angela	M	Coleman	NULL	1	NULL	<IndividualSurvey.xm...>
16	7138	IN	0	NULL	Angelica	NULL	Coleman	NULL	2	NULL	<IndividualSurvey.xm...>
17	3662	IN	0	NULL	Anna	NULL	Coleman	NULL	1	NULL	<IndividualSurvey.xm...>
18	7698	IN	0	NULL	Annina	NULL	Coleman	NULL	1	NULL	<IndividualSurvey.xm...>
19	12572	IN	0	NULL	Ashley	NULL	Coleman	NULL	2	NULL	<IndividualSurvey.xm...>
20	6242	IN	0	NULL	Austin	V	Coleman	NULL	0	NULL	<IndividualSurvey.xm...>
21	6699	IN	0	NULL	Benjamin	F	Coleman	NULL	0	NULL	<IndividualSurvey.xm...>

Query executed successfully.

BCIS4660-022 (15.0 RTM) STUDENTS\sm1763 (54) AdventureWorks2019 00:00:00 512 rows

Ready

147 AM 4/9/2023

SS 8: LastNames in Person.Person table that start with the letters 'Co'

The screenshot shows the Microsoft SQL Server Management Studio (SSMS) interface. The title bar reads "SQLQuery3.sql - BCIS4660-022.AdventureWorks2019 (STUDENTS\sm1763 (54))* - Microsoft SQL Server Management Studio". The main window has two panes: "Object Explorer" on the left and "SQL Query Editor" on the right.

In the Object Explorer, under the "AdventureWorks2019" database, the "Person" schema is expanded, showing tables like Person, PersonPhone, PersonPhoneType, PersonStateProvince, Production.BillOfMaterials, Production.Culture, Production.Document, Production.Illustration, Production.Location, and Production.Product. The "Production.Product" table is selected, and its columns (ProductID, Name, ProductNumber, MakeFlag, FinishedGoodsFlag, Color, SafetyStockLevel, ReorderPoint, StandardCost, ListPrice, Size, SizeUnitMeasureCode, WeightUnitMeasureCode, DaysToManufacture, ProductLine, Class, Style, ProductSubcategoryID, ProductModelID, SellStartDate, SellEndDate, DiscontinuedDate, rowguid,ModifiedDate) are listed.

The SQL Query Editor contains the following T-SQL code:

```
USE [AdventureWorks2019]
GO
SELECT COUNT(*) AS NumberOfRowsReturned FROM Person.Person
WHERE LastName LIKE 'Co%'
```

The results pane shows the output of the query:

NumberOfRowsReturned
512

Below the results, a message states "Query executed successfully." The status bar at the bottom right shows "BCIS4660-022 (15.0 RTM) STUDENTS\sm1763 (54) AdventureWorks2019 00:00:00 | 1 rows". The system tray at the bottom indicates the date and time as "4/9/2023 145 AM".

SS 9: Showing the total number of rows displayed with the above SS 8 SQL

Number of rows returned = 512