**Problem Statement**

Scenario: Adventureworks is a Bicycle company that sells bicycles worldwide. They work with a partner to handle their internet sales. Their partner exports the sales history on a monthly basis to Amazon S3 storage. The format of the files is Parquet and contains following attributes.

root

|-- OrderDate: date (nullable = true)

|-- ProductId: integer (nullable = true)

|-- ProductSubcategoryId: integer (nullable = true)

|-- DueDate: date (nullable = true)

|-- ShipDate: date (nullable = true)

|-- Promotion: integer (nullable = true)

|-- Currency: integer (nullable = true)

|-- SalesTerritory: struct (nullable = false)

| |-- SalesTerritoryId: integer (nullable = true)

| |-- SalesTerritoryAlternateKey: integer (nullable = true)

| |-- SalesTerritoryRegion: string (nullable = true)

| |-- SalesTerritoryCountry: string (nullable = true)

| |-- SalesTerritoryGroup: string (nullable = true)

|-- SalesOrderNumber: string (nullable = true)

|-- SalesOrderLineNumber: integer (nullable = true)

|-- RevisionNumber: integer (nullable = true)

|-- OrderQuantity: integer (nullable = true)

|-- UnitPrice: decimal(19,4) (nullable = true)

|-- ExtendedAmount: decimal(19,4) (nullable = true)

|-- UnitPriceDiscountPct: double (nullable = true)

|-- DiscountAmount: double (nullable = true)

|-- ProductStandardCost: decimal(19,4) (nullable = true)

|-- TotalProductCost: decimal(19,4) (nullable = true)

|-- SalesAmount: decimal(19,4) (nullable = true)

|-- TaxAmt: decimal(19,4) (nullable = true)

|-- Freight: decimal(19,4) (nullable = true)

|-- CarrierTrackingNumber: string (nullable = true)

|-- CustomerPONumber: string (nullable = true)

|-- Customer: struct (nullable = false)

| |-- CustomerId: integer (nullable = true)

| |-- Geography: struct (nullable = false)

| | |-- GeographyId: integer (nullable = true)

| | |-- City: string (nullable = true)

| | |-- StateProvinceCode: string (nullable = true)

| | |-- StateProvinceName: string (nullable = true)

| | |-- CountryRegionCode: string (nullable = true)

| | |-- EnglishCountryRegionName: string (nullable = true)

| | |-- SpanishCountryRegionName: string (nullable = true)

| | |-- FrenchCountryRegionName: string (nullable = true)

| | |-- PostalCode: string (nullable = true)

| | |-- SalesTerritoryId: integer (nullable = true)

| |-- CustomerAlternateId: string (nullable = true)

| |-- Title: string (nullable = true)

| |-- FirstName: string (nullable = true)

| |-- MiddleName: string (nullable = true)

| |-- LastName: string (nullable = true)

| |-- NameStyle: boolean (nullable = true)

| |-- EmailAddress: string (nullable = true)

| |-- HouseOwnerFlag: string (nullable = true)

| |-- NumberCarsOwned: integer (nullable = true)

| |-- AddressLine1: string (nullable = true)

| |-- AddressLine2: string (nullable = true)

| |-- Phone: string (nullable = true)

The folder structure for these files is shown below:

microsoft-airlift-azure-data-factory-ingestion

|-internetSales

||-05

|||OrderDate=2013-05-08

Adventureworks is also introducing new products to their portfolio in the upcoming months. They exported an entire catalog of their products to a shared BLOB storage location. The format of the files is delimited text. The structure of the table is shown below.

**Table Schema**

[ProductKey] [int] NOT NULL,

[ProductAlternateKey] [nvarchar](25) NULL,

[ProductSubcategoryKey] [int] NULL,

[WeightUnitMeasureCode] [nchar](3) NULL,

[SizeUnitMeasureCode] [nchar](3) NULL,

[EnglishProductName] [nvarchar](50) NOT NULL,

[SpanishProductName] [nvarchar](50) NULL,

[FrenchProductName] [nvarchar](50) NULL,

[StandardCost] [money] NULL,

[FinishedGoodsFlag] [bit] NOT NULL,

[Color] [nvarchar](15) NOT NULL,

[SafetyStockLevel] [smallint] NULL,

[ReorderPoint] [smallint] NULL,

[ListPrice] [money] NULL,

[Size] [nvarchar](50) NULL,

[SizeRange] [nvarchar](50) NULL,

[Weight] [float] NULL,

[DaysToManufacture] [int] NULL,

[ProductLine] [nchar](2) NULL,

[DealerPrice] [money] NULL,

[Class] [nchar](2) NULL,

[Style] [nchar](2) NULL,

[ModelName] [nvarchar](50) NULL,

[EnglishDescription] [nvarchar](400) NULL,

[FrenchDescription] [nvarchar](400) NULL,

[ChineseDescription] [nvarchar](400) NULL,

[ArabicDescription] [nvarchar](400) NULL,

[HebrewDescription] [nvarchar](400) NULL,

[ThaiDescription] [nvarchar](400) NULL,

[GermanDescription] [nvarchar](400) NULL,

[JapaneseDescription] [nvarchar](400) NULL,

[TurkishDescription] [nvarchar](400) NULL,

[StartDate] [datetime] NULL,

[EndDate] [datetime] NULL,

[Status] [nvarchar](7) NULL

Storage Account: demodatacasey

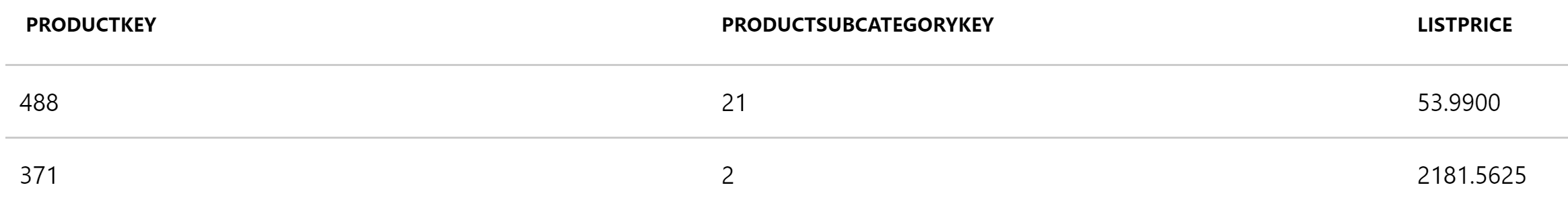
Container: demodata

Key starts with (on ribbon): ‘chYJlpTz…..’

File delimiter '|'

They would like to use this data to draw meaningful insights. They approached you to build an end to end solution that addresses data ingestion, cleansing, preparation, analysis, and orchestration.

You need build a Modern Data Warehouse (MDW) solution that meets the following criteria:

1. Ingest the Parquet data from S3 for the month of May (05) into an Azure BLOB container in your Azure subscription.
2. Perform necessary transformations to the ingested data and load the data into a table named ‘FactInternetSalesStg’ in your SQLDW database.
3. Load the products dump file into a table named ‘DimProductStg’ in the most efficient way possible.
4. Write a query to join both tables above to generate a report that looks similar to below:
5. Create a reporting user named ‘rptusr’ with absolute minimum privileges to execute the query above.
6. The query observed to be not performing optimally. Investigate the reasons for this performance issue and make necessary changes to the table structure so that query can perform optimally.

Bonus:

There is data for additional months stored in S3 storage. Develop your Azure infrastructure to ingest this data from each month into it’s corresponding folder in your Azure BLOB storage, Transform and load this data into FactInternetSalesStg table.

Use the following resources for guidance:

<https://docs.microsoft.com/en-us/azure/sql-data-warehouse/load-data-wideworldimportersdw>

<https://docs.microsoft.com/en-us/azure/data-factory/>

<https://docs.microsoft.com/en-us/azure/azure-databricks/>

<https://docs.microsoft.com/en-us/azure/sql-data-warehouse/sql-data-warehouse-manage-monitor>

<https://docs.microsoft.com/en-us/azure/sql-database/sql-database-manage-logins?toc=/azure/sql-data-warehouse/toc.json>

Hints:

Minimize any Data Movement during queries.