**Creating dimension tables**

CREATE TABLE dbo.DimCustomer

(

CustomerKey INT IDENTITY NOT NULL,

CustomerAlternateKey NVARCHAR(15) NULL,

CustomerName NVARCHAR(80) NOT NULL,

EmailAddress NVARCHAR(50) NULL,

Phone NVARCHAR(25) NULL,

StreetAddress NVARCHAR(100),

City NVARCHAR(20),

PostalCode NVARCHAR(10),

CountryRegion NVARCHAR(20)

)

WITH

(

DISTRIBUTION = REPLICATE,

CLUSTERED COLUMNSTORE INDEX

);

CREATE TABLE dbo.DimGeography

(

GeographyKey INT IDENTITY NOT NULL,

GeographyAlternateKey NVARCHAR(10) NULL,

StreetAddress NVARCHAR(100),

City NVARCHAR(20),

PostalCode NVARCHAR(10),

CountryRegion NVARCHAR(20)

)

WITH

(

DISTRIBUTION = REPLICATE,

CLUSTERED COLUMNSTORE INDEX

);

CREATE TABLE dbo.DimCustomer

(

CustomerKey INT IDENTITY NOT NULL,

CustomerAlternateKey NVARCHAR(15) NULL,

GeographyKey INT NULL,

CustomerName NVARCHAR(80) NOT NULL,

EmailAddress NVARCHAR(50) NULL,

Phone NVARCHAR(25) NULL

)

WITH

(

DISTRIBUTION = REPLICATE,

CLUSTERED COLUMNSTORE INDEX

);

### **Time dimension tables**

CREATE TABLE dbo.DimDate

(

DateKey INT NOT NULL,

DateAltKey DATETIME NOT NULL,

DayOfMonth INT NOT NULL,

DayOfWeek INT NOT NULL,

DayName NVARCHAR(15) NOT NULL,

MonthOfYear INT NOT NULL,

MonthName NVARCHAR(15) NOT NULL,

CalendarQuarter INT NOT NULL,

CalendarYear INT NOT NULL,

FiscalQuarter INT NOT NULL,

FiscalYear INT NOT NULL

)

WITH

(

DISTRIBUTION = REPLICATE,

CLUSTERED COLUMNSTORE INDEX

);

## Creating fact tables

The following code example creates a hypothetical fact table named **FactSales** that is related to multiple dimensions through key columns (date, customer, product, and store)

CREATE TABLE dbo.FactSales

(

OrderDateKey INT NOT NULL,

CustomerKey INT NOT NULL,

ProductKey INT NOT NULL,

StoreKey INT NOT NULL,

OrderNumber NVARCHAR(10) NOT NULL,

OrderLineItem INT NOT NULL,

OrderQuantity SMALLINT NOT NULL,

UnitPrice DECIMAL NOT NULL,

Discount DECIMAL NOT NULL,

Tax DECIMAL NOT NULL,

SalesAmount DECIMAL NOT NULL

)

WITH

(

DISTRIBUTION = HASH(OrderNumber),

CLUSTERED COLUMNSTORE INDEX

);

## Creating staging tables

The following code example creates a staging table for product data that will ultimately be loaded into a dimension table:

CREATE TABLE dbo.StageProduct

(

ProductID NVARCHAR(10) NOT NULL,

ProductName NVARCHAR(200) NOT NULL,

ProductCategory NVARCHAR(200) NOT NULL,

Color NVARCHAR(10),

Size NVARCHAR(10),

ListPrice DECIMAL NOT NULL,

Discontinued BIT NOT NULL

)

WITH

(

DISTRIBUTION = ROUND\_ROBIN,

CLUSTERED COLUMNSTORE INDEX

);

### **Using external tables**

-- External data source links to data lake location

CREATE EXTERNAL DATA SOURCE StagedFiles

WITH (

LOCATION = 'https://mydatalake.blob.core.windows.net/data/stagedfiles/'

);

GO

-- External format specifies file format

CREATE EXTERNAL FILE FORMAT ParquetFormat

WITH (

FORMAT\_TYPE = PARQUET,

DATA\_COMPRESSION = 'org.apache.hadoop.io.compress.SnappyCodec'

);

GO

-- External table references files in external data source

CREATE EXTERNAL TABLE dbo.ExternalStageProduct

(

ProductID NVARCHAR(10) NOT NULL,

ProductName NVARCHAR(200) NOT NULL,

ProductCategory NVARCHAR(200) NOT NULL,

Color NVARCHAR(10),

Size NVARCHAR(10),

ListPrice DECIMAL NOT NULL,

Discontinued BIT NOT NULL

)

WITH

(

DATA\_SOURCE = StagedFiles,

LOCATION = 'products/\*.parquet',

FILE\_FORMAT = ParquetFormat

);

GO

# Load data warehouse tables

COPY INTO dbo.StageProducts

(ProductID, ProductName, ProductCategory, Color, Size, ListPrice, Discontinued)

FROM 'https://mydatalake.blob.core.windows.net/data/stagedfiles/products/\*.parquet'

WITH

(

FILE\_TYPE = 'PARQUET',

MAXERRORS = 0,

IDENTITY\_INSERT = 'OFF'

);

**QUERYING A DATABASE**

SELECT ProductCategory,

ProductName,

ListPrice,

ROW\_NUMBER() OVER

(PARTITION BY ProductCategory ORDER BY ListPrice DESC) AS RowNumber,

RANK() OVER

(PARTITION BY ProductCategory ORDER BY ListPrice DESC) AS Rank,

DENSE\_RANK() OVER

(PARTITION BY ProductCategory ORDER BY ListPrice DESC) AS DenseRank,

NTILE(4) OVER

(PARTITION BY ProductCategory ORDER BY ListPrice DESC) AS Quartile

FROM dbo.DimProduct

ORDER BY ProductCategory;