

**DATA SCHEMA PROCEDURES**

Pertaining the report for the data star schema shown in the uploaded image, these are the key procedures required for building the star schema in SQL and ensuring data cleanliness.

1. **Understanding the Schema**

From the uploaded image, the star schema revolves around sales data from three years (2015, 2016, 2017), with customers, products, territories, and returns serving as dimensions. There is also a calendar table that connects the sales data by date.

**The central fact tables are:**

AdventureWorks\_Sales\_2015

AdventureWorks\_Sales\_2016

AdventureWorks\_Sales\_2017

AdventureWorks\_Returns

**The dimension tables are:**

AdventureWorks\_Products

AdventureWorks\_Customers

AdventureWorks\_Territories

AdventureWorks\_Product\_Subcategories

AdventureWorks\_Product\_Categories

AdventureWorks\_Calendar

2. **SQL Scripts to Build the Star Schema**

Step 1: Create Dimension Tables

Products Dimension

CREATE TABLE AdventureWorks\_Products (

ProductKey INTEGER PRIMARY KEY,

ProductSubcategoryKey INTEGER,

ProductSKU TEXT,

ProductName TEXT,

ModelName TEXT,

ProductDescription TEXT,

ProductColor TEXT,

ProductSize TEXT,

ProductCost REAL,

ProductPrice REAL

);

Customers Dimension

CREATE TABLE AdventureWorks\_Customers (

CustomerKey INTEGER PRIMARY KEY,

Prefix TEXT,

FirstName TEXT,

LastName TEXT,

BirthDate TEXT,

MaritalStatus TEXT,

Gender TEXT,

EmailAddress TEXT,

AnnualIncome REAL,

TotalChildren INTEGER,

EducationLevel TEXT,

Occupation TEXT,

HomeOwner TEXT

);

Territories Dimension

CREATE TABLE AdventureWorks\_Territories (

SalesTerritoryKey INTEGER PRIMARY KEY,

Region TEXT,

Country TEXT,

TerritoryGroup TEXT

);

Product Subcategories Dimension

CREATE TABLE AdventureWorks\_Product\_Subcategories (

ProductSubcategoryKey INTEGER PRIMARY KEY,

SubcategoryName TEXT,

ProductCategoryKey INTEGER

);

Product Categories Dimension

CREATE TABLE AdventureWorks\_Product\_Categories (

ProductCategoryKey INTEGER PRIMARY KEY,

CategoryName TEXT

);

Calendar Dimension

CREATE TABLE AdventureWorks\_Calendar (

Date TEXT PRIMARY KEY

);

Step 2: Creating Fact Tables

Sales Fact Table

CREATE TABLE AdventureWorks\_Sales\_2015 (

OrderDate TEXT,

StockDate TEXT,

OrderNumber TEXT,

ProductKey INTEGER,

CustomerKey INTEGER,

TerritoryKey INTEGER,

OrderLineItem INTEGER,

OrderQuantity INTEGER,

FOREIGN KEY (ProductKey) REFERENCES AdventureWorks\_Products(ProductKey),

FOREIGN KEY (CustomerKey) REFERENCES AdventureWorks\_Customers(CustomerKey),

FOREIGN KEY (TerritoryKey) REFERENCES AdventureWorks\_Territories(SalesTerritoryKey)

);

The same applies for AdventureWorks\_Sales\_2016 and AdventureWorks\_Sales\_2017.

Returns Fact Table

CREATE TABLE AdventureWorks\_Returns (

ReturnDate TEXT,

ProductKey INTEGER,

CustomerKey INTEGER,

TerritoryKey INTEGER,

ReturnQuantity INTEGER,

FOREIGN KEY (ProductKey) REFERENCES AdventureWorks\_Products(ProductKey),

FOREIGN KEY (CustomerKey) REFERENCES AdventureWorks\_Customers(CustomerKey),

FOREIGN KEY (TerritoryKey) REFERENCES AdventureWorks\_Territories(SalesTerritoryKey)

);

**3. Data Cleaning Procedures**

Cleaning the data is crucial to ensure that the data is in the right format, free from duplicates, and correctly linked. Here are some SQLbased data cleaning steps:

Step 1: Removed NULLs and replaced with NA in the prefix column at the AdventureWorks\_Customers table.

Step 2: Fill in Missing Dates

Ensure that the Calendar table contains all the dates found in the OrderDate fields:

Insert missing dates from the sales data into the Calendar table

INSERT INTO AdventureWorks\_Calendar (Date)

SELECT DISTINCT OrderDate FROM AdventureWorks\_Sales\_2015

WHERE OrderDate NOT IN (SELECT Date FROM AdventureWorks\_Calendar);

Step 3: Normalize Text Fields

Standardize inconsistent text data for uniformity and also updating the values in the database from abbreviated forms to full forms which helps make the data more readable and consistent.

UPDATE AdventureWorks\_Customers

SET Gender = REPLACE(REPLACE(Gender, 'M', 'Male'), 'F', 'Female');

UPDATE AdventureWorks\_Customers

SET MaritalStatus = REPLACE(REPLACE(MaritalStatus, 'M', 'Married'), 'S', 'Single');

UPDATE AdventureWorks\_Customers

SET HomeOwner = REPLACE(REPLACE(HomeOwner, 'Y', 'Yes'), 'N', 'No');

UPDATE AdventureWorks\_Customers

SET FirstName = UPPER(FirstName), LastName = UPPER(LastName);

Step 4: Text Cleaning: This involves identifying and correcting (or removing) corrupt or inaccurate records from a dataset. In your case, it's about removing invalid characters (e.g., �) from the FirstName, LastName and emailaddress fields in the SQLite database.

UPDATE AdventureWorks\_Customers

SET FirstName = REPLACE(FirstName, '�', ''),

LastName = REPLACE(LastName, '�', '')

WHERE FirstName LIKE '%�%' OR LastName LIKE '%�%';

UPDATE AdventureWorks\_Customers

SET EmailAddress = REPLACE(EmailAddress, '�', '')

WHERE EmailAddress LIKE '%�%';

Conclusion

This report provides the SQL scripts and data cleaning steps necessary to create and maintain a star schema as per the database structure shown in the image. Following these procedures ensures reproducibility and data integrity for transparent analysis. Each of the SQL queries addresses core aspects of the sales and returns data while ensuring that the dimensions like customers and products are correctly maintained.