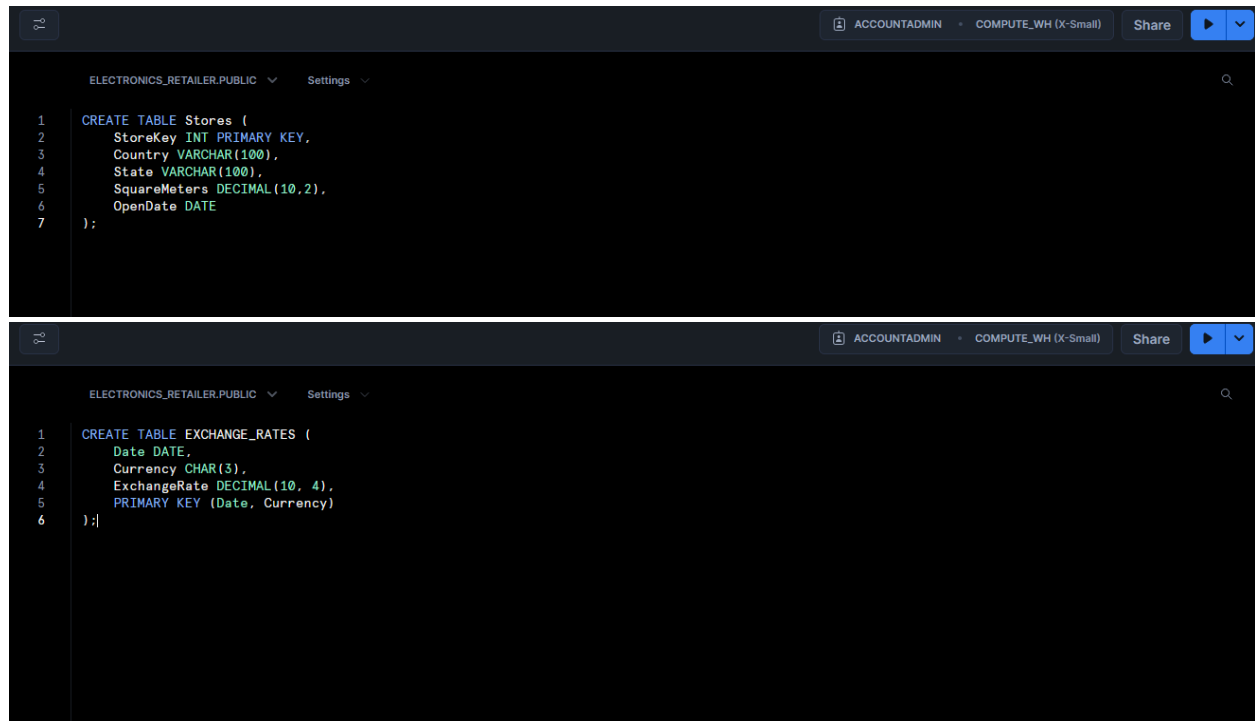


Logan Samples

04/30/2025

DDL Examples



The image displays two screenshots of a SQL development environment. The top screenshot shows the creation of a table named 'Stores' with columns: StoreKey (INT, PRIMARY KEY), Country (VARCHAR(100)), State (VARCHAR(100)), SquareMeters (DECIMAL(10,2)), and OpenDate (DATE). The bottom screenshot shows the creation of a table named 'EXCHANGE_RATES' with columns: Date (DATE), Currency (CHAR(3)), and ExchangeRate (DECIMAL(10,4)). The primary key for 'EXCHANGE_RATES' is defined as a composite key on (Date, Currency).

```
1 CREATE TABLE Stores (  
2   StoreKey INT PRIMARY KEY,  
3   Country VARCHAR(100),  
4   State VARCHAR(100),  
5   SquareMeters DECIMAL(10,2),  
6   OpenDate DATE  
7 );
```

```
1 CREATE TABLE EXCHANGE_RATES (  
2   Date DATE,  
3   Currency CHAR(3),  
4   ExchangeRate DECIMAL(10, 4),  
5   PRIMARY KEY (Date, Currency)  
6 );
```

The exchange rates table does not have a natural primary key. To prevent entries that have the same date and currency type, I needed to include a composition key. This key ensures that no duplicate entries for a currency on an existing date can occur.

ACCOUNTADMINCOMPUTE_WH (X-Small)Share

ELECTRONICS_RETAILER.PUBLICSettingsCode Versions

```
1 CREATE TABLE CUSTOMERS (  
2   CustomerKey INT PRIMARY KEY,  
3   Gender VARCHAR(10),  
4   Name VARCHAR(100),  
5   City VARCHAR(100),  
6   State_Code VARCHAR(10),  
7   State VARCHAR(100),  
8   Zip_Code VARCHAR(10),  
9   Country VARCHAR(100),  
10  Continent VARCHAR(100),  
11  Birthday DATE  
12 );
```

ResultsChart

status

1 Table CUSTOMERS successfully created.

Query Details

Query duration148ms

Rows1

Query ID01bc1be4-0002-02b2-0...

Show more

status100% filledAsk Copilot

ACCOUNTADMINCOMPUTE_WH (X-Small)Share

ELECTRONICS_RETAILER.PUBLICSettings

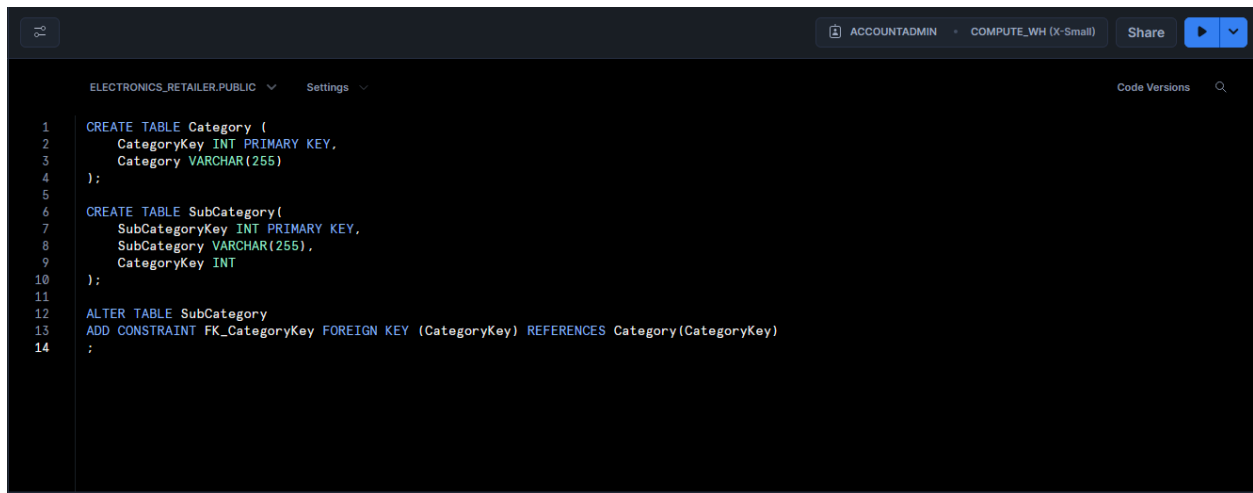
```
1 CREATE TABLE Products (  
2   ProductKey INT PRIMARY KEY,  
3   ProductName VARCHAR(255),  
4   Brand VARCHAR(255),  
5   Color VARCHAR(100),  
6   UnitCostUSD DECIMAL(10,2),  
7   UnitPriceUSD DECIMAL(10,2),  
8   SubcategoryKey INT,  
9   Subcategory VARCHAR(100),  
10  CategoryKey INT,  
11  Category VARCHAR(100)  
12 );
```

ACCOUNTADMINCOMPUTE_WH (X-Small)Share

ELECTRONICS_RETAILER.PUBLICSettings

```
1 CREATE TABLE Sales (  
2   OrderNumber INT,  
3   LineItem INT,  
4   OrderDate DATE,  
5   DeliveryDate DATE,  
6   CustomerKey INT,  
7   StoreKey INT,  
8   ProductKey INT,  
9   Quantity INT,  
10  CurrencyCode CHAR(3),  
11  PRIMARY KEY (OrderNumber, LineItem)  
12 );  
13  
14 ALTER TABLE Sales  
15 ADD CONSTRAINT FK_CUSTOMERS FOREIGN KEY (CustomerKey) REFERENCES Customers(CustomerKey);  
16  
17 ALTER TABLE Sales  
18 ADD CONSTRAINT FK_Stores FOREIGN KEY (StoreKey) REFERENCES Stores(StoreKey);  
19  
20 ALTER TABLE Sales  
21 ADD CONSTRAINT FK_Products FOREIGN KEY (ProductKey) REFERENCES Products(ProductKey);
```

At first glance, order number would seem to be a natural key, but an order can contain multiple which is represented with line items. A composition key ensures that there are no duplicate line items within an order that are created. The foreign key constraints ensure referential integrity between tables.



The screenshot shows a SQL IDE interface with a dark theme. At the top, there's a header bar with a user icon, the text 'ACCOUNTADMIN', a database name 'COMPUTE_WH (X-Small)', a 'Share' button, and a dropdown menu. Below the header, the main area displays SQL code for creating two tables: 'Category' and 'SubCategory'. The 'Category' table has a primary key 'CategoryKey' of type 'INT' and a 'Category' column of type 'VARCHAR(255)'. The 'SubCategory' table has a primary key 'SubCategoryKey' of type 'INT', a 'SubCategory' column of type 'VARCHAR(255)', and a foreign key 'CategoryKey' of type 'INT' that references the 'Category' table's 'CategoryKey'. The code is numbered from 1 to 14 on the left side. The IDE also shows 'ELECTRONICS_RETAILER.PUBLIC' in the top left and 'Settings' in the top right. A 'Code Versions' button is also visible in the top right.

```
1 CREATE TABLE Category (  
2     CategoryKey INT PRIMARY KEY,  
3     Category VARCHAR(255)  
4 );  
5  
6 CREATE TABLE SubCategory(  
7     SubCategoryKey INT PRIMARY KEY,  
8     SubCategory VARCHAR(255),  
9     CategoryKey INT  
10 );  
11  
12 ALTER TABLE SubCategory  
13 ADD CONSTRAINT FK_CategoryKey FOREIGN KEY (CategoryKey) REFERENCES Category(CategoryKey)  
14 ;
```

Category and SubCategory Need their own tables to ensure referential integrity. Without these tables, a category/subcategory key could be tied to 2 different categories/subcategories.