Lab 2: Building a Star Schema

# 1. Overview

This lab introduces how to use SQLite and operate a file-based database, which will be used later in other labs. After finishing this lab, you should:

* Understand how to convert a fact and its measures to a fact table
* Understand how to convert a dimension and its levels into a dimension table
* Understand how to create primary keys in fact table and dimension table
* Understand how to connect primary keys of the fact table to the primary keys to the dimension table.

# 2. The Sample Star Schema

Given the star schema below, we will build this sales schema with SQLite in this lab.

Diagram

Description automatically generated

# 3. Create Dimensions

Create a new SQLite database using the Command Prompt (Windows) or Terminal (Mac):

* Change into the directory where the SQLite Shell is located
* Run sqlite3 sales.db in the command line window
* In this case, a database file called sales.db is created

We need to create the tables for dimensions because the foreign keys in the fact table depends on these dimensions. Based on the figure shown in Section 2, we will firstly create the dimension date using a CREATE statement since it is the simplest one. Run the following statement in the SQLite Shell.

CREATE TABLE date (

"keyD" integer NOT NULL PRIMARY KEY,

date timestamp without time zone,

quarter varchar(255),

month varchar(255),

dow varchar(255),

year varchar(255)

);

**Note that you need to match the case of column name of the primary key.** For those names with upper-cased letters, we put double quotes around them. Otherwise, we cannot import data later when there are mismatches between column names.

In addition, we need to choose the right data types for the columns. For the purpose of completing this lab, you can choose the following data types based on the following hints:

* Keys – Use integer
* String – Use varchar(N) where N is the maximum length of that string
* Floating point numbers / decimal numbers – Use real
* Dates – Use timestamp without time zone

Similarly, we can create the dimension product using the below statement:

CREATE TABLE product (

"keyP" integer NOT NULL PRIMARY KEY,

brand varchar(255),

product varchar(255),

"SKU" varchar(50),

"SRP" real,

gross\_weight real,

net\_weight real,

type varchar(50),

category varchar(50),

department varchar(50),

family varchar(50),

lowfat varchar(255)

);

Now, run the .tables command in the SQLite Shell, you should be able to see the table names that you just created.

## Exercise

Create the rest of the dimensions, i.e., store and customer, by writing the appropriate CREATE statements. Refer to the schema figure in Section 2 for the necessary column names.

# 4. Create the Fact Table

The fact table follows the similar approach used for creating the dimension tables. However, we must add foreign keys to connect the fact table to the dimensions.

Inspect the CREATE statement below (especially the foreign key definitions). Then, identify what the keys and the measures are in this fact table. Finally, run it with SQLite.

CREATE TABLE sales (

"keyD" integer,

"keyC" integer,

"keyP" integer,

"keyS" integer,

receipts real,

cost real,

quantity real,

FOREIGN KEY(keyD) REFERENCES date(keyD),

FOREIGN KEY(keyC) REFERENCES customer(keyC),

FOREIGN KEY(keyP) REFERENCES product(keyP),

FOREIGN KEY(keyS) REFERENCES store(keyS)

);

Run the .tables command in the SQLite Shell again, and make sure you have all these tables created:

customer date product sales store

# 5. Importing Data

Download the data file provided from Canvas, which should be a zip file. Extract the content of the zip file, and you should obtain a SQL file called sales\_data.sql. Put this file in the **same directory** of the database file.

Use the command of SQLite Shell .read to import this file. For example, run .read sales\_data.sql

The data should be imported by now. Run a few SELECT statements to inspect the data. For example:

