CS 6400 Database Project: **S&E's Technology Superstore Data Warehouse**

Spring 2019

Project Overview

The purpose of this project is to analyze, specify, design, and implement a data warehouse for an up-and-coming computer and electronics store called S&E's Technology Superstore. The project will proceed in three phases as outlined in the Classical Methodology for Database Development: Analysis & Specification, Design, and Implementation & Testing. The system will be implemented using a Database Management System (DBMS) that supports standard SQL queries.

The S&E Data Warehouse

The S&E Technology Superstore is an up-and-coming computer and electronics retail business with stores throughout the United States. S&E specializes in all kinds of products including computers, software, televisions, sound systems, cameras, and electronic accessories.

Your team has been tasked with designing and building a data warehouse used by the S&E executive team to determine how S&E stores are doing and make major decisions about the future of the company. This section describes in detail the requirements for S&E's data warehouse (SEDW).

A data warehouse is a database system used for reporting, analysis, and other tasks required for decision support. Unlike *transactional* databases which are generally designed to record repetitive day-to-day business transactions (e.g., point of sale, buy and sell stock orders, online shopping carts, etc.), *data warehouses* are specially suited for reporting and analysis over millions of records to support enterprise-wide decision making. As an example, a large online merchant like amazon.com or bestbuy.com relies on a transactional (also called *operational*) database system for recording customer orders and payments in real time. A data analyst tasked with generating a report that compares sales of a certain product among the different regions of the United States will typically query a specially-designed data warehouse for the report instead of accessing the transactional databases directly. There are several reasons for this: the data warehouse can store data from multiple transactional databases in a consolidated form, the data warehouse schema is designed to support complex queries aggregating millions of rows, and queries against the data warehouse do not impact the performance of the transactional database which must support high transaction throughput.

For this project, you will design the database schema for S&E's Technology Superstore data warehouse and attach it to a rudimentary user interface. You need not be concerned with the transactional databases that we assume exist to support the point-of-sale system at each of S&E's stores. Instead, you will design the schema to support a consolidated view of the products offered and sold in all of S&E's stores across the country. What follows is a description of the requirements for the data warehouse in terms of what information must be stored to support a set of reports defined by S&E's executive team.

Even though some amount of redundancy is typically acceptable in a data warehouse schema, ***for this project you should create a normalized schema with as little redundancy as possible***.

Data Requirements

The S&E Data Warehouse (SEDW) maintains information about each *store*, including a unique *store number*, the store's *phone number*, and the store's *street address*. In addition, the store *manager name* and the manager *e-mail address* should be kept. A store may have more than one manager and managers may manage more than one store. Managers may become *inactive* if they stop being an S&E employee. Some stores may not have a manager assigned to them, but all active managers must be assigned to a store.

SEDW should also maintain information about each store's *city*, including the *city name*, the *state* in which the city is located, and the *population* of the city. It is possible that multiple stores are located in the same city.

SEDW contains information about every *product* for sale at S&E's stores. Products have a numeric unique identifier (*PID*), similar to a UPC barcode, as well as the *name* of the product. Assume that all products are available and sold at all stores—that is, there is no need to specify that a certain product is only available at a certain store.

Each product is related to a single *manufacturer.* Each manufacturer has a *name.* It is possible that multiple products are made by the same manufacturer.

To help identify the kinds of products that are popular, each product is assigned one or more *categories*. Each category has a *name*, which we assume to be unique. Every product must be in at least one category.

Every product has a retail *price*. The retail price is in effect unless there is a *sale*. SEDW maintains the *sale date* and *sale price* of any product that goes on sale. If a product is on sale for multiple days in a row, then a record is stored in the data warehouse for each day of the sale. It is possible that the same product goes on sale multiple times (i.e., different days) with different sale prices. **If a product goes on sale, it is on sale at the same price in all stores—i.e., stores are not allowed to hold sales independently or have store-specific sale prices.** The data warehouse should disallow sale prices that are higher than retail prices. Some manufacturers put a cap on the *maximum discount* that any retailer can apply to any of the manufacturer's products in terms of a percentage. For example, if a manufacturer has a maximum discount of 20%, then no product can be placed on sale for less than 80% of the retail price. A maximum discount of 0% means the product cannot be placed on sale. Even if a maximum discount is not specified by the manufacturer, as a general rule of S&E, no product can be discounted more than 90% of retail.

The S&E executive team would like the ability to compare sales data on *holidays* versus non-holidays, so SEDW should maintain information about which specific dates are holidays. The specific name of the holiday is also required.

Finally, SEDW stores information about which products are *sold*, including the *store* where it is sold, the *date* of the sale, and the *quantity* of the product purchased. The price of the sale is not stored explicitly, but can be derived based on the date purchased and the quantity. Assume there is no sales tax. Also, the data warehouse is not required to store which products were purchased together during a single sales transaction.

S&E's DBAs are working on an extract of sample data from their point-of-sale system for you to test in your data warehouse, however, to avoid revealing confidential information, S&E's data security team has directed them to use data from almost twenty years ago and refuse to allow newer data to be used. Retrieving the data from tape backup will take at least two to three months before it can be made available to you. (The contractor responsible for the tapes chose to store them in Greenland, and they are inaccessible until the glacier recedes from the vault entrance in the spring.) You will need to ensure that your schema design matches the data as described here so that any transformation prior to loading is kept to a minimum.

S&E Data Warehouse User Interface

All of your reports will be accessible from a “dashboard” UI that must be developed. Since this is the first version of the system, you do not need to concern yourself with configuring usernames or passwords to control access to the system, as S&E’s data security team will handle that, and in addition, the version you will demonstrate to S&E’s executive team will be populated with outdated information.

There should be a main menu screen which can be used to access all functionality of the system that has been described in this specification. On this main menu, the following statistics should be displayed along with any buttons/links to reports or functionalities: the count of stores, manufacturers, products, and managers in the data warehouse.

In addition to the reports, there are some relatively simple interfaces you should design and provide as part of maintaining the data warehouse. First, you must provide an interface for holidays to be maintained by the user. This interface must allow for viewing and adding holiday information directly within the user interface. Second, there must be a mechanism in the UI for managers’ information to be added, removed, and assigned/unassigned to stores, as this information may not be available from the source system or outdated. Note that a manager who has become inactive cannot be deleted from the system until they have been unassigned from all of their stores. Finally, your UI must allow for updating the population of any cities in the data warehouse, should a city’s population change.

S&E Data Warehouse Reports

S&E management has put your team in charge of developing the queries necessary to produce the following reports. Many of the reports have derived and/or aggregate data. As mentioned previously, these reports will be accessed with the user interface that you will create.

Some of the report queries are expensive to run given the large number of rows in the S&E Data Warehouse. Therefore, whenever possible you should include the filter conditions specified. For example, some reports ask for data from only a certain time period. If you leave off this filtering condition, the query will likely take a long time to return any results.

Report 1 – Manufacturer’s Product Report

For each manufacturer, return the manufacturer’s name, total number of products offered by the manufacturer, average retail price of all the manufacturer’s products, minimum retail price, and maximum retail price. Ignore all sale days (do not take into account the days the product is discounted). Sort the results by average price with the highest average price appearing first, for only the top 100 manufacturers based on average price.

This report should also have “drill-down” detail (in other words, each line in the master report should have a method for loading its detail, such as a hyperlink or a button) for the manufacturer, which shows in the report header the manufacturer’s details (name and maximum discount), the summary information from the parent report, and lists for each of the manufacturer’s products’ its product ID, name, category (or categories), and price, ordered by price descending (high to low). If a product has multiple categories it must not show up as multiple rows on the report, but as a single row with multiple categories concatenated together.

Report 2 – Category Report

For each category, return the category name, total number of products in that category, total number of unique manufacturers offering products in that category, and the average retail price (not including sale days) of all the products in that category, sorted by category name ascending.

Report 3 – Actual versus Predicted Revenue for GPS units

S&E executives want to predict whether offering items at a discount actually helps to increase revenue by encouraging a higher volume of sales. This report compares how much revenue was actually generated from a product’s sales to a predicted revenue if the product were never offered on sale. After speaking with some marketing consultants, S&E executives have learned that product discounts introduce on average a 25% increase in volume (quantity sold). Therefore we assume that if an item that was offered at a discount were instead offered at the retail price, the quantity of items sold would be reduced by 25%. However, it is still possible that the predicted revenue would be higher since the reduced volume of products would be sold at a higher price per product. Initially, the executives are only interested in seeing the report for products in the GPS category.

Here is a simple example:

Assume that Product Z has a retail price of $10. Assume that it was offered at a discount for on 6/1/2012 and 6/2/2012. Also assume the following transaction data for Product Z:

Date	Price	Quantity	Actual Revenue
5/1/2012	10.00	5	50.00
6/1/2012	8.00	10	80.00
6/2/2012	7.00	5	35.00
TOTALS		**20**	**$165.00**

Table 1 - Actual Revenue

The predicted revenue is calculated by assuming that the product is never offered at a discount and only 75% of the original quantity was actually sold on discounted days. Note that because this is just a predicted average, we assume that it is possible to sell a fraction of a product (e.g., 7.5 DVD players).

Date	**Price**	**Quantity**	**Predicted Revenue**
5/1/2012	10.00	5	50.00
6/1/2012	~~8.00~~ 10.00	10 * .75 = 7.5	75.00
6/2/2012	~~7.00~~ 10.00	5 * .75 = 3.75	37.50
TOTALS		**16.25**	**$162.50**

Table 2 - Predicted Revenue

In this example, the discounted prices resulted in slightly more revenue due to the higher volume of sales ($2.50 more).

Generate the following report: For each product in the GPS category, return the product ID, the name of the product, the product's retail price, the total number of units ever sold, the total number of units sold at a discount (i.e., during a sale day, the total number of units sold at retail price, the actual revenue collected from all the sales of the product, the predicted revenue had the product never been put on sale (based on 75% volume selling at retail price), and the difference between the actual revenue and the predicted revenue. If the difference is a positive number, it means that the discounts worked in favor of S&E because the predicted revenue is less than the actual revenue collected. If it is a negative number, it indicates that S&E would have been better off not offering the product discounts. Only predicted revenue differences greater than $5000 (positive or negative) should be displayed and sorted in descending order.

Report 4 –Store Revenue by Year by State

This report shows the revenue collected by stores per state grouped by year. The states available for querying should be presented in a drop down box. For example, the user would select "New York" and the system would show each store in New York state, show the store ID, store address, city name, sales year, and total revenue. Be sure the revenue calculation takes into account items that were sold at a discount. Sort the report first by year in ascending order and then by revenue in descending order.

Report 5 – Air Conditioners on Groundhog Day?

Some of the sales staff have noticed that air conditioner sales seem to spike on Groundhog Day (which falls on February 2 each year). They surmise that this is because customers begin thinking about the warm spring weather ahead. The S&E marketing team would like to know for sure if this is the case, so they have requested the following report.

For each year, return the year, the total number of items sold that year in the air conditioning category, the average number of units sold per day (assume a year is exactly 365 days), and the total number of units sold on Groundhog Day (February 2) of that year. Sort the report on the year in ascending order. The marketing team will use the report to determine if the total number of units sold on Groundhog Day each year is significantly higher than the average number of units sold per day.

Report 6 – State with Highest Volume for each Category

S&E management is planning to recognize all stores in the states that sell the most number of units for each category. They want to view this monthly, so the user interface must allow choosing a year and month from the available dates in the database before running the report. The report will return for each category: the category name, the state that sold the highest number of units in that category (i.e., include items sold by all stores in the state), and the number of units that were sold by stores in that state. This output shall be sorted by category name ascending. Note that each category will only be listed once unless two or more states tied for selling the highest number of units in that category. *The report can take a significant time to run, which may require tuned indices for the final implementation, but do not focus on their creation until the final phase.*

This report should also have “drill-down” detail (in other words, each line in the master report should have a method for loading its detail, such as a hyperlink or a button) using the criteria of state, category, and year/month to provide the IDs, addresses, and cities of all the stores and their managers names and email addresses so that they can be recognized for their efforts. This sub report should be ordered by store ID ascending and the header should include the original criteria from the parent report (category, year/month, state). If a store has more than one manager, it can be displayed as multiple rows.

Report 7 – Revenue by Population

To help forecast expansions into other cities, S&E management would like to see what the average revenue is for specific population categories, and to see if there is a trend for growth, the revenue should be broken down on an annual basis. The categories for city size are: Small (population <3,700,000), Medium (population >=3,700,000 and <6,700,000), Large (population >=6,700,000 and <9,000,000) and Extra Large (population >=9,000,000). There is some flexibility in formatting this report, in that it could be “pivoted” to present it with either years or city category as columns or as rows, so ensure that both elements are arranged in ascending order (oldest to newest for years, smallest to largest for city size) so that no matter how it is formatted it is properly organized and understandable.

Revision History

Version	Notes	Date
1.0	New version for Spring 2019	1/21/19
1.1	Clarification of active vs. inactive managers needing store assignment and their deletion, changing “loading” of holidays to “maintaining” with more detail, fixed error where store “name” was listed for report 6 instead of address, updated report 1 to exclude a field that is not actually present	2/1/19