

# CS 6400 Database Project: Peachtree Savings Club

---

*Fall 2025*

## **Project Overview**

The purpose of this project is to analyze, specify, design, and implement a data warehouse for an up-and-coming club store called Peachtree Savings Club. The project will proceed in three phases as outlined in the 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.

When reading through this project description, please make the following assumptions: unless otherwise specified as optional, all attributes are required; unless otherwise specified, if given a list of potential values, choices should be limited to that list; If a set of values is listed with "and/or", combinations of those values are possible, while "or" indicates only a single value is possible; that you should create normalized schemas, and minimize the use of NULL attributes whenever and wherever possible; ensure that you store non-numeric data that appear as numbers (such as street numbers, phone numbers, postal codes, etc.) as strings and not numeric data types; and avoid "catch-all" forms with unnecessary inputs that the user would leave empty or NULL. You also do not need to be concerned about handling concurrent operations that could conflict and introduce inconsistencies in your database, so transactions and locking will not be required. Any actions listed as "performed by the database administrator" will not be performed in our application, but rather manually, via SQL prompt or similar interface. Do not create any extra functionality that is not mentioned in this specification (such as email notifications, etc.) or attempt to enhance your final product beyond what the specification requires. Adding unwanted functionality can and will impact your grade!

## **Peachtree Savings Club Data Warehouse**

Peachtree Savings Club is a wholesale club business with stores throughout the United States. It sells all kinds of products. Your team has been tasked with designing and building a data warehouse used by the executive team to determine how the company stores are doing and make major decisions about the company's future. This section describes in detail the requirements for Peachtree Savings Club data warehouse.

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 especially 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 Peachtree Savings Club 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 the company stores. Instead, you will design a schema to support a consolidated view of the products offered and sold in all 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 the executive team.

Even though some 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

Peachtree Savings Club Data Warehouse (PSCDW) maintains information about each *store*, including a unique *store number*, the store's *phone number*, the store's *street address*, and the store's *zip code*.

PSCDW 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.

PSCDW contains information about every *product* for sale at the company stores. Products have a unique identifier (*PID*), as well as the *name* of the product. We will 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* and *unique numeric id*. 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 to 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*. PSCDW 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., on several different days) with different sale prices on different sale days. **At the same time, if a product goes on sale on a certain day, it is on sale at the same price on this day in all stores—i.e., stores are not allowed to hold sales independently or have store-specific sale prices.** The data warehouse disallows 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 coming from this manufacturer 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 at all. Even if a manufacturer does not specify a maximum discount, Peachtree Savings Club's general rule is that no product may be discounted more than 90% off the retail price. In addition, Savings Club administration does not allow any product to be on sale for more than 25 days in any calendar month. Be aware, however, that as with any retail store, pricing errors may occur for a variety of reasons, and such errors in historical sales data should not be corrected.

The Savings Club executive team also requires that our application maintain information about which specific dates are holidays. The specific name of the holiday is also required for each day designated as a holiday.

PSCDW 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. We will assume that our company does not need to deal with any sales tax. Also, the data warehouse is not required to store which products were purchased together during a single sales transaction.

Since PSCDW is a club warehouse, they sell *memberships*. For each new membership, the *member ID*, which will be unique, the *signup date*, the *signup store*, and the *membership type* are tracked. Memberships can only be purchased in-store; online or mail signups are not supported. Peachtree Savings Club currently offers four levels of membership: “Basic”, “Gold”, “Platinum” or “Diamond”. There is a possibility, however, that the executive team will add a few more levels. Therefore, the DBA should have an option to update the list of possible levels, although this update will not be performed through our application: the DBA will use a manual SQL command to implement it. You do not need to worry about storing details about existing memberships, our application will only need to be concerned with statistical data on new enrollments.

Savings Club’s data analytics group is currently working on extracting sample data from their point-of-sale system for us to test our data warehouse functionality. However, to avoid revealing confidential information, the data security team has directed them to use old data limited to certain categories and has refused to allow any newer data to be used. Retrieving the data from tape backup and sanitizing it will take at least two to three months before it can be made available to us. Therefore, we will need to ensure that by that time our schema design matches the data as described here so that any transformations prior to loading the data into the database are kept to a minimum.

## **Peachtree Savings Club Data Warehouse User Interface**

All reports will be accessible from a “dashboard” UI that must be developed.

There should be a main menu screen which can be used to access all functionality of the system that is 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 memberships sold.

In addition to the reports, there are some relatively simple interfaces we should design and provide as part of maintaining the data warehouse. First, we must provide an interface for holidays to be maintained by the user. This interface must allow for viewing and adding holiday information (but not editing or deleting holidays) directly within the user interface. Second, our UI must allow for updating the population of any city in the data warehouse, should a city’s population change.

## **Peachtree Savings Club Data Warehouse Reports**

Savings Club management has put our team in charge of developing the queries necessary to produce the following reports. As mentioned previously, these reports will be accessed with the user interface that we will create.

### **Report 1 – Manufacturer’s Product Report**

For each manufacturer, return the manufacturer id, the manufacturer’s name, total number of products offered by the manufacturer, average retail price of all the manufacturer’s products, minimum retail price, maximum retail price, retail price range (the difference of the maximum retail price minus the minimum retail price). Note that in this report, only retail prices are used in calculations. Actual transactions at discounted prices are not considered. Sort the results by average price with the highest average price appearing first, show only the top 100 manufacturers based on average price.

This report should also provide access to “drill-down” detail for each manufacturer. In other words, each line in the parent report should have a method for loading its detail, such as a hyperlink or a button on the manufacturer id, which would show in the report header the manufacturer’s details (id, name and maximum discount), the summary information from the parent report, and lists, for each of the manufacturer’s products, its product ID, product name, category (or categories) and price. This output should be ordered by price descending (high to low) and by product id ascending. 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 in alphabetical order, separated by forward slashes.

### **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, the average retail price,

total revenue for all products sold in that category (the revenue should certainly take into account actual selling price, i.e. if the product was sold at retail price or at discounted price in this case). The results should be sorted by category name ascending.

### Report 3 – Actual versus Predicted Revenue for Speaker units

The company executives want to predict whether offering items at a discount helps to increase revenue by encouraging a higher volume of sales. This report compares how much revenue was generated by actual product sales to a predicted revenue in the hypothetical situation when the product was never offered on sale. After speaking with some marketing consultants, the executives have learned that the lack of discounts typically results in a 25% decrease 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 unit. Initially, the executives are only interested in seeing the report for products in the Speaker category.

Here is a simple example:

Assume that Product Z has a retail price of \$10. Assume that it was offered at a discount on 6/1/2025 and 6/2/2025. Also assume the following transaction data for Product Z:

<u>Date</u>	<u>Price</u>	<u>Quantity</u>	<u>Actual Revenue</u>
5/1/2025	10.00	5	50.00
6/1/2025	8.00	10	80.00
6/2/2025	7.00	5	35.00
<b>TOTALS</b>		<b>20</b>	<b>\$165.00</b>

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 sold on discounted days. Because this is a prediction, we assume that it is possible to sell a fraction of a product (e.g., 7.5 Speakers).

<u>Date</u>	<u>Price</u>	<u>Quantity</u>	<u>Predicted Revenue</u>
5/1/2025	10.00	5	50.00
6/1/2025	<del>8.00</del> 10.00	10 * .75 = 7.5	75.00
6/2/2025	<del>7.00</del> 10.00	5 * .75 = 3.75	37.50
<b>TOTALS</b>		<b>16.25</b>	<b>\$162.50</b>

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 Speaker 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 sale days), 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 Peachtree Savings Club, because the predicted revenue is less than the actual revenue collected. If it is a negative number, it indicates that the company would have been better off not offering the 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 number, store address, store zip code, city name, sales year, total revenue, the revenue for products sold at retail price, and revenue for the products sold at discounted price. Sort the report first by year in descending order and then by revenue in descending order as well.

#### 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 company 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 always exactly 365 days, no matter how many days of the year are actually represented in the database), and the total number of units sold on Groundhog Day (February 2) of that year. Sort the report by the year in ascending order.

#### Report 6 – State with Highest Volume for each Category

The company management is planning to recognize the states that sell the greatest numbers of units for each category. They want to view this monthly, so the user interface must allow choosing a year and a 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, and the total number of units that were sold in all 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 are tied for selling the highest number of units in that category.

## Report 7 – Revenue by Population

To help forecast profitability of expansions into new cities, the company management would like to see what the average revenue is for all cities in specific population categories, and to determine 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  $\geq 3,700,000$  and <6,700,000), Large (population  $\geq 6,700,000$  and <9,000,000) and Extra Large (population  $\geq 9,000,000$ ). The report should show a row for every year present in the data and for each year should show the following columns: the year, the average revenue for small cities, the average revenue for medium cities, the average revenue for large cities and the average revenue for extra large cities. The output should be sorted by year in ascending order.

## Report 8 – Membership Trends

The management team wishes to track the number of new signups per city, as this indicates if more stores are needed in an area or if stores should be closed and/or consolidated.

This report will have three parts. First, the initial report will show each year present in the database in descending order (most recent to least recent year) and the total number of memberships sold for that year.

The second part is a drill-down (in other words, each line in the parent report should have a method for loading its detail, such as a hyperlink on the total or a button) for that year showing the top 25 cities that sold the most memberships, in descending order of numbers of memberships and ascending alphabetical order of city names, and another section showing the bottom 25 cities that sold the least memberships for that year in ascending order of numbers of memberships and ascending alphabetical order of city names. Cities which have sold 250 or more memberships should have their total highlighted with a green background. Cities which have sold 30 or fewer memberships should have their total highlighted with a red background. Since same city names can be used in different states, be sure to include the state the city is in. The header of this report should include the selected year.

The last part is a city to the store level drill-down. All lines in the city level drill-down report, **which correspond to the cities with more than one store**, should have a method for loading its detail, such as a hyperlink on the total or a button. This second, store level drill-down should show store number, street address, zip code, city, and state along with the total number of memberships sold during that year in this store. The header of the report should include the selected year.

## Revision History

<u>Version</u>	<u>Notes</u>	<u>Date</u>
1.0	New version for Fall 2025	08/31/25
1.1	Made updates for clarification based on feedback: <ul style="list-style-type: none"><li>• Made clearer that Revenue by Population should return average revenue over cities within each category</li><li>• Changed Manufacturer's Product Report to link to drilldown from main report using manufacturer ID instead of name</li></ul>	10/15/25