Table of Contents

Data Types

Store		
Attributes	Data Type	Nullable
StoreNumber	INT	Not Null
PhoneNumber	STRING	Not Null
StreetAddress	STRING	Not Null
StoreAffliates	STRING	Null
ChildcareLimit	INT	Not Null
City		
Attributes	Data Type	Nullable
ZipCode	INT	Not Null
States	STRING	Not Null
CitySize	STRING	Not Null
Population	LONG INT	Not Null
CityName	STRING	Not Null
DiscountPrice		
Attributes	Data Type	Nullable
DiscountPrice	FLOAT	Not Null
DiscountDate	DATE	Not Null

Date		
Attributes	Data Type	Nullable
Date	DATE	Not Null
AdCampaignDescription	STRING	Null
Holiday	STRING	Null
Sales		
Attributes	Data Type	Nullable
QuantitySold	INT	Not Null
Product		
Attributes	Data Type	Nullable
PID	INT	Not Null
ProductName	STRING	Not Null
RegularPrice	FLOAT	Not Null
ProductCategory		
Attributes	Data Type	Nullable
CategoryName	STRING	Not Null

Business Logic Constraints

Product:

- All products are available in all stores to be sold to all customers.
- The daily price of a product is the RegularPrice unless there is a DiscountPrice.
- RegularPrice should be greater than 0.

Sales:

- A sales represents the total quantity of a product sold on a certain day in a certain store.
- QuantitySold should always be greater or equal to 0.

Store:

- Each store chooses the time limit of childcare (ChildcareLimit) from predetermined values. ChildcareLimit is zero if the store does not provide childcare. Updating any of these limits is required once it changes or it is manually updated outside of a data load.
- StoreAffiliates can only be none, part or all of the three values: snackbar, restaurant, and childcare.
- All stores share the same discount price for a certain product on the same day.
- Phone number should be in the string format of: XXX-XXX-XXXX.

City:

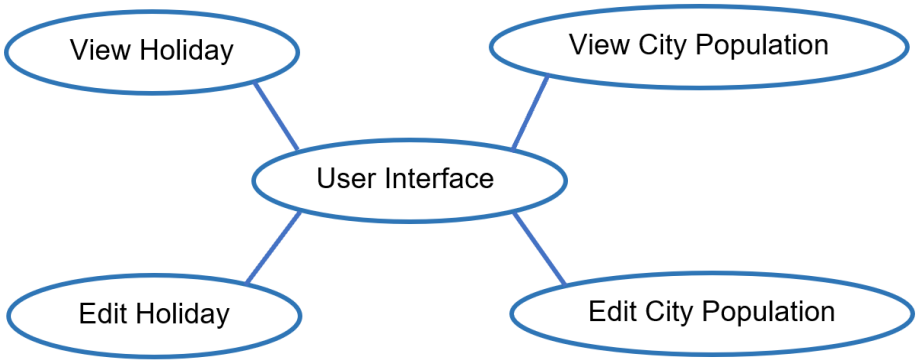
- CitySize is a categorical attribute derived from Population.
    - Small: Population < 3,700,000
    - Medium: Population >= 3,700,000 and Population < 6,700,000
    - Large: Population >= 6,700,000 and Population < 9,000,000
    - Extra large: Population >= 9,000,000
- Population should be greater than 0.
- State should be in one of (AL, AK, AZ, AR, CA, CO, CT, DE, DC, FL, GA, HI, ID, IL, IN, IA, KS, KY, LA, ME, MD, MA, MI, MN, MS, MO, MT, NE, NV, NH, NJ, NM, NY, NC, ND, OH, OK, OR, PA, RI, SC, SD, TN, TX, UT, VT, VA, WA, WV, WI, WY).
- ZipCode is in the 6-digit number format.

DiscountPrice:

- DiscountPrice should be less than RegularPrice.
- DiscountPrice should be greater than 0.
- The DiscountPrice of a product may change by date.

Task Decomposition with Abstract Code

User Interface



Task Decomp:

Lock Types : Read only Lockups: Category, Store, Product, Sales, DiscountPrice, 2 Read and write lockups: City and Date.
Number of Locks: Several different schema constructs are needed.
Enabling Conditions: None.
Frequency: All have the same frequency.
Consistency (ACID): Not critical, order is not critical.
Subtasks: All tasks must be done, but can be done in parallel. Mother Task is needed.

Abstract Code:

- Show “View Category Report”,“View Holiday”, “Edit Holiday”, “View City Population”, “Edit City Population”, “View Actual versus Predicted Revenue for Couches and Sofas Report”, “View Store Revenue by Year by State”, “View Outdoor Furniture on Groundhog Day”, “View State with Highest Volume for each Category”, “View Revenue by Population”, “View Childcare Sales Volume”, “View Restaurant Impact on Category Sales”, “View Advertising Campaign Analysis” tabs.
- Upon:
    - Click View Category Report button- Jump to View Category Report task.
    - Click View Holiday button- Jump to View Holiday task.
    - Click Edit Holiday button- Jump to Edit Holiday task.
    - Click View City Population button- Jump to View City Population task.
    - Click Edit City Population button- Jump to Edit City Population task.
    - Click View Actual versus Predicted Revenue for Couches and Sofas button- Jump to the View Actual versus Predicted Revenue for Couches and Sofas task.
    - Click View Store Revenue by Year by State button - Jump to ViewStore Revenue by Year by State task.
    - Click View Outdoor Furniture on Groundhog Day button- Jump to the View Outdoor Furniture on Groundhog Day task.

- Click View State with Highest Volume for each Category button- Jump to View State with Highest Volume for each Category task.
- Click View State with Revenue by Population button- Jump to View Revenue by Population task.
- Click View Childcare Sales Volume button- Jump to View Childcare Sales Volume task.
- Click View Restaurant Impact on Category Sales button- Jump to View Restaurant Impact on Category Sales task.
- Click View Advertising Campaign Analysis button- Jump to View Advertising Campaign Analysis task.

View Category Report

Task Decomp:

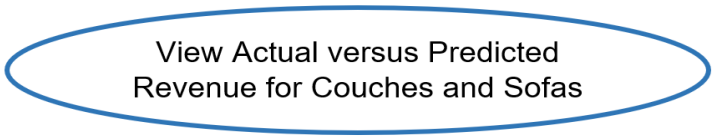
Lock Types : Lockup Category, Product, Date, all read only.
Number of Locks: Several different schema constructs are needed.
Enabling Conditions: Triggered by user.
Frequency: All have the same frequency.
Consistency (ACID): Not critical, order is not critical.
Subtasks: Mother Task is not needed. No decomposition needed.

Abstract Code:

- User clicked on ***View Category Report*** button from **Main Menu**:
- Run the **View Category Report** task: query for information about the category - category name, total number of products in that category, and the average regular price (not including discount days) of all the products in that category.
- Sort the results by category name ascending.

When ready, the user selects the next action from choices in the **Main Menu**.

View Actual versus Predicted Revenue for Couches and Sofas



Task Decomp:

Lock Types : Lockup Product, Sales, DiscountPrice, Category, all read only.
Number of Locks: Several different schema constructs are needed.
Enabling Conditions: Triggered by user.
Frequency: All have the same frequency.
Consistency (ACID): Order is not critical.
Subtasks: Mother Task is not needed. No decomposition needed.

Abstract Code:

- User clicked on ***View Actual versus Predicted Revenue for Couches and Sofas*** button from **Main Menu**:
- Run **View Actual versus Predicted Revenue for Couches and Sofas** task: query for the information about each product in Couches and Sofas category.
    - Display PID, ProductName.
    - Find the current price using the PID; display DiscountPrice, RegularPrice.
    - Find the total number of units ever sold.
        - Calculate the total number of units sold at a discount (DiscountPrice* QuantitySold during ad campaign).
        - Find the total number of units sold at regular price (RegularPrice* QuantitySold outside ad campaign).
    - Find the actual revenue collected from all the sales of the product, the predicted revenue of the product on sale (based on 75% volume selling at regular price), and the difference between the actual revenue and the predicted revenue.
        - Calculate the actual revenue collected from all the sales of the product (RegularPrice * QuantitySold).
        - Calculate the predicted revenue of the product on sale (based on 75% volume selling at retail price) (RegularPrice * QuantitySold * 75%).
        - Calculate the difference between the actual revenue and the predicted revenue.
        - Sort the difference in descending order.

When ready, the user selects the next action from choices in the **Main Menu**.

View Store Revenue by Year by State

Task Decomp:

Lock Types : Lockup City, Store, Sales, Product, DiscountPrice all read only.
Number of Locks: Several different schema constructs are needed.
Enabling Conditions: Triggered by user with selection of state.
Frequency: All have the same frequency.
Consistency (ACID): Not critical, order is not critical.
Subtasks: Mother Task is not needed. No decomposition needed.

Abstract Code:

- User clicked on ***View Store Revenue by Year by State*** button from **Main Menu**:
- Run the **View Store Revenue by Year by State** task: Display all the states in the Drop-down box.
    - Upon selection of the specified state.
    - Display StoreNumber, StreetAddress, CityName.

    - Filter date to the last 12 months.
    - For each month:
        - Calculate the total revenue for all stores for each month in the selected state (revenue calculation must consider the products were sold at a discounted price).
        - Sort the total revenue by each month in ascending order and then by revenue in descending order.

When ready, the user selects the next action from choices in the **Main Menu**.

View Outdoor Furniture on Groundhog Day

Task Decomp:

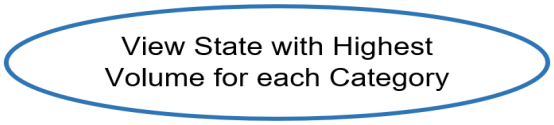
Lock Types : Lockup Category, Product, Date, Sales, all read only.
Number of Locks: Several different schema constructs are needed.
Enabling Conditions: Triggered by user.
Frequency: All have the same frequency.
Consistency (ACID): Not critical, order is not critical.
Subtasks: Mother Task is not needed. No decomposition needed.

Abstract Code:

- User clicked on ***View Outdoor Furniture on Groundhog Day*** button from **Main Menu**:
- Run the **View Outdoor Furniture on Groundhog Day** task:
    - Filter and display date (February 2) of each year.
    - Calculate the total number of units sold that year in the Outdoor Furniture category.
    - Sort the total number of units in the Outdoor Furniture category of each year in ascending order.

When ready, the user selects the next action from choices in the **Main Menu**.

View State with Highest Volume for each Category



Task Decomp:

Lock Types : 5 Read-only lookup of Sales, City, Product, Category and Store.

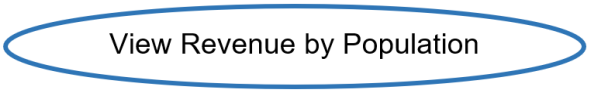
Number of Locks: Several different schema constructs are needed.
Enabling Conditions: Triggered by user's click with selection of year and month.
Frequency: All have the same frequency.
Consistency (ACID): Not critical.
Subtasks: Mother Task is not needed. No decomposition needed.

Abstract Code:

- User clicked on ***View State with Highest Volume for each Category*** button from **Main Menu**:
- Display Sales, City, Product, Category, Store.
- Filter and display the selected date.
- Run the **View State with Highest Volume for each Category task:**
    - Filter by each category.
    - Calculate the highest volume of units for that category of each state.
    - Sort the highest volume of units for that category of the state by states descending order.
    - User clicks ***drill-down detail*** button for category:
    - Drop-down detail for each rows filtered by state, category, and date to discover which store in each city has the highest sales volume.

When ready, the user selects the next action from choices in the **Main Menu**.

View Revenue by Population



Task Decomp:

Lock Types : Look up Sales, City, Product, and Store, all are read-only.
Number of Locks: Several different schema constructs are needed.
Enabling Conditions: Triggered by user's click.
Frequency: All have the same frequency.
Consistency (ACID): It's critical to have most updated city population size from city.
Subtasks: Mother Task is not needed. No decomposition needed.

Abstract Code:

- User clicked on ***View Revenue by Population*** button from **Main Menu**:
- Run the **View Revenue by Population** task:
    - Extract year from Date.date.
    - For **each** CitySize level (Small, Medium, Large, Extra Large):
        - Calculate annual revenue for every year.
        - Sort in ascending order:
            - years (oldest to newest).

- CitySize (small, medium, large, extra large).

When ready, the user selects the next action from choices in the **Main Menu**.

View Childcare Sales Volume

Task Decomp:

Lock Types : Look up Sales, Store, all are read-only.
Number of Locks: Several different schema constructs are needed.
Enabling Conditions: Triggered by user's click.
Frequency: All have the same frequency.
Consistency (ACID): not critical.
Subtasks: Mother Task is not needed. No decomposition needed.

Abstract Code:

- User clicked on ***Childcare Sales Volume*** button from **Main Menu**:
- Run the **View Childcare Sales Volume** task:
    - Filter SalesDate to the last 12 months.
    - Create ChildcareLimitCategory based on Store.ChildcareLimit as columns (When ChildcareLimit = 0, ChildcareLimitCategory = "No childcare").
    - Extract Date.Month from Date.Date.
    - for each Date.Month (as row) and ChildcareLimitCategory (as column):
        - Sum total sales.

When ready, the user selects the next action from choices in the **Main Menu**.

View Restaurant Impact on Category Sales

Task Decomp:

Lock Types : Look up Sales, Store, Product, Category, all are read-only.
Number of Locks: Several different schema constructs are needed.
Enabling Conditions: Triggered by user's click.
Frequency: All have the same frequency.
Consistency (ACID): Not critical.
Subtasks: Mother Task is not needed. No decomposition needed.

Abstract Code:

- User clicked on ***Restaurant Impact on Category Sales*** button from **Main Menu**:
- Run the **View Childcare Sales Volume** task:
    - Exclude CategoryName with no assigned product.
    - Create StoreType (with a restaurant, and stores without) based on Store.StoreAffiliates.
    - For each category and store_type:
        - Count Sales.PID as QuantitySold
        - Sort by CategoryName ascending, StoreAffliates (non-restaurant store data listed first)

When ready, the user selects the next action from choices in the **Main Menu**.

View Advertising Campaign Analysis

Task Decomp:

Lock Types : Look up Date, Product, DiscountPrice, Sales, all are read-only.
Number of Locks: Several different schema constructs are needed.
Enabling Conditions: Triggered by user's click.
Frequency: All have the same frequency.
Consistency (ACID): It's critical to have most updated date information.
Subtasks: Mother Task is not needed. No decomposition needed.

Abstract Code:

- User clicked on ***Advertising Campaign Analysis*** button from **Main Menu**:
- Run the **View Advertising Campaign Analysis** task:
    - Filter only discount product (products in Ad Campaign)
    - for each discount product:
        - Display PID, ProductName.
        - Sum total Sold During Campaign, Sold Outside Campaign:
            - Sum the QuantitySold of Sales (When Sales.Date = DiscountDate) as Sold During Campaign.
            - Sum the QuantitySold of Sales (When Date.Date not in DiscountDate) as Sold Outside Campaign.
        - Calculate the Difference between Sold During Campaign and Sold Outside Campaign.
        - Sort by Difference in descending (highest to lowest) order.
        - Filter Top 10 and Bottom 10.

When ready, the user selects the next action from choices in the **Main Menu**.