Phase 1

Table of Content

Data Types

Store

Attribute	Data Type	Nullable
store_number	Integer	Not Null
phone_number	String	Not Null
street_address	String	Not Null
has_restaurant	Boolean	Not Null
has_snack_bar	Boolean	Not Null

City

Attribute	Data Type	Nullable
city_name	String	Not Null
state	String	Not Null
population	Integer	Not Null

Childcare

Attribute	Data Type	Nullable
limit	Integer	Not Null

Product

Attribute	Data Type	Nullable
pid	Integer	Not Null
name	String	Not Null
retail_price	Float	Not Null

Category

Attribute	Data Type	Nullable
name	String	Not Null

Discount

Attribute	Data Type	Nullable
discount_price	Float	Not Null

Holiday

Attribute	Data Type	Nullable
name	String	Not Null

Sale

Attribute	Data Type	Nullable
quantity	Integer	Not Null

Date

Attribute	Data Type	Nullable
date_number	Date	Not Null

Advertising Campaign

Attribute	Data Type	Nullable
description	String	Not Null

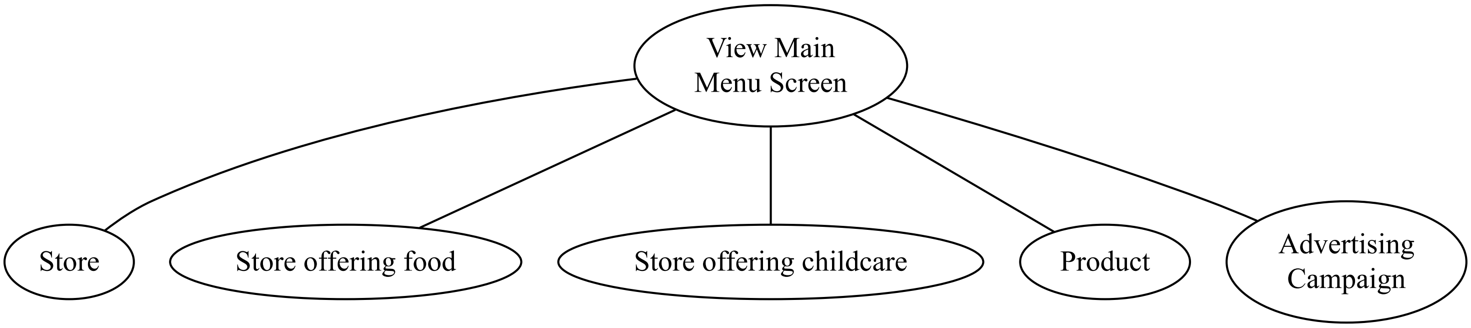
Business Logic Constraints

- Childcare: The limit is an integer index representation of the predetermined time limits (for example, "1" implies 45 minutes).
- Product: All products are available and sold at all stores; If a product is on sale for multiple days in a row, then a record is stored for each day of the sale; The retail price is effective unless there is a discount.
- Sale: Sales tax values are ignored; Only daily sale instead of individual transaction is stored.
- Holiday: If a day has multiple holidays, their names can be combined.
- Discount: Stores are not allowed to discount items independently or have store-specific discount prices; Discount Price should not be higher than the retail price.

Tasks

View Main Menu Screen

Task Decomposition



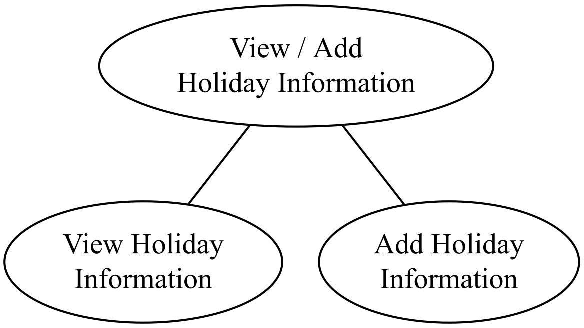
Lock Types: Read-only lookups: the count of stores, count of stores that offer food, count of stores offering childcare, count of products, and count of distinct advertising campaigns. All are read-only.
Number of Locks: Five. Several different schema constructs are needed.
Enabling Condition: All five are enabled by loading the main menu screen.
Frequency: All five have the same frequency.
Consistency (ACID): It is not critical.
Subtasks: Mother Task is needed. All subtasks can be executed in parallel.

Abstract Code

- Show statistics: count of stores, count of stores that offer food (have a restaurant, a snack bar, or both), count of stores offering childcare, count of products, count of distinct advertising campaigns;
- Show Buttons to reports and other functionalities;
- Upon:
  - Click ***Report 1 Category Report*** button - Jump to the **View Category Report** task;
  - Click ***Report 2 Actual versus Predicted Revenue for Couches and Sofas*** button - Jump to the **View Actual versus Predicted Revenue for Couches and Sofas** task;
  - Click ***Report 3 Store Revenue by Year by State*** button - Jump to the **View Store Revenue by Year by State** task;
  - Click ***Report 4 Outdoor Furniture on Groundhog Day*** button - Jump to the **View Outdoor Furniture on Groundhog Day** task;
  - Click ***Report 5 State with Highest Volume for each Category*** button - Jump to the **View State with Highest Volume for each Category By Year and Month** task;
  - Click ***Report 6 Revenue by Population*** button - Jump to the **View Revenue by Population** task;
  - Click ***Report 7 Childcare Sales Volume*** button - Jump to the **View Childcare Sales Volume** task;
  - Click ***Report 8 Restaurant Impact on Category Sales*** button - Jump to the **View Restaurant Impact on Category Sales** task;
  - Click ***Report 9 Advertising Campaign Analysis*** button - Jump to the **View Advertising Campaign Analysis** task.

View / Add Holiday Information

Task Decomposition



Lock Types: One read-only and write-only lookup of the holiday information.

Number of Locks: Two. One schema construct is needed.

Enabling Condition: Both are enabled by a user's request.

Frequency: Low. Viewing holiday information is more frequent.

Consistency (ACID): It is not critical, even if the holiday is being edited by the user while another user is looking at it

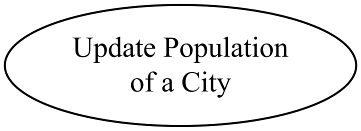
Subtasks: Mother task is required to coordinate two subtasks. Both subtasks can be performed in parallel.

Abstract Code

- Click ***View/Add Holiday Information*** button - jump to **View Holiday Information** and **Add holiday Information** screen;
- Click **View Holiday Information**, the holiday information is displayed. Holiday name and the date will be included;
- Click ***Back***, return to **View Holiday Information** and **Add holiday information** screen;
- Click ***Add holiday information*** button, a form is dispalyed. User is allowed to add holiday name and the date. After the new holiday information is added, click ***Save***. The new information is updated and **View Holiday Information** is displayed;
- If the user clicks ***Back***, return to **Main Menu**.

Update Population of a City

Task Decomposition



Lock Types: One write-only lookup of the population information.

Number of Locks: Single.

Enabling Condition: Enabled by a user's request by entering valid population for any cities in the system.

Frequency: Low.

Consistency (ACID): Not critical.

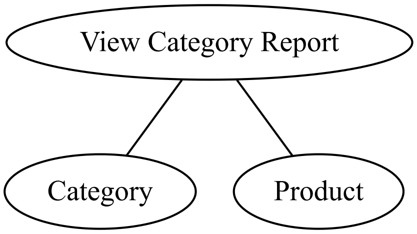
Subtasks: Mother task is not needed.

Abstract Code

- User clicked on ***Update Population of a City*** button from **Main Menu**;
- User selects a city to be updated;
- User enters *population* input field for the selected city, and clicks *Enter* button;
- Run **Update Population of a City** Task:
  - If entered data is not a positive integer, the data is invalid, so display error message;
  - If data validation is successful, update the city information in the database;
- When ready, the user selects next action from choices in **Main Menu**.

View Category Report

Task Decomposition



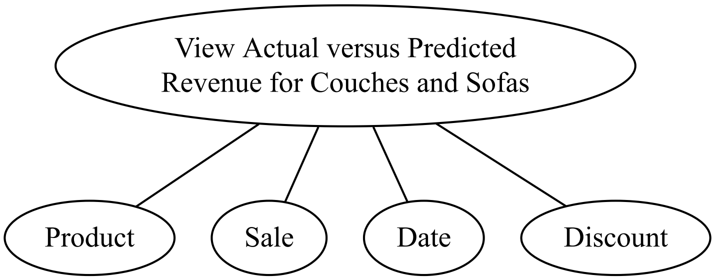
Lock Types: Read-only lookups of Product, Category.
Number of Locks: Two. Several different schema constructs are needed.
Enabling Condition: Enabled by clicking ***View Category Report*** Button.
Frequency: Low. Lookup of Product is more frequent than that of Category.
Consistency (ACID): Not critical.
Subtasks: Mother Task is required. Lookup of Category is followed by the lookup of corresponding Product.

Abstract Code

- User clicked on ***View Category Report*** button from **Main Menu**;
- Run the **View Category Report** task:
  - Run the **Category** task: Find name of each category;
  - For each category, using category name:
    - Run the **Product** task: Find product information that corresponds to certain category;
    - According to product information of certain category, calculate the minimum, maximum, and average retail price of all the products in that category;
- Sort the report by category name in ascending order;
- When ready, the user selects next action from choices in **Main Menu**.

View Actual versus Predicted Revenue for Couches and Sofas

Task Decomposition



Lock Types: Read-only lookups of Product, Sale, Date, and Discount.
Number of Locks: Four. Several different schema constructs are needed.
Enabling Condition: Enabled by clicking ***View Actual versus Predicted Revenue for Couches and Sofas*** Button.
Frequency: Low. Lookup of Sale is more frequent than of Product and Discount.
Consistency (ACID): Not critical.
Subtasks: Mother Task is required. Lookup of Product is followed by the lookup of Sale, which is followed by the lookup of Discount.

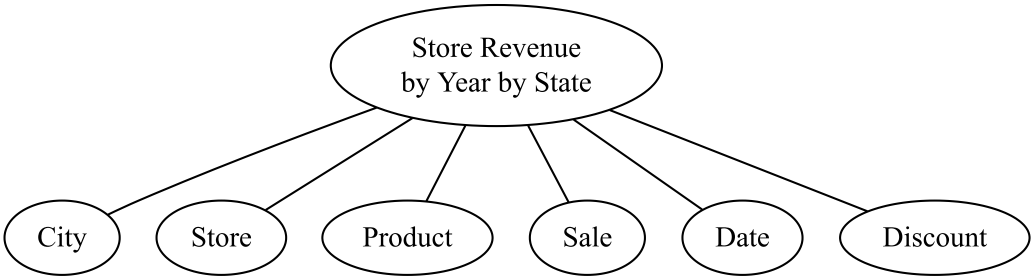
Abstract Code

- User clicked on ***View Actual versus Predicted Revenue for Couches and Sofas*** button from **Main Menu**;
- Run the **View Actual versus Predicted Revenue for Couches and Sofas** task:
  - Run the **Product** task:

        - Find the Product using "Couches and Sofas" category, then get the Product ID, the name of the Product, the retail price of the Product;
    - For each product of "Couches and Sofas", run the **Sale** task:
        - Find the Sale using Product ID;
    - For each Sale, run the **Date** task:
        - Find the Date corresponding to Sale;
    - For each Date and the Product, run the **Discount** task:
        - Find the Discount coresponding to the Date and the Product;
    - For each product in the "Couches and Sofas" category:
        - Calculate the actual revenue based on the price(either discount or retail price) and quantity;
        - Calculate the predicted revenue based on the retail price and assumed quantity (based on 75% volume actual selling);
        - Calculate the difference between the actual revenue and the predicted revenue;
        - Display and Store the predicted revenue differences greater than $5000 (positive or negative);
    - Sort the report by predicted revenue difference in descending order;
- When ready, the user selects next action from choices in **Main Menu**.

View Store Revenue by Year by State

Task Decomposition



Lock Types: Read-only lookups of Store, Sale, Discount, City, Product, Date.
Number of Locks: Six. Several different schema constructs are needed.
Enabling Condition: Enable by user's request by clicking the ***View Store Revenue by Year by State*** button. User must select *State* before viewing the report.
Frequency: Low. Lookups of Product, Sale, Date, and Discount are more frequent than of City and Store.
Consistency (ACID): Not critical.
Subtasks: Mother Task is required. Lookup of Product can be completed in parallel with lookup of City that is followed by lookup of Store. Then, the lookups of Sale and Date follow the lookups of Store and Product. Finally, lookup of the Discount is performed.

Abstract Code

- User clicked on ***View Store Revenue by Year by State*** button from **Main Menu**;

- User select *State* in the drop-down box;

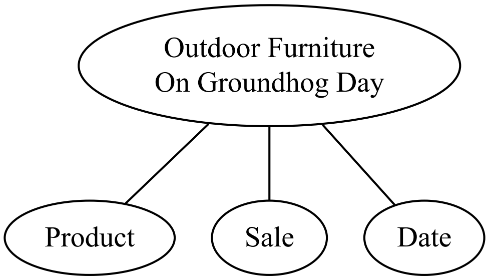
- Run the **Store Revenue by Year by State** task:
    - Run the **City** task:
        - Find all the City using *State*;
    - Run the **Product** task:
        - Find all Product;
    - For each City, run the **Store** task:
        - Find the Store using City;
    - For each Store and Product, run the **Sale** and **Date** task:
        - Find Sale and Date using Store and Product;

    - For each Sale and Date, run the **Discount** task:
        - Find the Discount corresponding to the Sale and Date;
    - For each store in the state:
        - Show the store ID, street address, city name, sales year;
        - Calculate total revenue;
    - Sort the report first by year in ascending order and then revenue in descending order;

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

View Report Outdoor Furniture on Groundhog Day

Task Decomposition



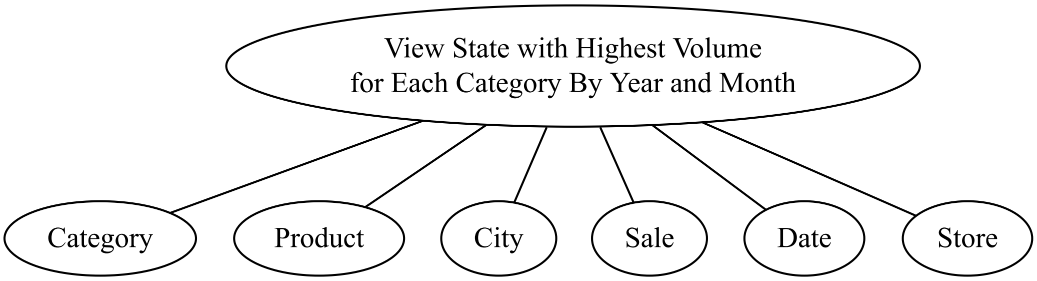
Lock Types: Read-only lookups of Product, Sale, Date.
Number of Locks: Three. Several different schema constructs are needed.
Enabling Condition: Enabled by clicking the ***View Outdoor Furniture on Groundhog Day*** button.
Frequency: Low. Lookups of Sale and Date are more frequent than of Product.
Consistency (ACID): Not critical. Lookups of Sale and Date are more frequent than the lookup of Product.
Subtasks: Mother task is required. Lookup of Product is followed by the lookup of Sale and Date.

Abstract Code

- User clicked on ***View Report Outdoor Furniture on Groundhog day*** button from **Main Menu**.
- Run the **View Outdoor Furniture on Groundhog Day** task:
    - Run the **Product** task:
        - Find the Product using "Outdoor Furniture" category;
    - For each Product, run the **Sale** and **Date** task:
        - Find Sale Quantity and Date of each Prodcut;
    - Using Date for each year:
        - Calculate the total number of items sold that year in the outdoor furniture category;
        - Calculate the average number of units sold per day that year(assume a year is exactly 365 days);
        - Calculte the total number of units sold on February 2 of that year;
- For each year, show if the total number of units sold on Groundhog Day (February 2) each year is significantly higher than the average number of units sold per day;
- Sort the report on the year in ascending order;
- When ready, the user selects next action from choices in **Main Menu**.

View State with Highest Volume for Each Category By Year and Month

Task Decomposition


View State with Highest Volume
for Each Category By Year and Month
Category
Product
City
Sale
Date
Store

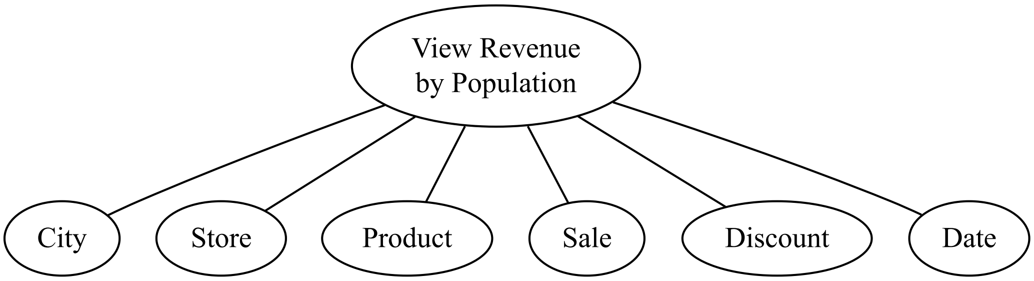
Lock Types: Read-only lookups of Category, Product, Sale, Date, Store and City.
Number of Locks: Six. Several different schema constructs are needed.
Enabling Condition: Enabled by clicking the ***View Outdoor Furniture on Groundhog Day*** button. User selected *Year* and *Month* from the drop-down menu.
Frequency: Low. Lookups of Sale, Store and Date are more frequent thant the other lookups.
Consistency (ACID): Not critical.
Subtasks: Mother task is required. Lookup of Date can be performed in parallel with the lookups of Category and of City. The lookup of Store follows the lookup of City, and the lookup of Product follows the lookup of Category. Finally, the lookup of Sale is performed.

Abstract Code

- User clicked on ***State with Highest Volume for Each Category By Year and Month*** button from **Main Menu**;
- User selected *Year* and *Month* from the drop-down menu;
- Run the **State with Highest Volume for Each Category By Year and Month** task:
  - Run the **Date** task:
    - Find all the Date;
  - Run the **Category** task:
    - Find all the category name;
  - For each Category, run the **Product** task
    - Find Product using Category name;
  - Run the **City** task:
    - Find all the City and corresponding state;
  - For each City, run the **Store** task:
    - Find all the Store;
  - Run the **Sale** task:
    - For each state:
      - Calculate the number of units sold in all stores of this state for each Category;
  - For each Category, display the state that sold the highest number of units in that Category and the number of units that were sold in that state;
- Sort by category name in ascending order
  - Each category will only be listed once unless two or more states tied for selling the highest number of units in that category;
- When ready, the user selects next action from choices in **Main Menu**.

View Revenue by Population

Task Decomposition



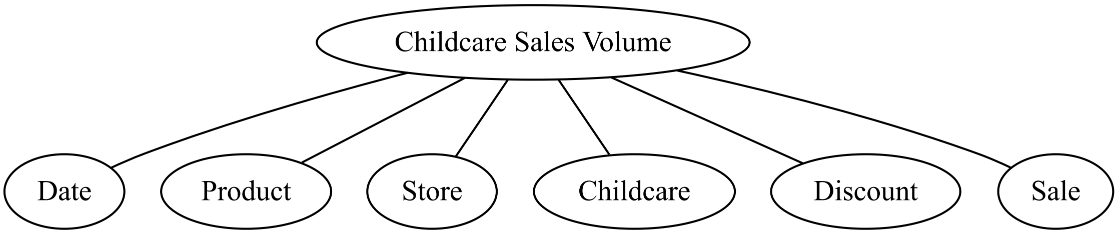
Lock Types: Read-only lookups of City, Store, Sale, Product, Date and Discount.
Number of Locks: Six. Several different schema constructs are needed.
Enabling Condition: Enabled by clicking the ***View Revenue by Population*** button.
Frequency: Low. Lookups of Sale, Discount and Date are more frequent than the lookups of City and Store.
Consistency (ACID): Not critical.
Subtasks: Mother Task is needed. The lookup of City can be performed in parallel with the lookups of Product and Date. Then, the lookup of Store follows the lookup of City, and the lookups of Sale and the Discount follow the lookup of Store.

Abstract Code

- User clicked on ***Revenue by Population*** button from **Main Menu**;
- Run the **Revenue by Population** task:
  - Run the **City** task:
    - Find all the City and sort them by "population": Small (population ♡,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);
  - Run the **Product** task:
    - Find all the Product;
  - For each year, run the **Date** task:
    - Find Date using year;
    - For each City:
      - Run the **Store** task:
        - Find Store using City;
      - For each Store and Product, run the **Sale** task:
        - Find Sale using Store and Product;
      - Run the **Discount** task:
        - Find Discount using Product and Date;
        - Find the retail price for each sold product;
      - Calculate the total revenue in each year;
  - Sort the row using year in ascending order and sort the column using city size in ascending order;
  - Display the total revenue for each year grouped by city categories;
- When ready, the user selects next action from choices in **Main Menu**.

View Childcare Sales Volume

Task Decomposition



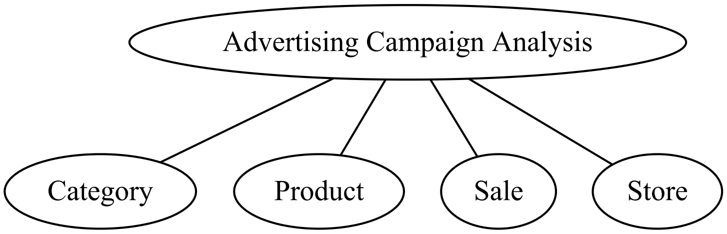
Lock Types: Read-only lookups of Store, Sale, Date, Product, Discount, and Childcare.
Number of Locks: Six. Several different schema constructs are needed.
Enabling Condition: Enabled by clicking the ***View Childcare Sales Volume*** button.
Frequency: Low. Lookups of Sale and Discount are more frequent than the other lookups.
Consistency (ACID): Not critical.
Subtasks: Mother Task is needed. Lookup of Date can be performed in parallel with the lookups of Product and Store. The lookup of Childcare follows the lookup of Store. Finally, the lookups of Discount and Sale are performed.

Abstract Code

- User clicks ***View Childcare Sales Volume*** button from **Main Menu**;
- Run the **Childcare Sales Volume** task:
    - Run the **Date** task:
        - Find all the Date of the last 12 months;
    - Run the **Product** task:
        - Find all the Product;
    - Run the **Store** task:
        - Find all the Store;
    - For each Store:
        - Run the **Childcare** task:
            - Find Timelimit using Childcare;
        - Run the **Sale**:
            - Find Sale using Store and Product;
        - Run the **Discount**:
            - Find Discount using Date, Sale and Product;
        - Calculate the total amount of sale of each month under each Timelimit category;
- Display the table with total amount of sale under each Timelimit for each month;
- When ready, the user selects next action from choices in **Main Menu**.

View Restaurant Impact on Category Sales

Task Decomposition



Lock Types: Read-only lookups of Category, Product, Sale, and Store.
Number of Locks: Four. Several different schema constructs are needed.
Enabling Condition: Enabled by clicking the ***View Restaurant Impact on Category Sales*** button.
Frequency: Low. Lookups of Sale and Product are more frequent than the other lookups.
Consistency (ACID): Not critical.
Subtasks: Mother Task is needed. The lookup of Category can be performed in parallel with the lookup of Store. Then the lookup of Product follows the lookup of Category. Finally, the lookup of Sale can be performed.

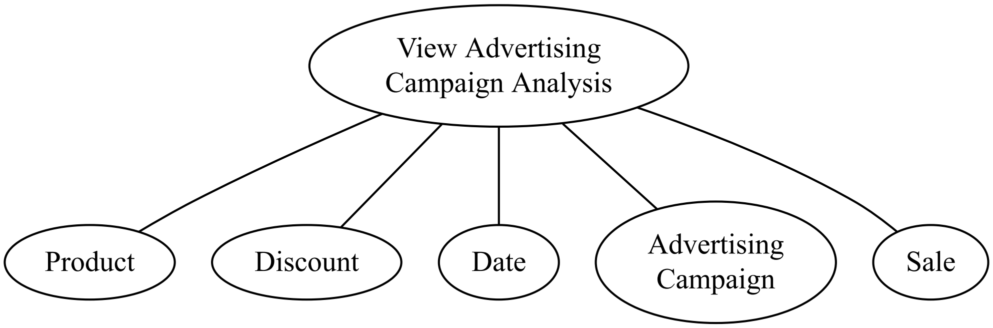
Abstract Code

- User clicked on ***View Restaurant Impact on Category Sales*** button from **Main Menu**;
- Run the **Restaurant Impact on Category Sales** task:

    - Run the **Category** and **Store** task:
        - Find the all the category name (may exclude any categories that are not assigned products) and if the store has a restaurant;
    - For each category:
        - Run the **Product** task:
            - Find the Product corresponding to the category;
            - For each Store without a restaurant:
                - Run the **Sale** task:
                    - Find the Sale using Store, Product;
            - For each Store with a restaurant:
                - Run the **Sale** task:
                    - Find the Sale using Store, Product;
        - Calculate the total quantity of all products sold in the stores without a restaurant; Display the store type as "Restaurant" and the total Quantity Sold;
        - Calculate the total quantity of all products sold in the stores with a restaurant; Display the store type as "Non-restaurant" and the total Quantity Sold;
        - Calculate the total quantity of all products sold in the category;
- Sort the the results by category name in ascending order and with non-restaurant store data listed first;
- When ready, the user selects next action from choices in **Main Menu**.

View Advertising Campaign Analysis

Task Decomposition



Lock Types: Read-only lookups of Date, Product, Sale, Discount and Advertising Campaign.
Number of Locks: Five. Several different schema constructs are needed.
Enabling Condition: Enabled by clicking the ***View Advertising Campaign Analysis*** button.
Frequency: Low. Lookups of Sale and Discount are more frequent than the other lookups.
Consistency (ACID): Not critical.
Subtasks: Mother Task is needed. The lookup of Product is followed by the lookup of Discount. Then, the lookups of Date can be performed and be followed by the lookup Advertising Campaign. Finally, the lookup of Sale can be performed.

Abstract Code

- User clicked on ***Advertising Campaign Analysis*** button from **Main Menu**;
- Run the **Advertising Campaign Analysis** task:
    - Run the **Product** task:
        - Find the Product ID and Product Name;
    - For each Product, run the **Discount** task:
        - Run the **Discount** task:
            - Find Discount using Product
    - Run the **Date** task:

    - Find Date using Discount;
  - For each Date, run the **Advertising Campaign**:
    - Find Advertising Campaign dates using Date;
  - For each Product, run the **Sale** task:
    - Find Sale Quantity;
    - Calculate total sale quantity when Advertising Campaign is active and when Advertising Campaign is not active;
    - Compute the difference between the two total sales;
  - Display the results for each Product：
    - Sort the the results by difference in descending order；
    - Only display the top 10 and the bottom 10 results;
- When ready, the user selects next action from choices in **Main Menu**.