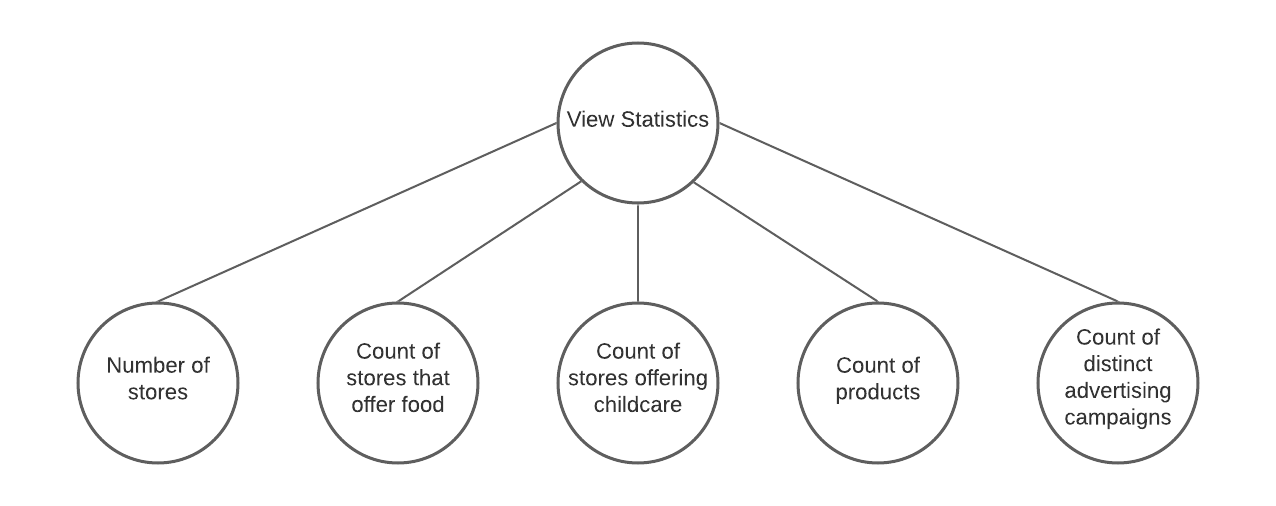
Statistics View



Task Decomp

**Lock Types**: Read-only on Store, Childcare, Product, and Advertising\_Campaigns constructs/entities/tables

**Number of Locks**: 4 – 4 different schema constructs/entities/tables are needed. All read-only.

**Enabling Conditions**: None – all four are available to show once the page is loaded and menu is displayed.

**Frequency**: High – all four statistics are shown every time when the main page is loaded

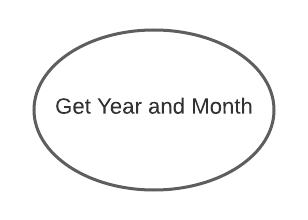
**Consistency (ACID)**: Not critical, order is not critical.

**Subtasks**: All tasks must be done but can be done in parallel. Mother task is required to coordinate subtasks. Order is not necessary.

Abstract Code

* Show *“****View Holidays****”, “****View Product by Category Report****”, “****View Actual vs. Predicted Revenue for Couches and Sofas Report****”, “****View Store Revenue by Year by State Report****”, “****View GroundHog Day Outdoor Furniture Report****”, “****View State with Highest Volume by Category Report****”, “****View Revenue by Population Report****”, “****View Childcare Sales Volume Report****”, “****View Restaurant Impact on Category Sales Report****”, “****View Advertising Campaign Analysis Report****”,* and *“****Maintain Population****”* buttons/links on the **View Statistics** form.
* Upon:
  + Click ***View Holidays*** button – Jump to the **View Holidays** task.
  + Click ***View Product by Category Report*** button – Jump to the **View Product by Category Report** task.
  + Click ***View Actual vs. Predicted Revenue for Couches and Sofas Report*** button – Jump to the **View Actual vs. Predicted Revenue for Couches and Sofas Report** task.
  + Click ***View Store Revenue by Year by State Report*** button – Jump to the **Get Available State List** task.
  + Click ***View GroundHog Day Outdoor Furniture Report*** button – Jump to the **View GroundHog Day Outdoor Furniture Report** task.
  + Click ***View State with Highest Volume by Category Report*** button – Jump to the **Get Year and Month List** task.
  + Click ***View Revenue by Population Report*** button – Jump to the **View Revenue by Population Report** task.
  + Click ***View Childcare Sales Volume Report*** button – Jump to the **View Childcare Sales Volume** **Report** task.
  + Click ***View Restaurant Impact on Category Sales Report*** button – Jump to the **View Restaurant Impact on Category Sales Report** task.
  + Click ***View Advertising Campaign Analysis Report*** button – Jump to the **View Advertising Campaign Analysis Report** task.
  + Click ***Maintain Population*** button – Jump to the **Maintain Population** task.
* Display statistics for “the 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”, and “count of distinct advertising campaigns” on the **View Statistics** form.
  + Show “the count of stores” .
    - Query for total count of Store\_Number in the Store table. Store that variable in (‘$NumberOfStores’) and return that as the count of stores.
  + Show “count of stores that offer food”.
    - Query for each store and check the value for Has\_Restaurant and Has\_Snack\_Bar, increment the count each time when a store has either Has\_Restaurant or Has\_Snack\_Bar as True. Store that total count in variable (‘$NumberOfStoresOfferingFood’) and return that variable.
  + Show “count of stores offering childcare”.
    - Query for each store and increment the count each time when a store has a Childcare center association. Store that total count in variable (‘$NumberOfStoresWithChildcare’) and return that variable.
  + Show “count of products”.
    - Query for the total count of Product\_Id (PID) in the Product table. Store that total count in variable (‘$NumberOfProducts’) and return that as the total count of products.
  + Show “count of distinct advertising campaigns”.
    - Query for the total count of Description in Advertising\_Campaigns table. Store that total count in (‘$NumberOfAdCampaigns’) and return that as the variable.

Get Year and Month List



Task Decomp

**Lock Types**: Read-only on Day table

**Number of Locks**: Single

**Enabling Conditions**: Triggered when ***View State with Highest Volume by Category Report*** button is clicked.

**Frequency**: Low – only triggered by user action.

**Consistency (ACID)**: Not critical, order is not critical.

**Subtasks**: Mother Task is not needed. No decomposition needed.

Abstract Code

* User clicked on the ***View State with Highest Volume by Category Report*** button from the **View Statistics** form.
* Run the **Get Year and Month** task: query for information about the available *year* and *month* fields from the Day table.
  + Store all *year* data in the array (‘$Year’) and all *month* data in the array (‘$Month’), both in descending order.
  + Return both (‘$Year’) and (‘$Month’) variables to the drop-down list UI form.
* On the drop-down list UI form, show ***Run Report*** button
* Upon:
  + Click ***Run Report*** button –
    - If both *year* and *month* are selected – Jump to the **View State with Highest Volume by Category Report** task.
    - If one or both fields are empty – display a message asking for user input.

View State with Highest Volume by Category Report



Task Decomp

**Lock Types**: Read-only on City, Day, Sale, Category, and Store table

**Number of Locks**: 5 – 5 different schema constructs/entities/tables are needed. All read-only.

**Enabling Conditions**: Triggered when ***Run Report*** button from the drop-down list UI form is clicked.

**Frequency**: Low – only triggered by user action.

**Consistency (ACID)**: Not critical, order is not critical.

**Subtasks**: Mother Task is not needed. No decomposition needed.

Abstract Code

* User clicked on the ***Run Report*** button from the drop-down list UI form and both the *year* and *month* fields are not null.
* Run the **View State with Highest Volume by Category Report** task:
  + Query all Sales data from the Sale table based on the *year* and *month* selected.
  + Get all Category\_Name data from the Category table and store that in an array (‘$CategoryName’).
  + For each Category\_Name in the (‘$CategoryName’) array
    - Find sales records with each Category\_Name
      * If the sales records from that Category\_Name have the same state, then aggregate their quantities sold together.
      * Find the state with the greatest number of units for each Category\_Name.
        + If two or more states tied for having the greatest number of units, then save all those states and their total units.
    - Store the states and their total units for each Category\_Name.
  + Display each category name, its corresponding states with the highest units sold and total units sold.
  + Sort by category name ascending.