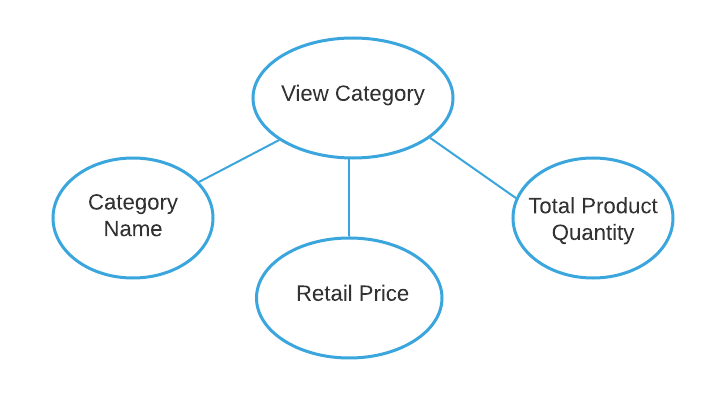
//Report 1

View Category

Task Decomp



**Lock Types**: 3 read-only lookups of Category Name, Retail Price, and Total Product Quantity

**Number of Locks:** Several different schema constructs are needed

**Enabling Conditions:** All 3 are enabled when report loading transaction is triggered

**Frequency**: Low- All 3 have the same frequency

**Consistency (ACID):** Not critical, even if the category is being edited while a LEOFURN executive is looking at it.

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

Abstract Code

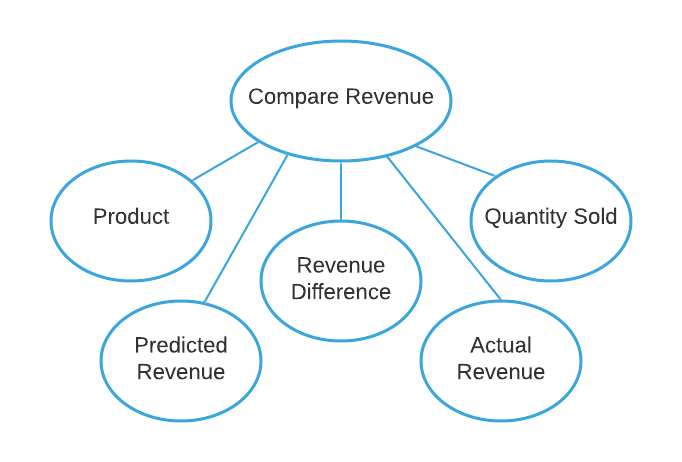
* User clicked on ***View Category button/Link*** from **Main Menu**:
* Run the **View Category** task: query for each category including those without products, find and display Category\_Name in Ascending Order.
  + Find minimum/average/maximum Retail Price for the PRODUCT using Retail\_Price; Display minimum/average/maximum Retail Price;
  + Find Total Product Quantity for the category using the SALE; Display Total\_Amount;

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

//Report 2

Compare Revenue

Task Decomp



**Lock Types**: 5 read-only lookups of Product, Quantity Sold, Predicted Revenue, Actual Revenue, and Revenue Difference for the Couches and Sofas category.

**Number of Locks:** Several different schema constructs are needed

**Enabling Conditions:** All 5 are enabled when report loading transaction is triggered

**Frequency**: Different frequencies

**Consistency (ACID):** Critical, the result of Revenue Difference depends on Predicted Revenue and Actual Revenue.

**Subtasks:** All 5 tasks must be done, so mother task is needed to coordinate subtasks. Order is necessary.

Abstract Code

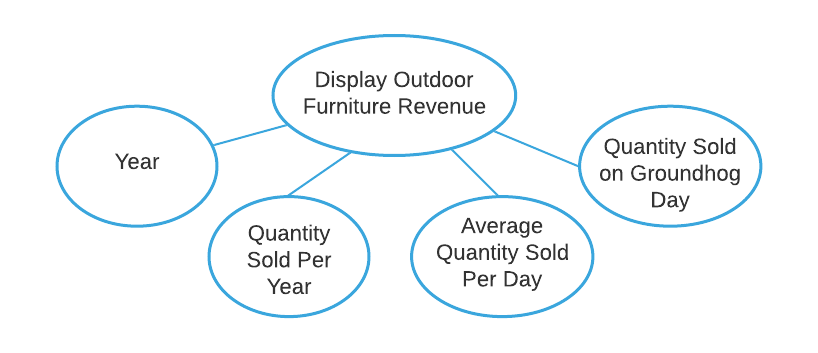
* User clicked on ***Compare Revenue button/Link*** from **Main Menu**:
* Run the **Compare Revenue** task: query for each product in the Couches and Sofas category.
  + Find each Product using the PRODUCT; Display PID, Product\_Name, and Retail\_Price;
  + Find Quantity Sold for the product using SALE and DISCOUNT; Display Quantity and Discount\_Price;
  + Find and display Actual Revenue:
    - Calculate Actual Revenue by multiply Discount\_Price and Quantity;
  + Find and display Predicted Revenue:
    - Calculate Predicted Revenue by multiply Retail\_Price and 75% Quantity;
  + Find Revenue Difference:
    - Calculate Revenue Difference by subtract Actual Revenue from Predicted Revenue;
    - If Revenue Difference greater than $5000 (positive or negative): Display Revenue Difference in Descending Order;

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

//Report 4

Display Outdoor Furniture Revenue

Task Decomp



**Lock Types**: 4 read-only lookups of Year, Quantity Sold Per Year, Average Quantity Sold Per Day, and Quantity Sold on Groundhog Day.

**Number of Locks:** Several different schema constructs are needed

**Enabling Conditions:** All 4 are enabled when report loading transaction is triggered

**Frequency**: Different frequencies

**Consistency (ACID):** Not critical, even if the outdoor furniture revenue is being edited while a LEOFURN executive is looking at it.

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

Abstract Code

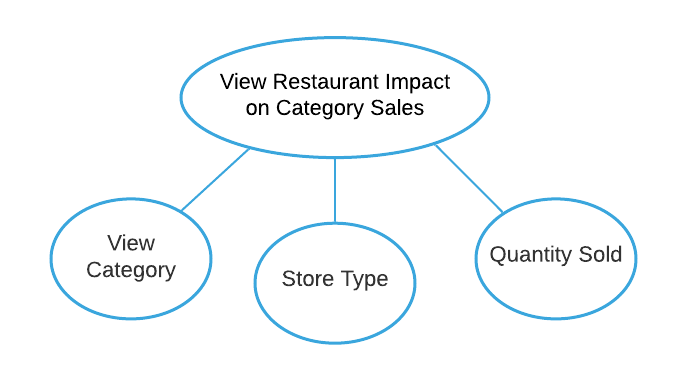
* User clicked on ***Display Outdoor Furniture Revenue button/Link*** from **Main Menu**:
* Run the **Display Outdoor Furniture Revenue** task: query for information about Quantity Sold Per Year and Quantity Sold on Groundhog Day (Feb 2) in the outdoor furniture category.
* Find and display Year using DAY in Ascending Order;
* For each year:
  + - Find Quantity Sold using the SALE;
    - Return Quantity and display Quantity Sold Per Year;
  + Find and display Average Quantity Sold Per Day:
    - Calculate Average Quantity Sold Per Day by divide Quantity Sold Per Year by 365;
  + Find the Groundhog Day using the DAY and return Date;
  + Find Quantity Sold occurred on Date using the SALE;
    - Return Quantity and display Quantity Sold on Groundhog Day;

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

//Report 8

View Restaurant Impact on Category Sales

Task Decomp



**Lock Types**: Read-only lookups of Store Type and Quantity Sold for each category (use: View Category)

**Number of Locks:** Several different schema constructs are needed

**Enabling Conditions:** All 3 are enabled when report loading transaction is triggered

**Frequency**: Low- All 3 have the same frequency

**Consistency (ACID):** Not critical, even if the category sale is being edited while a LEOFURN executive is looking at it.

**Subtasks:** All 3 tasks must be done, so mother task is needed to coordinate subtasks. Order is not matter.

Abstract Code

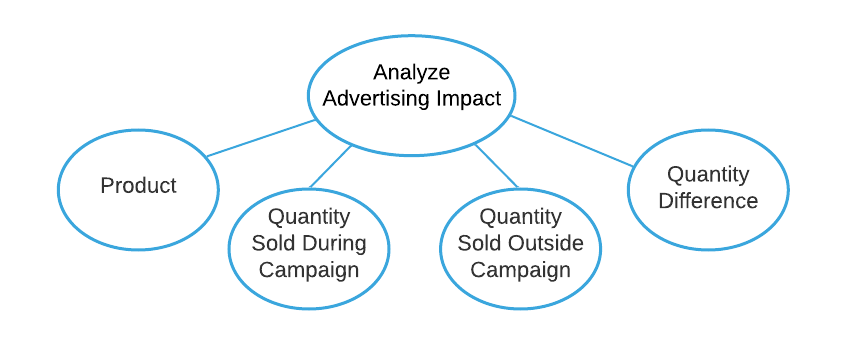
* User clicked on ***View Restaurant Impact on Category Sales button/Link*** from **Main Menu**:
* Run the **View Restaurant Impact on Category Sales**task: query for all sales data for each category.
* **View Category**; Populate all Category\_Name that has products in that category;
* Find Store Type and Quantity Sold for each category using STORE and SALE;
  + - If a store Has\_Restaurant is TRUE:
    - Display the Store Type as “Restaurant” and Quantity;
  + If a store Has\_Restaurant is FALSE:
    - Display the Store Type as “Non-Restaurant” and Quantity;
* Display Category\_name, Store Type, and Quantity by Category\_name in Ascending Order and with “Non-Restaurant” Store Type first;

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

//Report 9

Analyze Advertising Impact

Task Decomp



**Lock Types**: 4 read-only lookups of Product, Quantity Sold During Campaign, Quantity Sold Outside Campaign, and Quantity Difference for all products.

**Number of Locks:** Several different schema constructs are needed

**Enabling Conditions:** All 4 are enabled when report loading transaction is triggered

**Frequency**: Low- All 4 have the same frequency

**Consistency (ACID):** Critical, the result of Quantity Difference depends on Quantity Sold During Campaign and Quantity Sold Outside Campaign

**Subtasks:** All 4 tasks must be done, so mother task is needed to coordinate subtasks. Order is necessary.

Abstract Code

* User clicked on ***Analyze Advertising Impact button/Link*** from **Main Menu**:
* Run the **Analyze Advertising Impact**task: query for information about Quantity Sold During and Outside Campaign for all products.
* Find Discount\_Price for the product using DISCOUNT;
  + while a Product has Discount\_Price:
    - Find and display PID, Product\_Name using PRODUCT;
    - Find Quantity Sold During Campaign using SALE, DAY, and ADVERTISING\_CAMPAIGN; Display Quantity;
    - Find Quantity Sold Outside Campaign for the product using SALE and DAY; Display Quantity;
    - Find Quantity Difference by subtract Quantity Sold Outside Campaign from Quantity Sold During Campaign; Return Quantity Difference in Descending Order;
    - Display Top 10 followed by Bottom 10 Quantity Difference;

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