//Report 1

View Product by Category Report

Task Decomp



**Lock Types**: 2 Read-only on CATEGORY and PRODUCT tables

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

**Enabling Conditions**: Triggered when ***View Product by Category Report*** button is clicked.

**Frequency**: Low.

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

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

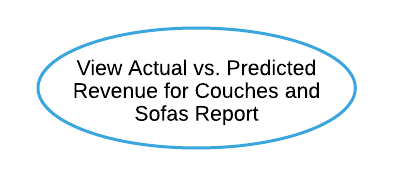
Abstract Code

* User clicked on the ***View Product by Category Report*** button from the **Dashboard** form.
* Run the **View Product by Category Report** task: query for each category from CATEGORY and PRODUCT tables, including those without products.
  + Get all Category\_Name data (from the CATEGORY table).
  + For each category:
    - Find minimum, average, and maximum Retail\_Price data for all products (from the PRODUCT table).
    - Find total number of products by counting their PID data (from the PRODUCT table).
  + Sort by category name in ascending order.
* When ready, user can click on the ***Return*** button to return to the **Dashboard** form.

//Report 2

View Actual vs. Predicted Revenue for Couches and Sofas Report

Task Decomp



**Lock Types**: 3 Read-only on PRODUCT, SALE, and DISCOUNT tables

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

**Enabling Conditions**: Triggered when ***View Actual vs. Predicted Revenue for Couches and Sofas Report*** button is clicked.

**Frequency**: Low.

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

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

Abstract Code

* User clicked on the ***View Actual vs. Predicted Revenue for Couches and Sofas Report*** button from the **Dashboard** form.
* Run the **View Actual vs. Predicted Revenue for Couches and Sofas Report**task: query for each product in the Couches and Sofas category from PRODUCT, SALE, and DISCOUNT tables.
  + Get PID, Product\_Name, and Retail\_Price data for each product (from the PRODUCT table).
  + Get Total\_Amount ever sold data (from the SALE table), sold at a discount data by using Quantity and Discount\_Price (from SALE and DISCOUNT tables), sold at retail price data by using Quantity and Retail\_Price (from SALE and PRODUCT tables),
  + Find actual revenue by multiplying Quantity and Discount\_Price (from SALE and DISCOUNT tables).
  + Find predicted revenue by multiplying 75% Quantity and Retail\_Price (from SALE and PRODUCT tables).
  + Find revenue difference by subtracting predicted revenue from actual revenue.
    - If revenue difference is greater than $5000 (positive or negative):
      * Display and sort revenue difference in descending order.
* When ready, user can click on the ***Return*** button to return to the **Dashboard** form.

//Report 4

View Groundhog Day Outdoor Furniture Report

Task Decomp



**Lock Types**: 2 Read-only on DAY and SALE tables

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

**Enabling Conditions**: Triggered when ***View Groundhog Day Outdoor Furniture Report*** button is clicked.

**Frequency**: Low.

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

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

Abstract Code

* User clicked on the ***View Groundhog Day Outdoor Furniture Report*** button from the **Dashboard** form.
* Run the **View Groundhog Day Outdoor Furniture Report** task: query for information about total number of products sold in the outdoor furniture category from DAY and SALE tables.
* In the outdoor furniture category, get Year data (from the DAY table).
* For each year:
  + - Get Total\_Amount data (from the SALE table).
    - Find average number of products sold per day by dividing Total\_Amount by 365;
    - Find total number of products sold on Groundhog Day (Feb 2) using Quantity and Date (from SALE and DAY tables).
  + Sort by Year in ascending order.
* When ready, user can click on the ***Return*** button to return to the **Dashboard** form.

//Report 8

View Restaurant Impact on Category Sales Report

Task Decomp



**Lock Types**: 2 Read-only on tables

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

**Enabling Conditions**: Triggered when ***View Restaurant Impact on Category Sales Report*** button is clicked.

**Frequency**: Low.

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

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

Abstract Code

* User clicked on the ***View Restaurant Impact on Category Sales Report*** button from the **Dashboard** form.
* Run the **View Restaurant Impact on Category Sales Report** task: query for information about all sales data for each category from CATEGORY, STORE, and SALE tables.
  + Get all Category\_Name data that have related sales data of Total\_Amount (from CATEGORY and SALE tables).
  + For each category:
    - Find Store\_Number data (from the STORE table) and quantity sold by using Total\_Amount data (from the SALE table).
    - If this store Has\_Restaurant (from the STORE table) is TRUE:
      * Display the store type as “Restaurant”.
    - Else:
      * Display the store type as “Non-Restaurant”.
* Sort by Category\_Name in ascending order and with “Non-Restaurant” store type first.
* When ready, user can click on the ***Return*** button to return to the **Dashboard** form.

//Report 9

View Advertising Campaign Analysis Report

Task Decomp



**Lock Types**: 5 Read-only on DISCOUNT, PRODUCT, SALE, DAY, and ADVERTISING\_CAMPAIGN tables

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

**Enabling Conditions**: Triggered when ***View Advertising Campaign Analysis Report*** button is clicked.

**Frequency**: Low.

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

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

Abstract Code

* User clicked on the ***View Advertising Campaign Analysis Report*** button from the **Dashboard** form.
* Run the **View Advertising Campaign Analysis Report** task: query for information about quantity sold during and outside campaign for all products from DISCOUNT, PRODUCT, SALE, DAY, and ADVERTISING\_CAMPAIGN tables.
  + Get Discount\_Price data for all products (from the DISCOUNT table).
  + while a Product has Discount\_Price:
    - Get PID, Product\_Name data (from the PRODUCT table).
    - Find quantity sold during and outside campaign by using Total\_Amount data (from SALE, DAY, and ADVERTISING\_CAMPAIGN tables).
    - Find difference by subtracting quantity sold outside campaign from quantity sold during campaign.
  + Sort by difference in descending order and only display the top 10 followed by the bottom 10.
* When ready, user can click on the ***Return*** button to return to the **Dashboard** form.