Table of Contents:

View Statistics

Abstract Code

- Show ***"Holiday Maintenance"***, ***"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 ***"Population Maintenance"*** buttons/links on the **Dashboard** form.
- Upon:
    - Click ***Holiday Maintenance*** button – Jump to the **Holiday Maintenance** form.
    - 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 ***Population Maintenance*** button – Jump to the **Population Maintenance** form.

- Display statistics for “the count of stores”, “count of stores offering 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 **Dashboard** form.
    - Show “the count of stores”.
        - Query for total count of Store_Number in the STORE table.
        - Display the total count.
    - Show “count of stores offering food”.
        - Query for the total count of Store_Number in the STORE table that has either or both Has_Restaurant and Has_Snack_Bar value as true.
        - Display the total count.
    - Show “count of stores offering childcare”.
        - Query for the total count of Store_Number in the STORE table that has a Childcare center association in the CHILDCARE table.
        - Display the total count.
    - Show “count of products”.
        - Query for the total count of PID in the PRODUCT table.
        - Display the total count.
    - Show “count of distinct advertising campaigns”.
        - Query for the total count of Description in ADVERTISING_CAMPAIGNS table.
        - Display the total count.

Get Holiday List

Abstract Code

- User clicked on the ***Holiday Maintenance*** button from the **Dashboard** form.
- Run the **Get Holiday List** task: query for information about the available Name field from the HOLIDAY table.
    - Display the holiday name list.
- Upon:
    - User enters Holiday Name (‘$HolidayName’) in input textbox and selects Date (‘$HolidayDate’) in Calendar Dropdown.
    - Click ***Add Holiday*** button –
        - Jump to the **Add Holiday** task.
- When ready, user can click on the ***Return*** button to return to the **Dashboard** form.

Add Holiday

Abstract Code

- User enters Holiday Name ('$HolidayName') in input textbox and selects Date ('$HolidayDate') in Calendar Dropdown.
- Click ***Add Holiday*** button.
- Run the **Add Holiday** task:
    - If data validation passed for both holiday name and date in Client Side, then:
        - If same holiday name and same holiday date exist:
            - Go back to **Holiday Maintenance** form and show the failure message that this holiday with this date existed.
        - If holiday name does not exist but date exists:
            - Store the holiday name in HOLIDAY table and link its date with DAY table.
            - Go back to **Holiday Maintenance** form and show success message.
        - If both holiday name and date do not exist:
            - Store the holiday name and date in both HOLIDAY and DAY tables.
            - Go back to **Holiday Maintenance** form and show success message.
    - Else: display the invalid error message in **Holiday Maintenance** form.
- When ready, user can click on the ***Return*** button to return to the **Dashboard** form.

Maintain Population

Abstract Code

- User clicked on the ***Population Maintenance*** button from the **Dashboard** form.
- Run the **Get City List** task: query for information about the available *State_Location*, *City_Name,* and *Population* fields from the CITY table.
    - Display *State_Location* list in ascending order in the drop-down list.
    - Select state and then display the available *City_Name* (from the CITY table) list in the drop-down list.
    - Select target *City_Name*, and display *Population* in view population textbox.

- User edits population textbox.
- Upon:
    - Click ***Update Population*** button –
        - Jump to the **Update Population** task.
- Run the **Update Population** task.
    - If data validation passed for *Population* in Client Side, then:
        - If the updated population entered is the same as the original population, do nothing.
        - Else if the updated population entered is different from the original population, update the *Population* and *Population_Size_Category* in the CITY table.
        - Else updated population is not entered, populate a message asking for user input.
    - Else: display the invalid error message in **Population Maintenance** form.
- When ready, user can click on the ***Return*** button to return to the **Dashboard** form.

View Product by Category Report

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, ASSIGNED, and PRODUCT tables, including those without products.
    - Find all Category_Name data (from the CATEGORY table).
    - For each Category_Name including those without products:
        - Find total number of products by counting their PID data (from the PRODUCT table).
        - Find minimum, average, and maximum Retail_Price data for all products (from the PRODUCT table).
    - Sort by Category_Name in ascending order.

(Sample TextBox – To Be Deleted)

```
SELECT first_name, last_name, gender, birthdate, current_city, home_town
FROM `User`
INNER JOIN RegularUser ON `User`.email=RegularUser.email
WHERE `User`.email='$UserID';
```

```
SELECT C.Category_Name, COUNT(P.PID) AS Cnt_Product,
MIN(P.Retail_Price) AS Min_RtlPrc, AVG(P.Retail_Price) AS Avg_RtlPrc,
MAX(P.Retail_Price) AS Max_RtlPrc
FROM CATEGORY AS C
LEFT JOIN ASSIGNED AS A ON C.Category_Name = A.Category_Name
LEFT JOIN PRODUCT AS P ON A.PID = P.PID
GROUP BY C.Category_Name
ORDER BY C.Category_Name ASC;
```

- When ready, user can click on the ***Return*** button to return to the **Dashboard** form.

View Actual vs. Predicted Revenue for Couches and Sofas Report

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:
    - Find the 'Couches and Sofas' category (from the CATEGORY table).
    - For each product in 'Couches and Sofas' category:
        - Find PID, Product_Name, and Retail_Price data (from the PRODUCT table).
        - Find the total number of units ever sold by aggregating Quantity (from the SALE table).
        - Find the total number of units sold at a discount by aggregating Quantity (from the SALE table) when the product has a Discount_Price (from the DISCOUNT table).

- Find the total number of units sold at retail price by aggregating Quantity (from the SALE table) when the product doesn't have a Discount_Price (from the DISCOUNT table).
- Find the actual revenue by aggregating Total_Amount (from the SALE table).
- Find the predicted revenue by aggregating the following multiplication result: Retail_Price (from the PRODUCT table) * Quantity (from the SALE table) * quantity multiplier. The quantity multiplier equals 0.75 when the product has a Discount_Price (from the DISCOUNT table), otherwise, the quantity multiplier equals 1.
- Find the revenue difference by subtracting predicted revenue from actual revenue.

- If the revenue difference is greater than $5000 (positive or negative): Display revenue difference and sort in descending order.

```
SELECT P.PID, P.Product_Name, P.Retail_Price,
SUM(IFNULL(S.Quantity,0)) AS Tot_UnitSold,
SUM(IF(D.Discount_Price IS NULL,0,1) * IFNULL(S.Quantity,0)) AS
Tot_UnitSold_AtDsct,
SUM(IF(D.Discount_Price IS NULL,1,0) * IFNULL(S.Quantity,0)) AS
Tot_UnitSold_AtRtl,
SUM(IFNULL(S.Total_Amount,0)) AS Act_Revenue,
SUM(P.Retail_Price * IFNULL(S.Quantity,0) * IF(D.Discount_Price IS NULL,
1, 0.75)) AS Pred_Revenue,
(SUM(IFNULL(S.Total_Amount,0)) - SUM(P.Retail_Price *
IFNULL(S.Quantity,0) * IF(D.Discount_Price IS NULL, 1, 0.75))) AS
Diff_Act_Pred_Revenue
FROM CATEGORY AS C
LEFT JOIN ASSIGNED AS A ON C.Category_Name = A.Category_Name
LEFT JOIN PRODUCT AS P ON A.PID = P.PID
LEFT JOIN SALE AS S ON P.PID = S.PID
LEFT JOIN DISCOUNT AS D ON S.Date = D.Date AND S.PID = D.PID
WHERE C.Category_Name = 'Couches and Sofas'
GROUP BY P.PID
HAVING Diff_Act_Pred_Revenue > 5000 OR Diff_Act_Pred_Revenue < -
5000
ORDER BY Diff_Act_Pred_Revenue DESC;
```

- When ready, user can click on the ***Return*** button to return to the **Dashboard** form.

Get Available State List

Abstract Code

- User clicked on ***View Store Revenue by Year by State Report*** button from the **Dashboard** form.
- Run the **Get Available State List** task: query for information about the available *State_Location* field from the CITY table.
    - Display *State_Location* list in ascending order on the drop-down list.
- On the drop-down list, show ***Run Report*** button.
- Upon:
    - Click ***Run Report*** button –
        - If *State_Location* is selected – Jump to the **View State Revenue by Year by State Report** task.
        - If *State_Location* field is empty – Display a message asking for user input.
- When ready, user can click on the ***Return*** button to return to the **Dashboard** form.

View Store Revenue by Year by State Report

Abstract Code

- User clicked on the ***Run Report*** button from the drop-down list.
- If data validation is successful for *State_Location* input field, then proceed.
- Run the **View State Revenue by Year by State Report** task:
    - Find all stores from the STORE table based on the *State_Location* (from the CITY table) selected.
    - On every sale *date* (from the DAY table), find each product's sale revenue based on Total_Amount (from the SALE table).
        - Total_Amount (from the SALE table) is calculated based on the Date purchased (from the DAY table), the Quantity (from the SALE table), and individual item price.
        - Individual item prices can be determined by Retail_Price (from the PRODUCT table) or Discount_Price (from the DISCOUNT table) when the product has a discount.

    - In each year, at each store in the selected state, find the total revenue by aggregating all products' sale revenue on every sale date in that year.
    - Sort by sale year ascendingly, then sort by total revenue in descending order for each store in the selected state, display Store_Number (from the STORE table), Street_Address (from the STORE table) of the store, and City_Name (from the CITY table).
- When ready, user can click on the ***Return*** button to return to the **Dashboard** form.

View Groundhog Day Outdoor Furniture Report

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:
    - Get and return the year (from the DAY table).
    - For each year:
        - Get the outdoor furniture category (from the CATEGORY table).
        - Get Quantity data for the products sold (from the SALE table) at the specific Date that year (from the DAY table); Find total number of products sold by aggregating Quantity in all sale days that year.
        - Find average number of products sold per day by dividing total number of products sold 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.

Get Year and Month List

Abstract Code

- User clicked on the ***View State with Highest Volume by Category Report*** button from the **Dashboard** form.
- Run the **Get Year and Month List** task: query for information about the available *year* and *month* fields from the DAY table.
    - Display both *year* and *month* lists in descending order on the drop-down list.

- On the drop-down list, 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.
- When ready, user can click on the ***Return*** button to return to the **Dashboard** form.

View State with Highest Volume by Category Report

Abstract Code

- User clicked on the ***Run Report*** button from the drop-down list.
- If data validation is successful for both *year* and *month* input fields, then proceed.
- Run the **View State with Highest Volume by Category Report** task:
    - Find all Sales data from the SALE table based on the *year* and *month* (from the DAY table) selected.
    - Get all categories by using Category_Name data (from the CATEGORY table).
    - In each category, aggregate each state's sales Quantity data (from the SALE table) by adding up all stores (from the STORE table)'s sales Quantity in each state (from the CITY table).
    - Find the states that sold the highest number of units in each category.
        - If two or more states tied for having the greatest number of units, then save all those states and their total units.
    - Sort by category name ascendingly, display each category name, its corresponding states with the highest units sold and total units sold.
- When ready, user can click on the ***Return*** button to return to the **Dashboard** form.

View Revenue by Population Report

Abstract Code

- User clicked on the ***View Revenue by Population Report*** button from the **Dashboard** form.
- Run the **View Revenue by Population Report** task:

    - Find the year, Population_Size_Category and sale's Total_Amount from the DAY, SALE, STORE, and CITY tables.
    - Group by year and Population_Size_Category, and then aggregate the Total_Amount as total revenue.
    - Display Total Revenue in a tabular form:
        - Row: year, sorted in ascending order.
        - Column: Population_Size_Category, sorted in ascending order.
- When ready, user can click on the ***Return*** button to return to the **Dashboard** form.

View Childcare Sales Volume Report

Abstract Code

- User clicked on the ***View Childcare Sales Volume Report*** button from the **Dashboard** form.
- Run the **View Childcare Sales Volume Report** task:
    - Find and aggregate the store monthly sales from the SALE and STORE tables based on the Store_Number (from the STORE table) and month (Date from the DAY and SALE tables) based on the last 12 months.
    - Find stores with and without childcare category from the STORE and CHILDCARE tables.
    - Group the total sales by month and by childcare category (no childcare offer will be grouped as category "No childcare").
    - Display Total Sale in a tabular form:
        - Row: month, sorted in ascending order.
        - Column: childcare category.
- When ready, user can click on the ***Return*** button to return to the **Dashboard** form.

View Restaurant Impact on Category Sales Report

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:
    - Get all Category_Name data that has product(s) assigned (from CATEGORY and PRODUCT tables).
    - For each category:

        - Get products in this category (from CATEGORY and PRODUCT tables).
        - Find stores with Has_Restaurant (from the STORE table) = TRUE:
            - Display store type with value “Restaurant”.
            - Find total quantity sold by aggregating all Quantity (from the SALE table) of all products for these stores.
        - Find stores with Has_Restaurant (from the STORE table) = FALSE:
            - Display the store type with value “Non-Restaurant”.
            - Find total quantity sold by aggregating all Quantity (from the SALE table) of all products for these stores.
    - Group by Category_Name ascendingly and with “Non-Restaurant” store data listed first.
- When ready, user can click on the ***Return*** button to return to the **Dashboard** form.

View Advertising Campaign Analysis Report

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 and Product_Name data (from the PRODUCT table).
        - Get Quantity data for this product sold during campaign (from SALE and ADVERTISING_CAMPAIGN tables) at the specific discount Date (from DAY and DSICOUNT tables); Find total quantity sold during campaign by aggregating Quantity in all discount sale days that hold a campaign.
        - Get Quantity data for this product sold outside campaign (from SALE and ADVERTISING_CAMPAIGN tables) at the specific discount Date (from DAY and DSICOUNT tables); Find total quantity sold outside campaign by aggregating Quantity in all discount sale days without a campaign.
        - 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.