LEOFUREN Sales Reporting System

Phase 1 Report

Team 017

Table of Content

Data Types:

State

Attribute	**Data Type**	**Nullable**
Name	String	Not Null

City

Attribute	**Data Type**	**Nullable**
Name	String	Not Null
Population	Integer	Not Null

Store

Attribute	**Data Type**	**Nullable**
Store ID	String	Not Null
Street Address	String	Not Null
Phone Number	String	Not Null

Service-Childcare

Attribute	**Data Type**	**Nullable**
Time Limit	Integer	Not Null

Sale

Attribute	**Data Type**	**Nullable**
Sold Quantity	Integer	Not Null

Date

Attribute	Data Type	Nullable
Date	String	Not Null

Holiday

Attribute	Data Type	Nullable
Name	String	Not Null

Campaign

Attribute	Data Type	Nullable
Description	String	Not Null

Product

Attribute	Data Type	Nullable
PID	String	Not Null
Name	String	Not Null

Category

Attribute	Data Type	Nullable
Name	String	Not Null

Price

Attribute	Data Type	Nullable
Date	String	Not Null

Business Logic Constraints

- LSRS takes sales data as inputs to generate reports for various business insights.
- All (Date) CANNOT be in the future
- Service time uses ‘minutes’ as unit
- Campaign started on the same day may not share the same description
- Campaign date may overlap.
- All products are available and sold at all stores .
- Every product has a retail price .
- Campaign dates may overlap.
- Campaign started on the same day may not share the same description.
- There is no sale tax.
- Quantity cannot be negative.
- Population cannot be negative.
- The name of the holidays should be a non-zero-length string and unique.
- Product category selling or sold in the store cannot be null.
- Retail Price and Discount Price cannot be NULL.
- Discount Price must be smaller than the Retail Price
- When holiday information is updated, nonholiday can change to holiday
- Date format MM/DD/YYYY
- If an item that was offered at a discount price were instead offered at the retail price, the quantity of items sold would be reduced by 25%

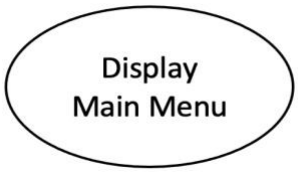
Task Decomposition with Abstract Code:

Notation:

- The Clickable Button is : "Underline"
  - Example: "Button"
- The Task is: **Bold**
  - Example: **Task Name**
- The Entity is in: Blue ink
  - Example: Entity Name

Display Main Menu

Task Decomposition:



Lock Types: Read only.

Number of Locks: Single.

Enabling Conditions: Display is enabled by default.

Frequency: High.

Consistency (ACID): Not critical.

Subtasks: NA (decomposition is not needed).

Abstract Code:

- Show "Statistic Count", "Statistics Reports", "Maintain Holidays", and "Update Populations" tabs.
- The "Statistics Count" tab is the default tab for initial viewing.
- Upon the "Statistics Count" tab:

- A clickable button named "Count Stores" allows the user to jump to the **Count Stores** task.
- A clickable button named "Count Stores Provide Food" allows the user to jump to the **Count Stores Provide Food** task.
- A clickable button named "Count Stores Provide Childcare" allows the user to jump to the **Count Stores Provide Childcare** task.
- A clickable button named "Count Products" allows the user to jump to the **Count Products** task.
- A clickable button named "Count distinct advertising campaigns " allows the user to jump to the **Count Distinct Advertising Campaigns** task.

- Upon the "Statistics Report" tab:
  - A clickable button named "Report category" allows the user to jump to the **Report Category** task.
  - A clickable button named "Report outdoor furniture on Groundhog Day" allows the user to jump to the **Report Air Conditioners on Groundhog Day** task.
  - A clickable button named "Report restaurant impact on category sales" allows the user to jump to the **Report Restaurant Impact on Category Sales** task.
  - A clickable button named "View childcare sales volume" allows the user to jump to the **View Child Care Sales Volume** task.
  - A clickable button named "View highest volume for each category" allows the user to jump to the **Select Year and Month** task.
  - A clickable button named "View actual revenue vs predicated" allows the user to jump to the **View Actual Revenue vs Predicated** task.
  - A clickable button named "Description and active date of advertising campaign" allows the user to jump to the **Advertising Campaign Analysis** task.
  - A clickable button named "View revenue by state by year" allows the user to jump to the **Revenue by State by Year** task.
  - A clickable button named "View revenue by population" allows the user to jump to the **Select Population Category** task.
- Upon the "Maintain Holidays" tab:

- A clickable button named "Upload holidays" allows the user to jump to the **Upload Holidays** task.

- Upon the "Update Population" tab:
  - A clickable button named "Edit city population" allows the user to jump to the **Edit City Population** task.

Count Stores

Task Decomposition

Lock Types: Read-only.

Number of Locks: Single.

Enabling Conditions: When count is needed by the user..

Consistency (ACID): Not critical.

Frequency: Low.

Subtasks: NA (decomposition is not needed).

Abstract Code:

- Click on the "Count Stores" button on the "Statistics Reports" tab from the "Main menu", run **Count Stores** task.
- Query for all the unique Stores in the database where Store ID is an unique identifier.
- Display number of stores.
- User clicked on "Main Menu" button: go back to main menu page

Count Stores Provide Food

Task Decomposition

Lock Types: Read-only.

Number of Locks: Single.

Enabling Conditions: When count is needed by the user.

Consistency (ACID): Not critical.

Frequency: Low.

Subtasks: NA (decomposition is not needed).

Abstract Code:

- Click on the "Count Stores Provide Food" button on the "Statistics Count" tab from the "Main menu", run the **Count Stores Provide Food** task.
- Query for all the unique Stores that provide Snack Bar or Restaurant in the database where Store ID is an identifier.
- Display number of stores.
- User clicked on "Main Menu" button: go back to main menu page

Count Stores Provide Childcare

Task Decomposition

Lock Types: Read-only.

Number of Locks: Single.

Enabling Conditions: When count is needed by the user.

Consistency (ACID): Not critical.

Frequency: Low.

Subtasks: NA (decomposition is not needed).

Abstract Code:

- Click on the "Count Stores Provide Childcare" button on the "Statistics Reports" from the "Main menu", run the **Count Stores Provide Childcare** task.
- Query for all the unique Stores provides Childcare in the database where Store ID is an identifier.
- Display number of stores.
- User clicked on "Main Menu" button: go back to main menu page

Count Products

Task Decomposition:

Lock Types: Read-only.

Number of Locks: Single.

Enabling Conditions: When count is needed by the user.

Consistency (ACID): Not critical.

Frequency: Low.

Subtasks: NA.

Abstract Code:

- When the user clicked on the "Count Products" button on the "Statistics Count" tab from the "Main menu", run the **Count Products task**.

- Query for all the Products in the database where PID is unique.
- Display count of products.
- User clicked on "Main Menu" button: go back to main menu page

Count Distinct Advertising Campaigns

Task Decomposition

Lock Type: Read only on Campaign, Date tables

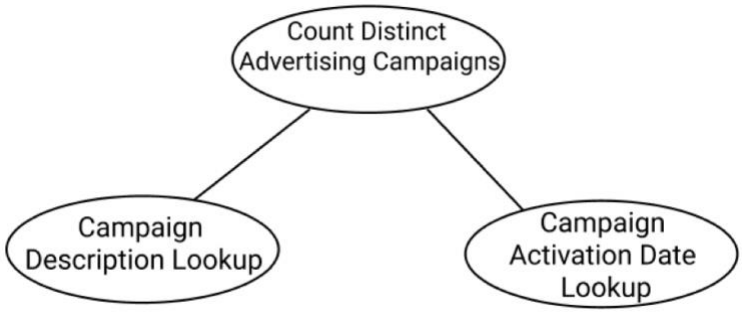
Number of Locks: Multiple

Enabling Conditions: When count is needed by the user

Frequency: Same frequency across two subtasks

Consistency: Critical

Subtasks: Campaign Description Lookup + Campaign Activation Date Lookup

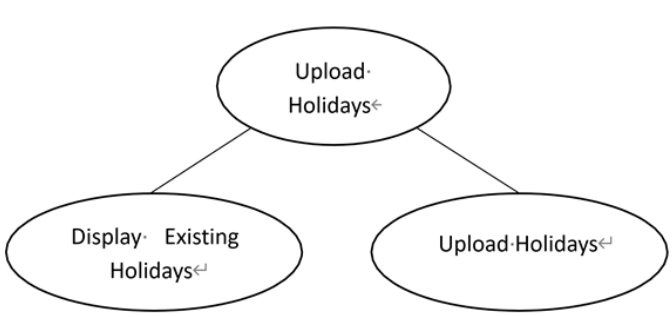


Abstract Code

- When need to get the count for distinct campaigns, run Count Distinct Advertising Campaigns
    - Look up the activation date for all available Campaigns in given time range
    - For Campaigns activated on the same Date, look up the descriptions to check if the description is the same.
    - Count the number of campaigns with distinct [date, description] combination for a given time period.
    - Return the count.

Upload Holidays

Task Decomposition:



Lock Type: A read-only lookups of **Holidays** (release after the display subtask is done); A update lock at the selected row to prevent multiple users trying to add a Holiday name for the same day;
Number of Locks:Two;
Enabling Conditions: Display is enabled by default, the upload Holidays function is enabled by selecting a valid date;
Frequency: Low by the frequency of displaying can be high.
Subtasks: Firstly, displaying existing Holidays, then the upload Holidays function is based on the user's demand.

Abstract Code:

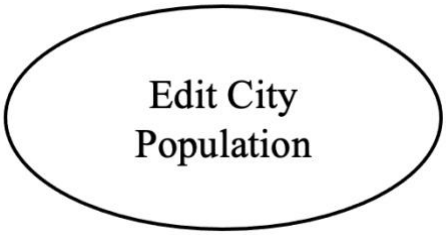
- User clicked on the "Upload holidays" in the "Main Menu", the **upload holiday** task is run.
    - The **Upload Holiday** task has two subtasks: **Display Existing Holidays** and **Upload Holiday**.
- The **Display Existing Holiday** task shows a list of Holidays already in the database.
    - All the Holidays should be unique in the database.
    - A refresh button is needed for reloading the Holiday information whenever the holiday information has been uploaded.
    - A scroll bar is needed in case the list of existing Holidays is too long to display in the window.
- The **Upload Holiday** task allows the user to add a new Holiday into the database.

- UI design: Two input boxes are designed for users to enter the new Date and holiday name. One button of "Apply" is designed to apply the changes to the database.
- The input restrictions:
  - The format of Date should be the same as the existing date in the database.
  - The name of the Holidays should be a non-zero-length string and unique.
- Error message reminders:
  - Any format errors of date and Holiday names will lead to an error message reminder.
  - If the name of the Holidays is not unique, an error message will show up.

- After entering the new Date and Holiday name, click "Apply" button to refresh.
- Click "Main Menu" button to go back to the Main Menu page

Edit City Population

Task Decomposition



Lock Types: Update lock at the row-level;

Number of Locks: One;

Enabling Conditions: Click the button of Edit city population;

Frequency: Low;

Consistency (ACID): Critical;

Subtasks: None;

Abstract Code:

- Click the button of "Edit city population" in the Main menu to run the **Edit City Population** task.
- Two drop-lists, and an input box for entry are needed for this task UI.
- The first drop-list contains a list of states available for selection. The user can select one of the States and click on "Select State".
- Once the state is selected, the user can select the city from the second drop-list and click on "Select City". The list of the City is within the selected state.
- The input box can be used to enter a new number for the population of the selected City. The input number expects to be a non-negative number. Any other data type will lead to an error message with the indication of the expected data type.
- A City should be selected before applying for the number from the input box. Otherwise, a warning will be issued for any violation.
- After finishing the task, click on "Apply" button to refresh.
- Click "Main Menu" button to go back to the Main Menu page

Report 1: Category Report

Task Decomposition

Lock Type: Read only on Category, Product, Price tables

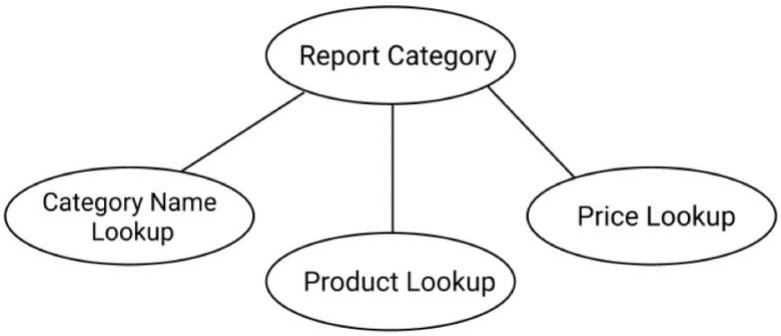
Number of Locks: Multiple

Enabling Conditions: When clicked View Category Report from main menu

Frequency: Same frequency across three subtasks

Consistency: Not Critical

Subtasks: Category Name Lookup + Product Lookup + Price Lookup

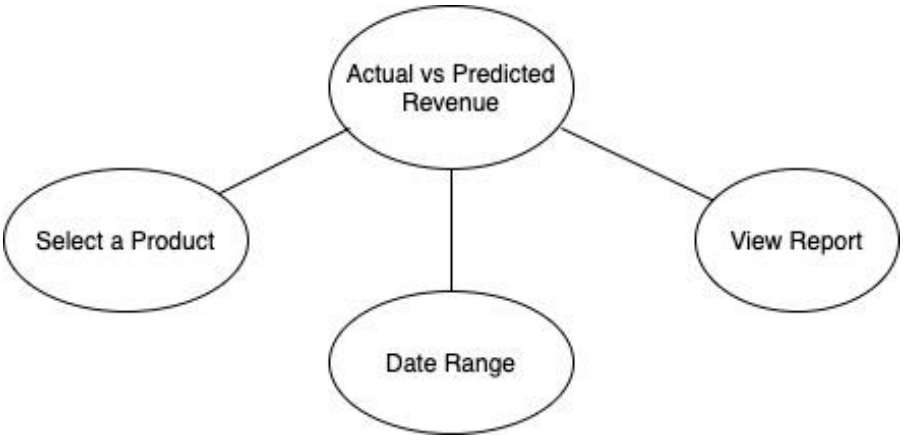


Abstract Code

- User clicked on the "View Category Report" button from the Main Menu.
- Run the **View Category Report** task
    - Get names of all available Categories
    - Sort by category Name ascendingly
        - Display category names
    - For each Category, find all the [Products]'s PID associated with that Category using category name as an identifier
    - Count the number of Product for each Category
        - Display total number of products in each category
    - For products in each Category, find Retail Price for all Products using PID as identifier.
    - Get minimum, maximum retail prices and calculate the average retail price
        - Display min, avg, and max retail price for products in each category
- User clicked on "Main Menu" button: go back to main menu page

Report 2 – Actual versus Predicted Revenue for Couches and Sofas

Task Decomposition:



Lock Type: Read-Only lookups

Enabling Conditions: The revenue compare report should be executed after user "**Select the Product**" and "**Date Range**"

Frequency: Low

Subtasks: All tasks must be done to perform revenue comparison between actual and predicted. The order is "**Select a Product**", "**Date Range**", and "**View Report**".

Abstract Code:

- The report can be executed by clicking the button "Actual vs Predicted Revenue" on main menu in displayed statistics information, the system will perform all the calculation relating to subtasks.
    - The user should click the tab "Select a Product" to specify the Product type.
    - The user should use "**Date Range**" to select a time period. Clicking the button should display a calendar.
    - The user can execute the report by clicking "View Report". The elements include: Product ID, Product Name, Retail Price, total number of units ever sold, total number of units sold at discount, total number of units sold at retail price, actual revenue in all sales, predicted revenue with no discount, and actual vs predicted revenue differences.
        - For each selected Date and Product, find Price and Quantity.
        - Calculate Actual Revenue using Price*Quantity.

- Display table 1-"Actual Revenue" with element Date, Price, Quantity and Actual Revenue.
- Create table 2-"Predicted Revenue" and fill the elements Date, Price, Quantity. The data is the same as table 1.
- Replace all Sale Price with Retail Price.
- Apply Quantity with multiplier 0.75.
- Calculate "Predicted Revenue":
- Quantity = Quantity*0.75
- Predicted Revenue = Price(Retail Price)*Quantity

  - Compute the difference between "Actual Revenue" and "Predicted Revenue", by taking Actual Revenue - Predicted Revenue.
  - Display and sort table 3-"Differences in Actual and Predicted Revenue" in descending order only when absolute value for difference in 1,4 is greater than $5000.

- When finish the task, click on "Main Menu" button and return to the main menu.

Report 3 – Store Revenue by Year by State

Task Decomposition:

Lock Type: 5 Read-Only lookups as shown in the diagram

Enabling Conditions: state should be selected to enable 5 subtasks

Frequency: the 5 subtasks have the same low frequency

Subtasks: All 5 tasks should be done and there should be correlations between **Store ID**, **Store Address** and **City Name** for consistency. Supertask "**Store Revenue by Year by State**" should coordinate subtasks. The order has two sorting requirements: sort by year in ascending order, and then sort by revenue in descending order.

Abstract Code:

- Execute the report generation by clicking the task "Store Revenue by Year by State" on main menu in displayed statistics information.
- Select the state through querying database
- The states available are presented in drop-down box
- After selecting the State information, there are two choices for display results: view by sales year(ascending order) or view by revenue(descending order).
    - For the selected state, calculate the annual revenue of all cities in that state.
    - For each City in that state, find all Stores in that city; and then calculate the annual revenue for all the stores in that city
    - In the selected state, sum all the revenues covering multiple sales years of the cities.
    - If user choose to display by year, then the system can display store information in the state with listed tasks: **store ID, store address, city name, total revenue** and **sales year** in ascending order; If user choose to display by total revenue, then the system can display store information in the state with tasks: store ID, store address, city name, sales year and total revenue in descending order.
- When finish the task, click on "Main Menu" button and return to the main menu.

Report 4 – Outdoor Furniture on Groundhog Day

Task Decomposition

Lock Type: Read only on Date, Sales, Product, Category tables

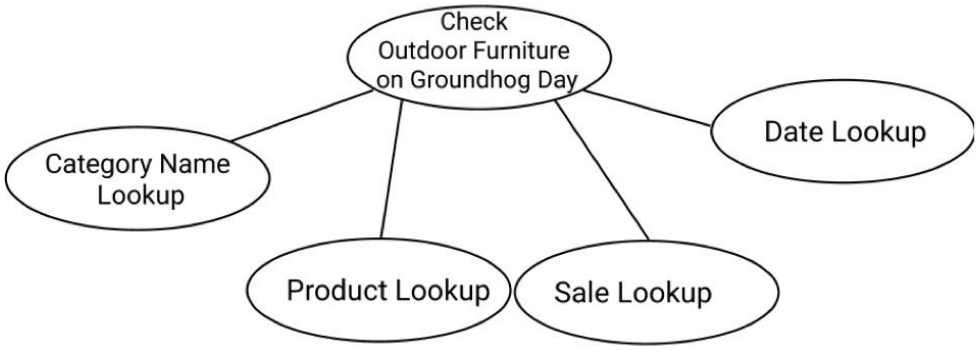
Number of Locks: Multiple

Enabling Conditions: When clicked View Report 4 – Outdoor Furniture on Groundhog Day is clicked from Main Menu

Frequency: Same frequency across the subtasks

Consistency: Critical

Subtasks: Category Lookup + Product Lookup + Sale Lookup + Date Lookup



Abstract Code

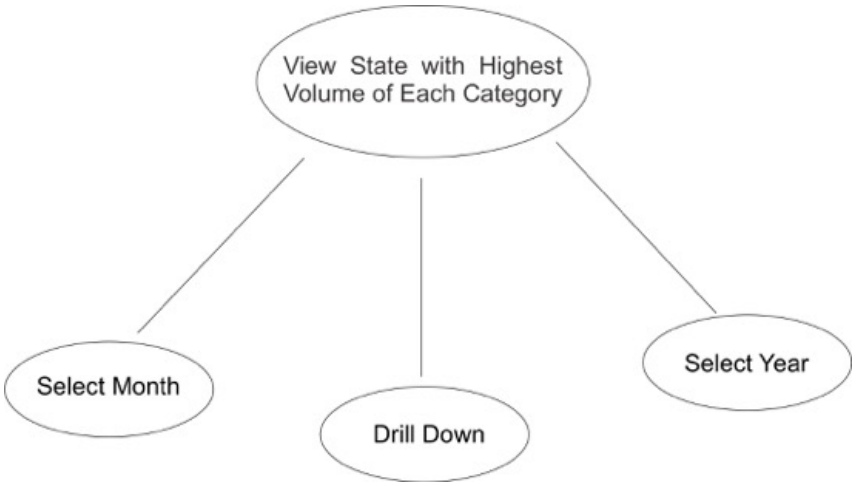
- User clicked on "View Report 4 – Outdoor Furniture on Groundhog Day" button from **Main Menu**.
- Run the **Outdoor Furniture on Groundhog Day** Task.
    - Look up category identifier for "Outdoor furniture" category from the Category table.
    - Look up all the Products' PID in "Outdoor furniture" category using category name as an identifier.
    - For all Product found above, find all the Sale record associated with the Product using PID as an identifier, and lookup the quantity sold.
    - Look up the Date for all the Sale records found above.
    - Group all records by year in Date
    - Rank by year in ascending order
        - Display Year
    - Sum all the quantity sold for each year
        - Display total number of items sold in outdoor furniture category that year
    - Divide the sum from previous step by 365
        - Display average number of units sold per day in that year
    - For all Sale record found previously, filter the date to Feb 2nd

- Sum all the quantity sold from previous step
  - Display Count

- User clicked on "Main Menu" button: go back to main menu page

Report 5 – View State with Highest Volume of Each Category

Task Decomposition:



Lock Types: 3 tasks on view state highest volume task;

Number of Locks: Several different schema constructs are needed;

Enabling Conditions: By clicking the select month, select year;

Frequency: low;

Consistency (ACID): Not critical;

Subtasks: Select month and select year must be clicked on to show year and month; Details in the "Drill-down" of categories must be clicked to show details;

Abstract Code:

- When the user clicked on the "View state highest volume for each category" button on the "statistics reports" tab from the "Main menu".
- The **View state highest volume for each category** task will show a list of years and months.

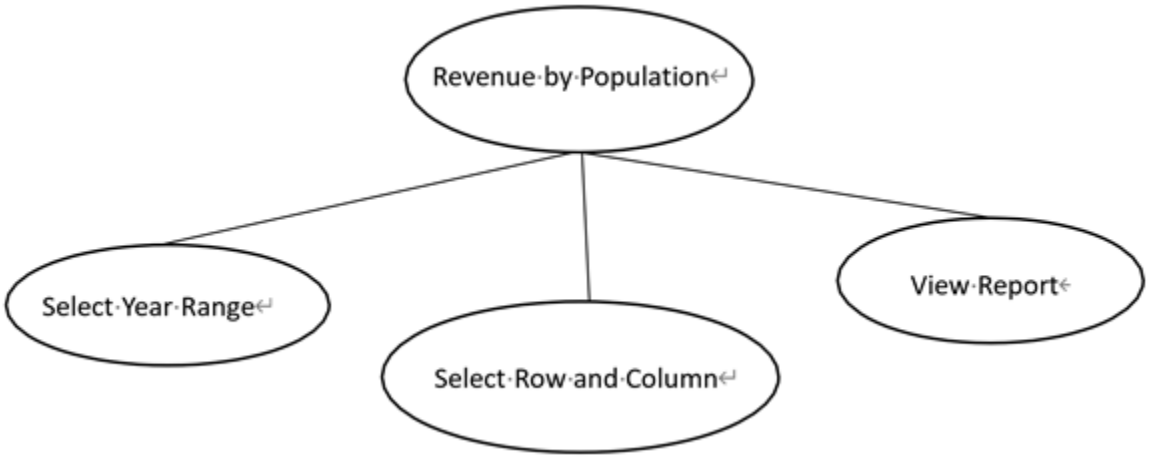
- From a UI point of view, there are two drop-levels to select the year and month and an apply button.
- After clicking the apply button, the master table like the following will be displayed.
- Sort by category name with ascending order.
- A drill-down detail table will be generated by clicking the button/hyperlink in the Category column. For example: clicking Table will lead to the following detail table.
- The header should include the category, month/year, and state info.

Category	State	The highest number of sold units
Tables	TX	…
...	...	…
Laptop	MA	…

- From a procedural point of view, the information in the master table will be generated based on the selected year and month.
  - For each category, find the state with the high volume.
  - Calculate the number of the sold units for the given state within the selected month and year;
  - Sum the number of sold units in all the stores in the state.
  - Find the state with the highest volume.
  - Display the category, state, and sold quantity.
  - Tied states will be listed in the table.
- When ready, user clicked on the “Main Menu” button: go back to the main menu page.

Report 6 – Revenue by Population

Task Decomposition:



Lock Type: read-only lock;

Number of locks: Three;

Enabling Conditions: Click the button of "View Revenue by Population";

Frequency: Low;

Consistency (ACID): Not critical;

Subtasks: Three subtasks of **Select Year Range**, **Select Row and Column** and **View Report** must be all done to finish this task.

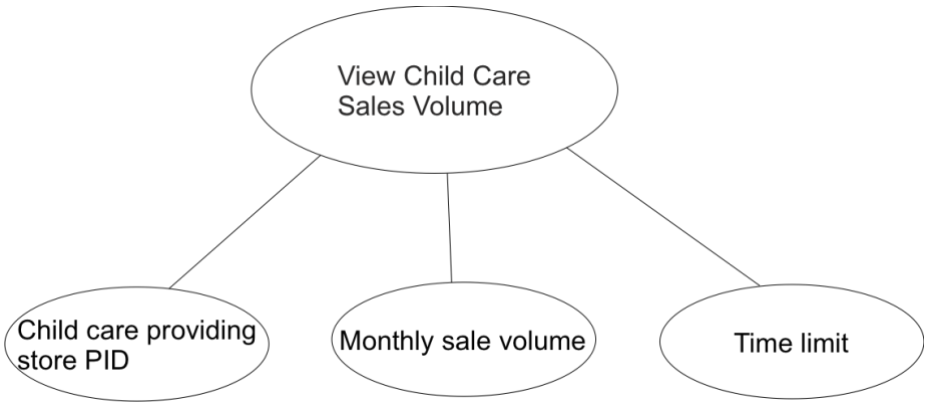
Abstract Code:

- Click on the "View Revenue by Population" button on the "Main menu" to show "**Select Year Range**" menu and "**Select Row and Column**" tabs.
- Draw a table and set years as row and population category as column.
- Use City Population to divide all cities into four groups: Small = CityPopulation < 3,700,000; Medium = CityPopulation>=3,700,000 and <6,700,000; Large = CityPopulation >=6,700,000 and <9,000,000; Extra Large = CityPopulation >=9,000,000

- Calculate Average Annual Revenue
    - Selecting the data based on the criteria of each category;
    - For each category of [Small, Medium, Large, Extra Large]:
        - Sum the annual revenues of all the Stores in the Cities under the category;
        - Count the number of Cities under the category;
        - Calculate the average annual revenue for the category;
- Click "View Report" to display the table including contents of year, population in ascending order and also display each corresponding revenue.
- After finishing the task, click on "Apply" button to refresh.
- Click "Main Menu" button to go back to the Main Menu page.

Report 7 – View Childcare Sales Volume

Task Decomposition:



Lock Type: Read only;

Number of Locks: Single;

Enabling Conditions: Display is enabled by default;

Frequency: low, childcare is not a must have service item, and different customer would have different requirement;

Consistency (ACID): Not critical;

Subtasks: No decomposition needed

Abstract Code:

- When the user clicked on the "View child care sales volume" button on the "Category Reports" from the "Main Menu", run the **View child care sales volume** task.
- The "View child care sales volume" task will create and display a table which lists the sales volume of each month with different child care limits at each store in the 12 months.
    - From a UI point of view, there is a table to show the static result.
    - Look up and find the store ID which provides the child care service.
    - If a store does not provide childcare service, it will show "No childcare" or "0 min".
    - Find all the Sale record associated with the Product using PID as an identifier, and
    - lookup the quantity sold.
    - Find the total number of sales volume/month (monthly sale volume) corresponding to all categories and each store with different childcare time limit category.
    - Sum all the quantities sold of each category in each month.
    - Sort by time (month) with ascending order and display the months.

Time/ Childcare limit time/sales volume	No childcare (0 min)	10 min	20 min	30 min	40 min	50 min	...
Jan	...	...	...	...	...	...	...
Feb	...	...	...	...	...	...	...
Mar	...	...	...	...	...	...	...
Apr	...	...	...	...	...	...	...
May	...	...	...	...	...	...	...

Jun	...	...	...	...	...	...	...
Jul	...	...	...	...	...	...	...
Aug	...	...	...	...	...	...	...
Sep	...	...	...	...	...	...	...
Oct	...	...	...	...	...	...	...
Nov	...	...	...	...	...	...	...
Dec	...	...	...	...	...	...	...

- When ready, user clicked on the "Main Menu" button: go back to the main menu page.

Report 8 – Report Restaurant Impact on Category Sales

Task Decomposition:

Lock Type: Read only

Number of Locks: Single

Enabling Conditions: Display is enabled by default

Frequency: High

Consistency (ACID): Not critical

Subtasks: No decomposition needed

Abstract Code:

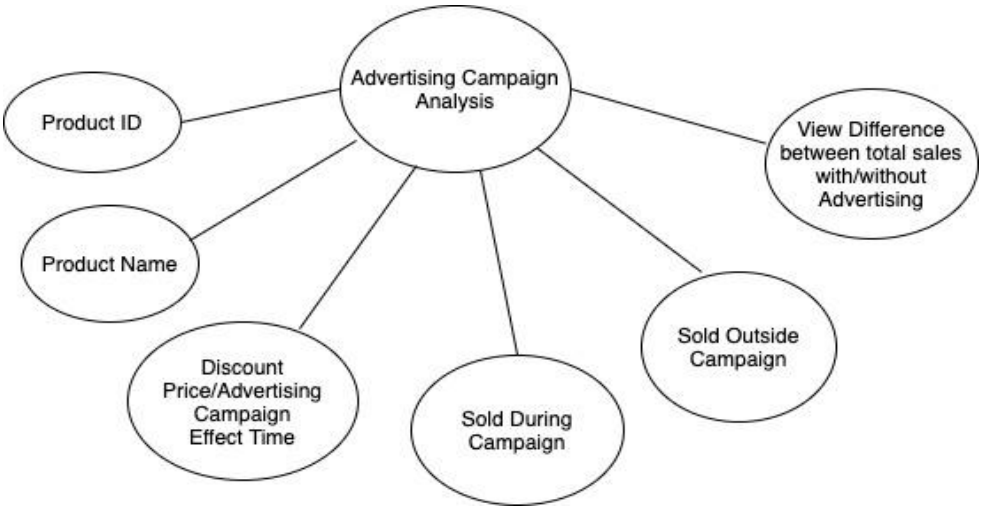
- When the user clicked on the "Report restaurant impact on category sales" button on the "Category Reports" from the "Main Menu", run the **Report restaurant impact on category sales** task.
- **Report restaurant impact on category sales** task will:
    - Display the categories which are selling in the store. Check in the "restaurant" or "snack bar" entity to determine if a store provides dining services.
    - Display if the store has the restaurant service or not.
    - Find all products belonging to each category, then get the sales records for all the products.
    - Display the total number of items sold belonging to each category under the condition of w/o restaurant service within one year.
    - Sorted by category with ascending order.

Category	W/O Restaurant	Quantity Sold/yr
Accent Chairs	Non-restaurant	…
	Restaurant	…
Sofas	Non-restaurant	…
	Restaurant	…
Outdoor furniture	Non-restaurant	…
	Restaurant	…

- When ready, user clicked on the "Main Menu" button: go back to the main menu page.

Report 9 – Advertising Campaign Analysis

Task Decomposition:



Lock Type: 5 Read-Only lookups as shown in the diagram.

Enabling Conditions: count of distinct advertising campaigns should be clicked to enable the analysis.

Frequency: Low.

Subtasks: **Product ID, Product Name and Discount Price Campaign Effect Time(Date)** should be done before performing calculation for **Sold During Campaign**, **Sold Outside Campaign** and **View Difference between total sales with/without Advertising**.

Abstract Code:

- The system will start to run the **Advertising Campaign Analysis** task when clicking on the "Advertising Campaign Analysis**"** tab .
- Query for all the products where the **product ID** is unique.
- For every product we query:
    - Display the information for **Product Name** together with **Product ID**.
    - Display the **discount price effect time**.
- Sum up total sold with active advertising campaign based on discount effect Date.
    - Sum up total sold when there is no advertising campaign, by excluding the effect discount price Date.
    - Compare and display the two totals between with/without Advertising Campaign by

- taking the difference.
- Sort the total Sales differences in descending order.
- The table will display the final report with features **Product ID, Product Name, Sold During Campaign, Sold Outside Campaign** and **Differences.**
- When finish the task, click on "Main Menu" button and return to the main menu.