Table of Contents:

Data Types:

Store

Attribute	Data Type	Nullable
Store_number	Integer	Not Null
Phone_number	Integer	Not Null
Address	String	Not Null
Restaurant	Integer	Not Null
Snack_bar	Integer	Not Null

City

Attribute	Data Type	Nullable
Population	Integer	Not Null
State_name	Integer	Not Null
City_name	String	Not Null

ChildCare

Attribute	Data Type	Nullable
ServiceID	Integer	Not Null
Limit_time	time	Not Null
Care_category	Integer	Not Null

Date

Attribute	Data Type	Nullable
Sell_date	Date	Not Null
Sell_amount	Integer	Not Null

Product

Attribute	Data Type	Nullable
PID	String	Not Null
Pname	String	Not Null
Price	float	Not Null

Discount

Attribute	Data Type	Nullable
Percentage	float	Not Null
Date	Date	Not Null

Category

Attribute	Data Type	Nullable
Name	String	Not Null

AD

Attribute	Data Type	Nullable
Name	String	Not Null
Date	date	Null

Festival

Attribute	Data Type	Nullable
Name	String	Not Null
Date	Date	Null

Business Logic Constraints:

Store

- Store_number must be positive number
- Phone_number must be 10-digit number
- Restaurant must be 0 or 1
- Snack_bar must be 0 or 1

City

- Population must be positive number

Childcare

- Care_category must be 0 or 1

Date

- Sell_amount must be positive

Product

- Price must be positive number

Discount

- Percentage must be positive

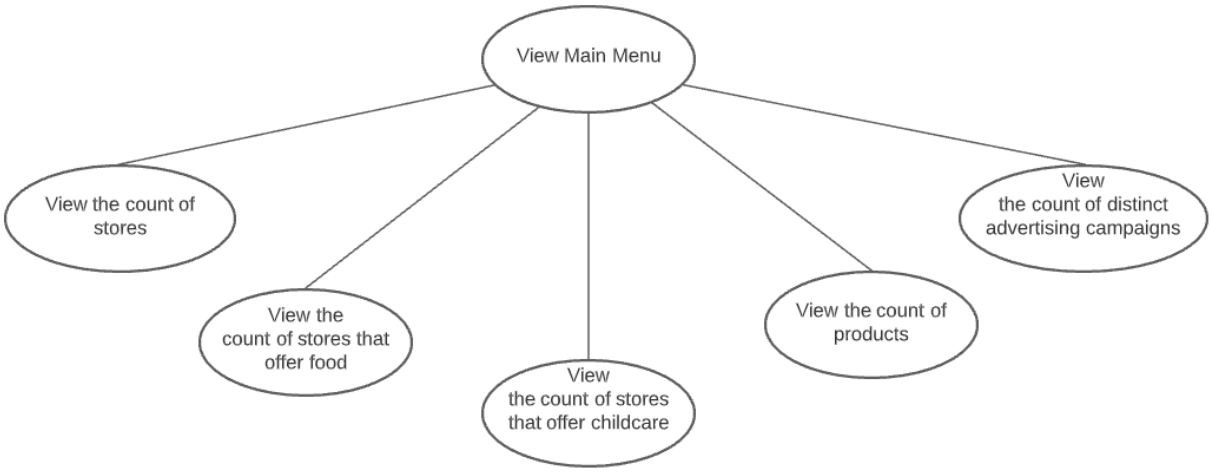
User

- Users must determine if the statistics \\is generally accurate before viewing reports.

View Main Menu

Task Decomp

- **Lock Types**:4 lookups of STORE, CHILDCARE, PRODUCT, AD; all are read-only
- **Number of Locks**: 4-STORE, CHILDCARE, PRODUCT, AD
- **Enabling Condition**: Enabled on main page
- **Frequency**: High - 200 times a day. This assumption is based on that various users might want to access any report or link on the main menu page. Each time the 4 lookups would need to display. All tasks have same frequency
- **Consistency (ACID)**: not critical.
- **Subtasks**: All tasks must be done; can be done in parallel. Mother task is required to coordinate subtasks. Order is not necessary.



Abstract Code

● Display sum count of Store, sum count of store who provide Restaurant or Snack_bar service, total count of store who provide Childcare service, total Count of Product, total Count of AD

● Show **“*View Holiday*”**, **“*Add Holiday*”**, **“*View Population of Cities*”**, **“*Update Population of Cities*”**, **“*View Category Report*”**, **“*View Actual versus Predicted Revenue for Couches and Sofas*”**, **“*View Store Revenue by Year by State*”**, **“*View Outdoor Furniture on Groundhog Day*”**, **“*View State with Highest Volume for each Category*”**, **“*View Revenue by Population*”**, **“*View Childcare Sales Volume*”**, **“*View Restaurant Impact on Category Sales*”**, **“*View Advertising Campaign Analysis*”** buttons.

● Upon:

- Click ***View Holiday*** button - Jump to **View Holiday** task.
- Click ***Add Holiday*** button - Jump to **Add Holiday** task.

● Click ***View Population of Cities*** button - Jump to **View Population of Cities** task.

● Click ***Update Population of Cities*** button - Jump to **Update Population of Cities** task.

● Click ***View Category Report*** button - Jump to **View Category Report** task.

● Click ***View Actual versus Predicted Revenue for Couches and Sofas*** button - Jump to **View Actual versus Predicted Revenue for Couches and Sofas** task.

● Click ***View Store Revenue by Year by State*** button - Jump to **Edit View Store Revenue by Year by State** task.

● Click ***View Outdoor Furniture Sale on Groundhog Day*** button - Jump to **View Outdoor Furniture Sale on Groundhog Day** task.

● Click ***View State with Highest Volume for each Category*** button - Jump to **View State with Highest Volume for each Category** task.

● Click ***View Revenue by Population*** button - Jump to **View Revenue by Population** task.

● Click ***View Childcare Sales Volume*** button - Jump to **View Childcare Sales Volume** task.

● Click ***View Restaurant Impact on Category Sales*** button - Jump to **View Restaurant Impact on Category Sales** task.

●Click ***View Advertising Campaign Analysis*** button - Jump to **View Advertising Campaign Analysis** tasks.

View Holidays

Task Decomp
- **Lock Types**: 1 look up of holiday based on holiday name; read-only
- **Number of Locks**: Single - FESTIVAL
- **Enabling Condition**: Enabled on Main Menu through button or link
- **Frequency**: Low
- **Consistency (ACID)**: not critical, order is not critical.
- **Subtasks**: Mother Task is not needed. No decomposition needed.

Abstract Code
- User clicked on ***View Holidays*** from **Main Menu**:
- Run the **View Holidays** task: query for information about FESTIVAL using the system from the HTTP Session/Cookie
    - For each Holiday of FESTIVAL:
        - Display FESTIVAL.Name, FESTIVAL.Date

Add Holidays

Task Decomp
- **Lock Types**: 1 look up of HOLIDAY; Add holiday date and holiday name
- **Number of Locks**: Single - HOLIDAY
- **Enabling Condition**: Enabled on Main Menu through button or link
- **Frequency**: Low
- **Consistency (ACID)**: not critical, order is not critical.
- **Subtasks**: Mother Task is not needed. No decomposition needed.

Abstract Code
- User clicked on ***Add Holiday*** from from **Main Menu**:
- Run the **Add Holiday** task: add information about FESTIVAL using the system from the HTTP Session/Cookie:
    - User selects date in a date drop down box
    - User enters FESTIVAL.Name for the Holiday input field
        - If data validation is successful for FESTIVAL.Name input field: Add Holiday into FESTIVAL
        - Else: Go back to **Main Menu**, with error message

View Population of Cities

Task Decomp
- **Lock Types**: 1 look up of city, state and population based on city name; all are read-only
- **Number of Locks**: Single-CITY
- **Enabling Condition**: Enabled on Main Menu through button or link
- **Frequency**: Low
- **Consistency (ACID)**: not critical, order is not critical.
- **Subtasks**: Mother Task is not needed. No decomposition needed.

Abstract Code
- User clicked on ***View Population*** from **Main Menu**:
- Run the **View population** task: query for information about POPULATION using the system from the HTTP Session/Cookie For each city of CITY:
    - Display CITY.City_name, City.Population
- Show ***Update Population*** button.

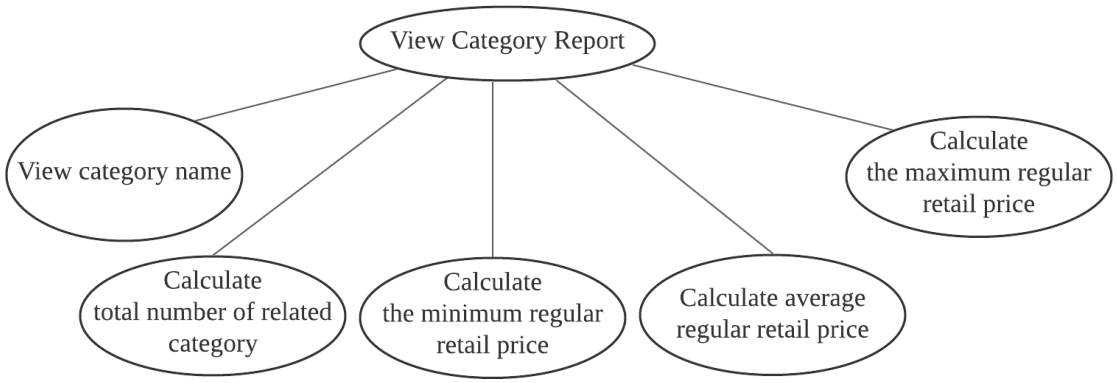
Update Population of Cities

Task Decomp
- **Lock Types**: 1 look up of CITY; CITY.City_name and CITY.State_name are read-only; edit of CITY.Population
- **Number of Locks**: Single - CITY
- **Enabling Condition**: Enabled on View Population page through button or link
- **Frequency**: Low
- **Consistency (ACID)**: not critical, order is critical.
- **Subtasks**: All tasks must be done; can be done in parallel. Mother task is required to coordinate subtasks.

Abstract Code
- User clicked on ***Update Population*** from **View Population**:
- Run the **Update Population** task: edit information about CITY using the system from the HTTP Session/Cookie:
    - User selects city in a city drop down box
    - User enters city name($city) and state name($state) for the City to be updated input field
    - User enters population($population) for the City to be updated input field
    - If data validation is successful for both City_name and Population input field:
        - Find current CITY using CITY.City_name == $city and CITY.State_name == $state
        - Edit CITY.Population = $population
    - Else:
        - Go back to **View Population** task, with error message

View Category Report

Task Decomp



Lock Types: Lookups of PRODUCT, CATEGORY, ASSIGNS all of them are read lock.
Number of Locks: 3 schema constructs are needed.
Enabling Condition: Enabled on main menu through button or link
Frequency: Medium - this query is ready for searching tasks at any time, and its priority level is medium when concurrency throughput is high.
Consistency: Not critical;
Sub-tasks: All must be done, mother task is needed; order is not critical.

Abstract Code :

● User click on ***View Category Report*** from **Main Menu**;

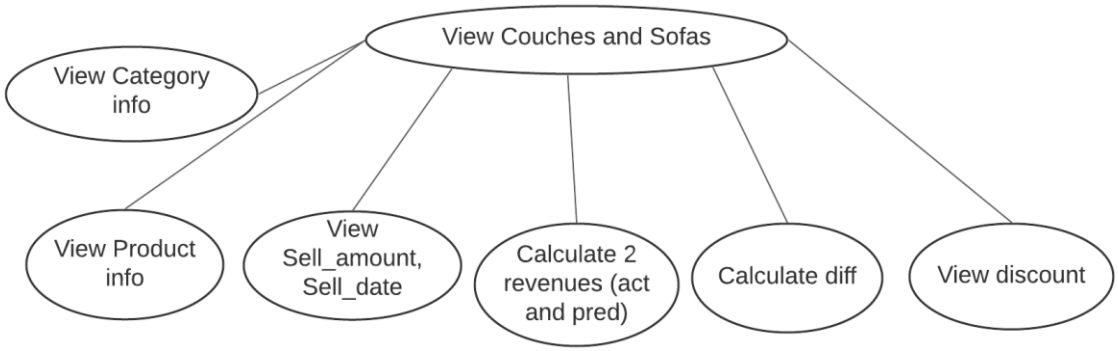
● Run the **View Category Report** task: query for information about CATEGORY using the system from the HTTP Session/Cookie;

- For each category of CATEGORY
  - Find CATEGORY.Name;
  - Find the total number of products belongs to this category;
  - Calculate the min price of products belongs to this category;
  - Calculate the average price of products belongs to this category;
  - Calculate the max price of products belongs to this category;
  - Order by category name ascending.

● Display all the above info in the **View Category Report.**

View Actual versus Predicted Revenue for Couches and Sofas Report

Task Decomp



Lock Types: Lookups of PRODUCT, ASSIGNS, DATE, DISCOUNT.all of them are read lock.
Number of Locks: 4 schema constructs are needed.
Enabling Condition: Enabled on main menu through button or link.
Frequency:Medium- this query is ready for search task at anytime, and its priority level is medium when concurrency throughput is high.
Consistency: not critical;
Sub-tasks: All must be done, mother task is needed; order is not critical.

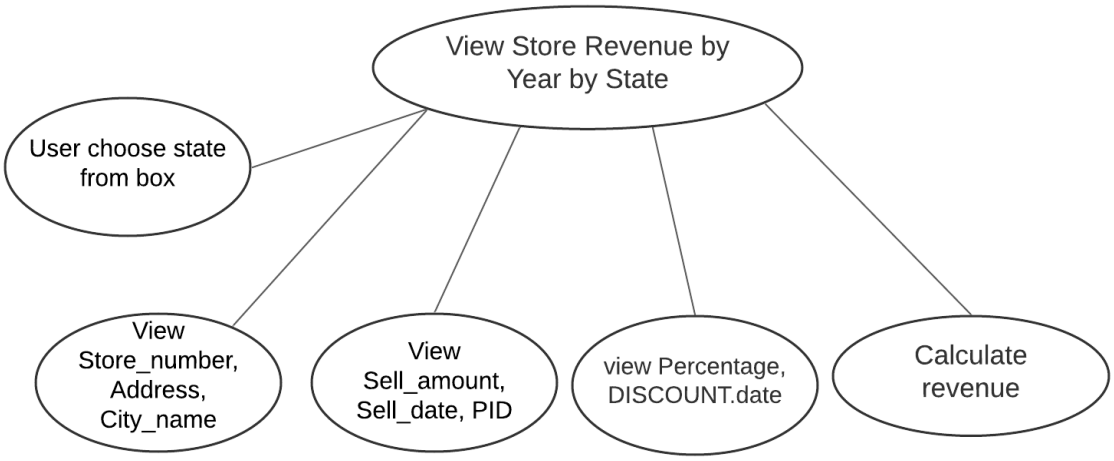
Abstract Code:

● User click on ***Actual versus Predicted Revenue for Couches and Sofas*** from **Main Menu**;
● Run **View Actual versus Predicted Revenue for Couches and Sofas** task:query for information about CATEGORY using the system from the HTTP Session/Cookie;
- ● Find PRODUCT.Pname, PRODUCT.PID in PRODUCT;
- ● Combine PRODUCT and CATEGORY to limit the search scope as 'Couches and Sofas' (mark it as temporary t1), and find the PID, Pname, Price;
- ● Combine t1 and DATE, we can get one new scope with DATE information and limited category scope 'Couches and Sofas'; then we can find PID, Pname, Sell_amount, Sell_date. We can rename all this info set as temporary t2.
- ● Combine t2 and DISCOUNT with Sell_date then we can get all the sold goods' info (Sell_date, PID, Pname, Price, Percentage, DISCOUNT.Date, Sell_amount) which we can use them to calculate the actual revenue and predicted revenue; if we find one product has discount date, we will set this date and this PID 's Sell_amount * 0.75 when calculate the predicted revenue.
- ● Calculate the gap between the actual revenue and predicted revenue, and list the gap > 5000 and order them by descending.

● Display all the above info in the **View Actual versus Predicted Revenue for Couches and Sofas.**

View Store Revenue by Year by State Report

Task Decomp:



Lock Types: Lookups of PRODUCT, DATE, CITY, STORE, DISCOUNT. all of them are read lock.
Number of Locks: 5 schema constructs are needed.
Enabling Condition: Enabled on main menu through button or link.
Frequency:Medium- this query is ready for search task at anytime, and its priority level is medium when concurrency throughput is high.
Consistency: Not critical;
Sub-tasks: All must be done, mother task is needed; order is not critical.

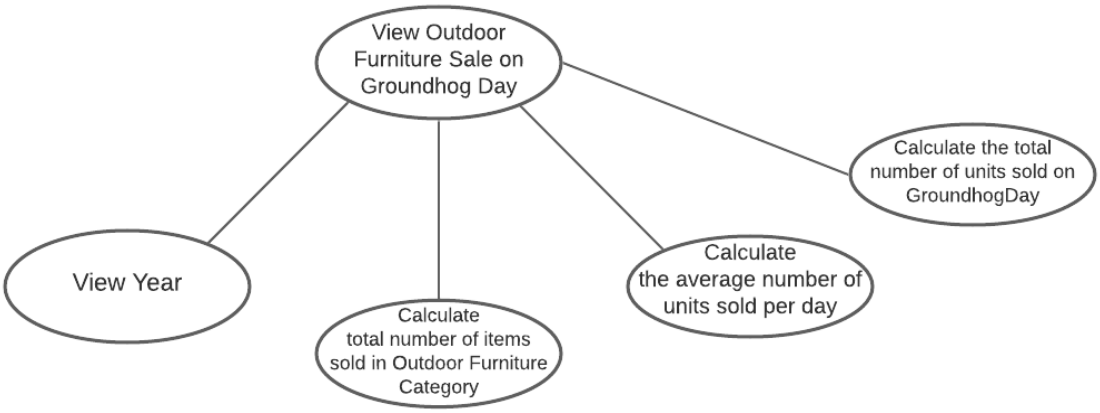
Abstract Code

● When User click ***View Store Revenue by Year by State*** button from **Main Menu**
● Run ***View Store Revenue by Year by State*** task: query for information about DATE using the system from the HTTP Session/Cookie;
- ● User selects stateName('$State_name') in a date drop down box;
- ● Find the Store_number, Address, City_name from STORE and CITY with stateName provided by user, and rename this result set as temporary t1;
- ● Find the Price,Sell_amount and Sell_date from t1 and DATE by combine them as the temporary set t2;
- ● Based on t2, we combine result of t2 and DISCOUNT then we can identify the price and discount percentage which we can use them to calculate the revenue;
- ● Order them by year(use inner function to get year from date.) ascending and then order revenue by descending.

● Display the year, revenue,Store_number, Address,City_name, in the ***View Store Revenue by Year by State.***

View Outdoor Furniture Sale on Groundhog Day Report

Task Decomp



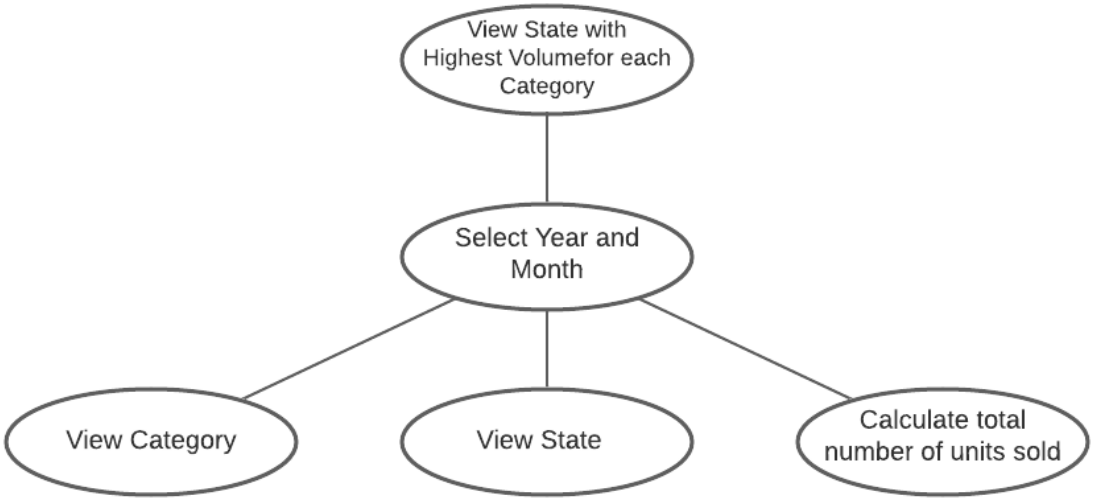
Lock Types: lookups of DATE, SELLS and ASSIGNS; all are read-only.
Number of Locks: 3 schema constructs are needed.
Enabling Conditions: Enabled on Main Menu through button or link.
Frequency: Low - Groundhog Day only takes place once per year, no report should be generated for the year without whole year's data. All tasks have the same frequency.
Consistency (ACID): not critical.
Subtasks: Mother task is required. All subtasks must be done,but can be done in parallel. Order is not necessary.

Abstract Code

● User clicked on ***Outdoor Furniture Sale on Groundhog Day*** button from **Main Menu**:
● Run the **View Outdoor Furniture Sale on Groundhog Day** task: query for information regarding sales of Outdoor Furniture category using the system from the HTTP Session/Cookie;
● Find PRODUCT.PID in Outdoor Furniture category through ASSIGNS;
● Combine with SELLS on PID and calculate the sum, group by year;
● Find PRODUCT.PID in Outdoor Furniture category through ASSIGNS;
    ● Combine with SELLS on PID on 2nd, February, the Groundhog Day;
    ● Divide Outdoor Furniture annual sale with 365 to get average daily sale;
● Order in ascending.
● When ready, user selects the next action from choices in the **Main Menu**.

View State with Highest Volume for each Category Report

Task Decomp



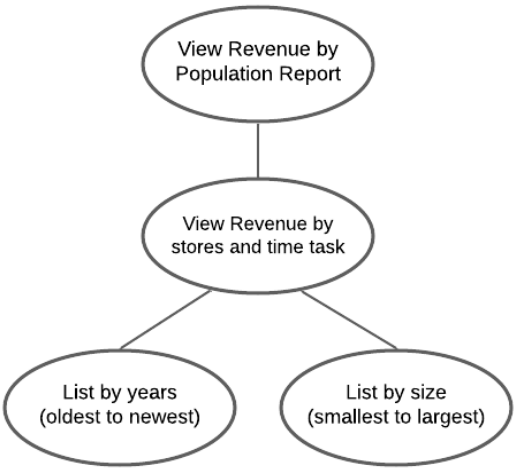
Lock Types: lookups of SELLS, STORE, ASSIGNS, CITY, DATE; all are read-only.
Number of Locks: 5 schema constructs are needed.
Enabling Condition: Enabled on Main Menu through button or link.
Frequency: Medium - monthly; all tasks have the same frequency.
Consistency (ACID): not critical.
Subtasks: Mother task is required. All subtasks must be done. User needs to select a month and year before performing other subtasks. Order for other subtasks is not necessary.

Abstract Code

● User clicked on ***State with Highest Volume for each Category*** from **Main Menu**:
● Run the **State with Highest Volume for each Category** task: query for information about CATEGORY using the system from the HTTP Session/Cookie;
● User selects year ('$Year') and month ('$Month') in a date drop down box;
    ● Find entries matching year and month selected on SELLS;
    ● For each category of CATEGORY matching in ASSIGNS, combine with STORE and CITY, find the products sold, sum the sales amount that group by state within the time frame;
    ● List category name and list twice if there is a tie of 2 or more states, with separate state names;
    ● Order by category name in ascending.
● When ready, user selects the next action from choices in the **Main Menu**.

View Revenue by Population Report

Task Decomp：



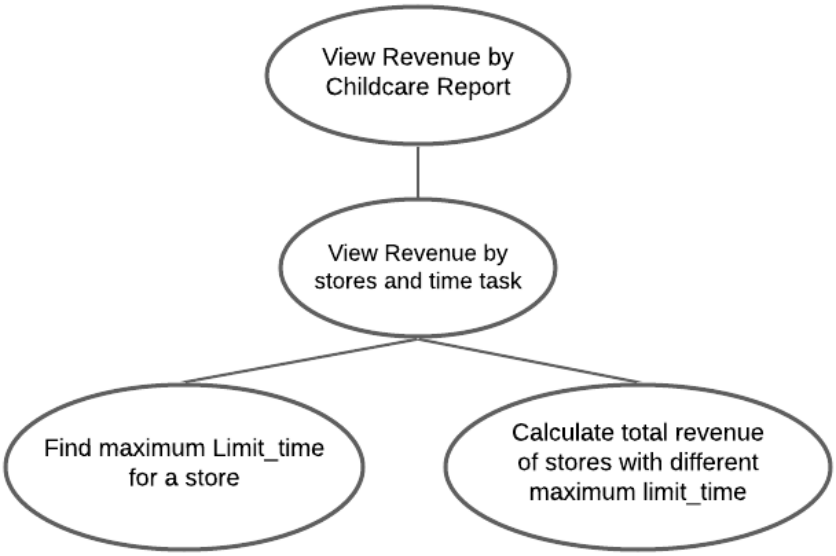
Lock Types: Lookups of PRODUCT, DATE, CITY, STORE, all are read-only.
Number of Locks: 4 schema constructs are needed.
Enabling Condition: Enabled on Main Menu through button or link.
Frequency: Medium - monthly; all tasks have the same frequency.
Consistency (ACID): not critical.
Sub-tasks: All must be done. Order is critical. Mother task is needed.

Abstract Code：

- User click on ***View Revenue by Population Report*** from the **Main Menu.**
- Run the **View Revenue by stores and time task**:
  - For every SELLS, find the PID, Sell_date and Sell_amount of it from DATE and PRODUCT, and the Percentage of DISCOUNT if the Sell_date is equal to the DISCOUNT.Date, so the sell price of a single item could be calculated.
  - Sum up all revenues of all products by stores
  - From every STORE.Store_number, identify which city the store is located.
  - Sum up all revenues by different cities with order of DATE.Sell_date from oldest to newest.
- Show buttons of two options: ***List by years (oldest to newest***) and ***List by size (smallest to largest).***
  - If the user clicks **List by years (oldest to newest):**
    Display the revenues summed up by years from oldest to newest.
  - If the user clicks **List by size (smallest to largest):**
    Display the revenues summed up by size of city, with requirements that small, represented city population <3,700,000); medium, represented city population >=3,700,000 and <6,700,000; large, represented city population >=6,700,000 and <9,000,000 and extra-large, represented city population >=9,000,000; respectively.

View Childcare Sales Volume Report

Task Decomp：



Lock Types: Lookups of Product, Date, Store, ChildCare, all are read-only.
Number of Locks: 4 schema constructs are needed.
Enabling Condition: Enabled on Main Menu through button or link.
Frequency: Medium - monthly; all tasks have the same frequency.
Consistency (ACID): not critical.
Sub-tasks: All must be done, order of subtasks is critical. Mother task is needed.

Abstract Code：

- User click on ***View Revenue by Childcare Report*** from **Main Menu.**
- From SELLS, find the sell record if DATE.Selll_data within the last 12 month.

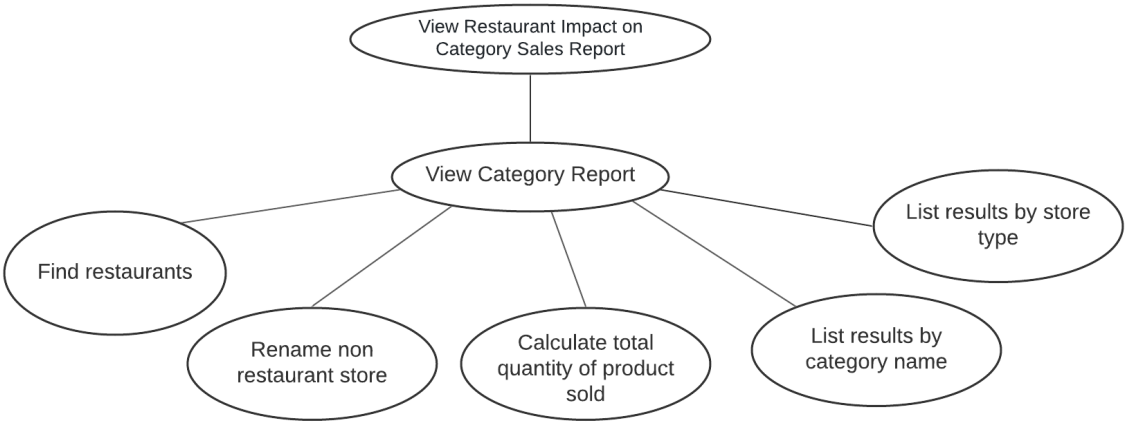
Run the **View Revenue by stores and time task**:

- For every SELLS, find the PID, Sell_date and Sell_amount of it from DATE and PRODUCT, and the Percentage of DISCOUNT if the Sell_date is equal to the DISCOUNT.Date, so the sell price of a single item could be calculated.
- Sum up all revenues of all products by stores
- From every STORE.Store_number, identify which city the store is located.
- Sum up all revenues by different cities with order of DATE.Sell_date from oldest to newest.

- Find all CHILDCARE.SeviceID corresponding each STORE and extract the service which has maximum Limit_time.
- Sum up all the revenues by different maximum Limit_time of all STORE
- Display how the revenues grow under different childcare Limit_time values.

View Restaurant Impact on Category Sales Report

Task Decomp:



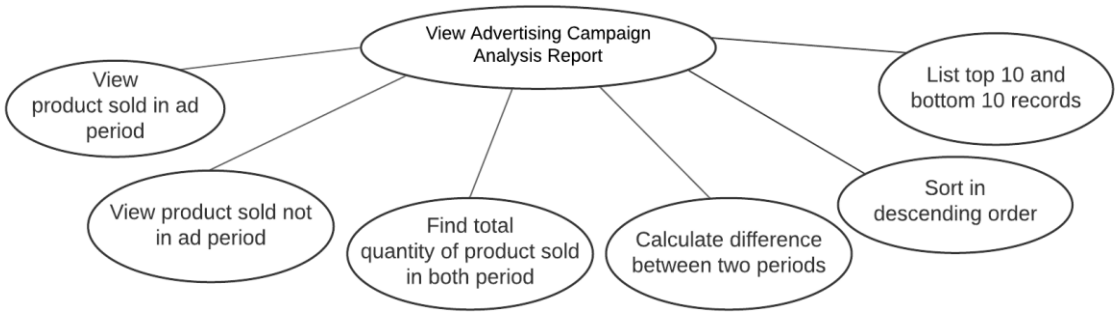
Lock Types: Lookups of STORE, DATE, CATEGORY. All are read-only.
Number of Locks: 3 schema constructs are needed.
Enabling Condition: Enabled on main menu through button or link
Frequency: Low, this task is an explore of a certain pattern
Consistency: not critical;
Sub-tasks: Mother task is needed; All must be done; Order is not critical.

Abstract Code：

- User click **View Restaurant Impact on Category Sales Report** button from Main Menu;
- Run **View Restaurant Impact on Category Sales Report** task:
  - Click View Category Report
  - Find CATEGORY.Name, STORE.Restaurant and number of products from Category Report
  - Find Restaurant records
  - Rename non-restaurant store data as '$Non-restaurant'
  - Calculate total quantity of products sold for both '$Restaurant' and '$Non-restaurant'
  - Order the report by category name
  - Order the report by Store_Type to list non-restaurant data first

View Advertising Campaign Analysis Report

Task Decomp:



Lock Types: Lookups of AD, DISCOUNT, PRODUCT, DATE. All are read-only.
Number of Locks: 4 schema constructs are needed.
Enabling Condition: Enabled on Main Menu through button or link
Frequency: Medium, analysing campaign effect would be done before and after each campaign.
Consistency: not critical;
Sub-tasks: Mother task is needed; All must be done; Order is not critical.

Abstract Code：

- User click **View Advertising Campaign Analysis Report** from main menu;
- Run **View Advertising Campaign Analysis Report** task:
  - Find all PRODUCT.PID, PRODUCT.PName of product and DATE.Sell_amount in two cases:
    - Product sold time in AD.DATE
    - Product sold time not in AD.DATE
  - Calculate total quantity of product sold for each case
  - Calculate difference between two totals
  - Sort the difference in descending order
  - Select top 10 and bottom 10 records
  - Display selected records in single report