

# Team 104 Phase 1 Report

## CS6400 Spring 2021

Abhik Choudhury <achoudhury64@gatech.edu>

Joseph Tadros <jtadros3@gatech.edu>

Rituraj Chauhan <rroj35@gatech.edu>

Shashvat Sinha <ssinha96@gatech.edu>

### Table of Contents

<i>Data Types</i> .....	2
<i>Business Logic Constraints</i> .....	3
<i>Task Decomposition and Abstract Code</i> .....	4
Main Menu: .....	4
Update Populations.....	4
Update Holidays: .....	5
Report 1: .....	5
Report 2 .....	6
Report 3 .....	7
Report 4 .....	9
Report 5 .....	10
Report 6 .....	11
Report 7 .....	13
Report 8 .....	14
Report 9 .....	15

## Data Types

Entity	Attribute	Data Type	Description	Example
Store	Store Number	Integer	10-digit integer	1324605879
Store	Phone Number	Varchar	10-digit integer	5162243274
Store	Street Address	Varchar	Max 60-digit string	123 abc rd
Store	Childcare Limit	Integer	Max 3-digit integer	60
Store	Snack Bar	Boolean	(Y/N) or (True/False)	Y or True
Store	Restaurant	Boolean	(Y/N) or (True/False)	Y or True
City	Name	Varchar	Max 30-character string	New York City
City	State	Varchar	2 character string	NY
City	Population	Integer	Max 15-digit integer	1528235
Sales	Qty	Integer	Max 15-digit integer	1420
Product	PID	Integer	Max 10-digit integer	1902348675
Product	Name	Varchar	Max 30-character string	Super Comfy Chair
Product	Retail Price	Decimal	Max 15-digit decimal with 2 decimal places	132.40
Category	Name	Varchar	Max 30-character string	Sofas
Discount	Discount Price	Decimal	Max 15-digit Decimal with 2 decimal places	98.23
Date	Date	Date	YYYY-MM-DD	2021-02-15
Campaign	Description	Varchar	Max 60-character string	Summer Radio Ads
Holiday	Holiday Name	Varchar	Max 30-character string	Christmas

We have used foundational data types. However, some data types may change in later phases of the project depending on our choice of database. For example, MySQL does not have a Boolean data type whereas PostgreSQL does.

## Business Logic Constraints

1. A **Store's Childcare Limit** must not be below 0
2. A **Store's Store Nbr, Phone Number, Street Address, Childcare Limit, Snackbar, Restaurant** cannot be Null
3. **City Name** must be in United States
4. **City State** must be in United States
5. **City Name, State, and Population** cannot be Null
6. **Product** cannot be sold at **Retail Price** and **Discount Price** for a single date
7. The **Discount Price** that is related to a **Product** must be less than the **Retail Price** of the **Product**
8. **Product PID, Name, and Retail Price** cannot be Null

## Task Decomposition and Abstract Code

*Main Menu:*

Task Decomposition

**Lock Types:**

**Number of Locks:**

**Enabling Conditions:**

**Frequency:**

**Consistency (ACID):**

**Subtasks:**

Abstract Code

Get List of **Categories** sorted alphabetically

For Each **Category**

*Update Populations*

Task Decomposition

**Lock Types:**

**Number of Locks:**

**Enabling Conditions:**

**Frequency:**

**Consistency (ACID):**

**Subtasks:**

Abstract Code

Get List of **Categories** sorted alphabetically

For Each **Category**

*Update Holidays:*  
Task Decomposition

**Lock Types:**

**Number of Locks:**

**Enabling Conditions:**

**Frequency:**

**Consistency (ACID):**

**Subtasks:**

Abstract Code

Get List of **Categories** sorted alphabetically  
For Each **Category**

*Report 1:*  
Task Decomposition

**Lock Types:**

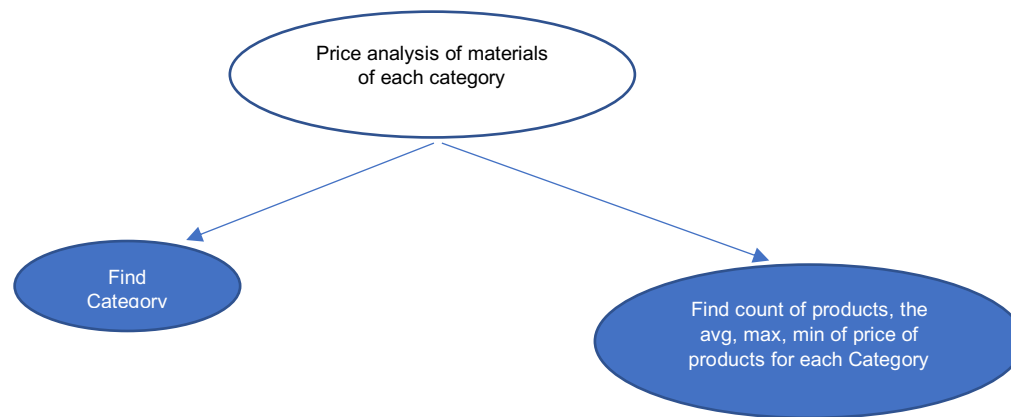
**Number of Locks:**

**Enabling Conditions:**

**Frequency:**

**Consistency (ACID):**

**Subtasks:**



## Abstract Code

Get List of **Categories** sorted alphabetically

For Each **Category**

    Get all **Products** and **Prices** that match the category

        Count all **Products**

        Average their **prices**

        Find the Max **Price**

        Find the Min **Price**

    Display the four numbers

## *Report 2*

### Task Decomposition

**Lock Types:**

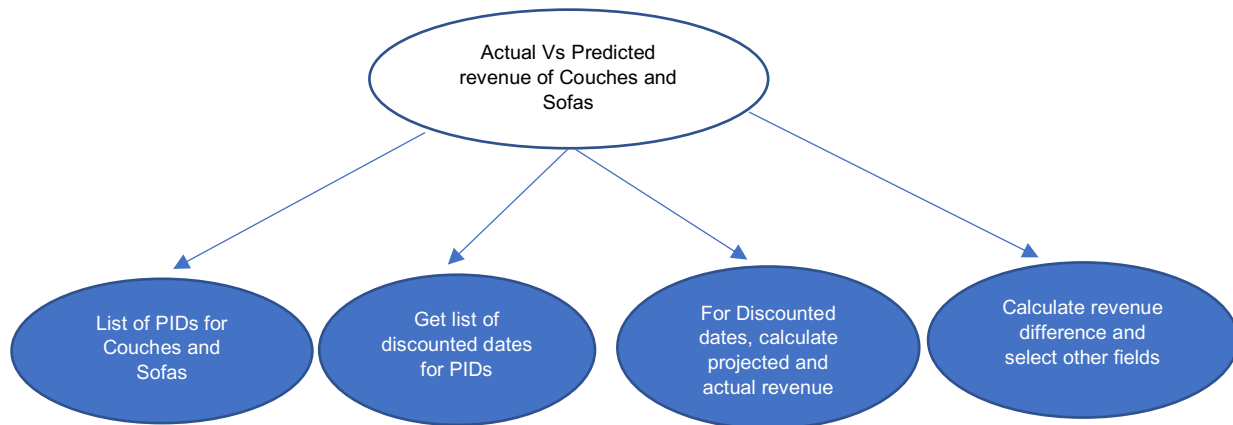
**Number of Locks:**

**Enabling Conditions:**

**Frequency:**

**Consistency (ACID):**

**Subtasks:**



## Abstract Code

Get a list of **PID** in category "Couches and Sofas"

For each **PID**:

    Get the list of **discounted dates**

    For each **discounted date**:

        Get the **Sales Qty** for this PID for this date

        Get the **Discounted Price**, **Retail Price** for this **PID** and **date**

        Calculate Actual Revenue as **Discounted Price** \* **Sales Qty**

        Calculated Projected Qty as **Sales Qty**\*0.75

        Calculate Projected Revenue as **Retail Price** \* **Projected Qty**

    Sum up the Actual Revenue, Projected Revenue, Sales Qty, Projected Qty

    Calculate Revenue Difference as Actual Revenue - Projected Revenue

    Display **PID**, **Product Name**, **Retail Price**, **Sales Qty**, Actual Revenue, Predicted Qty, Projected Revenue

Order by Diff Desc

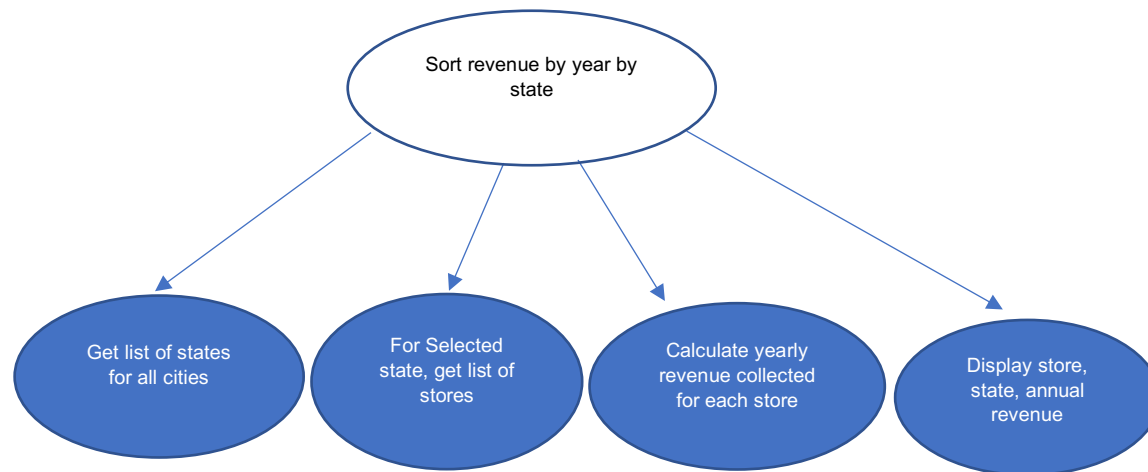
## Report 3

### Task Decomposition

**Lock Types:**

**Number of Locks:**

**Enabling Conditions:**

**Frequency:****Consistency (ACID):****Subtasks:****Abstract Code**

Get list of **States** for all **Cities**

Display **States**

When a button is pushed, then do the following:

Get list of Stores using the selected **State**

For each **Store**:

    Get **Store Nbr** and **Store Address**

    Get **City Name** from **City**

    Get list of **Sales** for **Store**

    Get list of **Dates** from **Sales**

    Calculate list of Years from **Dates**

    For each Year:

        For each **Sale** within Year:

            Get **Date** of **Sale**

            Get **PID** of **Sale**

            If **Discount** of **Date** and **PID** exists:

                Get **Discount Price** as Price



Else:

Get **Retail price** as Price

Get **Sales Qty** of **Sale**

Sum (**Sales Qty** \* Price) for each Year AS Total Year Revenue

Sum Total Year Revenue for each **Store** AS Total Revenue

Display **Store Nbr**, **Store Address**, **City Name**, Year, Total Revenue

Sort by Total Revenue descending and Year ascending;

## Report 4

### Task Decomposition

**Lock Types:**

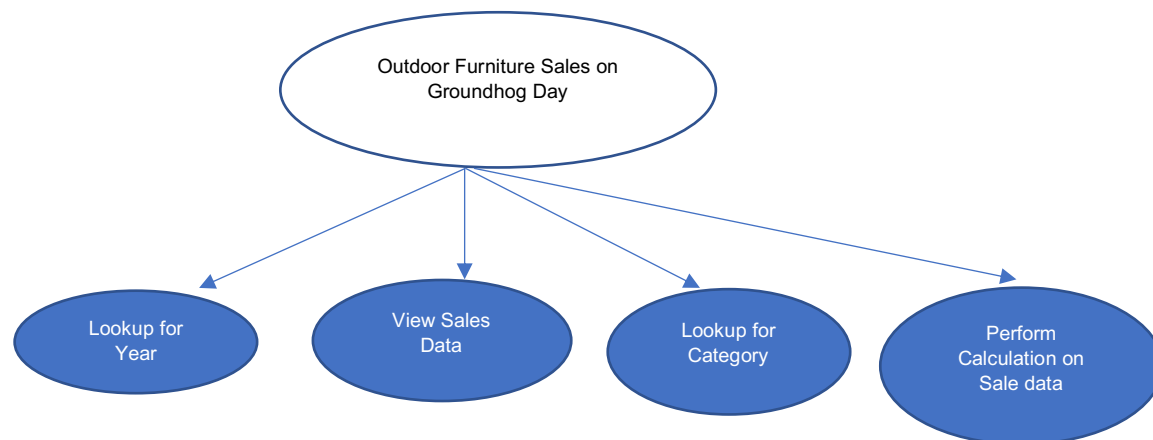
**Number of Locks:**

**Enabling Conditions:**

**Frequency:**

**Consistency (ACID):**

**Subtasks:**



### Abstract Code

Get list **Dates** from **Calendar**

Calculate list of Years from **Dates**

Get the list of **PIDs** that belong to the Category **Name** "Outdoor Furniture"

For Each **PID**:

    Get the Sales for that **PID**

    For each **Date** from the Sales

        Get the Sale **Qty**

        Add the Sale to the bucket for that Year

        If the date of the sale is February 2:

            Add the sale to the bucket for "Groundhog Day" for that year.

Divide the totals in each Year bucket by 365 to get Year Daily Average

Display Year Daily Average and "Groundhog Day" numbers.

## *Report 5*

### Task Decomposition

**Lock Types:**

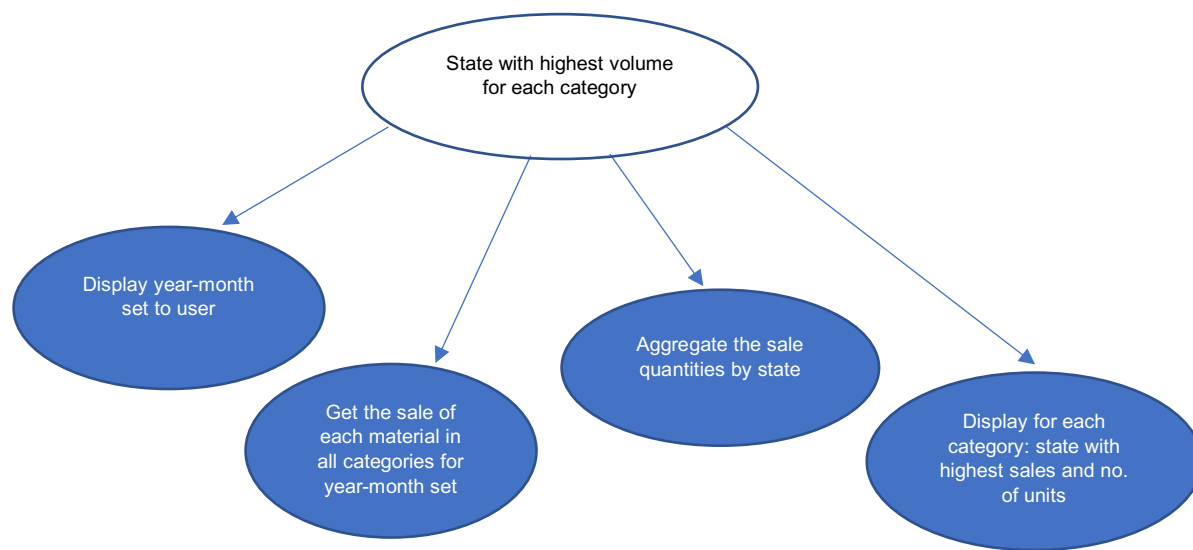
**Number of Locks:**

**Enabling Conditions:**

**Frequency:**

**Consistency (ACID):**

**Subtasks:**



## Abstract Code

```
Get list Dates from Calendar
Calculate list of Months and Year from Dates
Display the years and months
Let the user select one year-month set.
Get the list of Categories in ascending order.
With each Category
    Get the list of PIDs for that category
    With each PID
        Get the Sales for that PID for the required Year and Month.
        For each Sale
            Get the Store Number for the sale
            Get the State for that Store Number
        Sum up all the Sales quantities by State
    Select the State (or states) with the Highest Total Unit Sale
List out the top State or States
```

## *Report 6*

### Task Decomposition

**Lock Types:**

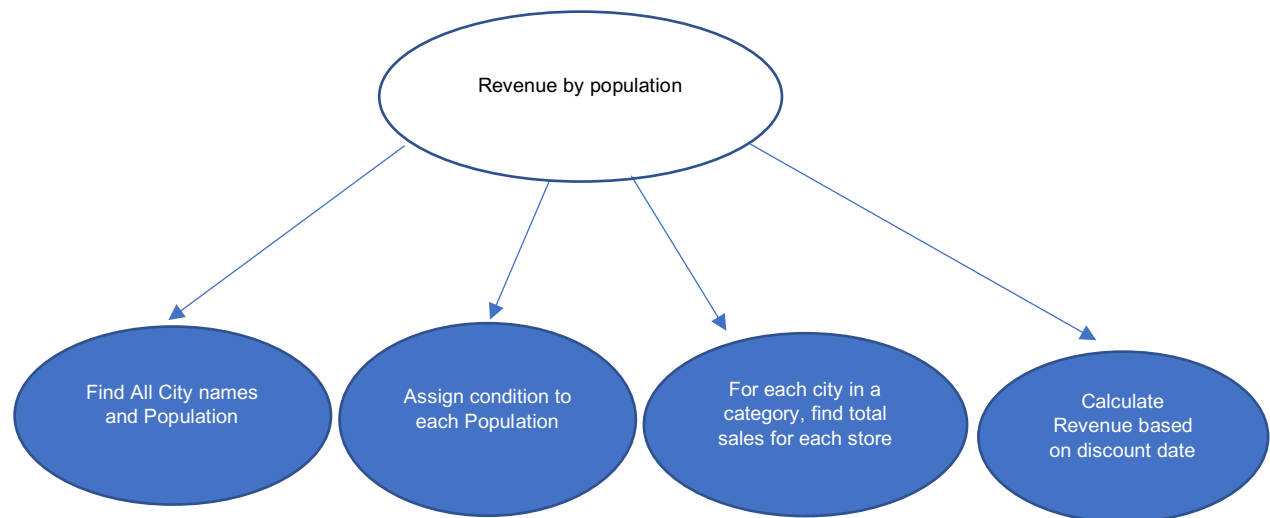
**Number of Locks:**

**Enabling Conditions:**

**Frequency:**

**Consistency (ACID):**

**Subtasks:**



## Abstract Code

Get list of all **City Names** and their **population**

For each **City Name**:

Assign Each **city** to a population category using condition\*\*

For each population category:

Get the list of **cities** for that category

For each **city**:

Get the list of stores in the city

For each **Store**:

Get the total sales by **PID** by date

For each **PID**:

If the **sales** happed on a discount date:

Then Revenue is **Sales\* Discount price**

Else:

Revenue is **Sales\* Retail price**

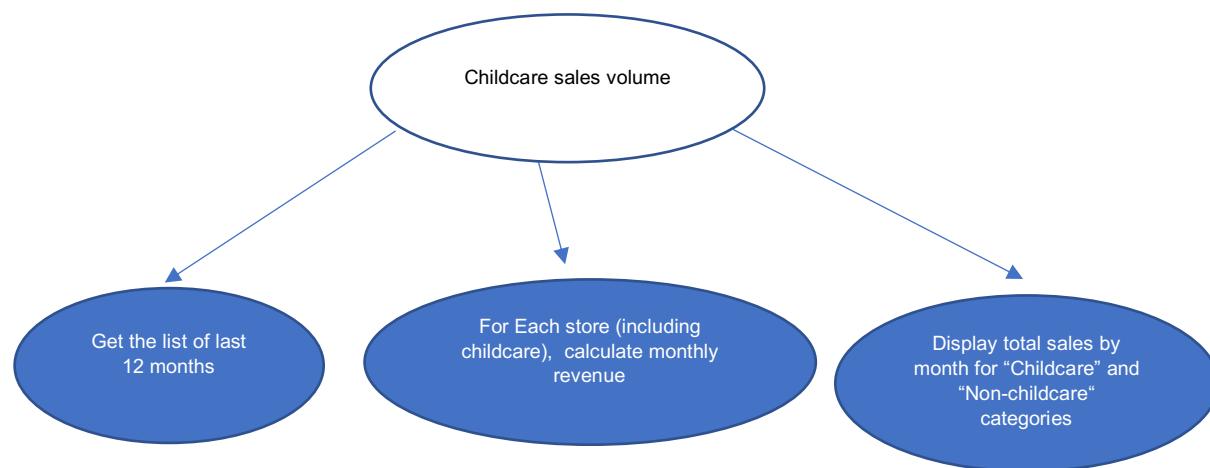
Sum up revenue by year

Sum up annual revenues for this category

\*\* The categories for city size are: Small (population <3,700,000), Medium (population >=3,700,000 and <6,700,000), Large (population >=6,700,000 and <9,000,000) and Extra Large (population >=9,000,000).

*Report 7*

## Task Decomposition

**Lock Types:****Number of Locks:****Enabling Conditions:****Frequency:****Consistency (ACID):****Subtasks:**

## Abstract Code

Get list **Dates** of **Sales**

Calculate list of Months from **Dates**

Limit Month list to most recent 12 months

Get list of **Childcare Limits**

For each **Childcare Limit**:

    Get list of **Stores** with **Childcare Limit**

    For each Month:

        For each **Store**:

            Get list of **Sales**

            For each **Sale**:

```

Get Date of Sale
Get PID with Sale
If Discount of Date and PID exists:
    Get Discount Price as Price
Else:
    Get Retail price as Price
    Get Sales Qty of Sale
Sum (Sales Qty * Price) as Total Revenue
Display Childcare Limit, Month, Total Revenue
IF Childcare Limit = 0:
    Childcare Limit = "No Childcare"

```

## Report 8

### Task Decomposition

**Lock Types:**

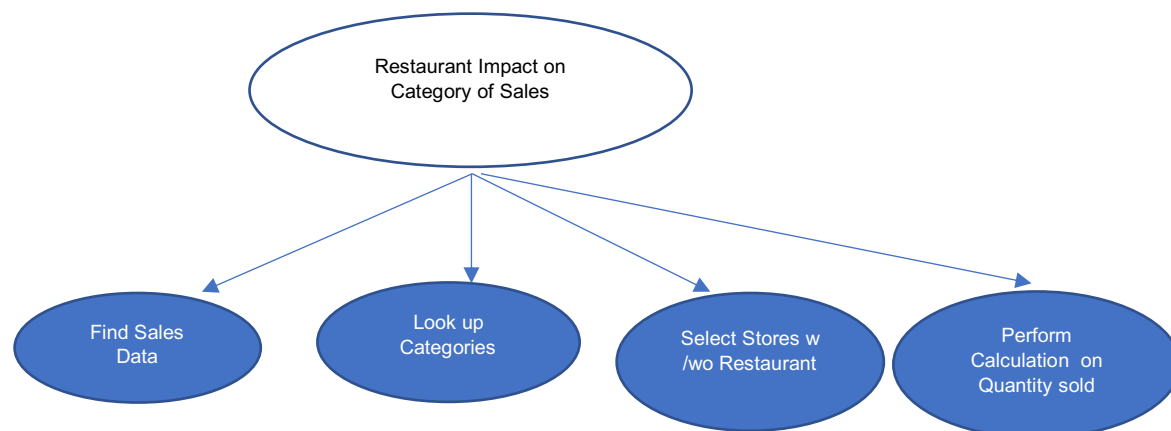
**Number of Locks:**

**Enabling Conditions:**

**Frequency:**

**Consistency (ACID):**

**Subtasks:**



Abstract Code

Get the List of **Categories**

For each **Category**

    Get the list of **PIDs** for the **Category**

    For each **PID**:

        Get all the **Sales**

        For each **Sale**:

            Get the **Store**

            Check if the Store has a **Restaurant Y/N**.

            If the **Store** has a **Restaurant**

                Add the Sale Qty to the Restaurant bucket for  
                that Category

            Else:

                Add the Sale Qty to the Non-Restaurant bucket  
                for that Category

    Display the total **Sale Qty** for the Restaurant and non-Restaurant bucket  
    for this **Category**

## *Report 9*

### Task Decomposition

**Lock Types:**

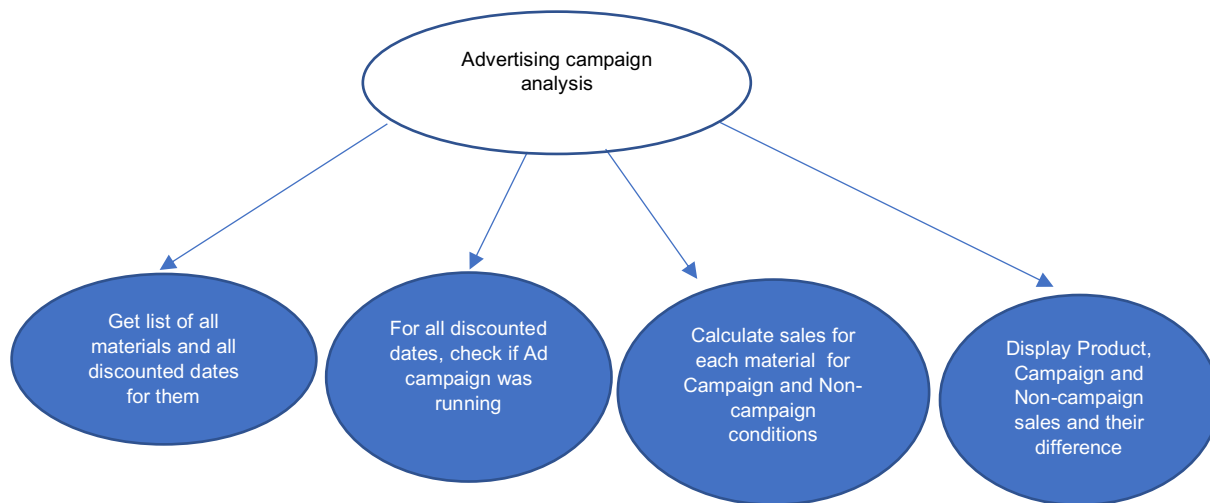
**Number of Locks:**

**Enabling Conditions:**

**Frequency:**

**Consistency (ACID):**

**Subtasks:**



## Abstract Code

Get the list of all **PIDs**

For each **PID**

    Get the **Discount Dates** for the **PID**

    For each **Discount Date**

        Get the **Sales** for that **PID** for that **Date**.

        Check to see if the **Discount/Sale Date** also had an **advertising campaign** running on the same **Date**.

        Assign the **Sales** to either the Campaign or non-Campaign bucket.

    Sum up the **sales** for each bucket.

    Get the differences for each bucket

Put all the **PID, Product Names**, campaign, non-campaign and difference numbers in descending order by difference.

Display the top 10 and bottom 10.