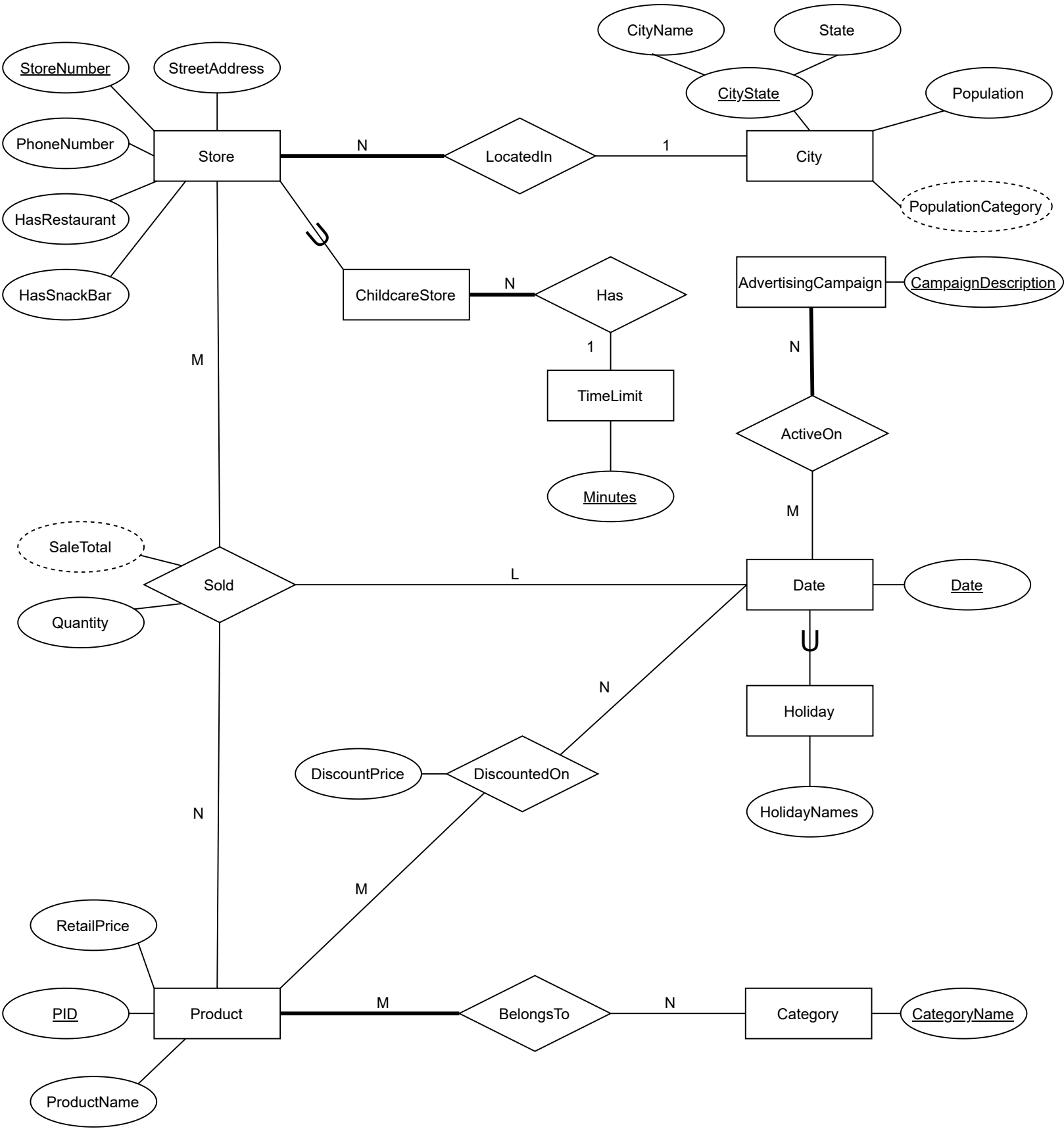
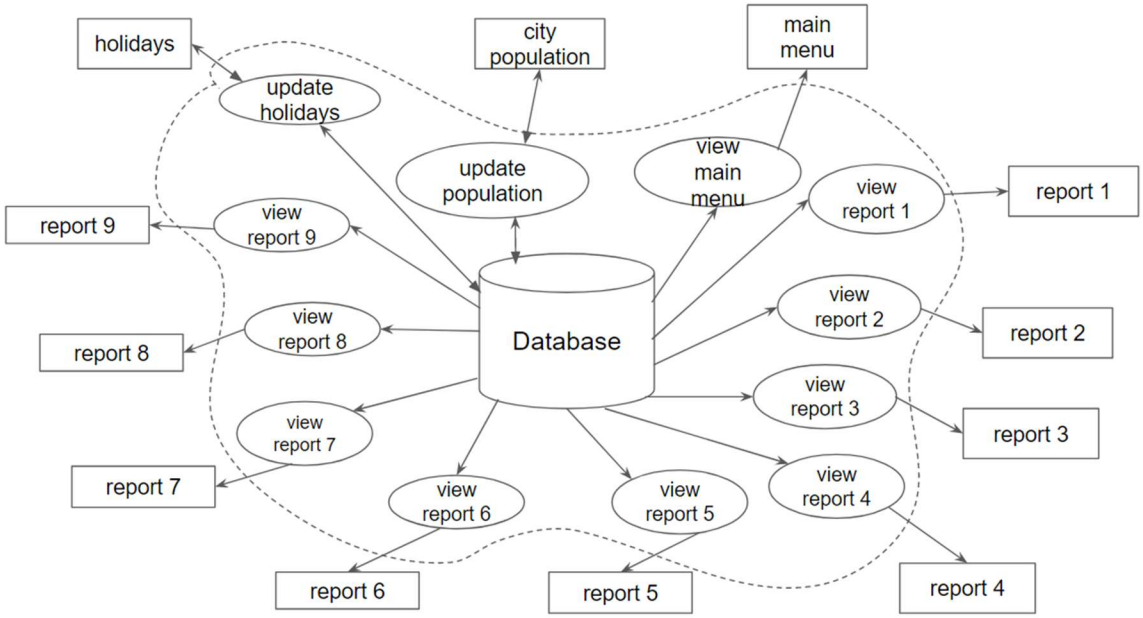
Table of Contents

EER Diagram - Team 032



Information Flow Diagram



Data Types

Store / ChildcareStore

Attribute	Data Type	Nullable	Notes
StoreNumber	Integer	Not Null	Unique Identifier
PhoneNumber	String	Not Null	Require Specific Format
StreetAddress	String	Not Null	
HasRestaurant	Boolean	Not Null	
HasSnackBar	Boolean	Not Null	

Time Limit

Attribute	Data Type	Nullable	Notes
Minutes	Integer	Not Null	Only required for childcare stores

City

Attribute	Data Type	Nullable	Notes
CityState	String	Not Null	Composite + Unique Identifier
CityName	String	Not Null	Component of CityState
State	String	Not Null	Component of CityState
Population	Integer	Not Null	
PopulationCategory	String	Not Null	Derived

Product

Attribute	Data Type	Nullable	Notes

PID	Integer	Not Null	Unique Identifier (Product ID)
ProductName	String	Not Null	
RetailPrice	Float	Not Null	

Category

Attribute	**Data Type**	**Nullable**	**Notes**
CategoryName	String	Not Null	Unique Identifier

DiscountedOn

Attribute	**Data Type**	**Nullable**	**Notes**
DiscountPrice	Float	Not Null	

Date

Attribute	**Data Type**	**Nullable**	**Notes**
Date	Date	Not Null	Unique Identifier

Holiday

Attribute	**Data Type**	**Nullable**	**Notes**
HolidayNames	String	Not Null	Only required for holiday subtype

Sold

Attribute	**Data Type**	**Nullable**	**Notes**
SaleTotal	Float	Not Null	Derived from discount price/retail price and quantity
Quantity	Integer	Not Null	

AdvertisingCampaign

Attribute	**Data Type**	**Nullable**	**Notes**
CampaignDescription	String	Not Null	Unique Identifier

Constraints

LSRS Product

- RetailPrice for any Product is in effect unless there is a discount price
- RetailPrice must be greater or equal to 0

LSRS DiscountedOn

- DiscountPrice for a product must be less than the RetailPrice of that product
- DiscountPrice must be greater or equal to 0

LSRS City

- Population must be greater than or equal to 0

LSRS TimeLimit

- TimeLimit can only be changed to certain predefined valid values offered by LSRS system

LSRS Sold

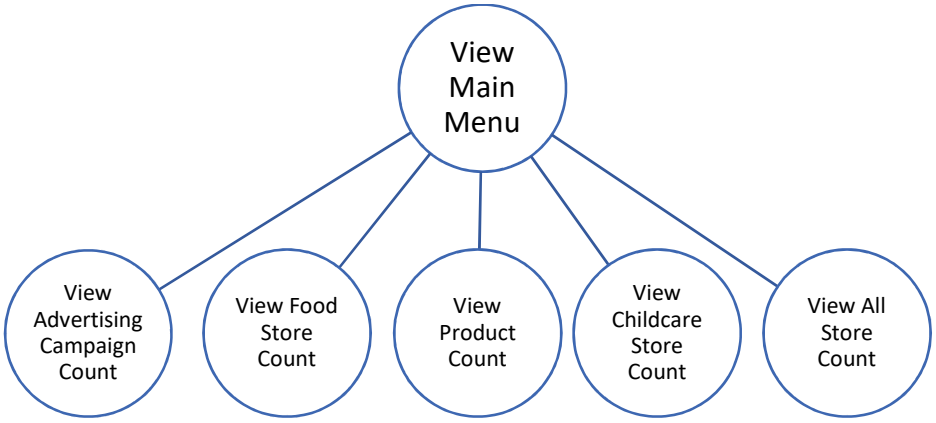
- Quantity of product sold must be greater than 0

Task Decomposition with Abstract Code

NOTE: Variables in abstract code are preceded with "$" (e.g. "$UserId"). Variable assignment is conveyed by placing the variable name in parentheses at the end of the line describing the value to be stored (e.g. "Sum of all sales ($SaleTotal)")

View Main Menu

Task Decomposition



Lock Types: 5 read-only lookups of Stores, Products, and Advertising Campaigns
Number of Locks: Several different schema constructs are needed
Enabling Conditions: None, all data visible on the initial view of the system
Frequency: All 5 have the same frequency
Consistency (ACID): order/consistency is not critical; data is not being updated regularly
Subtasks: All tasks must be done but can be done in parallel. Mother task is required

Abstract Code

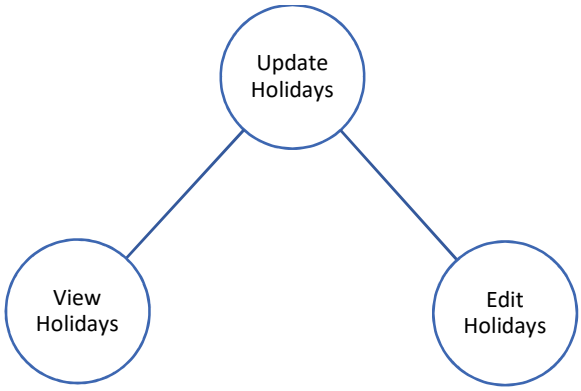
User **Views Main Menu**.

- Display "**Advertising Campaign Count"**
  - Find count of AdvertisingCampaign entities
  - Display count of AdvertisingCampaign entities
- Display **"Food Store Count"**
  - Find count of Store entities where HasRestaurant is true
  - Display count of these Store entities
- Display **"Product Count"**
  - Find count of Product entities
  - Display count of Product entities
- Display "**Childcare Store Count"**
  - Find count of ChildcareStore entities

    - Display count of ChildcareStore entities
- Display "**All Store Count"**
    - Find count of Store entities
    - Display count of Store entities
- Show "**Update Holidays**", "**Update City Population**", "**View Report 1**", "**View Report 2**", "**View Report 3**", "**View Report 4**", "**View Report 5**", "**View Report 6**", "**View Report 7**", "**View Report 8**", "**View Report 9**"
    - Click **Update Holidays** button – Jump to the **Update Holidays** task
    - Click **Update City Population** button – Jump to the **Update City Population** task
    - Click **View Report 1** button – Jump to the **View Report 1** task
    - Click **View Report 2** button – Jump to the **View Report 2** task
    - Click **View Report 3** button – Jump to the **View Report 3** task
    - Click **View Report 4** button – Jump to the **View Report 4** task
    - Click **View Report 5** button – Jump to the **View Report 5** task
    - Click **View Report 6** button – Jump to the **View Report 6** task
    - Click **View Report 7** button – Jump to the **View Report 7** task
    - Click **View Report 8** button – Jump to the **View Report 8** task
    - Click **View Report 9** button – Jump to the **View Report 9** task

Update Holidays

Task Decomposition



Lock Types: Lookup of holidays, edit of holidays
Number of Locks: Only one schema construct is needed, but two locks (one for reading, one for writing)
Enabling Conditions: Navigate to/open Holiday Management interface; click "save" after editing
Frequency: Viewing and editing of holidays can happen with (slightly) different frequencies
Consistency (ACID): Consistency is not critical (holidays do not need to be *absolutely* up to date when read)
Subtasks: Lookup must be performed before any update operations. Mother task is required

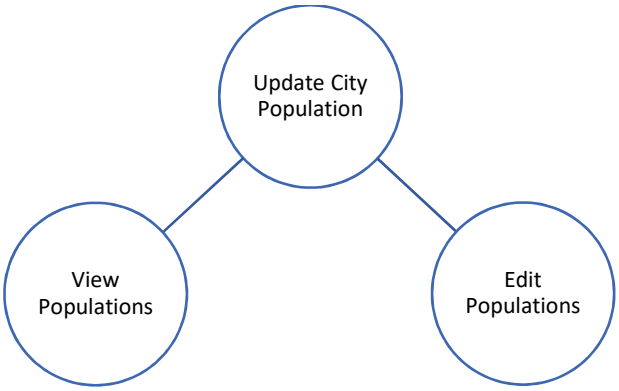
Abstract Code

User clicked on ***Update Holidays*** button from **Main Menu**.

- Display all Date and HolidayNames from Holiday entities
- While no buttons are pushed, do nothing.
- Click ***Save Holidays*** button – **Edit Holidays**.

Update City Population

Task Decomposition



Lock Types: Lookup of city populations, edit of city populations
Number of Locks: Only one schema construct is needed
Enabling Conditions: Navigate to/open City Population Management interface; click "save" after editing
Frequency: Viewing and editing of populations can happen with (slightly) different frequencies
Consistency (ACID): Consistency is not critical (populations do not need to be *absolutely* up to date when read)
Subtasks: Lookup must be performed before any update operations. Mother task is required

Abstract Code

User clicked on ***City Population*** button from **Main Menu**.

- Display all CityNames, State, and Populations from City entities
- While no buttons are pushed, do nothing.
- Click ***Save City Populations*** button – **Edit Populations**.

Report 1 – Category Report

Task Decomposition

Lock Types: 1 read-only lookup of query/view that joins Product and Category entities
Number of Locks: 3 read locks on Product, BelongsTo, and Category
Enabling Conditions: Navigate to Report 1 page
Frequency: no variation in frequency
Consistency (ACID): consistency is not critical
Subtasks: No mother task or decomposition needed

Abstract Code

User clicked on ***View Report 1*** button from **Main Menu**.

- For each Category, return:

    - CategoryName
    - The count of Product entities in the Category
    - Minimum RetailPrice across all products in the Category
    - Average RetailPrice across all products in the Category
    - Maximum RetailPrice across all products in the Category
- Sort results by CategoryName, ascending

Report 2 – Actual versus Predicted Revenue for Couches and Sofas

View Report 2

Task Decomposition

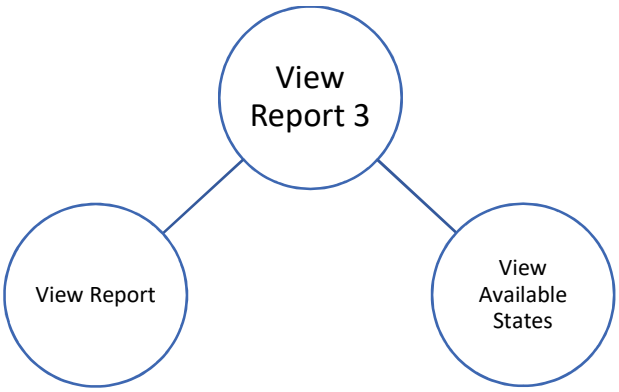
Lock Types: 1 read-only lookup joining Product, Category, DiscountedOn, Date, and Sold
Number of Locks: 5 read locks on Product, Category, DiscountedOn, Date, and Sold
Enabling Conditions: Navigate to Report 2 page
Frequency: no variation in frequency
Consistency (ACID): consistency is not critical
Subtasks: No mother task or decomposition needed

Abstract Code

User clicked on ***View Report 2*** button from **Main Menu**.

- For each Product in the "Couch" or "Sofa" Categories, return:
    - PID
    - ProductName
    - RetailPrice
    - Sum of all Quantities ever Sold ($TotalSold)
    - Sum of all Quantities Sold when the Product was DiscountedOn the sale Date
    - Sum of all Quantities Sold when the Product was *not* DiscountedOn the sale Date
    - Sum of all associated SaleTotals ($ActualRevenue)
    - Sum of $TotalSold * 0.75 * RetailPrice ($PredictedRevenue)
    - Difference of $ActualRevenue - $PredictedRevenue ($RevenueDifference).
- Filter the results to rows where the absolute value of $RevenueDifference > $5000
- Sort the results by $RevenueDifference, descending.

Report 3 – Store Revenue by Year by State



Task Decomposition

Lock Types: 2 read-only lookups: (1) query/view that joins Date, Store, City, and Product entities; (2) query of available states (from City entity)
Number of Locks: 6 read locks on Date, Store, Product, Sold, City, LocatedIn
Enabling Conditions: Navigate to Report 3 page
Frequency: "Available States" query only needs to be run when the report page first loads. The report query itself may be run multiple times thereafter for each page visit.
Consistency (ACID): consistency is not critical
Subtasks: Mother task required.

Abstract Code

User clicked on ***View Report 3*** button from **Main Menu**.

- Return unique States from City Entities
- User clicked on a State
    - For all Stores in the State selected, return:
        - StoreNumber
        - StreetAddress
        - CityName
        - Sum of SaleTotal* for sales within each Date year ($YearSaleTotal)
        - Date year
    - Sort results by Date year, ascending, then by YearSaleTotal, descending.

*SaleTotal is a derived attribute that calculates the total revenue of a sale by first looking to see if the Product was DiscountedOn the sale Date, then multiplying the Quantity by the DiscountPrice if on sale, and the RetailPrice otherwise.

Report 4 – Outdoor Furniture on Groundhog Day?

Task Decomposition

Lock Types: 1 read-only lookup of query/view that joins Date, Product, Category entities
Number of Locks: 6 read locks on Product, Sold, Date, DiscountedOn, BelongsTo, Category

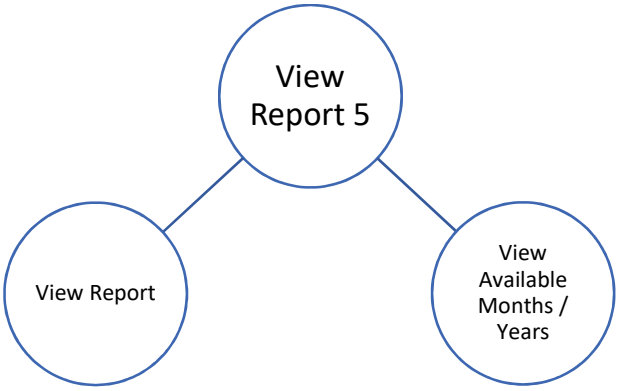
Enabling Conditions: Navigate to Report 4 page
Frequency: no variation in frequency
Consistency (ACID): consistency is not critical
Subtasks: No mother task or decomposition needed

Abstract Code

User clicked on ***View Report 4*** button from **Main Menu**.

- For each Date year, return:
  - Date year
  - Sum of Quantity Sold for Products in the "Outdoor Furniture" Category ($TotalUnitsSold),
  - Average Quantity sold per day assuming a 365-day year ($AvgQuantity = $TotalUnitsSold / 365)
  - Quantity Sold on the Date February 2nd for Products in the "Outdoor Furniture" category
- Sort the result by Date year, ascending.

Report 5 – State with Highest Volume for Each Category



Task Decomposition

Lock Types: 2 read-only lookups: (1) query/view that joins Date, Store, City, Category, and Product entities; (2) query of available Date Months and Years
Number of Locks: 7 read locks on Date, Category, BelongsTo, Product, Store, LocatedIn, City
Enabling Conditions: Navigate to Report 5 page
Frequency: "Available Dates" query only needs to be run when the report page first loads. The report query itself may be run multiple times thereafter for each page visit.
Consistency (ACID): consistency is not critical
Subtasks: Mother task required.

Abstract Code

User clicked on ***View Report 5*** button from **Main Menu**.

- User chooses year and month from Date entities
- For each Category in specified year and month return:
  - CategoryName
  - The State with largest sum of Quantity sold in that Category
  - Number of Products sold by Stores in that State
- Sort output by CategoryName, ascending

Report 6 – Revenue by Population

View Report 6

Task Decomposition

Lock Types: 1 read-only lookup of query/view that joins City, Store, Product, and Date entities
Number of Locks: 7 read locks on Product, DiscountedOn, Date, Sold, Store, LocatedIn, City
Enabling Conditions: Navigate to Report 6 page
Frequency: no variation in frequency
Consistency (ACID): consistency is not critical
Subtasks: No mother task or decomposition needed

Abstract Code

User clicked on ***View Report 6*** button from **Main Menu**.

- For each Date year and PopulationCategory* return:
  - The Date year
  - PopulationCategory
  - The sum of SaleTotals for that Date year in Cities with that PopulationCategory ($YearCategoryRevenue)
- Sort results by Date year, ascending, and by PopulationCategory in the order [“Small”, “Medium”, “Large”, “Extra Large”]
- Pivot the results such that the PopulationCategories form the columns and the Date years form the row index, with one $YearCategoryRevenue per each PopulationCategory in each Date year’s row.

*PopulationCategory is a derived attribute on City, calculated using the logic: if Population < 3.7 million, then “Small”; if Population >= 3.7 million and < 6.7 million, then “Medium”; if Population >= 6.7 million and < 9 million then “Large”; otherwise, if Population >= 9 million, “Extra Large”

Report 7 – Childcare Sales Volume

View Report 7

Task Decomposition

Lock Types: 1 read-only lookup of query/view that joins Date, Childcare Store, and Time Limit entities
Number of Locks: 8 read locks on Store, ChildcareStore, Has, TimeLimit, Sold, Date, DiscountedOn, Product
Enabling Conditions: Navigate to Report 7 page
Frequency: no variation in frequency
Consistency (ACID): consistency is not critical
Subtasks: No mother task or decomposition needed

Abstract Code

User clicked on ***View Report 7*** button from **Main Menu**.

- For each Date month in the last 12 months, return:

    - Date month
    - TimeLimit Minutes (or, for non-ChildcareStores, "No childcare") ($ChildcareCategory)
    - Sum of SaleTotal* for that Date month in stores with that $ChildcareCategory ($MonthCategorySales)
- Pivot the results such that the $ChildcareCategories form the columns and the Date months form the row index, with one $MonthCategorySales value per each $ChildcareCategory in each Date month's row

*SaleTotal is a derived attribute that calculates the total revenue of a sale by first looking to see if the Product was DiscountedOn the sale Date, then multiplying the Quantity by the DiscountPrice if on sale, and the RetailPrice otherwise.

Report 8 – Restaurant Impact on Category Sales

View Report 8

Task Decomposition

Lock Types: 1 read-only lookup of query/view that joins Product, Category and Store entities
Number of Locks: 5 read locks on Category, BelongsTo, Product, Sold, Stores
Enabling Conditions: Navigate to Report 8 page
Frequency: no variation in frequency
Consistency (ACID): consistency is not critical
Subtasks: No mother task or decomposition needed

Abstract Code

User clicked on ***View Report 8*** button from **Main Menu**.

- For each Category, return:
    - Category
    - Sum of Quantity Sold in Stores where HasRestaurant = True ($Restaurant)
    - Sum of Quantity Sold in Stores where HasRestaurant = False ($NonRestaurant)
- Un-pivot the Restaurant and NonRestaurant columns into a "StoreType" column and a "TotalQuantitySold" column, grouped by Category.
- Sort the results by Category, ascending and StoreType, ascending

Report 9 – Advertising Campaign Analysis

View Report 9

Task Decomposition

Lock Types: 1 read-only lookup of query/view that joins Product and Advertising Campaign entities
Number of Locks: 6 read locks on AdvertisingCampaign, ActiveOn, Date, DiscountedOn, Product, Sold
Enabling Conditions: Navigate to Report 9 page
Frequency: no variation in frequency
Consistency (ACID): consistency is not critical
Subtasks: No mother task or decomposition needed

Abstract Code

User clicked on ***View Report 9*** button from **Main Menu**.

- Construct $CampaignSaleQuantities dataset by returning, for each Product:
  - PID
  - ProductName
  - Sum of Quantity Sold on Dates where an AdvertisingCampaign was ActiveOn that Date and the Product was DiscountedOn that Date ($SoldDuringCampaign)
  - Sum of Quantity Sold on Dates where an AdvertisingCampaign was *not* ActiveOn that Date and the Product was DiscountedOn that Date ($SoldOutsideCampaign)
  - The difference between $SoldDuringCampaign and $SoldOutsideCampaign ($AdDifference)
- Sort $CampaignSaleQuantities dataset by $AdDifference, descending. Limit the output to only return 10 results ($CSQTop10).
- Sort $CampaignSaleQuantities dataset by $AdDifference, ascending. Limit the output to only return 10 results ($CSQBottom10). Re-sort this result by $AdDifference, descending.
- Return the union of $CSQTop10 and $CSQBottom10