View Revenue by Population Report

Task Decomp



**Lock Types**: 3 Read-only on CITY, STORE and SALE table

**Number of Locks**: Several different schema constructs are needed.

**Enabling Conditions**: Triggered when ***View Revenue by Population Report*** button is clicked.

**Frequency**: Low.

**Consistency (ACID)**: Not critical, order is not critical.

**Subtasks**: Mother Task is not needed. No decomposition needed.

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 store annual revenue from the SALE and STORE table based on the Store\_Number (from the STORE table) and *year* (date from the SALE table)
  + Find the Population\_ Size\_Category where the store located from the CITY and STORE table
  + Summary and group the Total Revenue based on Population\_ Size\_Category and year
  + Sort by year and Population\_ Size\_Category in ascending order
  + Display Total Revenue, City Size Category, Year in a tabular form
* When ready, user can click on the ***Return*** button to return to the **Dashboard** form.

View Childcare Sales Volume Report

Task Decomp



**Lock Types**: 3 Read-only on STORE, SALE and CHILDCARE table

**Number of Locks**: Several different schema constructs are needed.

**Enabling Conditions**: Triggered when ***View Childcare Sales Volume Report*** button is clicked.

**Frequency**: Medium – monthly report.

**Consistency (ACID)**: Not critical, order is not critical.

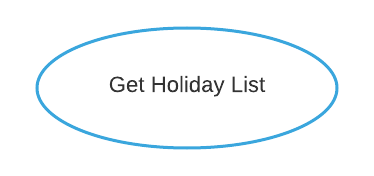
**Subtasks**: Mother Task is not needed. No decomposition needed.

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 summary the store monthly sales from the SALE and STORE table based on the Store\_Number (from the STORE table) and *month* (date from the SALE table) based on the last 12 months
  + Find the store with childcare category from the STORE and CHILDCARE table
  + Group the total sales by moth and by childcare category
  + Display Total Sale, Childcare Category and Month in a tabular form
* When ready, user can click on the ***Return*** button to return to the **Dashboard** form.

Get Holiday List

Task Decomp



**Lock Types**: Read-only on HOLIDAY table

**Number of Locks**: Single

**Enabling Conditions**: Triggered when ***Holiday Maintenance*** button is clicked.

**Frequency**: Low.

**Consistency (ACID)**: Not critical, order is not critical.

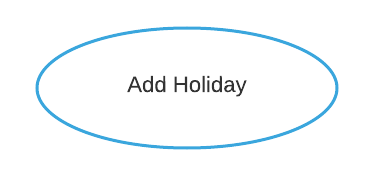
**Subtasks**: Mother Task is not needed. No decomposition needed.

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 (‘$HolidayName’) in input textbox and select Date (‘$HolidayDate’) in Calendar Dropdown
  + Click ***Add Holiday*** button –
    - If holiday is added successfully – display success message
    - If holiday is not added successfully – display failure message
* When ready, user can click on the ***Return*** button to return to the **Dashboard** form.

Add Holiday

Task Decomp



**Lock Types**: 2 write exclusive-locked on HOLIDAY and DAY table

**Number of Locks**: Several different schema constructs are needed.

**Enabling Conditions**: Triggered when ***Add Holiday*** button is clicked on **Holiday Maintenance** form

**Frequency**: Low.

**Consistency (ACID)**: Not critical, order is not critical.

**Subtasks**: Mother Task is not needed. No decomposition needed.

Abstract Code

* User enters Holiday Name (‘$HolidayName’) in input textbox and select Date (‘$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 this holiday with this date existed
  + - If Holiday Name is not existed 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.