Get Available State List

Task Decomp

Get Available State List

**Lock Types**: Read-only on CITY table.

**Number of Locks**: Single.

**Enabling Conditions**: Triggered when ***View Store Revenue by Year by State 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 ***View Store Revenue by Year by State Report*** button from the **Dashboard** form.
* Run the **Get Available State List** task: query for information about the available State\_Locationfields from the CITY table.
  + Display *State\_Location* list in ascending order on the drop-down list.
* On the drop-down list, show ***Run Report*** button.
* Upon:
  + Click ***Run Report*** button –
    - If *state* is selected – Jump to the **View State Revenue by Year by State Report** task.
    - If *state* field is empty – display a message asking for user input.
* When ready, user can click on the ***Return*** button to return to the **Dashboard** form.

View Store Revenue by Year by State Report

Task Decomp

View Store Revenue by Year by State Report

**Lock Types**: 6 Read-only on CITY, STORE, SALE, PRODUCT, DAY, and DISCOUNT table

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

**Enabling Conditions**: Triggered when ***Run Report*** button from the drop-down list 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 ***Run Report*** button from the drop-down list.
* If data validation is successful for *state* input fields, then proceed.
* Run the **View State Revenue by Year by State Report** task:
  + Find all stores from the STORE table based on the *state* (from the CITY table) selected.
  + On every sale *date* (from the DAY table), find each product’s sale revenue based on Total\_Amount (from the SALE table) and individual item price.
    - Total\_Amount (from the SALE table) is calculated based on the Date purchased (from the DAY table) and the Quantity (from the SALE table).
    - Individual item prices can be determined by Retail\_Price (from the PRODUCT table) or Discount\_Price (from the DISCOUNT table) when the product has a discount.
  + In each year, at each store in the selected state, find the total revenue by aggregating all products’ sale revenue on every sale date in that year.
  + Sort by sale year ascending, then sort by total revenue in descending order for each store in the selected state, display Store\_Number (from the STORE table), Street\_Address (from the STORE table) of the store, and City\_Name (from the CITY table).
* When ready, user can click on the ***Return*** button to return to the **Dashboard** form.

Maintain Population

Task Decomp

Maintain Population

Update Population

Get City List

**Lock Types**: Read and write on CITY table.

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

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

**Frequency**: Low – Both two have the same frequency.

**Consistency (ACID)**: Critical. If the city population is updated by the user, population category in Revenue by Population Report (Report 6) needs to be updated to make the database consistent.

**Subtasks**: Mother Task is required to coordinate subtasks. Order is necessary. **Get City List** first followed by **Update Population**.

Abstract Code

* User clicked on the ***Population Maintenance*** button from the **Dashboard** form.
* Run the **Get City List** task: query for information about the available *State\_Location* and *City\_Name* fields from the CITY table.
  + Display *State\_Location* list in ascending order in the drop-down list.
  + Select state and then display the available *City\_Name* (from the CITY table) list in the drop-down list.
  + Select target *City\_Name*, query for information about the Population (from the CITY table) and display population in view population textbox.
* On the drop-down list, show ***Run Report*** button.
* Upon:
  + User selects State (‘$StateName’) and then City (‘$CityName’) in the drop-down list.
  + Click ***Run Report*** button –
    - If city is selected – Jump to the **Update Population** task.
    - If city is not selected – display a message asking for user input.
* Run the **Update Population** task.
  + If data validation is successful for city name in Client Side, then proceed.
  + User enters updated population in input textbox.
  + Click on the ***Save*** button -
    - If the updated population entered is the same as the original population, do nothing.
    - If the updated population entered is different from the original population, update the *Population*and *Population\_Size\_Category* in the CITY table.
    - If updated population is not entered, populate a message asking for user input.
* When the ***Cancel*** button is pushed, return to the **Population Maintenance** form.
* When ready, user can click on the ***Return*** button to return to the **Dashboard** form.