**WiPayDBMS[standalone project]**

**Purpose:** This project is used to take the backup of the tables.

* First we start the program from WiPayDBMSMain class. Here in main () method we initialize **start\_time** to current date. We take the connection using CommonUtility.getWiPayConnection(); method and assign to **wipayCon** Connection variable.
* After that we call wipayTablesBackup (wipayCon); method by passing this connection. Then the control goes to this method which is in the same class. Here we call Utility.getWiPayBackupTables(); by passing the wipayCon [connection] and the returned value from this method is assigned to variable **tables\_set** [hashset].
* So, the control goes to getWiPayBackupTables() which is in Utility class. Here we initialize two variables; one is class level variable **wiPayBackupTable** which is used to set the values. And other is **WiPayBackupTable\_set** [HashSet] which is used to store the values and return.
* After that, we execute a query by selecting all the values from the table **wipay\_backup\_tables** where **wbt\_status='Y'**. **if(wbt\_day\_of\_the\_month == - 1 || db\_day\_of\_the\_month == wbt\_day\_of\_the\_month)**, if this condition is true, then we will set the values of **wipay\_backup\_tables** table to **wiPayBackupTable.** And we add these values to **WiPayBackupTable\_set** and return this to calling method and store in **tables\_set.**
* After that we check the condition if the **tables\_set** is empty or not. If the table is empty we will print “No tables\_set Exists”. Else we use for each loop and and assign these **tables\_set** to **table.** Here two new variables are initialized one is by getting the value from getting **source\_table** [for eg: request], other one is **dest\_table** by getting value from **Dest\_table**[for eg: request\_history].
* In this loop, we call a method, **backupTable()** by passing the **wipayCon**[connection] and this **table** as parameters.
* Here in this method, we initialize a Boolean variable **flag** to **false**. After that we initialize a **timestamp** variable **last\_day**. It will get the value from **Source\_table\_last\_ts**. After that, another data variable **from\_date** and takes the value from the **last\_day.**
* Next we take the **Calendar** object with **cal\_start** variable and set this variable with **settime**(from\_date). Similarly, we take **cal\_end** variable of Calendar object. Here we initialize end\_date variable to **current date**, and set this value to **cal.end.setTime(end\_date**), like that we set HOUR\_OF\_DAY, MINUTE, SECOND, MILLISECOND to 0, DAY\_OF\_WEEK to **table.getBackup\_before\_days()** [for eg: 2 days, 7 days etc.,]
* After that it will enter into while loop by checking the condition **while (cal\_start.before(cal\_end)),** again it will set all the parameters mentioned above.
* Next, it will check the **condition if (table.getBackup\_type().equals("DB") || table.getBackup\_type().equals("ALL"))** is true, then execute the query by inserting the source table values into destination table based on source index column(which is commonly timestamps for eg: time\_of\_req).
* After that we set connection to **autocommit(false**) by calling the method **wipayCon.setAutoCommit(false); [which means if one of the transaction or insertion of data got failed, then the entire table data updation also stops from the start]**. If the execution of all the data completes then we set the connection to commit by calling the method **wipayCon.commit();** **[which means updates the operation**]. To know the count of records got updated we print the count.
* **if(table.getBackup\_type().equals("CSV") || table.getBackup\_type().equals("ALL")),** if this condition is true, then it will initialize the value for the filename because we update the data as a **csv** file. Hence by using the **source\_file, csv\_location, startdate and end date**, we create a file name and execute the query as a csv file and committing operations are same as mentioned in the above condition [db condition].
* After that, we execute the condition of **if (table.getDelete\_source().equals("Y"))** is true, then we delete the data from the **source table after backup**.
* After deletion we update the source table last time stamp by calling a method **WiPayBackupTableDAO.updateSourceTableLastTs()** in which updates the last date.
* So the control enters into WiPayBackupTableDAO class and in **updateSourceTableLastTs**() method we update source table timestamps by executing the update query by getting the value from table.getSource\_table\_last\_ts(). After the updation, it will return to calling method in main class and updates the calendar and loop repeats. Then main class ends.

**Summary:**

Here we only use one table ie., **wipay\_backup\_tables.** So, we set the WiPayBackupTable .java [which is a Bean class] with all the values from the table by executing the query and setting those values to this bean. After that based on time stamps we insert the data in to the backup table [as db or csv based on requirement]. After inserting into backup tables, we delete the data in the source table [which is actual table] if necessary based on time stamps.