Primitive Operations:

loadDimensions():

Loads the different attributes present in the data (Time, Location, Account Type, etc.).

This could be done by fetching the meta from database. We can hand-code/hard-code at the very beginning.

setDefaults():

Sets default the dimension(s) on the cube. For example, we could as time as one of the default dimensions.

setAggregator(aggregator):

Sets how the values are combined in the table. Possibilities are Sum, Max, Min, Average, +/- variance

setDimension(dimension, axis):

Sets a dimension (Time, etc.) to one of the axes of the cube.

We will first support add one to the cube in a default order, then we will consider dragging.

removeAxis(axis):

Removes an axis from the cube (cube -> square -> line) and trigger an update to the underlying data representation (no need to query database at this moment)

drillDownOn(dimension):

Splits a dimension into finer-grained subdivisions, if present. Ex: Years -> Months -> Days -> Hours.

Drill down. formly spilt

rollUpOn(dimension):

Reverse of drillDown().

Roll up, formly merge

selectRange(dimension1…, coordinate1...):

Selects a bar/square/sub-cube on one the cube (defined by [dimension1…]).

If there is a third dimension, the selection will project along that dimension in the cube.

deselectRange(dimension1…, coordinate1...):

Deselect the current selected bar/square/sub-cube

cutCube(dimension1,...):

Creates a new cube comprised of the selected portion of the current cube.

slicingCube(dimension1)

dicingCube(dimension1,...)

showCube(cube):

Go back to the status that is specified by a cube in the history

pivot(dimension1, dimension2)

Swap the two dimensions

undo():

Undoes an action that modified the cube (drill down, roll up, etc).

redo():

Reverse of Undo(). Only possible after calling undo() and without modifying the cube.

Additional Interaction:

Scale:

Allows the user to scale the cube (essentially zooming into it).

Rotate:

Allows the user to rotate the cube.

Move:

Grab and move the cube.

Screenshot?

User Scenario:

1. User starts the program. An initial cube is shown without any dimensions, showing the total value in the database. A set of dimensions are visible in a floating menu.
   * loadDimensions()
   * setDefaults()
   * SELECT SUM(ACCT\_KPI\_TYPE\_TXN\_VAL) as 'Value' FROM BDC\_TXN\_FACT\_MA\_MORE base, BDC\_KPI\_DIM\_MORE KPI WHERE base.KPI\_TYPE\_ID = KPI.KPI\_TYPE\_ID;
2. User selects the Time dimension, and assigns it to the X axis.
   * setDimension(Time-year, x)
   * **SELECT DATE\_FORMAT (SNAPSHOT\_D, '%Y') as 'Time',** SUM(ACCT\_KPI\_TYPE\_TXN\_VAL) as 'Value' FROM BDC\_TXN\_FACT\_MA\_MORE base, BDC\_KPI\_DIM\_MORE KPI WHERE base.KPI\_TYPE\_ID = KPI.KPI\_TYPE\_ID **GROUP BY DATE\_FORMAT (SNAPSHOT\_D, '%Y')** ORDER BY 'Time';

(Instead of all of these abode, we simply start the program and load the defaults. The user will be able to see a YOY barchart on the cube with Money as the height.)

1. User selects the year 2016.
   * selectRange(Time, ‘2016’)
2. User cuts to just show that year.
   * cutCube(Time-year)
   * drillDownOn(Time-year) -> month
     1. Automatically split one dimension lower if possible and there is only one value for that dimension.
   * SELECT DATE\_FORMAT (SNAPSHOT\_D, '%Y') as 'Year', DATE\_FORMAT (SNAPSHOT\_D, '**%M'**) as 'Time', SUM(ACCT\_KPI\_TYPE\_TXN\_VAL) as 'Value' FROM BDC\_TXN\_FACT\_MA\_MORE base, BDC\_KPI\_DIM\_MORE KPI **WHERE base.KPI\_TYPE\_ID = KPI.KPI\_TYPE\_ID AND SNAPSHOT\_D >= "2016-01-01" AND SNAPSHOT\_D <= "2016- 12-31"** GROUP BY DATE\_FORMAT (SNAPSHOT\_D, '%Y**%M**') **ORDER BY DATE\_FORMAT (SNAPSHOT\_D, '%m')** ;
3. User undoes or go back to seeing yearly values.
   * undo() or showCube()
4. User adds the Business dimension.
   * setDimension(Business, y)
   * drillDownOn(Business) -> ????
   * SELECT **KPI\_BUSINESS\_NM as 'Business',** DATE\_FORMAT (SNAPSHOT\_D, '%Y') as 'Time', SUM(ACCT\_KPI\_TYPE\_TXN\_VAL) as 'Value' FROM base, BDC\_KPI\_DIM\_MORE kpi WHERE base.KPI\_TYPE\_ID = kpi.KPI\_TYPE\_ID GROUP BY DATE\_FORMAT (SNAPSHOT\_D, '%Y') ORDER BY 'Time';
5. User selects “New\_money” under Business, on the one-dimensional side.
   * selectRange(Business, “New\_Money”)
6. User cuts to just “New\_Money”
   * cutCube()
   * drillDownOn(Business) -> ????
   * SELECT KPI\_BUSINESS\_NM as 'Business', DATE\_FORMAT (SNAPSHOT\_D, '%Y') as 'Time', SUM(ACCT\_KPI\_TYPE\_TXN\_VAL) as 'Value' FROM BDC\_TXN\_FACT\_MA\_MORE base, BDC\_KPI\_DIM\_MORE kpi WHERE base.KPI\_TYPE\_ID = kpi.KPI\_TYPE\_ID **AND KPI\_BUSINESS\_NM = 'New\_money'** GROUP BY DATE\_FORMAT (SNAPSHOT\_D, '%Y') ORDER BY 'Time';
7. User selects 2015 for the year.
   * selectRange(Time-year, ‘2015’)
8. User cuts the slice of 2015.
   * cutCube()
   * drillDownOn(Time-year) -> monthly
   * SELECT DATE\_FORMAT (SNAPSHOT\_D, '%Y') as 'Year', DATE\_FORMAT (SNAPSHOT\_D, **'%M'**) as 'Time', SUM(ACCT\_KPI\_TYPE\_TXN\_VAL) as 'Value' FROM BDC\_TXN\_FACT\_MA\_MORE as base, BDC\_KPI\_DIM\_MORE as kpi WHERE base.KPI\_TYPE\_ID = kpi.KPI\_TYPE\_ID AND KPI\_BUSINESS\_NM = 'New\_money' **AND SNAPSHOT\_D >= "2015-01-01" AND SNAPSHOT\_D <= "2015-12-31"** GROUP BY DATE\_FORMAT (SNAPSHOT\_D, '%Y**%M**') ORDER BY DATE\_FORMAT (SNAPSHOT\_D, **'%m'**);
9. User rolls up on the Time dimension in order to see yearly values.
   * rollUpOn(Time) -> yearly
   * SELECT DATE\_FORMAT (SNAPSHOT\_D, '**%Y**') as 'Year', DATE\_FORMAT (SNAPSHOT\_D, '%Y') as 'Time', SUM(ACCT\_KPI\_TYPE\_TXN\_VAL) as 'Value' FROM BDC\_TXN\_FACT\_MA\_MORE as base, BDC\_KPI\_DIM\_MORE as kpi WHERE base.KPI\_TYPE\_ID = kpi.KPI\_TYPE\_ID AND KPI\_BUSINESS\_NM = 'New\_money' GROUP BY DATE\_FORMAT (SNAPSHOT\_D, '%Y') ORDER BY DATE\_FORMAT (SNAPSHOT\_D, **'%Y'**);
10. User adds the Money\_Category dimension.
    * setDimension(Money\_Category, z)
    * drillDownOn(Money\_Category) -> coarse
    * SELECT KPI\_BUSINESS\_NM as 'Business', **KPI\_CATEGORY\_COARSE\_X as 'Money\_Category'**, DATE\_FORMAT (SNAPSHOT\_D, '%Y') as 'Time', SUM(ACCT\_KPI\_TYPE\_TXN\_VAL) as 'Value' FROM BDC\_TXN\_FACT\_MA\_MORE as base, BDC\_KPI\_DIM\_MORE as kpi WHERE base.KPI\_TYPE\_ID = kpi.KPI\_TYPE\_ID AND KPI\_BUSINESS\_NM = 'New\_money' GROUP BY DATE\_FORMAT (SNAPSHOT\_D, '%Y'), KPI\_BUSINESS\_NM ORDER BY 'Time';
11. User switches the aggregate function to Max.
    * setAggregator(Max)
    * SELECT KPI\_BUSINESS\_NM as 'Business', KPI\_CATEGORY\_COARSE\_X as 'Money\_Category', DATE\_FORMAT (SNAPSHOT\_D, '%Y') as 'Time', **MAX**(ACCT\_KPI\_TYPE\_TXN\_VAL) as 'Value' FROM BDC\_TXN\_FACT\_MA\_MORE as base, BDC\_KPI\_DIM\_MORE as kpi WHERE base.KPI\_TYPE\_ID = kpi.KPI\_TYPE\_ID AND KPI\_BUSINESS\_NM = 'New\_money' GROUP BY DATE\_FORMAT (SNAPSHOT\_D, '%Y'), KPI\_BUSINESS\_NM ORDER BY 'Time';
12. User addes/replaces Money\_Category with Account\_Type
    * setDimension(Account\_Type, z)
    * drillDownOn(Account\_Type) -> something...
    * SELECT DATE\_FORMAT (SNAPSHOT\_D, '%Y') as 'Year', **REG\_ABBREV\_C as 'Account\_Type'**, KPI\_BUSINESS\_NM as 'Business', DATE\_FORMAT (SNAPSHOT\_D, '%M') as 'Time', MAX(ACCT\_KPI\_TYPE\_TXN\_VAL) as 'Value' FROM BDC\_TXN\_FACT\_MA\_MORE base, BDC\_KPI\_DIM\_MORE kpi, BDC\_ACCOUNT\_MINI\_DIM mini\_dim WHERE base.KPI\_TYPE\_ID = kpi.KPI\_TYPE\_ID AND mini\_dim.ACCT\_MINI\_DIM\_ID = base.ACCT\_MINI\_DIM\_ID AND KPI\_BUSINESS\_NM = 'New\_money' GROUP BY DATE\_FORMAT (SNAPSHOT\_D, '%Y') ORDER BY DATE\_FORMAT (SNAPSHOT\_D, '%Y') ;
13. User selects ‘IRA’ from Account\_Type and ‘2017’ from Time
    * selectRange([Account\_Type,Time], [‘IRA’,’2017’])
14. User cuts the cube to only show IRA accounts in 2017.
    * cutCube()
    * drillDownOn(Time) -> monthly
    * SELECT DATE\_FORMAT (SNAPSHOT\_D, '%Y') as 'Year', REG\_ABBREV\_C as 'Account\_Type', KPI\_BUSINESS\_NM as 'Business', DATE\_FORMAT (SNAPSHOT\_D, '%M') as 'Time', MAX(ACCT\_KPI\_TYPE\_TXN\_VAL) as 'Value' FROM BDC\_TXN\_FACT\_MA\_MORE base, BDC\_KPI\_DIM\_MORE kpi, BDC\_ACCOUNT\_MINI\_DIM mini\_dim WHERE base.KPI\_TYPE\_ID = kpi.KPI\_TYPE\_ID AND mini\_dim.ACCT\_MINI\_DIM\_ID = base.ACCT\_MINI\_DIM\_ID **AND REG\_ABBREV\_C='IRA'** AND KPI\_BUSINESS\_NM = 'New\_money' **AND SNAPSHOT\_D >= '2017-01-01' AND SNAPSHOT\_D <= '2017-12-31' GROUP BY DATE\_FORMAT (SNAPSHOT\_D, '%M') ORDER BY DATE\_FORMAT (SNAPSHOT\_D, '%m') ;**
15. User switches to averaging values.
    * setAggregator(Avg)
    * SELECT DATE\_FORMAT (SNAPSHOT\_D, '%Y') as 'Year', REG\_ABBREV\_C as 'Account\_Type', KPI\_BUSINESS\_NM as 'Business', DATE\_FORMAT (SNAPSHOT\_D, '%M') as 'Time', **AVG**(ACCT\_KPI\_TYPE\_TXN\_VAL) as 'Value' FROM BDC\_TXN\_FACT\_MA\_MORE base, BDC\_KPI\_DIM\_MORE kpi, BDC\_ACCOUNT\_MINI\_DIM mini\_dim WHERE base.KPI\_TYPE\_ID = kpi.KPI\_TYPE\_ID AND mini\_dim.ACCT\_MINI\_DIM\_ID = base.ACCT\_MINI\_DIM\_ID AND REG\_ABBREV\_C='IRA' AND KPI\_BUSINESS\_NM = 'New\_money' AND SNAPSHOT\_D >= '2017-01-01' AND SNAPSHOT\_D <= '2017-12-31' GROUP BY DATE\_FORMAT (SNAPSHOT\_D, '%M') ORDER BY DATE\_FORMAT (SNAPSHOT\_D, '%m') ;