|  |
| --- |
| Sajiv FrancisNovember 10, 2018 Data Warehouse  Assignment Three |
|  |

# Data Warehouse – Design Assignment

1. You should identify dimensions, map dimensions to data sources, and specify dimension hierarchies. For each dimension, you should identify its data sources and attributes in each data source. For hierarchical dimensions, you should indicate the levels from broad to narrow.

**Solution**: The dimension tables used are:

**Note**: ERD represents Entity Relationship Diagram - Snapshot

1. **Dim\_franchise**

* **Franchid:** ERD
* **Franchregionkey:** Derived from *region* column in *Dimgeography* based on lookup *postalcode* with postal code column in *Dimgeography.*
* **Postalcodekey:** Derived from *Geographykey* column in *Dimgeography* based on lookup *postalcode* with *postal code* column in *Dimgeography.*
* **Franchmodeltype:** ERD

1. **Dim\_geography**

* **Geographykey:** Primary key
* **Postalcode:** Derived from column postal code (tables *franchise* and *member).*
* **County:** Since geography is a universal dimension, this field is derived from the given postal code.
* **State:** Since geography is a universal dimension, this field is derived from the given postal code.
* **Country:** Derived similar to above.
* **Region:** Derived similar to above. 🡪Hierarchy: Region > Country > State > County

1. **Dim\_membertype**

* **Memtypeid:** ERD
* **Memtypename:** ERD
* **Memtypeprice:** ERD

1. **Dim\_date**

* **Datekey:** Primary Key
* **Date:** Dates have a universal dimension and are generated based on calendar dates. These have been used later as Role Playing Dimension for various dates in ERD.
* **Monthofyear:** Derived similar to above.
* **Weekofyear:** Derived similar to above.
* **Dayofweek:** Derived similar to above.
* **Calendarquarter:** These have been derived mainly using the universal dimension of dates.
* **Calendaryear:** Similar to above except checks can be made to add restrictions to limit data input to only certain criteria’s. 🡪Hierarchy: Calendaryear > Calendarquarter > Monthofyear > Date

1. **Dim\_servicecategory**

* **Servcatid:** Entity Relationship applied
* **Servcatname:** Entity Relationship applied
* **Servcatprice:** Entity Relationship applied

1. **Dim\_merchandise**

* **Merchid:** Entity Relationship applied as-is.
* **Merchname:** Entity Relationship applied as-is.
* **Merchprice:** Entity Relationship applied as-is.
* **MerchType:** Entity Relationship applied as-is.

1. **Dim\_member**

* **MemberID:** Entity Relationship applied as-is.
* **Membrname:** Entity Relationship applied as-is.
* **GeographyKey:** Derived key by looking *memberzipcode* with *postal code* column in *dim geography*
* **MembrtypeID:** Entity Relationship applied as-is.
* **Membrdatekey:** The key value is Derived by looking up *memberdate* with *date* in *dim date* table.
* **Franchid:** Entity Relationship applied as-is.
* **Mmbremail:** Entity Relationship applied as-is.

1. You should specify measures, related data sources, and measure aggregation properties.

**Solution:** The Measures derived from given Sources are:

1. **Quantity:** Derived from *contains* table from source - additional measure.
2. **Merchprice:** Derived from Merchandise table from source - additional measure.
3. **Salesamount:** It’s a calculated column derived from *Quantity* multiplied by *merchprice* – its an additional measure.
4. **Catprice:** Used as service price. It is recorded at the time of purchase from servicecatprice column in ServiceCategory table from Source – its an additional measure.
5. Identify the grain in your dimensional design using the business needs as a guideline. You should then indicate relative storage requirements for the grain using the statistics for the data sources. Using the cardinality estimates provided, you should determine either the fact table size or sparsity and then compute the unknown grain size variable. For example, you should compute sparsity if the fact table size is given.

**Solution:** This design uses a two fact table design - one is *fact\_merchandisesales table and the other is the fact\_service table.*

1. **Fact\_merchandisesales**

* **Member:** 50,000 rows/year
* The detailed grain is combination of memberID, saledate, merchID and franchID
* **Saledate:** 365 days/year
* **MerchID:** Derived from contains table – 450000/year.
* **Product of Dimensions:** 50000\*365\*450000
* **Sale:** This fact has been derived from table Sale – 150000/year.

**Fact\_merchandisesales..continued**

* **Sparsity Estimate:**

1-(fact table size/ Product of dimensions)

1-(150000/(50000\*365\*450000)) = 0.99

* The data cube has a few missing cells which are slightly more than 1% of cells with non zero values.

1. **Fact\_Service**

* **Member:** 50,000 rows/year
* The grain is defined by memberID,Servicepurchasedate.
* **Servicepurchasedate:** 365 days/year
* **Product of Dimension:** 50000\*365
* The fact is derived from service purchase : 100,000
* The worksheet used for fact: 60,000
* **Sparsity Estimate:**

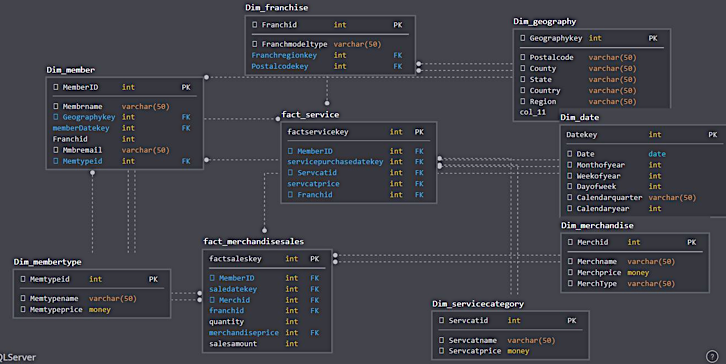
1-(fact table size/ Product of dimensions)

1-(160000/50000\*365) = 0.998

* The data cube has mostly missing cells with slightly more than 1% of cells with non zero values.

1. Extend your analysis to design a star schema (or variation) to support inventory analysis. For each table, you should define the table name, primary key, and columns. You do not need to write complete CREATE TABLE statements.

**Solution:** This Data Schema has been designed for the data ware house as per the requirements



1. You should populate your data warehouse tables based on the data in the sample tables and spreadsheet. You do not need to write SQL INSERT statements or insert data into database tables. You can just show table listings in your solution. You should indicate mappings from data sources into tables. For example, a mapping may involve generating new primary key values for a data warehouse table or using a default value for a missing value.

**Solution:**

The data warehouse tables have been derived from sample data available in this document.

**DimDate**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Datekey | date | Monthof year | Weekofyear | Dayofweek | Calendarquarter | Calendaryear |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |

Dimgeography

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Geographykey | Postalcode | County | State | Country | Region |
|  |  |  |  |  |  |
|  |  |  |  |  |  |

Dimfranchise

|  |  |  |  |
| --- | --- | --- | --- |
| Franchid | Franchmodeltype | Franchregionkey | Postalcodekey |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

Dimmerchandise

|  |  |  |  |
| --- | --- | --- | --- |
| Merchid | Merchname | Merchprice | MerchType |
|  |  |  |  |
|  |  |  |  |

Dimservicecategory

|  |  |  |
| --- | --- | --- |
| Servcatid | Servcatname | Servcatprice |
|  |  |  |
|  |  |  |

Dimmembertype

|  |  |  |
| --- | --- | --- |
| memtypeid | memtypename | memtypeprice |
|  |  |  |
|  |  |  |

DimMember

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| MemberID | Membrname | Geographykey | memberDatekey | Franchid | Mmbremail | Memtypeid |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |

Factmerchandisesales

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| factsaleskey | MemberID | saledatekey | Merchid | franchid | quantity | merchandiseprice | salesamount |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |

**Note:**

The data from Excel source is loaded into the above source tables in the prestage layer.

* From the excel source the corp custcode gives the memberid
* from customer name column member name is derived
* From customer location we get memberzip code
* Event type and event name loaded to dimension service category table
* And event date and event price Derived to fact\_purchase table