**Design Document**

For

**The Kambi Grill**

**DWH Modelling Process**

Author : Sayani Roy

Contents:

Page No.

1. Introduction ........................................................................................... 03

2. Source data details ................................................................................. 04

3. Table Structures ...................................................................................... 05

4. Data Mapping .......................................................................................... 10

5. Proposed Data Warehouse Design .......................................................... 11

6. Conclusion ................................................................................................. 13

1. Introduction

The purpose of this document is to record all the design details for Kambi Grill data warehouse. Here, we will discuss about the table structures, data mapping and proposed design data warehouse model. This warehouse will make the data analysis process easier and efficient.

1. Source data details:

Below is complete list for stage table name mapped to source file name.

|  |  |
| --- | --- |
| **Stage table Name** | **Source File Name** |
| membership\_type | membership\_type.csv |
| customer | customer.csv |
| customer\_preference | customer\_preference.csv |
| preference | preference.csv |
| campaign | campaign.csv |
| store | store.csv |
| order | order.csv |
| order\_list | order\_list.csv |
| order\_status | order\_status.csv |
| item | item.csv |
| customer\_audit\_log | Table Audit Data |
| campaign\_audit\_log | Table Audit Data |
| order\_audit\_log | Table Audit Data |

1. Table Structures:

Below are the table structures where data has been loaded.

|  |  |
| --- | --- |
| **Table Name :** | membership\_type |
| Column Name | Data Type |
| membership\_type\_id | INT |
| membership\_name | VARCHAR(30) |
| created\_date | TIMESTAMP |
| updated\_date | TIMESTAMP |
| PK : membership\_type\_id | |

|  |  |
| --- | --- |
| **Table Name :** | customer |
| Column Name | Data Type |
| customer\_id | INT |
| first\_name | VARCHAR(30), |
| last\_name | VARCHAR(30), |
| loyalty\_number | VARCHAR(30), |
| created\_date | TIMESTAMP, |
| updated\_date | TIMESTAMP, |
| membership\_type\_id | INT, |
| membership\_valid\_from\_date | TIMESTAMP, |
| membership\_valid\_to\_date | TIMESTAMP |
| PK : customer\_id FK : membership\_type\_id | |

|  |  |
| --- | --- |
| **Table Name :** | customer\_preference |
| Column Name | Data Type |
| customer\_preference\_id | INT, |
| customer\_id | INT, |
| prefrence\_id | INT, |
| prefrence\_name | VARCHAR(30), |
| prefrence\_value | VARCHAR(30), |
| created\_date | TIMESTAMP, |
| updated\_date | TIMESTAMP |
| PK : customer\_preference\_id FK : customer\_id FK : prefrence\_id | |

|  |  |
| --- | --- |
| **Table Name :** | preference |
| Column Name | Data Type |
| preference\_id | INT, |
| preference\_name | VARCHAR(30), |
| preference\_type | VARCHAR(30), |
| created\_date | TIMESTAMP, |
| updated\_date | TIMESTAMP |
| PK : preference\_id | |

|  |  |
| --- | --- |
| **Table Name :** | customer\_audit\_log |
| Column Name | Data Type |
| audit\_id | INT, |
| table\_name | VARCHAR(30), |
| primary\_key | VARCHAR(30), |
| column\_name | VARCHAR(30), |
| old\_value | VARCHAR(30), |
| new\_value | VARCHAR(30), |
| created\_date | TIMESTAMP |
| PK : customer\_audit\_log | |

|  |  |
| --- | --- |
| **Table Name :** | campaign |
| Column Name | Data Type |
| campaign\_id | INT, |
| campaign\_name | VARCHAR(50), |
| membership\_type\_id | INT, |
| store\_id | INT, |
| is\_expired | VARCHAR(5), |
| discount\_rate\_1 | INT, |
| discount\_rate\_2 | INT, |
| valid\_from\_date | TIMESTAMP, |
| valid\_to\_date | TIMESTAMP, |
| created\_date | TIMESTAMP, |
| updated\_date | TIMESTAMP, |
| created\_by | VARCHAR(30), |
| updated\_by | VARCHAR(30) |
| PK : campaign\_id FK : membership\_type\_id FK : store\_id | |

|  |  |
| --- | --- |
| **Table Name :** | store |
| Column Name | Data Type |
| store\_id | INT, |
| store\_name | VARCHAR(30), |
| address | VARCHAR(100), |
| city | VARCHAR(30), |
| country | VARCHAR(30), |
| tax\_rate\_1 | INT, |
| tax\_rate\_2 | INT, |
| created\_date | TIMESTAMP, |
| updated\_date | TIMESTAMP |
| PK : store\_id | |

|  |  |
| --- | --- |
| **Table Name :** | campaign\_audit\_log |
| Column Name | Data Type |
| campaign\_audit\_id | INT, |
| table\_name | VARCHAR(30), |
| primary\_key | VARCHAR(30), |
| column\_name | VARCHAR(30), |
| old\_value | VARCHAR(30), |
| new\_value | VARCHAR(30), |
| created\_date | TIMESTAMP |
| PK : campaign\_audit\_log | |

|  |  |
| --- | --- |
| **Table Name :** | order\_list |
| Column Name | Data Type |
| order\_list\_id | INT, |
| order\_id | INT, |
| item\_id | INT, |
| sale\_price | INT, |
| created\_date | TIMESTAMP, |
| updated\_date | TIMESTAMP |
| PK : order\_list\_id FK : order\_id FK : item\_id | |

|  |  |
| --- | --- |
| **Table Name :** | order |
| Column Name | Data Type |
| order\_id | INT, |
| customer\_id | INT, |
| campaign\_id | INT, |
| store\_id | INT, |
| order\_status\_id | INT, |
| sub\_total | INT, |
| total | INT, |
| tax\_total\_1 | INT, |
| tax\_total\_2 | INT, |
| discount\_total\_1 | INT, |
| discount\_total\_2 | INT, |
| created\_date | TIMESTAMP, |
| updated\_date | TIMESTAMP |
| PK : order\_id FK : customer\_id FK : campaign\_id FK : store\_id FK : order\_status\_id | |

|  |  |
| --- | --- |
| **Table Name :** | order\_status |
| Column Name |  |
| order\_status\_id | INT, |
| order\_status\_name | VARCHAR(30), |
| created\_date | TIMESTAMP, |
| updated\_date | TIMESTAMP |
| PK : order\_status\_id | |

|  |  |
| --- | --- |
| **Table Name :** | item |
| Column Name | Data Type |
| item\_id | INT, |
| item\_name | VARCHAR(30), |
| parent\_item\_id | VARCHAR(30), |
| is\_available | VARCHAR(5), |
| sale\_price | INT, |
| cost\_price | INT, |
| created\_date | TIMESTAMP, |
| updated\_date | TIMESTAMP |
| PK : item\_id FK : parent\_item\_id | |

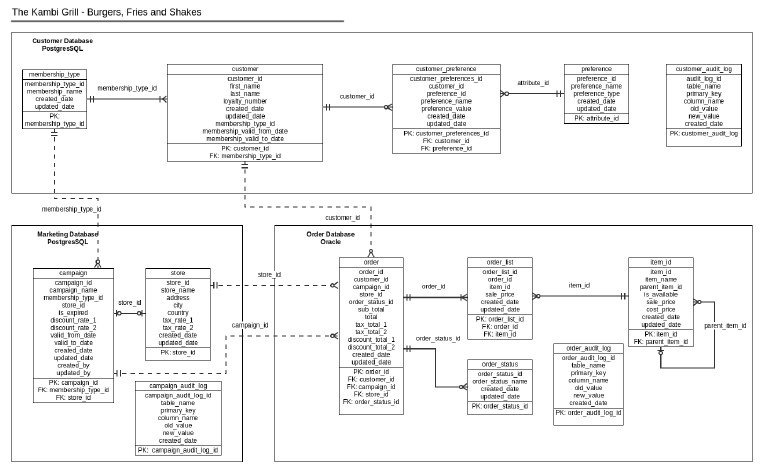
|  |  |
| --- | --- |
| **Table Name :** | order\_audit\_log |
| Column Name | Data Type |
| order\_audit\_log\_id | INT, |
| table\_name | VARCHAR(30), |
| primary\_key | VARCHAR(30), |
| column\_name | VARCHAR(30), |
| old\_value | VARCHAR(30), |
| new\_value | VARCHAR(30), |
| created\_date | TIMESTAMP |
| PK : order\_audit\_log\_id | |

1. Data Mapping:

Kambi Grill are working with three devOps domain, i.e. –

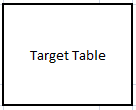
* Members Domain
* Marketing Domain
* Order Domain

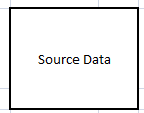
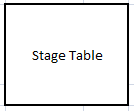
Above three domain data are mapped as per the below design.

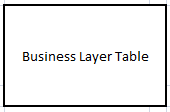


1. Proposed Data Warehouse Design:

The purpose of the development is to build a data warehouse where all data from different databases can be loaded in a single Warehouse Database. In that case, data accessibility from different table of different domain will be easier. Also, history data could be maintained properly and capturing the changes in data will be easier. It will help to analyse data more efficiently.







1. Source Data :

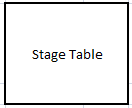
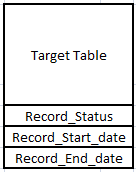
Source data can be loaded from different sources, like – several database or flat files. For this project, I have prepared dummy data for data visualization.

1. Stage Table :

Stage table structure will be same as the source data structure. Few audit columns can be added to track data load. It will load raw data from source.

1. Target Table :

Target table structure will be partially same as the Stage table structure. Audit columns will be present to track data load. The most important part of this layer will capturing data change. History data will be maintained properly in this layer. For that we can implement SCD TYPE 2 logics. New records will be added as “I” for indication, updated records will be mentioned as ‘U’ and deleted records will deactivated with ‘D’ value in record status column. In that case, records will not be deleted. Rather it will be present in table as deactivated. Record start date and end date column will help to extract data for past dates.



1. Business Layer Table :

The purpose of the business layer is to present BI data. It can describe as logical layer as well. After joining several tables and implementing business logics, the data will be loaded in a single table to be analysed more efficiently.

There are many examples from this project can be used where business layer could be helpful. Here, several problems need to be solved by analysing data which will extracted by using aggregation functions. These functions make query slower. So, if those queries are getting used frequently then we should build a business layer table where required dataset will be loaded in certain interval. Then data can be analysed from that table directly.

1. Conclusion:

In the previous sections, I have recorded all the table details along with mapping relationship among them. Also, a brief design for new data warehouse has been portrayed. The areas which can be improved by building a data warehouse have been focused.