DATAWARE HOUSE ASSESMENT -2

A) Category of a product may change over a period of time.Historical category information (current category as well as all old categories) has to be stored.Which SCD type will be suitable to implement this requirement? What kind of structure changes are required in a dimension table to implement SCD type 2 and type 3?

A Slowly Changing Dimension (SCD) is a dimension that stores and manages both current and historical data over time in a data warehouse. It is considered and implemented as one of the most critical ETL tasks in tracking the history of dimension records.

Structure table changes are required in a dimension table to implement the SCD Types

SCD Type 1:

In a Type 1 SCD the new data overwrites the existing data. Thus the existing data is lost as it is not stored anywhere else. This is the default type of dimension you create. You do not need to specify any additional information to create a Type 1 SCD.

Before Record

PROD_KEY	PROD_NAME	PRICE
1	Mobile	15,000
2	Footware	2000

Updated the price of prod_key 2 to 3000 in updated record it will change the Price of updated record to 3000 of Prod_Key 2.

Updated Record

PROD_KEY	PROD_NAME	PRICE
1	Mobile	15,000
2	Footware	3000

SCD Type 2:

A Type 2 SCD retains the full history of values. When the value of a chosen attribute changes, the current record is closed. A new record is created with the changed data values and this new record becomes the current record. Each record contains the effective time and expiration time to identify the time period between which the record was active.

Before Record

PROD_KEY	PROD_NAME	PRICE	START_DATE	END_DATE
1	Mobile	15,000	01-05-2015	Null
2	Footware	2000	02-08-2016	Null

Updated the Start_Date of prod_key 1 in this the End_Date of before record will be changed to updated record of Start_Date.

Updated Record

PROD_KEY	PROD_NAME	PRICE	START_DATE	END_DATE
1	Mobile	15,000	01-05-2015	08-08-2015
2	Footware	2000	02-08-2016	Null
1	Mobile	15,000	08-08-2015	Null

SCD Type 3:

A Type 3 SCD stores two versions of values for certain selected level attributes. Each record stores the previous value and the current value of the selected attribute. When the value of any of the selected attributes changes, the current value is stored as the old value and the new value becomes the current value.

Before Record

PROD_KEY	PROD_NAME	PRICE	PREVIOUS PRICE
1	Mobile	15,000	
2	Footware	2000	

Updated the price of Prod_Key 2 to 4500 so in Updated Record the previous price will be changed to 4500 in Prod_Key 2.

Updated Record

PROD_KEY	PROD_NAME	PRICE	PREVIOUS PRICE
1	Mobile	15,000	
2	Footware	2000	4500

B) What is surrogate key? Why it is required?

Surrogate Key:

- Surrogate keys are widely used and accepted design standard in data warehouses. It is sequentially generated unique number attached with each and every record in a Dimension table in any Data Warehouse.
- Surrogate keys join between the fact and dimension tables and is necessary to handle changes in dimension table attributes.
- Surrogate Key (SK) is sequentially generated meaningless unique number attached with each and every record in a table in any Data Warehouse (DW).
- It is UNIQUE since it is sequentially generated integer for each record being inserted in the table.
- It is MEANINGLESS since it does not carry any business meaning regarding the record it is attached to in any table.
- It is SEQUENTIAL since it is assigned in sequential order as and when new records are created in the table, starting with one and going up to the highest number that is needed.

Why it is required:

- Basically Surrogate key is an artifical key that is used as a substitute for natural key(NK) defined in data warehouse tables.
- we can use natural key or business keys as a primary key for tables.
- Natural keys(NK) or Business Keys are generally aplhanumeric values that is not suitable for index as traversing become slower.

Example:

prod123,prod231 etc.

- Business keys are often reused aftersometime.It will cause the problem as in data warehouse we maintain historic data as well as current data.

Example:

- product codes can be revised and reused after few years.
- It will become difficult to differentiate current products and historic products.

C) Stores are grouped in to multiple cluster.A store can be part of one or more clusters.Design tables to store this store-cluster mapping information?

Consider a table of Store which are located in bengaluru like “Hebbal”,”Yelhanaka”,”Malleswaram” etc.In this the one store will be related to another store like “Hebbal” and “Yelhanka” will be a clustered store.In the below table we have created clustered table with the primary key of store_id that key is present in Store table.In clustered mapping table the store_id will be same for some location and make a fact of Clustered mapping table.

CLUSTERED TABLE

CLUSTER_ID	NAME	LOCATION	STORE_ID
101	Udit	Hebbal	01
102	Suraj	Malleswaram	02
103	Javid	Yelhanka	03

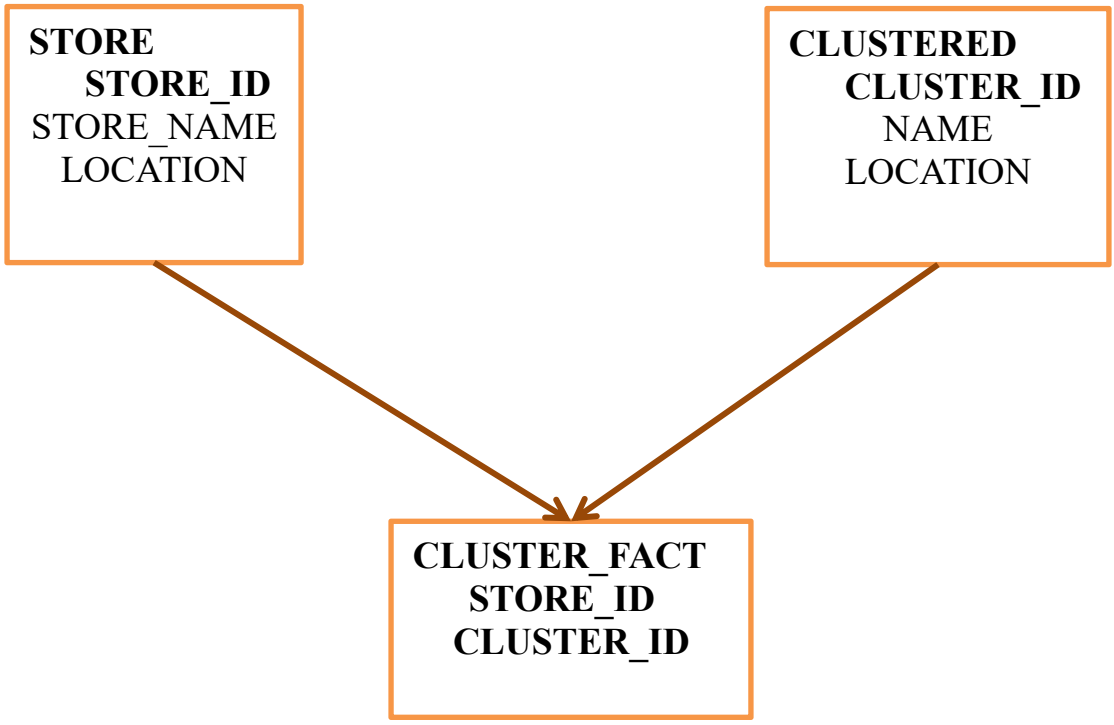
STORE TABLE

STORE_ID	STORE_NAME	LOCATION
01	Reliance Trends	Hebbal
02	More	Malleswaram
03	Brand Factory	Yelhanka

CLUSTERED MAPPING TABLE

CLUSTER_ID	STORE_NAME	LOCATION	STORE_ID
101	Reliance Trends	Hebbal	01
103	Brand Factory	Malleswaram	02
103	Brand Factory	Yelhanka	02

DIMENSION FOR CLUSTERED MAP TABLE



D) **What is a semi-additive measure? Give an example?**

Semi-Additive measures:

- Semi-additive facts are facts that can be summed up for some of the dimensions in the fact table, but not the others.
- Inventory levels and financial account balances are semi-additive because they are additive across all dimensions except time

Inventory Example:

We have stock levels say 1000(qty of Item A) on Monday...I sell 200(qty of Item A) on Tuesday I further sell 300(qty of Item A) on Wednesday...going by basic math On Thursday I should be left with 500(qty of Item A, assuming no inventory has flown in) to obtain current stock level I cannot aggregate the Stock sales across time dimension hierarchy...If done I will have inappropriate outcomes

Bank Account Example:

DATE
Account
Balance

Here the Current Balance measure is semi-additive because adding up all the current balances for a given account for each day of the month doesn't give any useful information.

In order to get further clarity, let's assume that on the first day of the week, you have $50 in your account. Next two days, the balance remains unchanged. On the fourth day, you deposit another $50 so the balance is now $100. The account has no further activity through the end of the week.This case we can't merely add up the daily balances during the week and declare that the ending balance is $550 (based on $50 + $50 + $50 + $100 + $100+ $100 + $100). The most useful way to combine account balances is to find average daily balance on a monthly account summary. Because in this example table, Current Balance represents the balance amount at one point in time