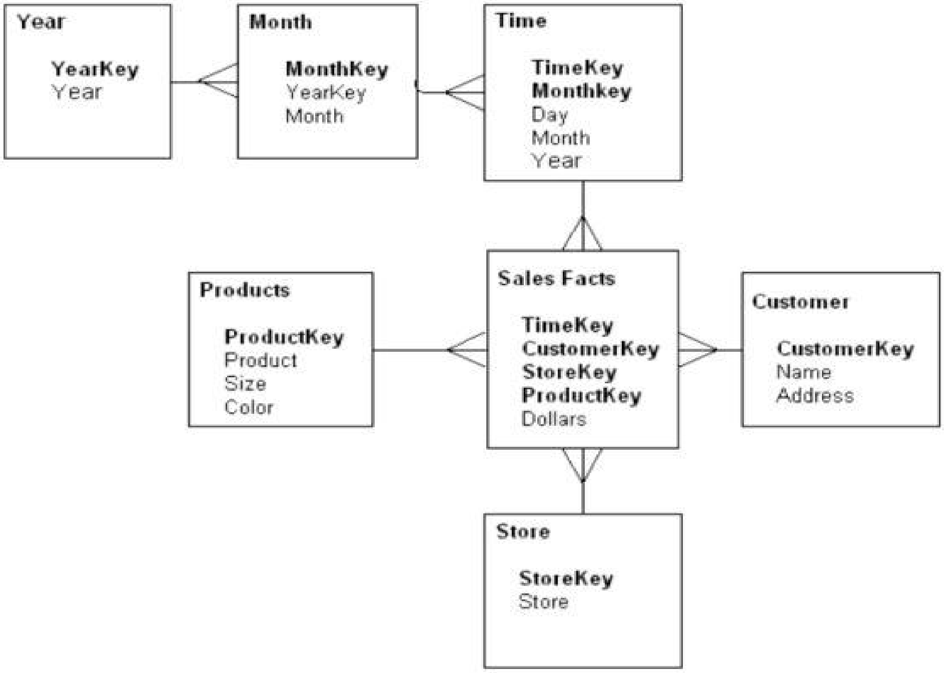
Name:ANKIT SINGH

::::*DATA WAREHOUSE ASSESSMENT*::::

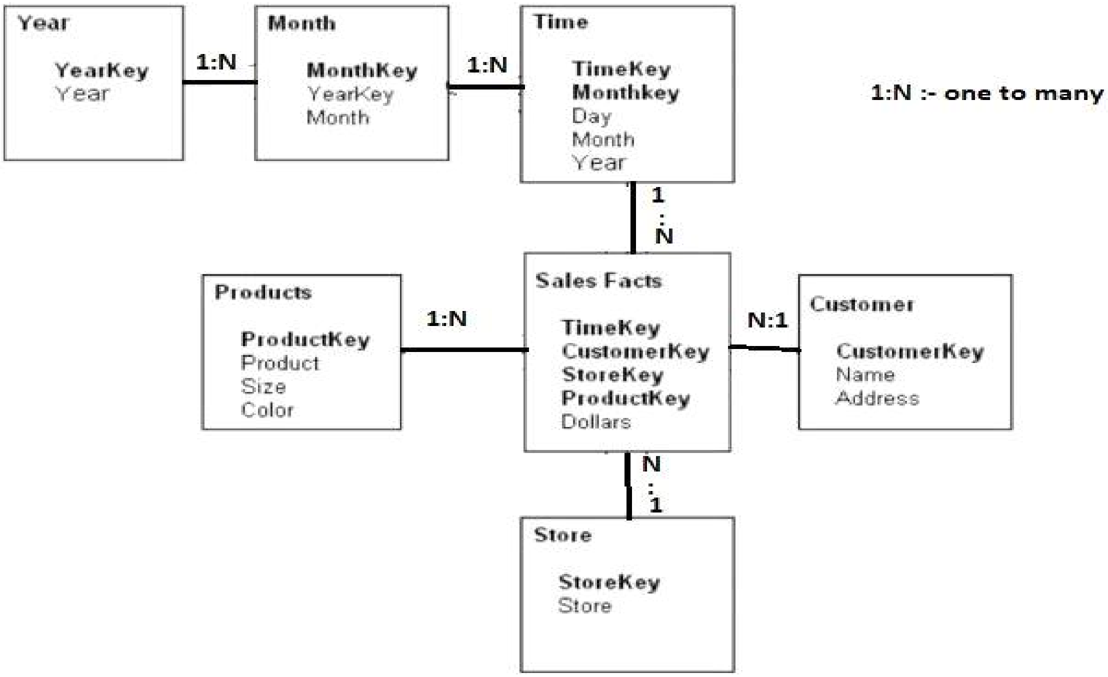
1. For the given Dimensional Modelling, please identify the following:



> How many dimensions and fact tables are present?

Ans:- There are 6 dimensions table and 1 fact table

> Please identify the cardinality between each table.



> How to create Sales_Aggr fact using following structure (SQL Statement) ?

Sales_Aggr

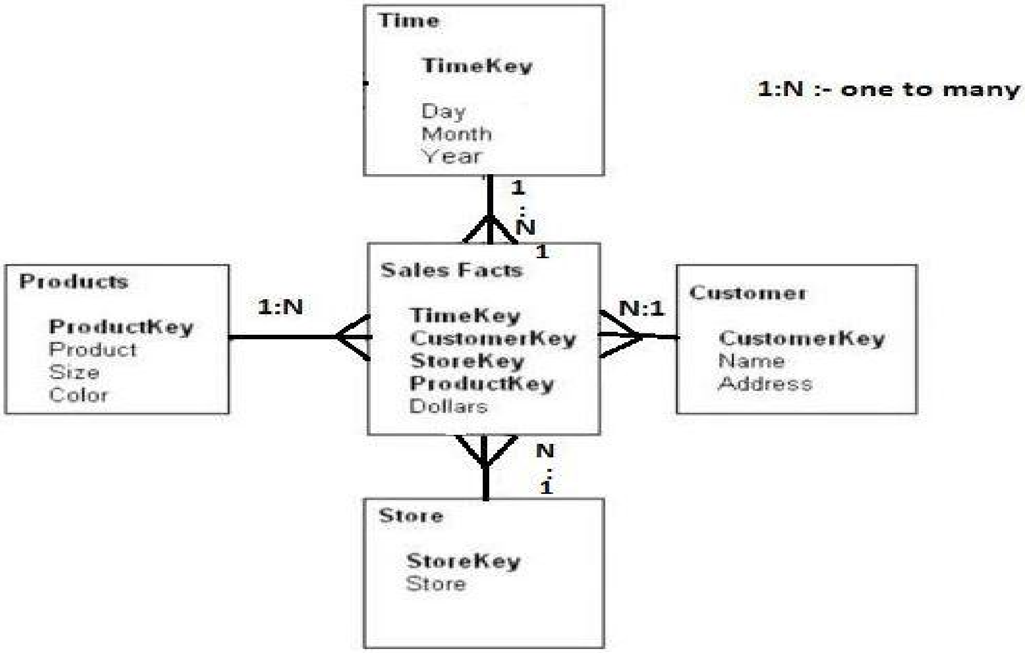
Year_ID
Customer_Key
Store_key
Product_key
Dollars

Ans:-

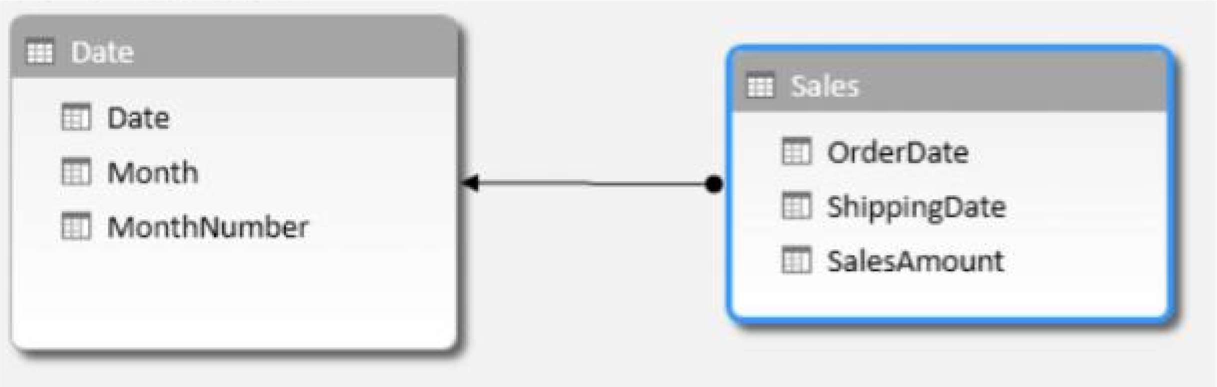
```
CREATE TABLE Sales_Aggr (Year_ID INT, Customer_Key INT, Store_Key INT,
    Product_Key INT, Dollars , FOREIGN KEY (Year_ID) REFERENCES
    Year(YearKey), FOREIGN KEY (Customer_Key)      REFERENCES
    Customer(CustomerKey), FOREIGN KEY (Store_Key) REFERENCES
    Store(StoreKey), FOREIGN KEY (Product_Key) REFERENCES
    Product(ProductKey), PRIMARY KEY (Year_ID, Customer_Key, Store_Key,
    Product_key) );

    INSERT INTO Sales_Aggr (Year_ID, Customer_Key, Store_Key, Product_Key,
    Dollars) VALUES ( SELECT Y.YearKey, C.CustomerKey, S.StoreKey,
    P.ProductKey,F.Dollars FROM Year Y, Customer C, Store S, Product P,
    Sales Facts F);
```

> Can you please modify the above snowflake schema to star schema and draw the dimension model, showing all the cardinality ?

Ans:- The Star Schema of the given snow-flake schema is shown below:-



2. For the following dimension Model can you please give an example of Circular Join and how to avoid it:



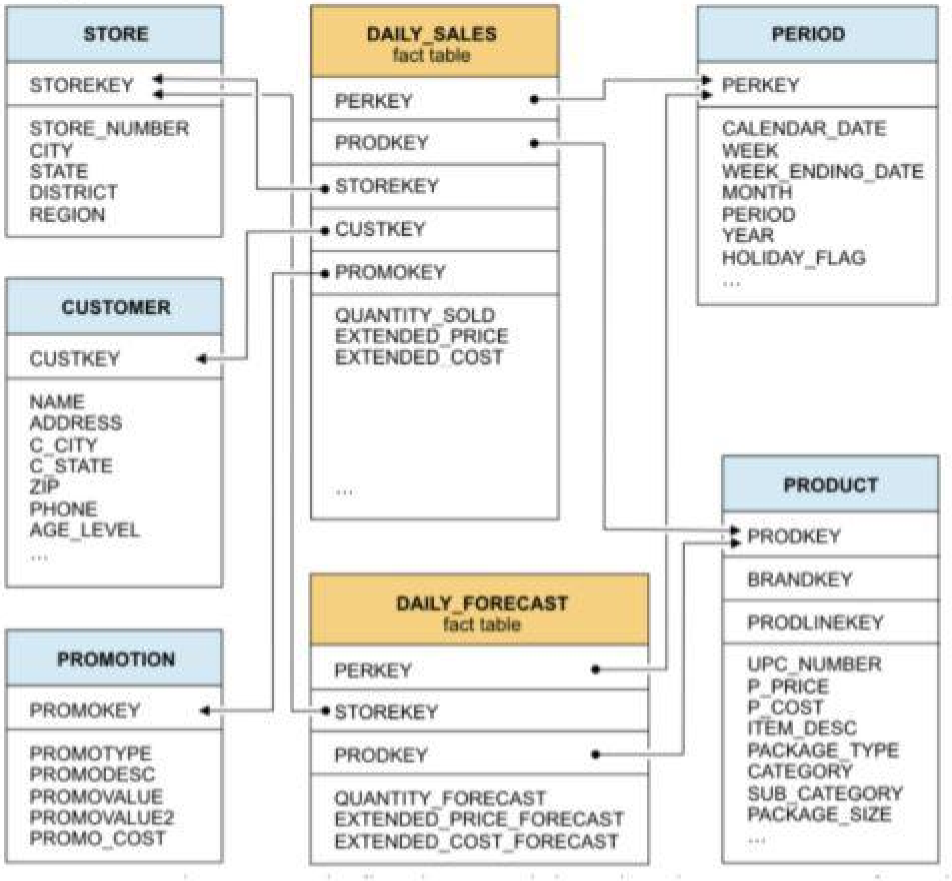
Ans: An example of circular join is shown below :-

```
SELECT S.OrderDate, S.Shipping Date
FROM Sales S,Date D
WHERE D.Date=S.OrderDate AND D.Date=S.ShippingDate;
```

To avoid circular join in given tables we need to execute below query :-

```
SELECT S.OrderDate, S.Shipping Date
FROM Sales S,Date D1, Date D2
WHERE D1.Date=S.OrderDate AND D2.Date=S.ShippingDate;
```

3. For the given Dimension Model, can you please generate a sql to get the total divergence between Quantity sold and Quantity Forecast for the current month for all the stores:



Ans:-

```
SELECT SUM(F.QUANTITY_FORECAST)-SUM(S.QUANTITY_SOLD) AS DIVERGENCES
        FROM DAILY_SALES S, DAILY_FORECAST F, PERIOD P
        WHERE S.PERKEY=P.PERKEY AND F.PERKEY=P.PERKEY AND
        MONTH(P.MONTH)= MONTH(CURRENT_DATE()) AND
          MONTH(P.YEAR)=YEAR(CURRENT_DATE())
```

4. For the above mentioned dimension model, please identify the conformed and Non-conformed dimensions . Additionally , identify the measures types.

Ans:- Conformed Dimensions are:

- STORE
- PERIOD
- PRODUCT

Non-Conformed Dimensions are:

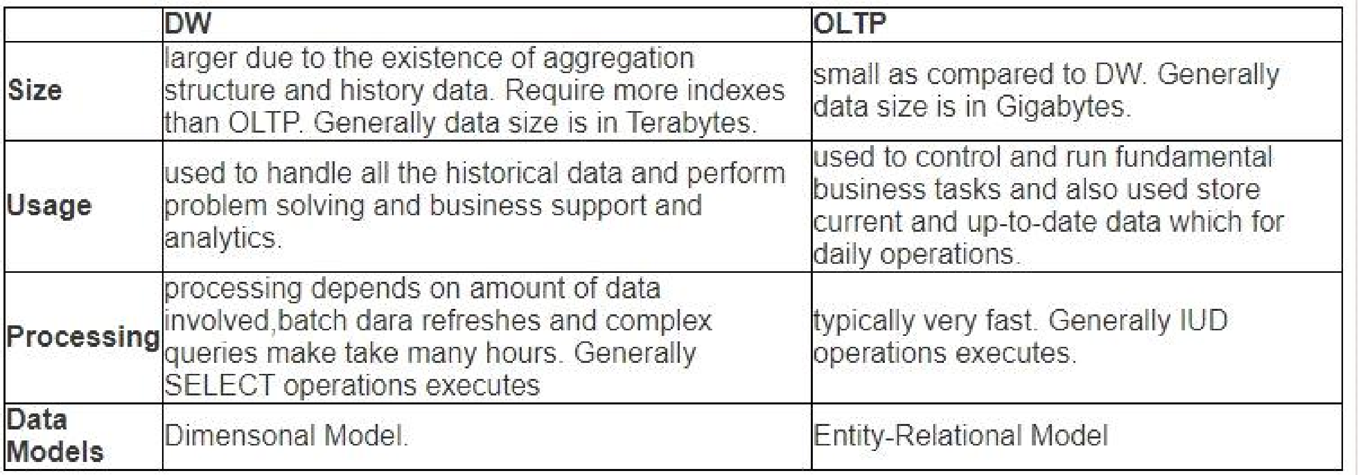
- CUSTOMER
- PROMOTION

Measure types are:

QUANTITY_SOLD :- Additive measure
EXTENDED_PRICE :- Semi-additive measure
EXTENDED_COST :- Semi-additive measure
QUANTITY_FORECAST :-Additive measure
EXTENDED_PRICE_FORECAST :-Semi-additive measure
EXTENDED_COST_FORECOST :-Semi-additive measure

5. Make list of differences between DW and OLTP based on size, usage, processing, data models.

Ans:- The differences between DW and OLTP is listed below :-



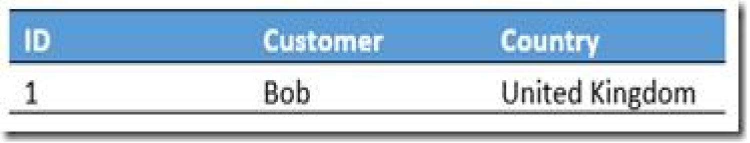
	DW	OLTP
Size	larger due to the existence of aggregation structure and history data. Require more indexes than OLTP. Generally data size is in Terabytes.	small as compared to DW. Generally data size is in Gigabytes.
Usage	used to handle all the historical data and perform problem solving and business support and analytics.	used to control and run fundamental business tasks and also used store current and up-to-date data which for daily operations.
Processing	processing depends on amount of data involved,batch dara refreshes and complex queries make take many hours. Generally SELECT operations executes	typically very fast. Generally IUD operations executes.
Data Models	Dimensonal Model.	Entity-Relational Model

6. 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.

Ans:- 1st part: If category of a product changes over a period of time then we use SCD type 2.SCD type 2 track the changes as version records with current flag and active dates and other metadata.

2nd part: If we implement SCD type 2 then add a new record encompassing the change and mark the old record as inactive.

Example: We have a very simple 'customer' dimension, with just 2 attributes – Customer Name and Country:



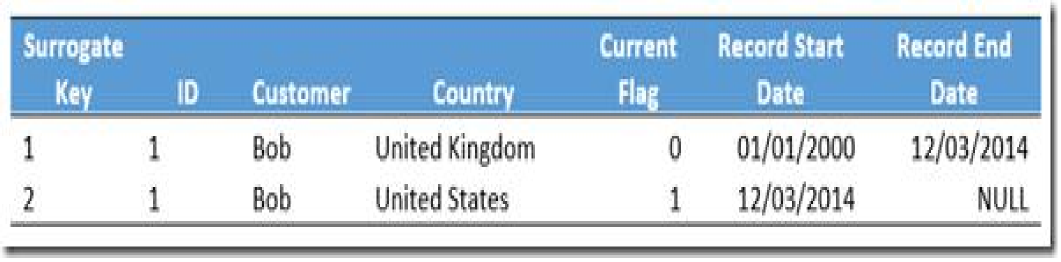
ID	Customer	Country
1	Bob	United Kingdom

However, Bob has just informed us that he has now moved to the US and we want to update our dimension record to reflect this.

In order to support type 2 changes, we need to add four columns to our table:

- Surrogate Key – the original ID will no longer be sufficient to identify the specific .
- Current Flag – A quick method of returning only the current version of each record
- Start Date – The date from which the specific historical version is active
- End Date – The date to which the specific historical version record is active

With these elements in place, our table will now look like:

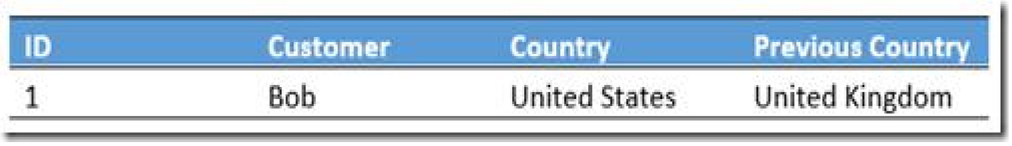


Surrogate Key	ID	Customer	Country	Current Flag	Record Start Date	Record End Date
1	1	Bob	United Kingdom	0	01/01/2000	12/03/2014
2	1	Bob	United States	1	12/03/2014	NULL

This method is very powerful – ***you maintain the history for the entire record and can easily perform change-over-time analysis.*** However, it also comes with more maintenance overhead, increased storage requirement and potential performance impacts if used on very large dimensions.
Type 2 is the most common method of tracking change in data warehouses.

2nd part: If we implement SCD type 3 then Track change to a specific attribute, add column to show the previous value, which is updated as further changes occur.

Example: Here, we add a new column called "Previous Country" to track what the last value for our attribute was.



ID	Customer	Country	Previous Country
1	Bob	United States	United Kingdom

Note how this will only provide a single historical value for Country. If the customer changes his name, we will not be able to track it without adding a new column. Likewise, if Bob moved country again, we would either need to add further "Previous Previous Country" columns or lose the fact that he once lived in the United Kingdom.

6. b) What is surrogate key? Why it is required?

Ans :- 1st part: It is sequentially generated unique number attached with each and every record in a Dimension table in any Data Warehouse. It join between the fact and dimension tables and is necessary to handle changes in dimension table attributes.

- 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.

2nd part: Basically, surrogate key is an artificial 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. However, it is not recommended because of following reasons:

- **Natural keys (NK)** or **Business keys** are generally alphanumeric values that is not suitable for index as traversing become slower. For example, prod123, prod231 etc
- Business keys are often reused after sometime. It will cause the problem as in data warehouse we maintain historic data as well as current data.

For example, product codes can be revised and reused after few years. It will become difficult to differentiate current products and historic products. To avoid such a situation, surrogate keys are used.

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

STORE TABLE:

Store_id	Store_name
1	S1
2	S2
3	S3
4	S4

PRODUCT TABLE:

Product_id	Product_name
1	P1
2	P2
3	P3

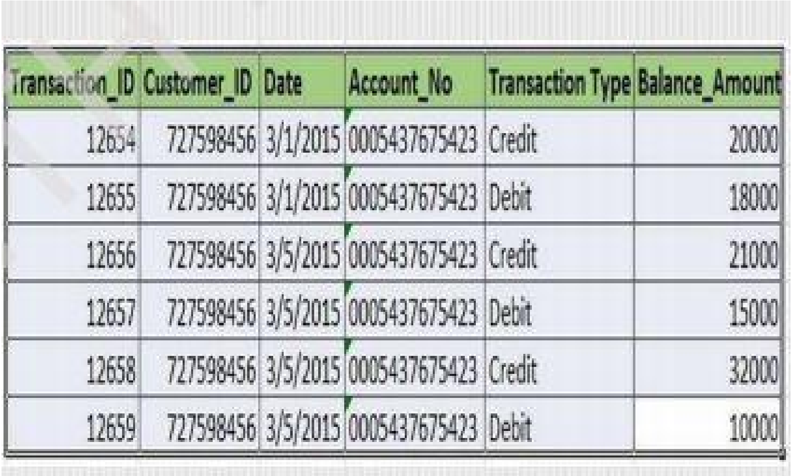
STORE-PRODUCT TABLE:

Store_id	Store_name	Product_id	Product_name
1	S1	1	P1
2	S2	1	P1
4	S4	2	P2
4	S4	3	P3

6. d) What is a semi-additive measure? Give an example.

Ans:- These are those specific class of fact measures which can be aggregated across all dimension and their hierarchy except the time dimension.

Example: Daily balances fact can be summed up through the customers dimension but not through the time dimension.



Transaction_ID	Customer_ID	Date	Account_No	Transaction Type	Balance_Amount
12654	727598456	3/1/2015	0005437675423	Credit	20000
12655	727598456	3/1/2015	0005437675423	Debit	18000
12656	727598456	3/5/2015	0005437675423	Credit	21000
12657	727598456	3/5/2015	0005437675423	Debit	15000
12658	727598456	3/5/2015	0005437675423	Credit	32000
12659	727598456	3/5/2015	0005437675423	Debit	10000

Example of Semi-additive measures