Experiment 1

Aim :

Build Data Warehouse/Data Mart for a given problem statement (One case study) Write Detailed Problem statement and design dimensional modeling (creation of star and snowflake schema)
i) Identifying the source tables and populating sample data
ii) Design dimensional data model i.e. Star schema, Snowflake schema and Fact Constellation schema (if applicable)
iii) Implementation of all dimension table and fact table based on experiment 1 case study

Reference: https://www.codeproject.com/Articles/652108/Create-First-Data-WareHouse Introduction

Data warehouse is integrated, non volatile, subject oriented and time variant storage of data. Whenever your data is distributed across various databases, application or at various places stored in different formats and you want to convert this data into useful information by integrating and creating unique storage at a single location for these distributed data at that time, you need to start thinking to use data warehouse.

In another case, if your daily transactional data entry is very huge in your database, maybe millions or billions of records, then you need to archive these data to another Archive database which holds your historical data to remove load from live database and if you are creating your two dimensional report on this archive database then your report generation is very slow on that data it may take couple of minutes to couple of hours or it can give you timeout error. On this two dimensional data, even you cannot do any type of trend analysis on your data, you cannot divide your data into various time buckets of the day or cannot do study of data between various combination of year, quarter, month, week, day, weekday-weekend. In this scenario to take perfect decision on the basis of your historical data, you have to think to go for designing of data warehouse as per your requirement, so you can study data using multiple dimensions and can do better analysis to take accurate decision.

Designing of data warehouse helps to convert data into useful information, it provides multiple dimensions to study your data, so higher management can take Quick and accurate decision on the basis of statistics calculated using this data, this data can also be utilized for data mining, forecasting, predictive analysis, quicker reports, and Informative Dash board creation, which also helps management in day to day life to resolve various complex queries as per their requirement.

Now a day's users need to have self service BI (Business Intelligence) capabilities so they can create reports on their own (Ad-Hoc reports) and can do analysis of data without much technical knowledge. Data warehousing is a business analyst's dream - all the information about the organization's activities gathered in one place, open to a single set of analytical tools. But how do you make the dream a reality? First, you have to plan your data warehouse system. So modeling of data warehouse is the first step in this direction.

Scenario

X-Mart is having different malls in our city, where daily sales take place for various products. Higher management is facing an issue while decision making due to non availability of integrated data they can't do study on their data as per their requirement. So they asked us to design a system which can help them quickly in decision making and provide Return on Investment (ROI).

Let us start designing of data warehouse; we need to follow a few steps before we start our data warehouse design.

Developing a Data Warehouse

The phases of a data warehouse project listed below are similar to those of most database projects, starting with identifying requirements and ending with executing the T-SQL Script to create data warehouse:

1. **Identify and collect requirements**
2. **Design the dimensional model**
3. **Execute T-SQL queries to create and populate your dimension and fact tables Identify**

and Collect Requirements

We need to interview the key decision makers to know, what factors define the success in the business? How does management want to analyze their data? What are the most important business questions, which need to be satisfied by this new system?

We also need to work with persons in different departments to know the data and their common relations if any, document their entire requirement which need to be satisfied by this system.

Let us first identify the requirement from management about their requirements. 1. Need to see daily, weekly, monthly, quarterly profit, sales of each store. 2. Comparison of sales and profit on various time periods.

3. **Comparison of sales in various time bands of the day.**
4. **Need to know which product has more demand on which location?**
5. **Need to study trend of sales by time period of the day over the week, month, and year? 6. On what day sales is higher?**
7. **On every Sunday of this month, what is sales and what is profit?**
8. **What is trend of sales on weekday and weekend?**
9. **Need to compare weekly, monthly and yearly sales to know growth and KPI?**

Design the Dimensional Model

We need to design Dimensional Model to suit requirements of users which must address business needs and contains information which can be easily accessible. Design of model should be easily extensible according to future needs. This model design must supports OLAP cubes to provide "instantaneous" query results for analysts.

Let us take a quick look at a few new terms and then we will identify/derive it for our requirement. **Dimension**

The dimension is a master table composed of individual, non-overlapping data elements. The primary functions of dimensions are to provide filtering, grouping and labeling on your data. Dimension tables contain textual descriptions about the subjects of the business.

Let me give you a glimpse on different types of dimensions available like confirmed dimension, Role Playing dimension, Degenerated dimension, Junk Dimension.

Slowly changing dimension (SCD) specifies the way using which you are storing values of your dimension which is changing over a time and preserver the history. Different methods / types are available to store history of this change E.g. SCD1, SCD2, and SCD3 you can use as per your requirement.

Let us identify dimensions related to the above case study.

Product, Customer, Store, date, Time, Sales person

Measure

A measure represents a column that contains quantifiable data, usually numeric, that can be aggregated. A measure is generally mapped to a column in a fact table. For your information, various types of measures are there. **E.g.** Additive, semi additive and Non additive.

Let us define what will be the Measures in our case.

Actual Cost, Total Sales, Quantity, Fact table record count

Fact Table

Data in fact table are called measures (or dependent attributes), Fact table provides statistics for sales broken down by customer, salesperson, product, period and store dimensions. Fact table usually contains historical transactional entries of your live system, it is mainly made up of Foreign key column which references to various dimension and numeric measure values on which aggregation will be performed. Fact tables are of different types, **E.g.** Transactional, Cumulative and Snapshot.

Let us identify what attributes should be there in our Fact Sales Table.

1. **Foreign Key Column**

Sales Date key, Sales Time key, Invoice Number, Sales Person ID, Store ID, Customer ID 2. **Measures**

Actual Cost, Total Sales, Quantity, Fact table record count

Design the Relational Database

We have done some basic workout to identify dimensions and measures, now we have to use appropriate schema to relate this dimension and Fact tables.

Few popular schemas used to develop dimensional model are as follows:

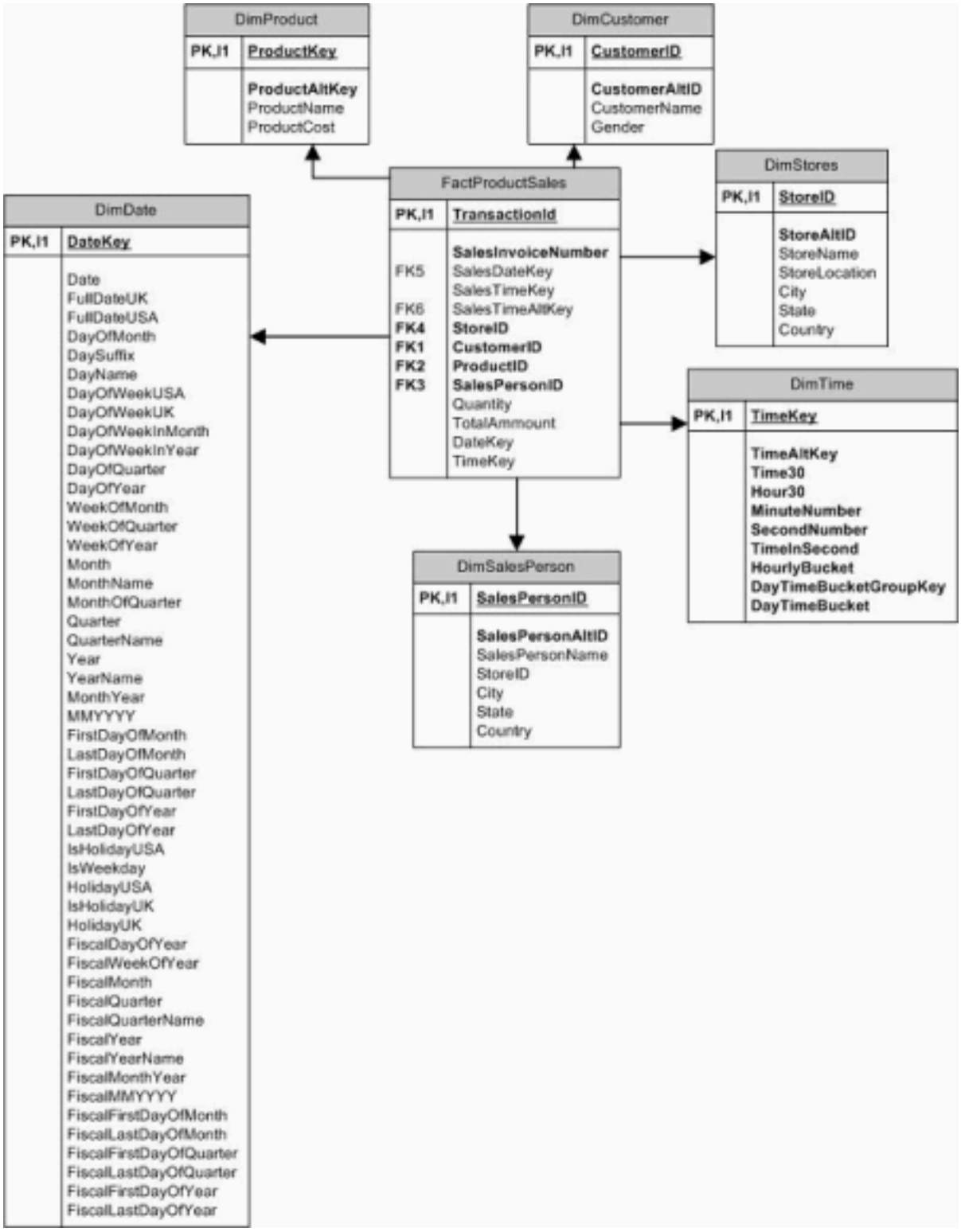
E.g. Star Schema, Snow Flake Schema, Star Flake Schema, Distributed Star Schema, etc. In a different article, we

will discuss all these schemas, dimension types, measure types, etc., in detail.

Personally, I will first try to use Star schema due to hierarchical attribute model it provides for analysis and speedy performance in querying the data.

Star schema the diagram resembles a star, with points radiating from a center. The center of the star consists of fact table and the points of the star are the dimension tables.

Let us create Our First Star Schema, please refer to the below figure:



Using the Code

Let us execute our T-SQL Script step by step to create table and populate them with appropriate test values.

Follow the given steps to run the query in SSMS (SQL Server Management Studio). 1. Open

SQL Server Management Studio

2. Connect Database Engine

3. Open New Query editor

4. Copy paste Scripts given below in various steps in new query editor window one by one 5. To run the given SQL

Script, press F5

Step 1

Create database for your Data Warehouse in SQL Server:

Hide Copy Code

```
Create database Sales_DW
Go


Use Sales_DW
Go
```

Step 2

Create **Customer dimension** table in Data Warehouse which will hold customer personal details. Hide Copy

Code

```
Create table DimCustomer
(
CustomerID int primary key,
CustomerAltID varchar(10) not null,
CustomerName varchar(50),
Gender varchar(20)
)
go
```

Fill the **Customer dimension** with sample Values

Hide Copy Code

```
Insert into DimCustomer(CustomerAltID,CustomerName,Gender)values ('IMI-001','Henry
Ford','M'),
```

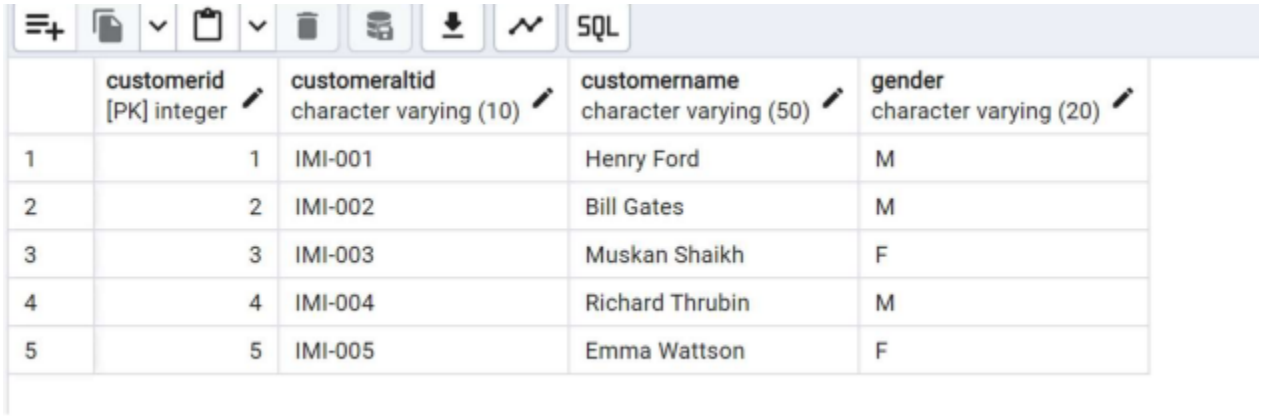
('IMI-002','Bill Gates','M'),

('IMI-003','Muskan Shaikh','F'),

('IMI-004','Richard Thrubin','M'),

('IMI-005','Emma Wattson','F');

Go



	customerid [PK] integer	customeraltid character varying (10)	customername character varying (50)	gender character varying (20)
1	1	IMI-001	Henry Ford	M
2	2	IMI-002	Bill Gates	M
3	3	IMI-003	Muskan Shaikh	F
4	4	IMI-004	Richard Thrubin	M
5	5	IMI-005	Emma Wattson	F

Step 3

Create basic level of **Product Dimension** table without considering any Category or Subcategory Hide Copy Code

Create table DimProduct

(

ProductKey int primary key identity,

ProductAltKey varchar(10)not null,

ProductName varchar(100),

ProductActualCost money,

ProductSalesCost money

)

Go

Fill the **Product dimension** with sample Values

Hide Copy Code

Insert into DimProduct(ProductAltKey,ProductName, ProductActualCost, ProductSalesCost)values

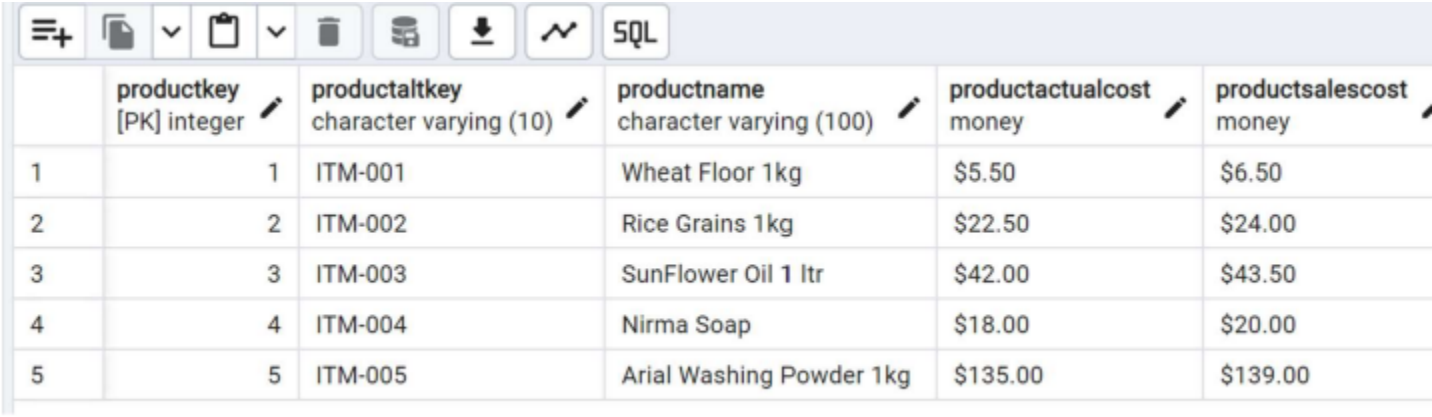
('ITM-001','Wheat Floor 1kg',5.50,6.50),

('ITM-002','Rice Grains 1kg',22.50,24),

('ITM-003','SunFlower Oil 1 ltr',42,43.5),
('ITM-004','Nirma Soap',18,20),

('ITM-005','Arial Washing Powder 1kg',135,139);

GO



	productkey [PK] integer	productaltkey character varying (10)	productname character varying (100)	productactualcost money	productsalescost money
1	1	ITM-001	Wheat Floor 1kg	$5.50	$6.50
2	2	ITM-002	Rice Grains 1kg	$22.50	$24.00
3	3	ITM-003	SunFlower Oil 1 ltr	$42.00	$43.50
4	4	ITM-004	Nirma Soap	$18.00	$20.00
5	5	ITM-005	Arial Washing Powder 1kg	$135.00	$139.00

Step 4

Create **Store Dimension** table which will hold details related stores available across various places. Hide Copy

Code

Create table DimStores

(

StoreID int primary key,

StoreAltID varchar(10)not null,

StoreName varchar(100),

StoreLocation varchar(100),

City varchar(100),

State varchar(100),

Country varchar(100)

)

Go

Fill the **Store Dimension** with sample Values

Hide Copy Code

Insert into DimStores(StoreAltID,StoreName,StoreLocation,City,State,Country)values

('LOC-A1','X-Mart','S.P. RingRoad','Ahmedabad','Guj','India'),

('LOC-A2','X-Mart','Maninagar','Ahmedabad','Guj','India'),

('LOC-A3','X-Mart','Sivranjani','Ahmedabad','Guj','India');

Go

Data Output | Messages | Notifications

SQL

	storeid [PK] integer	storealtid character varying (10)	storename character varying (100)	storelocation character varying (100)	city character varying (100)	state character varying (100)	country character var
1	100	LOC-A1	X-Mart	S.P. RingRoad	Ahmedabad	Guj	India
2	101	LOC-A2	X-Mart	Maninagar	Ahmedabad	Guj	India
3	102	LOC-A3	X-Mart	Sivranjani	Ahmedabad	Guj	India

Step 5

Create **Dimension Sales Person** table which will hold details related stores available across various places.

Hide Copy Code

```
Create table DimSalesPerson
(

SalesPersonID int primary key identity,

SalesPersonAltID varchar(10)not null,

SalesPersonName varchar(100),

StoreID int,

City varchar(100),

State varchar(100),

Country varchar(100)

)

Go
```

Fill the Dimension **Sales Person** with sample values:

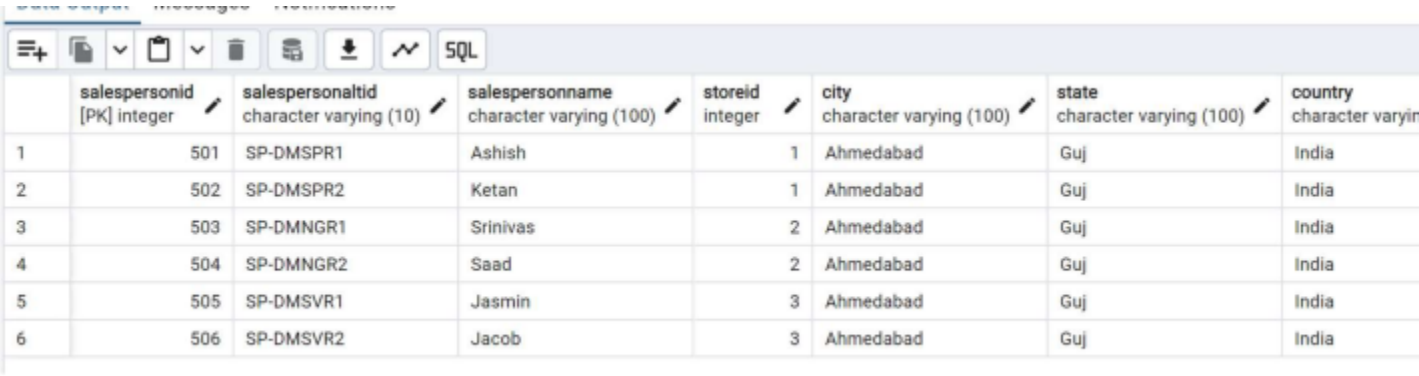
Hide Copy Code

```
Insert into DimSalesPerson(SalesPersonAltID,SalesPersonName,StoreID,City,State,Country )values

('SP-DMSPR1','Ashish',1,'Ahmedabad','Guj','India'),

('SP-DMSPR2','Ketan',1,'Ahmedabad','Guj','India'),

('SP-DMNGR1','Srinivas',2,'Ahmedabad','Guj','India'),

('SP-DMNGR2','Saad',2,'Ahmedabad','Guj','India'),
```

('SP-DMSVR1','Jasmin',3,'Ahmedabad','Guj','India'),

('SP-DMSVR2','Jacob',3,'Ahmedabad','Guj','India');

Go



	salespersonid [PK] integer	salespersonaltid character varying (10)	salespersonname character varying (100)	storeid integer	city character varying (100)	state character varying (100)	country character varyin
1	501	SP-DMSPR1	Ashish	1	Ahmedabad	Guj	India
2	502	SP-DMSPR2	Ketan	1	Ahmedabad	Guj	India
3	503	SP-DMNGR1	Srinivas	2	Ahmedabad	Guj	India
4	504	SP-DMNGR2	Saad	2	Ahmedabad	Guj	India
5	505	SP-DMSVR1	Jasmin	3	Ahmedabad	Guj	India
6	506	SP-DMSVR2	Jacob	3	Ahmedabad	Guj	India

Step 6

Create **Date Dimension** table which will create and populate date data divided on various levels. (For this, you have to refer my article on CodeProject Create and Populate Date Dimension. Download the script and run it in this database for creating and filling of date dimension with values.) **Step 7**

Create **Time Dimension** table which will create and populate Time data for the entire day with various time buckets.

For this, you have to refer to my article on Code Project, Create & Populate Time Dimension with 24 Hour+ Values
Download the script and run it in this database for creating and filling of time dimension with values. **Step 8**

Create **Fact table** to hold all your transactional entries of previous day sales with appropriate foreign key columns which refer to primary key column of your dimensions; you have to take care while populating your fact table to refer to primary key values of appropriate dimensions.

e.g.

Customer Henry Ford has purchase purchased 2 items (sunflower oil 1 kg, and 2 Nirma soap) in a single invoice on date 1-jan-2013 from D-mart at Sivranjani and sales person was Jacob , billing time recorded is 13:00, so let us define how will we refer to the primary key values from each dimension.

Before filling fact table, you have to identify and do look up for primary key column values in dimensions as per given example and fill in foreign key columns of fact table with appropriate key values.

Attribute Name	**Dimension Table**	**Primary Key Column/Value**
Date (1-jan-2013), Sales Date Key (20130101)	Dim Date	Date Key: 20130101

Time (13:00:00) Sales Time Alt Key (130000)	Dim Time	Time Key: 46800
Composite key (Sales Person Alt ID+ Name) for ('SP DMSVR1'+'Jacob')	Dim Sales Person	Sales Person ID: 6
Product Alt Key of (Sunflower Oil 1kg)'ITM-003'	Dim Product	Product ID: 3
Product Alt Key (Nirma Soap) 'ITM-004'	Dim Product	Product ID: 4
Store Alt ID of (Sivranjani store) 'LOC-A3'	Dim Store	Store ID: 3
Customer Alt ID of (Henry Ford) is 'IMI-001'	Dim Customer	Customer ID: 1

```
Create Table FactProductSales

(

TransactionId bigint primary key identity,

SalesInvoiceNumber int not null,

SalesDateKey int,

SalesTimeKey int,
```

SalesTimeAltKey int,

StoreID int not null,
CustomerID int not null,

ProductID int not null,

SalesPersonID int not null,

Quantity float,

SalesTotalCost money,

ProductActualCost money,

Deviation float

)

Go

Add Relation between Fact table and dimension tables:

-- Add relation between fact table foreign keys to Primary keys of Dimensions AlTER TABLE

FactProductSales ADD CONSTRAINT _

FK_StoreID FOREIGN KEY (StoreID)REFERENCES DimStores(StoreID);

AlTER TABLE FactProductSales ADD CONSTRAINT _

FK_CustomerID FOREIGN KEY (CustomerID)REFERENCES Dimcustomer(CustomerID); AlTER

TABLE FactProductSales ADD CONSTRAINT _

FK_ProductKey FOREIGN KEY (ProductID)REFERENCES Dimproduct(ProductKey); AlTER

TABLE FactProductSales ADD CONSTRAINT _

FK_SalesPersonID FOREIGN KEY (SalesPersonID)REFERENCES Dimsalesperson(SalesPersonID); Go

AlTER TABLE FactProductSales ADD CONSTRAINT _

FK_SalesDateKey FOREIGN KEY (SalesDateKey)REFERENCES DimDate(DateKey); Go

AlTER TABLE FactProductSales ADD CONSTRAINT _

FK_SalesTimeKey FOREIGN KEY (SalesTimeKey)REFERENCES DimDate(TimeKey); Go

Populate your **Fact table** with historical transaction values of sales for previous day, with proper values of dimension key values.

Insert into FactProductSales(SalesInvoiceNumber,SalesDateKey,_
SalesTimeKey,SalesTimeAltKey,StoreID,CustomerID,ProductID ,_

SalesPersonID,Quantity,ProductActualCost,SalesTotalCost,Deviation)values --1-jan-2013

--SalesInvoiceNumber,SalesDateKey,SalesTimeKey,SalesTimeAltKey,_

StoreID,CustomerID,ProductID ,SalesPersonID,Quantity,_

ProductActualCost,SalesTotalCost,Deviation)

(1,20130101,44347,121907,1,1,1,1,2,11,13,2),

(1,20130101,44347,121907,1,1,2,1,1,22.50,24,1.5),

(1,20130101,44347,121907,1,1,3,1,1,42,43.5,1.5),

(2,20130101,44519,122159,1,2,3,1,1,42,43.5,1.5),

(2,20130101,44519,122159,1,2,4,1,3,54,60,6),

(3,20130101,52415,143335,1,3,2,2,2,11,13,2),

(3,20130101,52415,143335,1,3,3,2,1,42,43.5,1.5),

(3,20130101,52415,143335,1,3,4,2,3,54,60,6),

(3,20130101,52415,143335,1,3,5,2,1,135,139,4),

--2-jan-2013

--SalesInvoiceNumber,SalesDateKey,SalesTimeKey,SalesTimeAltKey,_ StoreID,CustomerID,ProductID

,SalesPersonID,Quantity,ProductActualCost,SalesTotalCost,Deviation)

(4,20130102,44347,121907,1,1,1,1,2,11,13,2),

(4,20130102,44347,121907,1,1,2,1,1,22.50,24,1.5),

(5,20130102,44519,122159,1,2,3,1,1,42,43.5,1.5),

(5,20130102,44519,122159,1,2,4,1,3,54,60,6),

(6,20130102,52415,143335,1,3,2,2,2,11,13,2),

(6,20130102,52415,143335,1,3,5,2,1,135,139,4),
(7,20130102,44347,121907,2,1,4,3,3,54,60,6),

(7,20130102,44347,121907,2,1,5,3,1,135,139,4),

--3-jan-2013

```
--SalesInvoiceNumber,SalesDateKey,SalesTimeKey,SalesTimeAltKey,StoreID,_ CustomerID,ProductID

,SalesPersonID,Quantity,ProductActualCost,SalesTotalCost,Deviation)

(8,20130103,59326,162846,1,1,3,1,2,84,87,3),

(8,20130103,59326,162846,1,1,4,1,3,54,60,3),



(9,20130103,59349,162909,1,2,1,1,1,5.5,6.5,1),

(9,20130103,59349,162909,1,2,2,1,1,22.50,24,1.5),


(10,20130103,67390,184310,1,3,1,2,2,11,13,2),

(10,20130103,67390,184310,1,3,4,2,3,54,60,6),


(11,20130103,74877,204757,2,1,2,3,1,5.5,6.5,1),

(11,20130103,74877,204757,2,1,3,3,1,42,43.5,1.5)

Go
```

After executing the above T-SQL script, your sample data warehouse for sales will be ready, now you can create OLAP Cube on the basis of this data warehouse.

In real life scenario, we need to design SSIS ETL package to populate dimension and fact table of data warehouse with appropriate values, we can schedule this package for daily execution and daily processing and populating of previous day data in dimension and fact tables, so our data will get ready for analysis and reporting.

References:

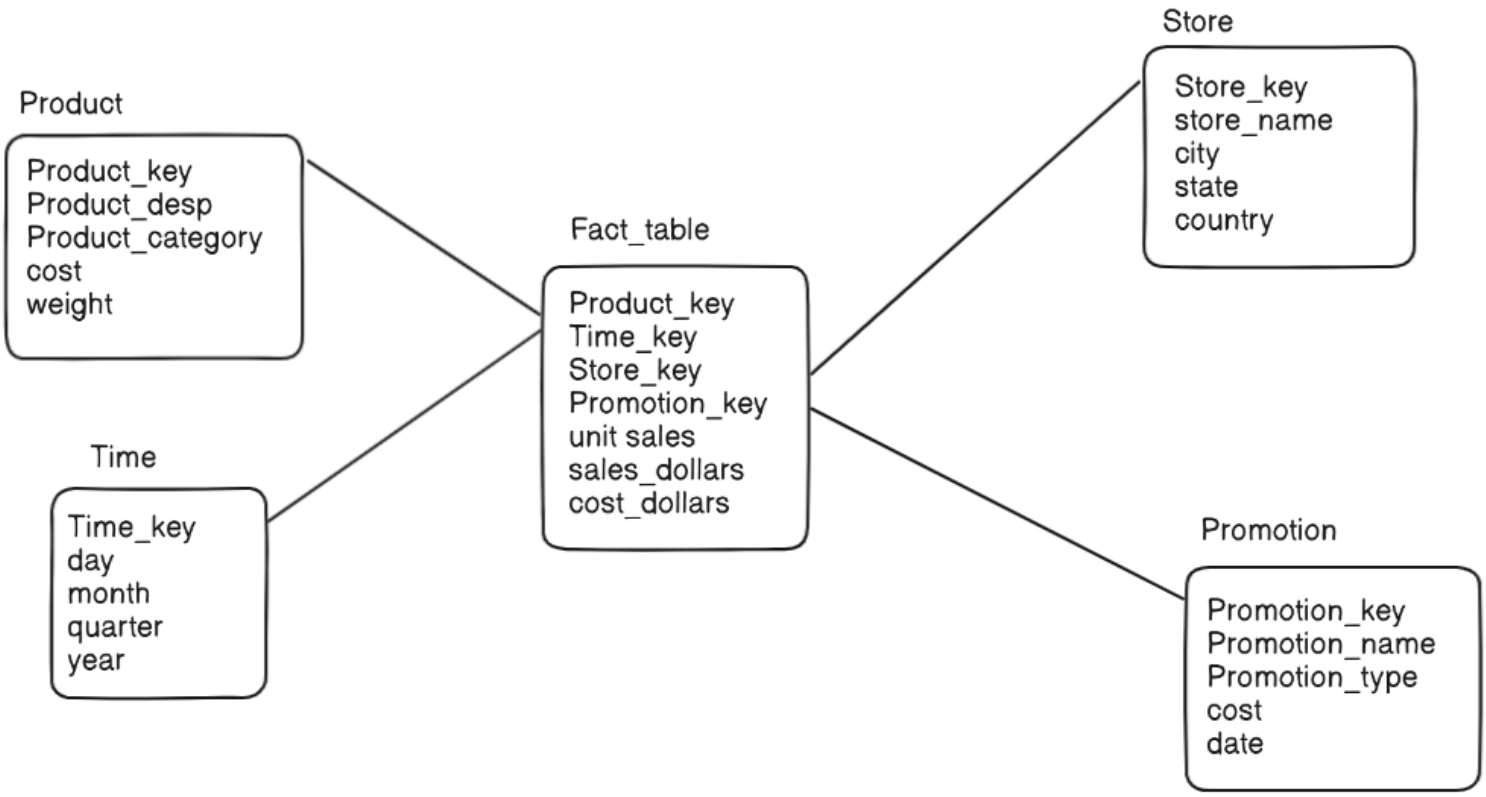
■ What are Slowly Changing Dimensions? | Datawarehouse4u.info ■ Data Warehouse Design Techniques - Slowly Changing Dimensions – (nuwavesolutions.com)

■ Slowly Changing Dimensions (SCDs) In The Age of The Cloud Data Warehouse (holistics.io)

■ It's All About ORACLE: Rapidly Changing Dimension (RCD) in Data Warehouse (its-all-about oracle.blogspot.com)

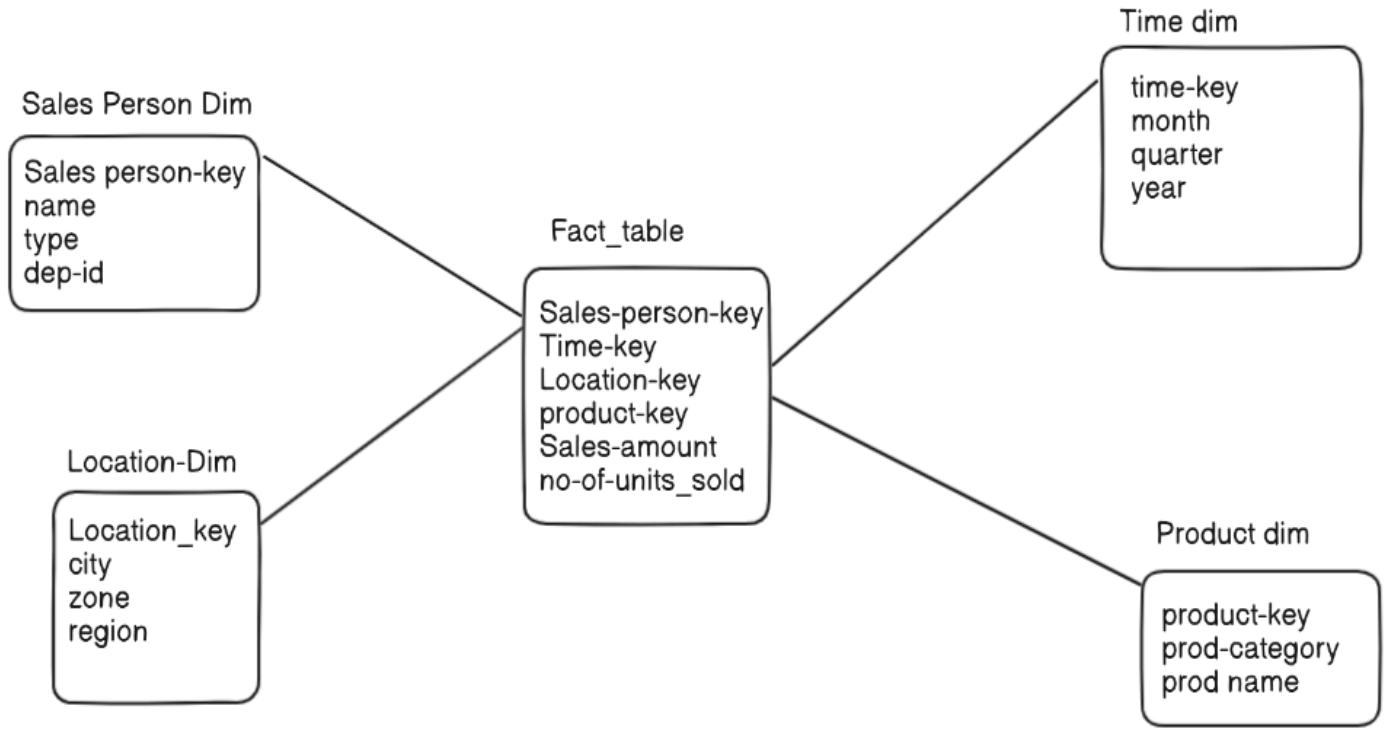
■ Advanced Dimensional Data Warehouse Design Techniques – (nuwavesolutions.com) ■ Types of Keys in Data Warehouse Schema – GeeksforGeeks ■ What is a FACTLESS FACT TABLE?Where we use Factless Fact (dwhlaureate.blogspot.com)

Q1 a) Consider following dimensions for a Hypermarket chain: Product, Store, Time and Promotion. With respect to this business scenario, answer the following questions. Clearly state any reasonable assumptions you make. Design a star schema. Whether the star schema can be converted to snowflake schema? Justify your answer and draw snowflake schema for the data warehouse (clearly mention the Fact table(s), Dimension table(s), their attributes and measures).



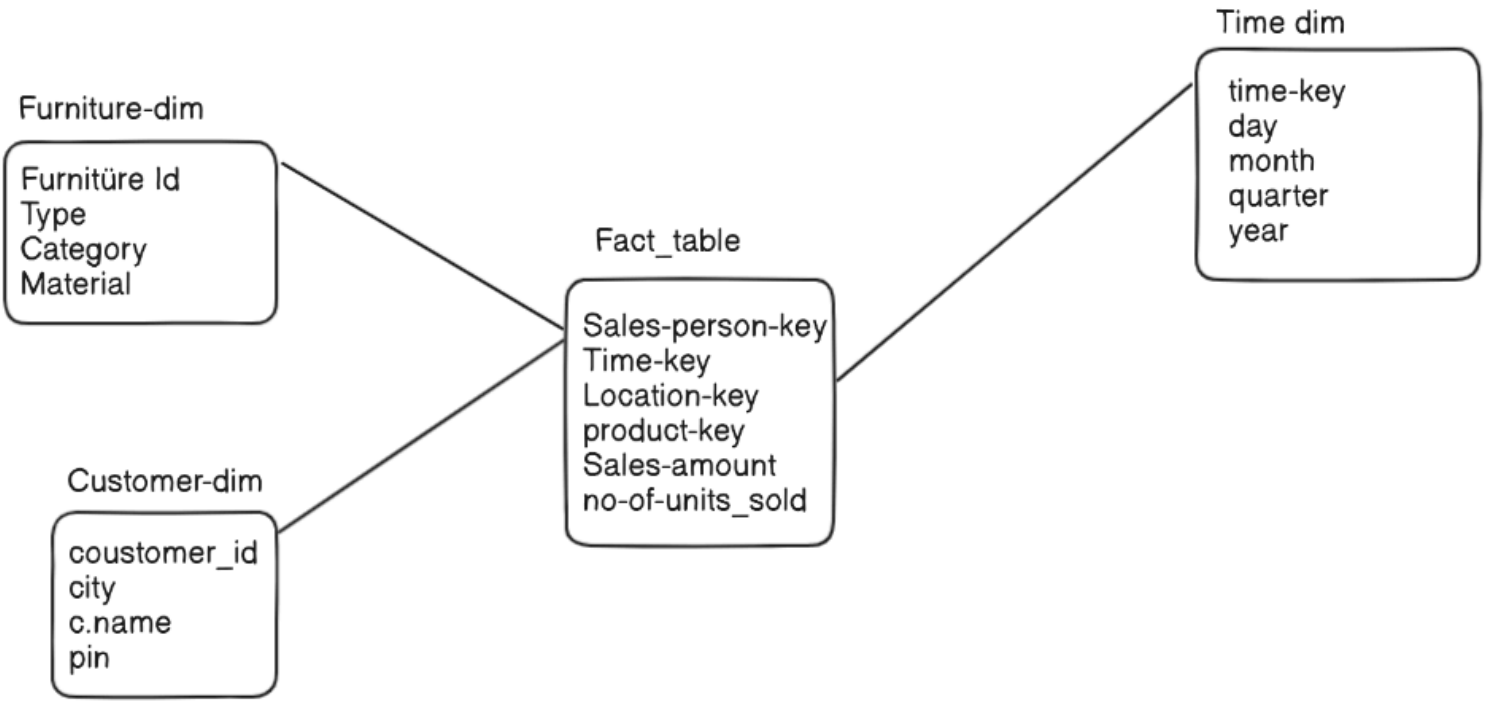
Q2

A) A manufacturing company has a huge sales network. To control the sales, it is divided into regions. Each region has multiple zones. Each zone has different cities. Each sales person is allocated different cities. The objective is to track sales figure at different granularity levels of region and to count no. of products sold. Design a star schema by considering granularity levels for region, sales person and time. Convert the star schema to snowflake schema.



Q.3

What is dimensional modelling? Design the data warehouse for wholesale furniture Company. The data warehouse has to allow analysing the company's situation at least with respect to the Furniture, Customer and Time. More ever, the company needs to analyse: The furniture with respect to its type, category and material. The customers with respect to their spatial location, by considering at least cities, regions and states. The company is interested in learning the quantity, income and discount of its sales.



Q.4

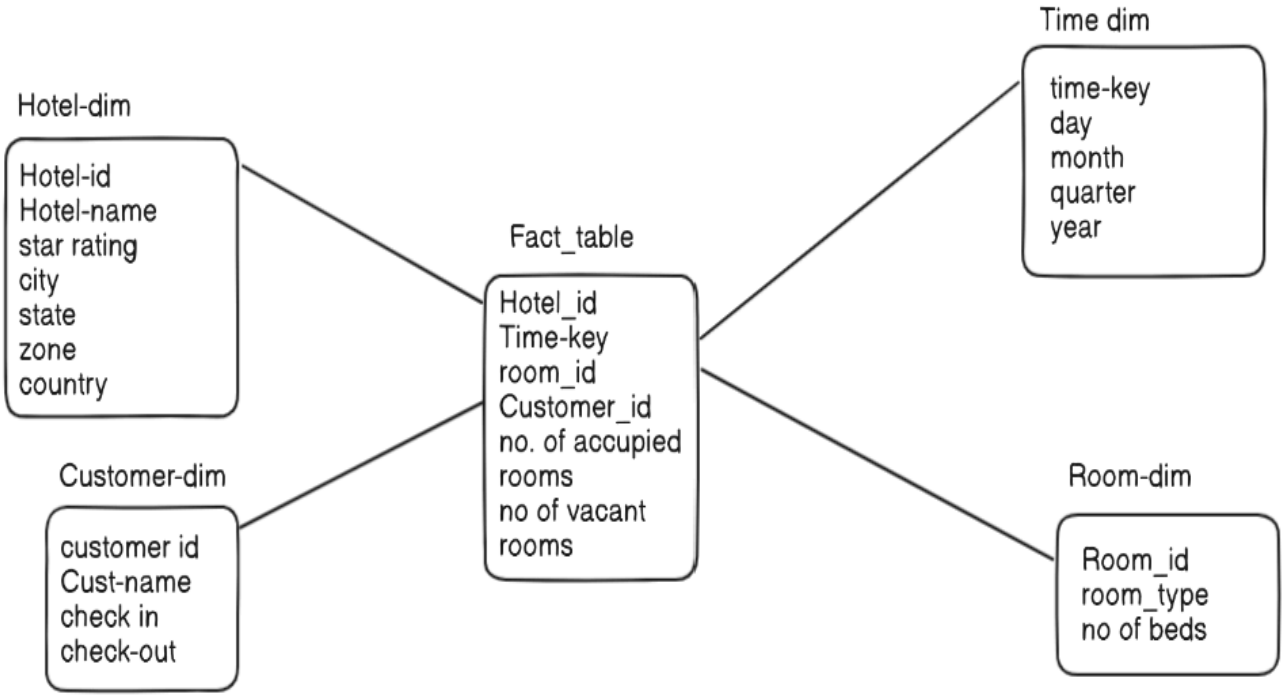
i. Design star & snowflake schema for "Hotel Occupancy" considering [dimensions like Time, Hotel, Room, etc.

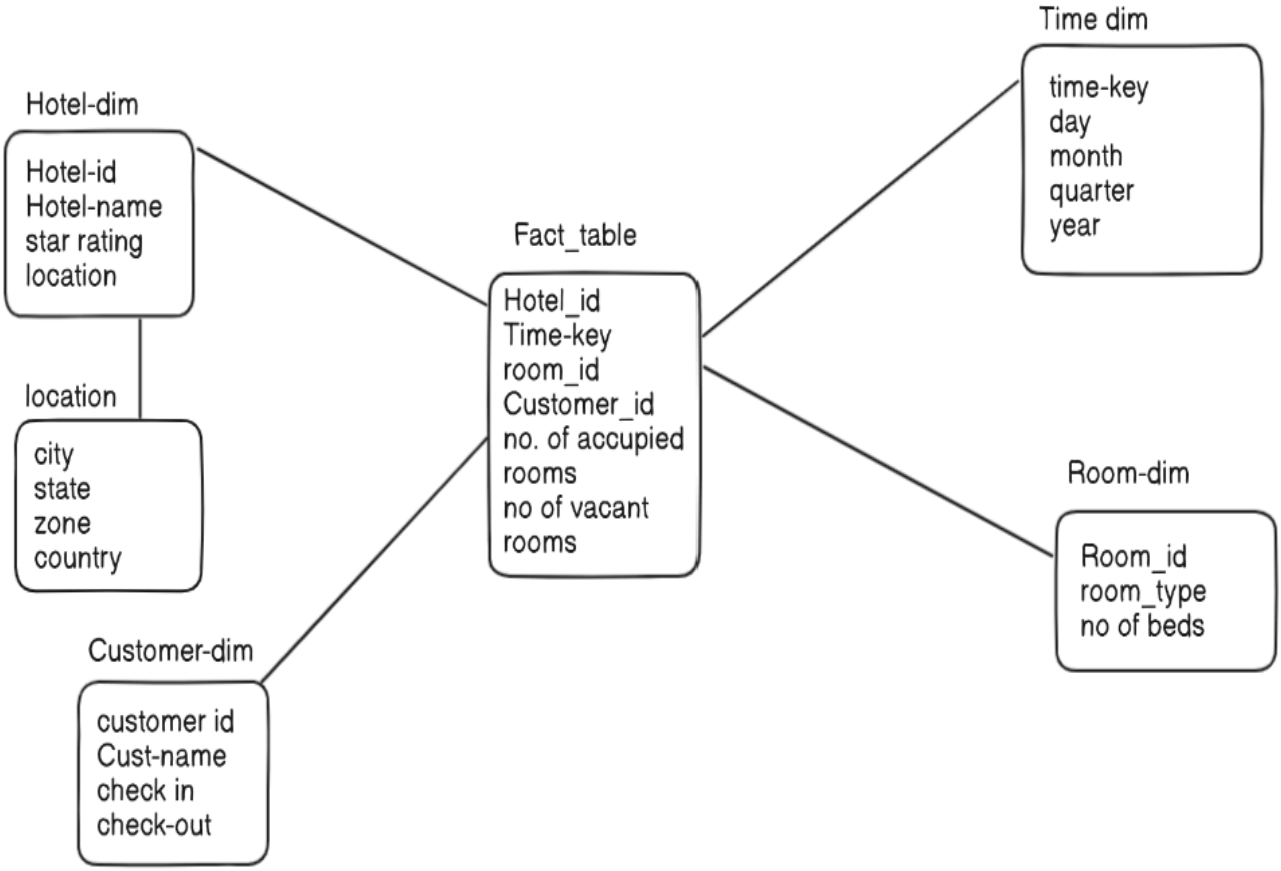
ii. Calculate the maximum number of base fact table records for the values given below:

Time period: 5 years

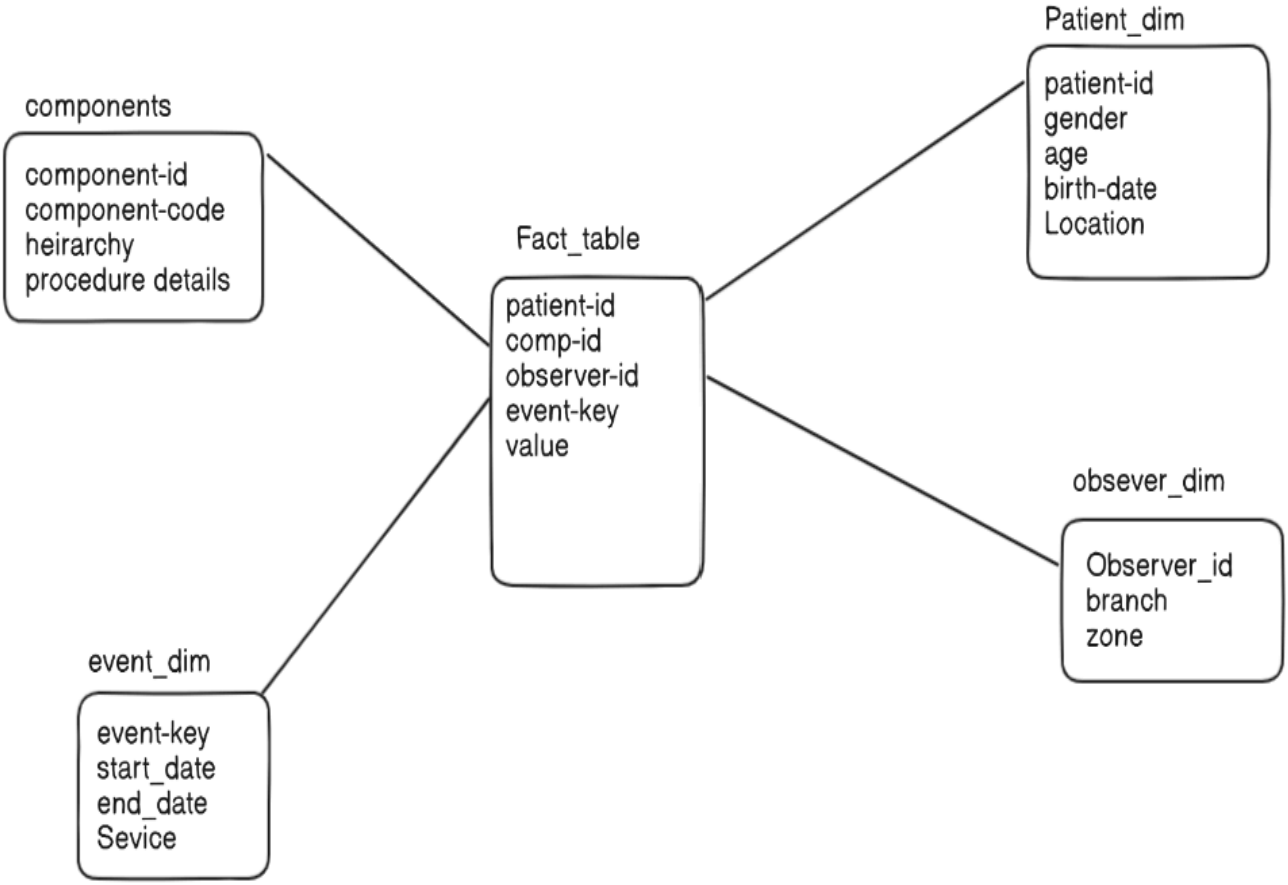
Hotels: 150

Rooms: 750 rooms in each Hotel (about 400 occupied in each hotel daily).

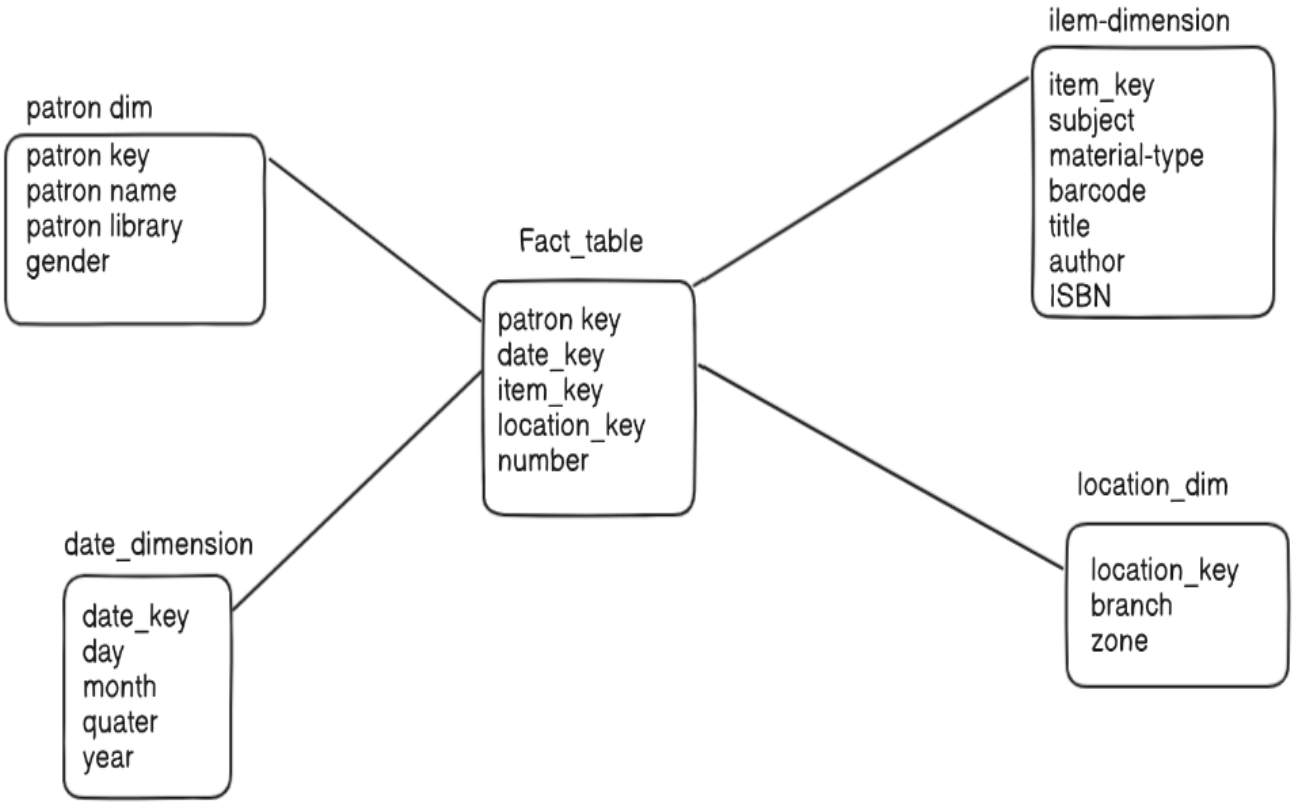




Q.5 Hospital DWH



Q.6 library management



Post Lab Questions

1. Justify whether slowly changing dimension modeling is required for the problem selected?

SCD modeling is required when you need to track changes in data over time and maintain historical accuracy,so it will be required for the problem selected

2. Justify why Fact-less fact table modeling is not required for the problem selected? Give the examples of fact less fact tables with its type.

Fact-less fact table modeling is not required for the problem selected because our problem focuses on analyzing measurable data rather than just tracking events or states
ex:-
Tracking student participation in hackathons.
Tracking student attendance in class.

3. Justify your approach to deal with large dimension tables for the problem selected.

Partitioning
Partitioning a large dimension table can improve performance and manageability by dividing the table into smaller, more manageable pieces

Indexing
Creating indexes on key columns can significantly enhance performance by allowing faster data retrieval.

Compression
Compressing data in large dimension tables can save storage space and improve performance

4. Justify the use of junk dimension and surrogate key for the problem selected.

A junk dimension is a collection of random attributes such as flags and other miscellaneous data that do not fit into the main dimensions.

A surrogate key is an artificial key used on a tab;e to uniquely identify rows