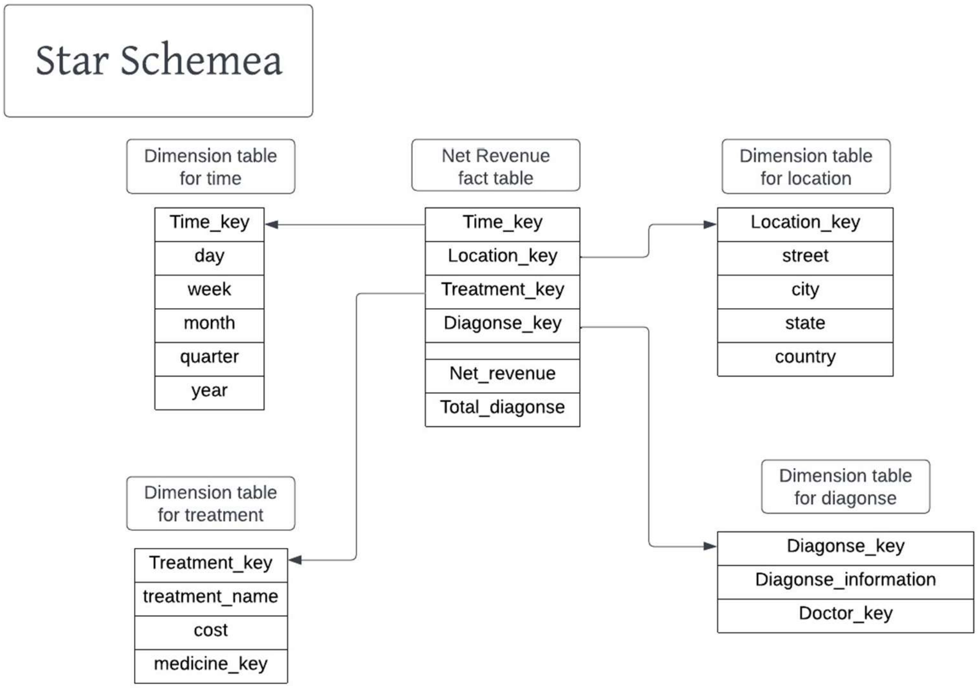
Topic: Dental Clinical Data System

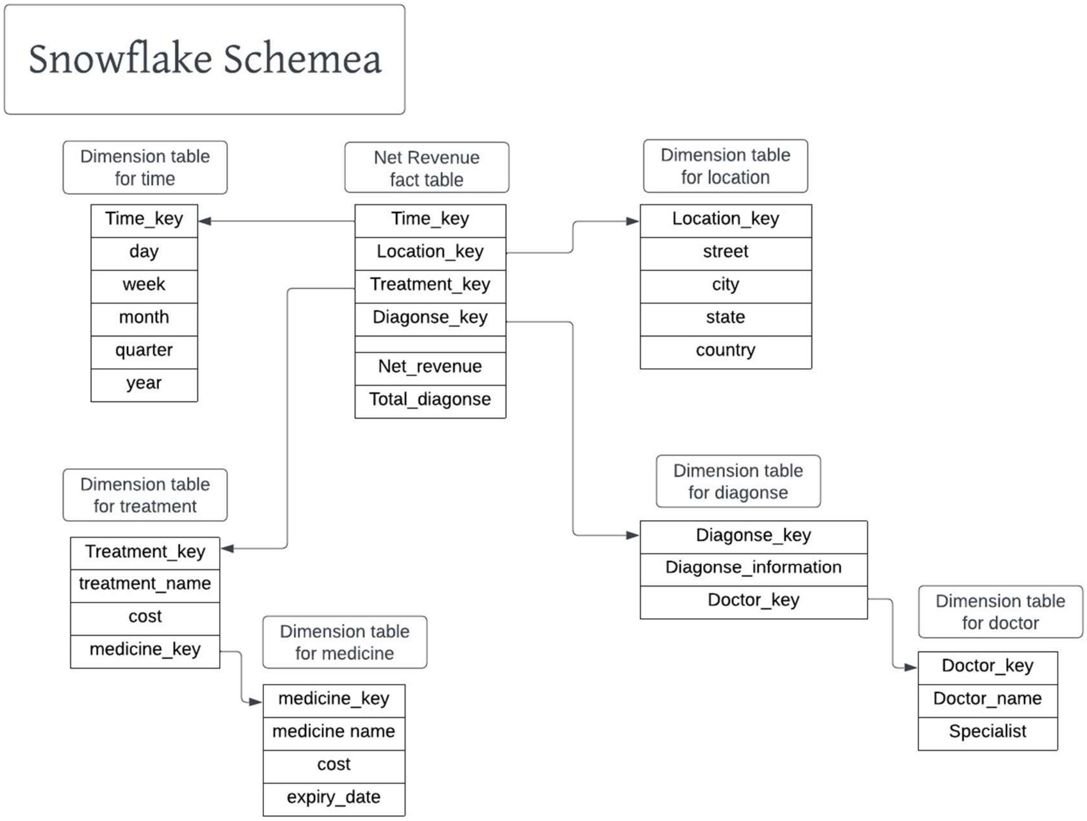
Goal: Our goal is to analyze the clinical data through revenue generation of each branch through Treatment analysis of each patient in every branch

We are creating dental clinical system to keep a proper track of patients under services. We also need them for the follow up sessions of every patient. This system will increase the patients under our services at different location thus increasing the profit. And we can also get to know the profit of different branches. Thus, to achieve the goal of better decision-making using dimensions like Time, Location, Treatment, Diagnosis and facts included are: Revenue generated, Profit, Expense, No. of treatments.

Information Diagram:

Business Process: Treatment Data Analysis				
Time	Location	Treatment	Disease Diagnosed	
Day	Street	Treatment key	Disease name	
Week	City	Treatment name	Disease information	
Month	State	Treatment cost	Doctor key	
Quarter	Country	Medicines used		
years				
Facts: Revenue generated, Profit, Expense, No. of treatments.				





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

Ans:

Slowly Changing Dimension (SCD) Modeling: SCD modeling is typically used in data warehousing and business intelligence when you have data that changes over time, such as historical customer addresses or product prices. The choice of whether to implement SCD modeling for a particular problem depends on the nature of the problem. Here are some justifications for and against using SCD modeling:

Justification for SCD Modeling:

- **Historical Analysis**: If your problem requires historical analysis, SCD modeling is crucial. For instance, analyzing how customer demographics change over time or tracking product price fluctuations over the years.
- **Auditing and Compliance**: When you need to maintain an audit trail and ensure compliance with historical data changes, SCD modeling helps you keep a record of all changes.

Justification against SCD Modeling:

- **Static Data**: If the data in your problem domain doesn't change over time, there's no need for SCD modeling. For example, if you're dealing with reference data that remains constant, like a list of countries and their attributes, you don't need SCD modeling.

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

Ans:

Fact-less Fact Table Modeling: Fact-less fact tables are used to represent events or facts that don't contain any measures or numeric values. These tables are primarily used to establish relationships between dimensions. Whether fact-less fact tables are required depends on the problem you're dealing with. Here are some justifications for not using fact-less fact table modeling:

Justification against Fact-less Fact Table Modeling:

- **Lack of Relationships**: If your problem doesn't involve establishing relationships between dimensions, there's no need for fact-less fact tables. For instance, in a simple sales reporting system, you might not need fact-less fact tables if you're only interested in sales figures without intricate dimension relationships.

Examples of Fact-less Fact Tables:

- **Date Dimension Bridge**: A fact-less fact table could be used to establish relationships between date dimensions. For example, to track holidays, weekends, or special events for analytical purposes without containing any measures.
- **Student Enrollment**: In an educational context, a fact-less fact table could be used to track student enrollments, helping to identify enrollment trends and relationships between students and courses.

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

Ans:

Large Dimension Table Management: When dealing with large dimension tables, you may encounter performance and storage challenges. To justify your approach for managing large dimension tables in your problem domain, consider the following:

- **Data Partitioning**: Use data partitioning techniques to break down large dimension tables into smaller, more manageable pieces. For instance, you can partition a time dimension table by year or month.
- **Indexing**: Implement appropriate indexing strategies to optimize query performance for large dimensions. This could involve using clustered and non-clustered indexes.
- **Aggregations and Summarizations**: Consider pre-computing aggregations and summaries to reduce the complexity and size of your dimension tables, especially when dealing with historical data.
- **Caching**: Implement caching mechanisms to store frequently accessed parts of large dimension tables in memory for faster retrieval.

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

Ans:

Junk Dimension and Surrogate Key Usage: Junk dimensions are used to combine several low-cardinality attributes into a single dimension to simplify the data warehouse structure. Surrogate keys are used as unique identifiers for dimension members. Their use depends on the specific problem domain:

- **Junk Dimension Justification**: If your problem domain contains multiple low-cardinality attributes that don't need to be analyzed separately, but their combinations are meaningful, you can use a junk dimension. For example, in a retail environment, you could use a junk dimension to combine different discount types and payment methods.
- **Surrogate Key Justification**: Surrogate keys are valuable when you need to maintain a stable and unique identifier for dimension members, especially when dealing with slowly changing dimensions. These keys help in tracking and auditing changes to dimension attributes over time. For example, in a customer dimension, a surrogate key could be used to uniquely identify customers, even if their names or addresses change.