3. The rule that is applied to place an entity in first normal form is that each column should contain atomic (indivisible) values and there should be no repeating groups or arrays. To revise the given data model into first normal form, we need to remove the repeating group of attributes for item name, quantity ordered, item unit, quantity shipped, item out, and quantity received, and create a separate entity for it. Thus, the revised data model for first normal form would look like:

Inventory Order

- Order number (identifier)
- Order date
- Customer Name
- Street Address
- City
- Customer type
- Initials
- District name
- Region name

Item

- Order number (foreign key)
- Item name
- Quantity ordered
- Item unit
- Quantity shipped
- Item out
- Quantity received

ii. The rule that is applied to place an entity into second normal form is that it should be in first normal form, and all non-key attributes should be dependent on the entire primary key. In the revised data model for first normal form, the primary key of the Inventory Order entity is the order number. All the attributes are dependent on the order number, so it is already in second normal form.

iii. The rule that is applied to place an entity into third normal form is that it should be in second normal form, and all non-key attributes should be dependent only on the primary key, and not on any other non-key attributes. In the revised data model for second normal form, all attributes in both entities are dependent only on their respective primary keys, so it is already in third normal form.

iv. Other guidelines and rules that can be followed to validate that the data model is in good form are as follows:

- Avoid redundant data by ensuring that each piece of data is stored in only one place.
- Ensure that all attributes have appropriate data types and constraints.
- Ensure that entity names and attribute names are clear, concise, and unambiguous.
- Ensure that relationships between entities are well-defined and appropriate for the data being modeled.
- Ensure that the data model is easy to understand, maintain, and update over time.

Qn 04;

4. refer page 28 in slides notes.

Despite these differences, there are also some similarities between databases and data warehouses. Both can store data, enforce data integrity and security, support data querying and reporting, and provide an interface for accessing data. Additionally, both databases and data warehouses can be managed using database management systems (DBMS) such as Oracle, SQL Server, and MySQL, and both can be hosted on-premise or in the cloud.

Question 5

5. Many companies in industry prefer the update-driven approach for integrating multiple heterogeneous information sources because it offers several advantages over the query-driven approach:

1. Performance: The update-driven approach is often faster and more efficient than the query-driven approach, as it pre-processes data from multiple sources and stores it in a central location (data warehouse). This eliminates the need to access and query multiple sources every time a user requests information.

2. Consistency: The update-driven approach ensures that all data in the data warehouse is consistent and up-to-date, as it periodically updates the data from multiple sources. This eliminates the risk of data inconsistencies that may arise from querying multiple sources at different points in time.

3. Data quality: The update-driven approach can also improve data quality, as it allows for data cleaning, transformation, and standardization before storing the data in the data warehouse. This helps to eliminate errors and inconsistencies that may exist in the original data sources.

4. Scalability: The update-driven approach is often more scalable than the query-driven approach, as it can handle large volumes of data and can be easily expanded to accommodate additional data sources.

However, there are situations where the query-driven approach is preferable over the update-driven approach. For example:

1. Real-time data: If the data sources contain real-time data that needs to be accessed and analyzed immediately, the query-driven approach may be more suitable. This is because the update-driven approach involves a delay in updating the data in the data warehouse, which may not be suitable for real-time applications.

2. Ad hoc queries: If users need to perform ad hoc queries on the data sources, the query-driven approach may be more appropriate. This is because the query-driven approach provides more flexibility in querying multiple sources and retrieving data as needed.

3. Limited data sources: If there are only a few data sources that need to be integrated, the query-driven approach may be more cost-effective and efficient than the update-driven approach, as it eliminates the

need to build and maintain a data warehouse.

In summary, the choice between the update-driven and query-driven approaches depends on the specific requirements and constraints of the integration project. The update-driven approach is often preferred for large-scale, centralized integration projects that require high performance, data consistency, and data quality, while the query-driven approach may be more suitable for real-time applications, ad hoc queries, and small-scale integration projects.

Qn 06;

a.

- Snowflake schema, fact constellation, and starnet query model are all data modeling techniques used in data warehousing.

- Snowflake schema is a normalized form of a star schema, where dimensions are normalized into multiple related tables. For example, a sales data warehouse may have a separate table for customer addresses, customer demographics, and customer purchasing history, rather than a single customer dimension table.

- Fact constellation is a schema where multiple fact tables share the same dimension tables. This allows for more complex relationships between facts, such as comparing sales revenue to production costs across different time periods and regions.

- Starnet query model is a method of querying data from multiple fact tables that share common dimensions. In this model, a central table acts as a hub for all related fact tables, and queries can be executed by traversing the hub to the relevant fact table.

b.

- Data cleaning, data transformation, and refresh are all steps in the ETL (Extract, Transform, Load) process for data integration.

- Data cleaning involves identifying and correcting errors and inconsistencies in the source data, such as missing values, duplicates, and outliers.

- Data transformation involves converting and restructuring the source data into a format that can be integrated into the target system. For example, this may involve converting date formats, combining data from multiple tables, or aggregating data by a certain criteria.

- Refresh refers to the process of updating the target system with new or changed data from the source system. This can be done on a scheduled basis, such as nightly or weekly, or in real-time as new data becomes available.

c.

- Discovery-driven cube, multifeature cube, and virtual warehouse are all approaches to data warehousing and data analysis.

- Discovery-driven cube is a type of OLAP (Online Analytical Processing) cube that allows users to explore and analyze data in an ad-hoc manner, without predefined hierarchies or dimensions. For example, a user may be able to drill down from a high-level summary of sales revenue to individual transactions, without knowing the specific dimensions or levels in advance.

- Multifeature cube is a type of OLAP cube that incorporates multiple measures and attributes into a single cube. This allows for more complex and comprehensive analysis, such as comparing sales revenue and customer satisfaction scores across different product categories and regions.

- Virtual warehouse is a cloud-based data warehousing solution that allows users to store and analyze large amounts of data without investing in physical infrastructure. Users can access the data warehouse from anywhere with an internet connection and pay only for the resources they use.

Qn 07:

a. Three classes of schemas that are popularly used for modeling data warehouses are:

Star Schema

Snowflake Schema

Fact Constellation Schema

b. Here is a schema diagram for the above data warehouse using the Star Schema:

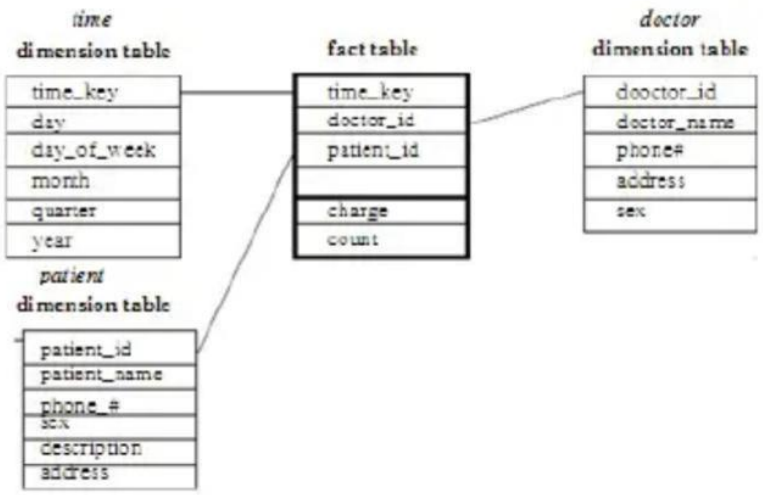

Figure 3.1: A star schema for data warehouse of Exercise 2.3.

c. To list the total fee collected by each doctor in 2015, we need to perform the following OLAP operations:

>Roll up the time dimension to the year level.

>Filter for the year 2015.

>Drill down to the doctor dimension.

>Aggregate the charge measure by summing it up for each doctor.

d. SELECT doctor, SUM(charge) as total_fee

FROM fee

WHERE year = 2015

GROUP BY doctor;

Qn 08;

a.

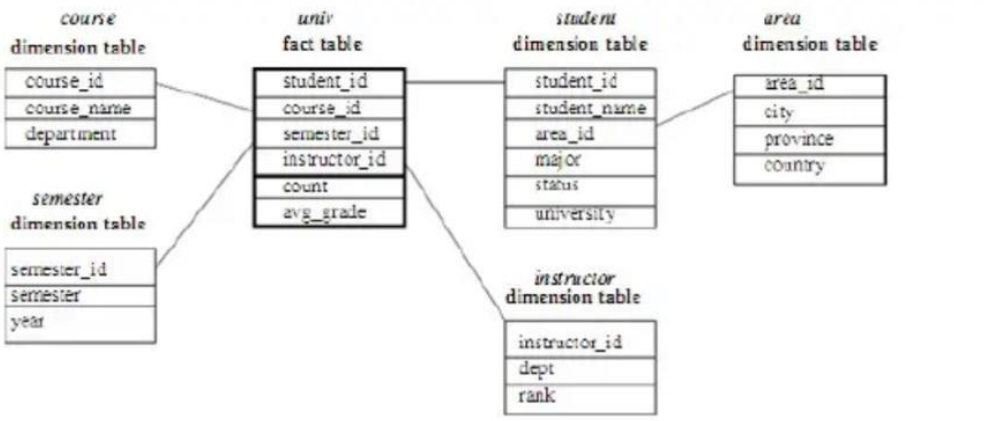

Figure 3.2: A snowflake schema for data warehouse of Exercise 2.4.

b. The specific OLAP operations that should be performed in order to list the average grade of CS courses for each Big-University student starting from the base cuboid [student, course, semester, instructor] are as follows:

>Drill-down on course dimension to the CS course level.

>Roll-up on semester dimension to the year level.

>Drill-across on instructor dimension to all instructors.

c. Each dimension has five levels, including all, so the number of cuboids in this cube, including the base and apex cuboids, is:

>(5 levels for Student) x (5 levels for Course) x (5 levels for Semester) x (5 levels for Instructor) = 625 cuboids.

Qn 09;

b. To list the total charge paid by student spectators at GM Place in 2015, the following OLAP operations should be performed:

>Select the slice of the cube for the year 2015 and the location GM Place.

>Roll-up the Spectator dimension to the category level.

>Drill-down to the Student category.

>Retrieve the Charge measure for the resulting cell.

c. Bitmap indexing is a technique used in data warehousing to speed up query processing. It has several advantages, such as:

>It requires less disk space compared to other indexing techniques.

>It can quickly perform boolean operations, such as AND, OR, and NOT, on large datasets.

>It can efficiently handle low-cardinality attributes, such as the Spectator dimension in the given data warehouse.

However, there are also some problems associated with bitmap indexing, including:

>It can become inefficient for high-cardinality attributes, such as the Game dimension, because the bitmap index can become very large.

>It can be slow for updates and inserts, as the bitmap index needs to be updated for each change.

It may not perform well for range queries, as it requires scanning multiple bitmaps to compute the result.

Qn 10;

Star schema and snowflake schema are two common modeling techniques used for data warehousing.

The similarities between the two models are that they both consist of a fact table and dimension tables, and they are both used to optimize OLAP queries.

The main difference between the two models is in the way they model the dimensions. In a star schema, each dimension is represented by a single table, while in a snowflake schema, a dimension can be represented by a hierarchy of tables, where each level of the hierarchy represents a more detailed attribute of the dimension.

The advantages of the star schema are that it is simpler and easier to understand and maintain, and it has better query performance due to the fewer number of tables involved in the queries. On the other hand, the snowflake schema provides more flexibility in representing hierarchies and can save space by eliminating duplicate data.

The disadvantages of the star schema are that it may require denormalization, leading to redundant data and possible data inconsistencies, and it may be less flexible in representing complex hierarchies. The snowflake schema, on the other hand, can be more complicated and harder to understand and maintain, and it may have a performance impact due to the increased number of tables involved in the queries.

In terms of empirical usefulness, the choice between star schema and snowflake schema depends on the specific needs of the organization. For simpler data warehousing needs, the star schema may be more practical and easier to maintain. For more complex hierarchies and larger data volumes, the snowflake schema may be more appropriate. Ultimately, the decision should be based on a careful analysis of the specific needs and requirements of the organization.