Data Warehousing Assignment

This problem set consists of two data modeling scenarios. You will be asked to analyze the strengths and weaknesses of some design alternatives for each scenario. Short answers are fine – one or two paragraphs per question would be an appropriate length.

**Scenario I** In this scenario, we are interested in modeling student enrollment in Stanford courses. We would like to answer questions such as:

* Which courses are most popular? Which instructors are most popular?
* Which courses are most popular among graduate students? Undergraduates? • Are there courses for which theassigned classrooms is too large or too small?

We are planning to have a course enrollment fact table with the grain of one row per student per course enrollment. In other words, if a student enrolls in 5 courses there will be 5 rows for that student in the fact table. We will use the following dimensions: Course, Department, Student, Term, Classroom, and Instructor. There will be a single fact measurement column, EnrollmentCount. Its value will always be equal to 1.

We are considering several options for dealing with the Instructor dimension. Interesting attributes of instructors include FirstName, LastName, Title (e.g. Assistant Professor), Department, and TenuredFlag. The difficulty is that a few courses (less than 5%) have multiple instructors. Thus it appears we cannot include the Instructor dimension in the fact table because it doesn’t match the intended grain. Here are the options under consideration:

**OptionA**

**Option B**

**Option C**

ModifytheInstructordimensionbyaddingspecialrowsrepresentinginstructorteams.Forexample,CS276ais taught by Manning and Raghavan, so there will be an Instructor row representing “Manning/Raghavan” (as well as separate rows for Manning and Raghavan, assuming that they sometimes teach courses as sole instructors). In this way, the Instructor dimension becomes true to the grain and we can include it in the fact table.

Change the grain of the fact table to be one row per student enrollment per course per instructor. For example, there will be two fact rows for each student enrolled in CS 276a, one that points to Manning as an instructor and one that points to Raghavan. However, each of the two rows will have a value of 0.5 in the EnrollmentCount field instead of a value of 1, in order to allow the fact to aggregate properly. (Enrollments are “allocated” equally among the multiple instructors.)

Create two fact tables. The first has the grain of one row per student enrollment per course and doesn’t include the Instructor dimension. The second has the grain of one row per student enrollment per course per instructor and includes the Instructor dimension (as well as all the other dimensions). Unlike Option B, the value of EnrollmentCount will be 1 for all rows in the second fact. Tell warehouse users to use the second fact table for queries involving attributes of the instructor dimension and the first fact table for all other queries.

Please answer the following questions.

**Question 1.** What are the strengths and weaknesses of each option?

Ans: Option A:

Strengths: Modifying the Instructor dimension allows it to match the intended grain of the fact table. It enables inclusion of instructors in the fact table and supports analysis related to instructor attributes.

Weaknesses: Increases complexity of the data model. Querying and aggregating data involving instructors may become more complicated and slower due to the presence of special rows representing instructor teams.

Option B:

Strengths: Changes the fact table's grain to accommodate multiple instructors per course enrollment, ensuring accurate representation. Enables proper allocation of enrollments among instructors.

Weaknesses: Introduces fractional values (0.5) in the EnrollmentCount field, potentially complicating calculations and reporting. Requires additional handling of fractional values in queries and analyses.

Option C:

Strengths: Provides separate fact tables for different query needs. The first fact table simplifies general queries, while the second fact table with the Instructor dimension allows detailed analysis of instructors.

Weaknesses: Increased complexity in managing and maintaining two fact tables. Possible redundancy in storage if dimensional attributes are duplicated in both tables.

**Question 2.** Which option would you choose and why?

Option C would be the preferred choice. It balances performance and analytical capabilities by separating queries involving instructor attributes from other queries. This optimization allows faster response times for general queries and detailed analysis when needed.

**Question 3.** Would your answer to Question 2 be different if the majority of classes had multiple instructors? How about if only one or two classes had multiple instructors? (Explain your answer.)

Would your answer to Question 2 be different if the majority of classes had multiple instructors? How about if only one or two classes had multiple instructors? (Explain your answer.)

If the majority of classes had multiple instructors, Option B would become more suitable. It accommodates the common occurrence of multiple instructors per course enrollment without requiring fractional values. Option A could still be viable if only one or two classes had multiple instructors, as it simplifies the data model for such cases.

**Question 4. [OPTIONAL]** Can you think of another reasonable alternative design besides Options A, B, and C? If so, what are the advantages and disadvantages of your alternative design?

Option D (Alternative Design): Create a separate fact table for instructor-related metrics, maintaining the original fact table without the Instructor dimension.

Advantages: Simplifies the fact table, enables detailed instructor analysis, and optimizes query performance for instructor attributes.

Disadvantages: Potential data redundancy and increased complexity in managing multiple fact tables.

**Scenario II** In this scenario, we are building a data warehouse for an online brokerage company. The company makes money by charging commissions when customers buy and sell stocks. We are planning to have a Trades fact table with the grain of one row per stock trade. We will use the following dimensions: Date, Customer, Account, Security (i.e. which stock was traded), and TradeType.

The company’s data analysts have told us that they have developed two customer scoring techniques that are used extensively in their analyses.

· Each customer is placed into one of nine Customer Activity Segments based on their frequency of transactions, average transaction size, and recency of transactions.

· EachcustomerisassignedaCustomerProfitabilityScorebasedontheprofitsearnedasaresultofthatcustomer’s trades. The score can be either 1,2,3,4, or 5, with 5 being the most profitable.

These two scores are frequently used as filters or grouping attributes in queries. For example:

· How many trades were placed in July by customers in each customer activity segment?

· What was the total commission earned in each quarter of 2003 on trades of IBM stock by customers with a profitability score of 4 or 5?

There are a total of 100,000 customers, and scores are recalculated every three months. The activity level or profitability level of some customers changes over time, and users are very interested in understanding how and why this occurs.

We are considering several options for dealing with the customer scores:

**OptionA Option B Option C**

**Option D**

ThescoresareattributesoftheCustomerdimension.Whenscoreschange,theoldscoreisoverwrittenwiththe new score (Type 1 Slowly Changing Dimension).

The scores are attributes of the Customer dimension. When scores change, new Customer dimension rows are created using the updated scores (Type 2 Slowly Changing Dimension).

The scores are stored in a separate CustomerScores dimension which contains 45 rows, one for each combi- nation of activity and profitability scores. The Trades fact table includes a foreign key to the CustomerScores dimension.

The scores are stored in a CustomerScores outrigger table which contains 45 rows. The Customer dimension includes a foreign key to the outrigger table (but the fact table does not). When scores change, the foreign key column in the Customer table is updated to point to the correct outrigger row.

Please answer the following questions.

**Question 5.** What are the strengths and weaknesses of each option?

Option A: Strengths: Type 1 Slowly Changing Dimension simplifies the data model by overwriting the old scores with new scores. It maintains a single Customer dimension.

Weaknesses: Loss of historical data, inability to track changes over time.

Option B: Strengths: Type 2 Slowly Changing Dimension creates new Customer dimension rows with updated scores, preserving historical data and allowing analysis of changes over time.

Weaknesses: Increased storage requirements due to additional rows for each score change.

Option C: Strengths: Scores stored in a separate CustomerScores dimension, allowing easy filtering and grouping based on scores. Simplifies queries involving scores.

Weaknesses: Limited flexibility if new scores or combinations are introduced.

Option D: Strengths: Scores stored in a CustomerScores outrigger table. Customer dimension includes a foreign key to the outrigger table. Allows easy filtering and grouping based on scores.

Weaknesses: The fact table does not directly reference scores, potentially limiting query performance.

**Question 6.** Which option would you choose and why?

Option B would be the preferred choice. It preserves historical data and enables tracking of score changes over time. This allows for comprehensive analysis of customer behavior and understanding changes in activity and profitability levels.

**Question 7.** Would your answer to Question 6 be different if the number of customers and/or the time interval between score recalculations was much larger or much smaller? (Explain your answer.)

If the number of customers is significantly larger or the time interval between score recalculations is much smaller, it would impact the choice of the design option.

In such cases, Option A (Type 1 Slowly Changing Dimension) might become more favorable. Overwriting the old scores with new scores simplifies the data model and reduces the storage requirements. With a larger number of customers or frequent score recalculations, managing a large number of historical rows (as in Option B) could become resource-intensive and increase storage costs. Type 1 dimension would be suitable when historical changes are not crucial and real-time analysis is more important.

On the other hand, if the number of customers is relatively small, and the time interval between score recalculations is longer, Option B (Type 2 Slowly Changing Dimension) remains a better choice. It allows preserving the historical data and provides a comprehensive view of customer changes over time, even if the storage requirements increase.

**Question 8. [OPTIONAL]** Can you think of another reasonable alternative design besides Options A, B, C, and D? If so, what are the advantages and disadvantages of your alternative design?

Option E (Alternative Design): Create a hybrid approach by combining Type 1 and Type 2 Slowly Changing Dimensions. Maintain the current Customer dimension but introduce additional columns to track score changes. Each row in the Customer dimension represents a customer's current state, while the additional columns store the historical scores and associated timestamps.

Advantages: Retains the simplicity of the existing Customer dimension while capturing historical changes in scores. Provides a balance between storage efficiency and historical analysis.

Disadvantages: Requires additional columns and increased complexity in managing and querying historical score changes. May require additional processing to extract and analyze historical data effectively.

The choice between different design options depends on the specific requirements, data volumes, query patterns, and trade-offs associated with storage, performance, and historical analysis needs.