**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 questionwould 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 the assigned 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 Tenured Flag. 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**

**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.**

**Option B**

**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.)**

**Option C**

**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?**

Option A:

Strengths:

The Instructor dimension can be included in the fact table, maintaining the intended grain.

Allows capturing the instructor teams for courses with multiple instructors.

Weaknesses:

Creates additional rows in the Instructor dimension, which may increase the dimension table size.

May require additional complexity in handling queries involving instructors.

Option B:

Strengths:

Maintains the intended grain of one row per student enrollment per course per instructor.

Allows capturing the allocation of enrollments among multiple instructors.

Weaknesses:

Requires assigning fractional values (e.g., 0.5) in the EnrollmentCount field, which can be confusing.

Increases the number of rows in the fact table, potentially impacting performance and storage.

Option C:

Strengths:

Provides separate fact tables to handle different types of queries efficiently.

Allows users to choose the appropriate fact table based on their query requirements.

Weaknesses:

Requires maintaining and managing two separate fact tables, which may increase complexity.

Users need to be aware of which fact table to use for specific queries, leading to potential confusion.

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

The choice of option depends on the specific requirements and priorities of the data warehouse implementation.

If the ability to include the Instructor dimension in the fact table is crucial for analysis and reporting, Option A would be a suitable choice. It maintains the intended grain and allows capturing instructor teams.

If the allocation of enrollments among multiple instructors is essential and the fractional values in the EnrollmentCount field can be effectively handled, Option B can be considered.

If the separation of queries involving the Instructor dimension and other queries is a priority, Option C provides a structured approach by using two separate fact tables.

**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.)**

If the majority of classes had multiple instructors, Option B may become more attractive as it allows capturing the allocation of enrollments among instructors. The fractional values in the EnrollmentCount field can be managed effectively when dealing with a significant number of classes with multiple instructors.

If only one or two classes had multiple instructors, the impact of using Option B would be minimal. In such cases, the choice between Option A and Option C would depend on other factors such as the need to include the Instructor dimension directly in the fact table or the separation of queries.

**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?**

**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 oftransactions, 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**

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

**Option B**

**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).**

**Option C**

**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.**

**Option D**

**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:

Simple implementation, as the scores are stored as attributes of the Customer dimension.

Weaknesses:

Overwrites the old scores, leading to a loss of historical information.

Doesn't capture the changes in customer activity or profitability over time.

Option B:

Strengths:

Preserves historical information by creating new rows for each score change.

Allows tracking changes in customer activity and profitability.

Weaknesses:

Increases the size of the Customer dimension with each score change, potentially impacting performance and storage.

Option C:

Strengths:

Separates the scores into a dedicated dimension, allowing easy filtering and grouping in queries.

Provides a compact representation with 45 rows covering all combinations of activity and profitability scores.

Weaknesses:

Requires joining with the CustomerScores dimension for every query involving scores, which may increase query complexity.

Option D:

Strengths:

Stores the scores in an outrigger table, reducing the size of the Customer dimension.

Allows updating the foreign key column in the Customer table for score changes, preserving historical information.

Weaknesses:

Requires an additional join with the outrigger table for queries involving scores, which may impact query performance.

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

The choice of option depends on the specific requirements and priorities of the data warehouse implementation.

If preserving historical information and tracking changes in customer activity and profitability over time is crucial, Option B would be a suitable choice. It allows creating new rows for each score change, maintaining a comprehensive record of customer scores.

If query performance and simplicity are priorities, Option A can be considered. It provides a straightforward implementation by storing the scores as attributes of the Customer dimension, although it sacrifices historical information.

**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, the impact of Option B on the size of the Customer dimension may become a concern. The increased number of rows in the Customer dimension can potentially impact performance and storage. In such cases, considering Option A or Option C, which don't increase the size of the Customer dimension, may be more favorable.

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?