Scenario 1

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 the assigned classrooms are 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:

Option A

Option B

Option C

ModifytheInstructordimensionbyaddingspecialrowsrepresentinginstructorteams.Forexample,CS276a is 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.

Q 1: What are the strengths and weaknesses of each option?

Option A: Strengths:

- Allows the Instructor dimension to be included in the fact table while maintaining the intended grain of one row per student per course enrollment.
- Avoids the need to change the fact table grain or to create multiple fact tables.
- Can accommodate courses with multiple instructors.

Weaknesses:

- Could lead to a larger and more complex dimension table.
- It may be challenging to handle changes in instructor teams over time.
- May require more complex queries to identify individual instructors versus instructor teams.

Option B: Strengths:

- Allows for a more detailed analysis of course enrollment by individual instructors.
- Can accommodate courses with multiple instructors.
- Avoids the need to create multiple fact tables.

Weaknesses:

- Requires a change in the fact table grain, which could impact existing reports and queries.
- Could be more complex for users to understand, given that the EnrollmentCount field would be used to allocate enrollments across instructors.
- May require additional effort to ensure that the EnrollmentCount field is always allocated correctly across instructors.

Option C: Strengths:

- Separates queries for different levels of granularity, making it easier for users to know which fact table to use.
- Can provide both a high-level view of course enrollment and a more detailed view by instructor.

Weaknesses:

- Requires the creation of two fact tables, which could be more complex to manage and maintain.
- May lead to increased storage requirements due to the need to maintain two fact tables.
- Requires users to be aware of which fact table to use for different queries.

Q 2: Which option would you choose and why?

The best option depends on the specific needs of the organization and the users of the data warehouse. That being said, I would recommend Option A, which involves modifying the Instructor dimension by adding special rows representing instructor teams.

Option A provides the benefit of including the Instructor dimension in the fact table while maintaining the intended grain of one row per student per course enrollment. This makes it easier to query and analyze the data, as users can easily see course enrollments by individual instructors or by instructor teams. Additionally, this option can accommodate courses with multiple instructors, which is a common occurrence at universities.

While Option A may result in a larger and more complex dimension table, it is still a manageable solution that does not require significant changes to the fact table or the creation of multiple fact tables. Overall, Option A strikes a good balance between ease of use and data accuracy.

Q 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)

Yes, if the majority of classes had multiple instructors, then Option A may become less feasible since the Instructor dimension table would become very large and potentially difficult to manage. In that case, Option B or Option C may be more appropriate since they allow for a more granular analysis of enrollments by individual instructors.

On the other hand, if only one or two classes had multiple instructors, then Option A could still be a good choice since it allows for a flexible representation of instructor teams without overly complicating the dimension table. In this case, the additional rows for instructor teams would make up a small percentage of the overall Instructor dimension table, making it manageable for querying and analysis.

In general, the choice of which option to use should be based on a careful consideration of the data requirements and the trade-offs between data accuracy, complexity, and ease of use.

Q 4: 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?

One alternative design could be to create a separate fact table for courses with multiple instructors. In this design, the original fact table would remain the same, but for courses with multiple instructors, a new fact table would be created with a grain of one row per student enrollment per course per instructor. The Instructor dimension would be included in this new fact table.

This design has the advantage of separating out the data for courses with multiple instructors, which can simplify the original fact table and make it easier to analyze. The new fact table can also provide a more detailed analysis of enrollments by individual instructors for these courses.

However, this design may also have some disadvantages. For example, it may lead to duplication of data, as some enrollments will be represented in both the original fact table and the new fact table. Additionally, it may make it more difficult to analyze enrollment data across all courses, as users will

need to switch between two different fact tables. Finally, it may be more complex to maintain and update, as changes to enrollment data for courses with multiple instructors will need to be reflected in both fact tables.

Overall, while this alternative design may be reasonable in some cases, it may not be the best choice for all situations. The choice of which design to use should depend on the specific needs of the organization and the users of the data warehouse.

Scenario 2

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.EachcustomerisassignedaCustomerProfitabilityScorebasedontheprofitsearnedasaresult ofthatcustomer'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:

Option A

Option B

Option C

Option D

ThescoresareattributesoftheCustomerdimension.Whenscoreschange,theoldscoreisoverwrittenwitht he 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.

Q 1: What are the strengths and weaknesses of each option?

Option A: In Option A, the customer scores are stored as attributes of the Customer dimension, and when the scores change, the old score is overwritten with the new score. This approach is simple and easy to understand, and it requires no additional dimension or table. However, it does not allow historical analysis of customer scores.

Option B: In Option B, the customer scores are also stored as attributes of the Customer dimension, but new rows are created in the dimension when scores change. This approach allows historical analysis of customer scores, but it increases the size of the dimension and makes it more complex.

Option C: In Option C, the customer scores are stored in a separate CustomerScores dimension with 45 rows, one for each combination of activity and profitability scores. This approach reduces the size and complexity of the Customer dimension and allows for historical analysis of scores. However, it requires the creation and maintenance of an additional dimension, and it may make queries more complex.

Option D: In Option D, the customer scores are stored in a CustomerScores outrigger table, and the Customer dimension includes a foreign key to the outrigger table. This approach is similar to Option C but does not require a foreign key in the Trades fact table. It also allows historical analysis of scores and reduces the size and complexity of the Customer dimension. However, it may still make queries more complex and requires the creation and maintenance of an additional table.

Q 2: Which option would you choose and why?

The choice of option will depend on the specific needs and requirements of the brokerage company. However, based on the given scenario and the potential advantages and disadvantages of each option, here is what I would suggest:

Option B, where the customer scores are attributes of the Customer dimension, with new Customer dimension rows created when scores change (Type 2 Slowly Changing Dimension), seems to be the most appropriate option.

One of the main advantages of this option is that it provides a historical record of changes to the customer scores, which is particularly useful for analyzing how and why the activity level or profitability level of some customers changes over time. It also allows for easy filtering and grouping of data by customer scores, which is a common requirement of the brokerage company's data analysts.

On the other hand, the main disadvantage of Option B is that it requires more storage space and may have some performance impact, as new rows are created in the Customer dimension when scores change. However, with the current data volumes and the fact that scores are only recalculated every three months, this is unlikely to be a significant issue.

Overall, Option B strikes a good balance between functionality, performance, and storage requirements, and is the most appropriate option for the given scenario.

Q 3: 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)

Yes, my answer to question 6 would be different if the number of customers and/or the time interval between score recalculations was much larger or much smaller. If the number of customers was much larger, then Option A or D might be more appropriate since they would result in a smaller data warehouse. This would make queries faster and reduce storage costs. However, if the time interval between score recalculations was much larger, then Option B or C might be more appropriate since they would allow for a more accurate historical record of customer scores.

On the other hand, if the number of customers was much smaller, then Option B or C might be more appropriate since they would allow for a more detailed and accurate record of customer scores. This would enable better analysis and decision-making. However, if the time interval between score recalculations was much smaller, then Option A or D might be more appropriate since they would result in a smaller data warehouse and allow for faster queries.

In general, the choice of option would depend on the specific requirements and constraints of the data warehouse project.

Q 4: 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?

One alternative design for storing the customer scores could be to use a hybrid approach. In this approach, the scores are attributes of the Customer dimension, and when scores change, a new row is created in the Customer dimension using the updated scores. However, the fact table would include a foreign key to the previous row in the Customer dimension, so that historical data can be easily tracked.

Advantages of this approach:

- Historical data can be easily tracked.
- Customers can be easily grouped based on their previous scores.
- It allows for more efficient querying of data.

Disadvantages of this approach:

- It requires additional complexity in the fact table to track historical data.
- It may not be as easy to query as Option C because the scores are still in the Customer dimension.

The choice of design ultimately depends on the specific requirements of the project and the preferences of the stakeholders.