**Scenario I**

**ANSWER:**

**Option A:** Modifying the Instructor dimension by adding special rows representing instructor teams seems like a reasonable option. This way, the Instructor dimension becomes true to the grain, and we can include it in the fact table. However, this option may make the fact table more complex and difficult to maintain. Moreover, it may not be easy to decide how to group instructors into teams, and some courses may have multiple teams of instructors, which can further complicate the model.

**Option B:** Changing the grain of the fact table to be one row per student enrolment per course per instructor also seems like a reasonable option. This option will accurately capture the fact that some courses have multiple instructors. However, the value of EnrollmentCount will need to be adjusted to 0.5 for each instructor, which can make aggregating the data more complicated.

**Option C:** Creating two fact tables with different grains is a good option for handling the Instructor dimension. The first fact table will not include the Instructor dimension, and the second fact table will include it. This approach will simplify the model and make it easier to query. However, it may increase the complexity of the ETL process as data needs to be loaded into two different fact tables. Additionally, it may be confusing for warehouse users who need to decide which fact table to use for each query.

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

**ANSWER:**

Option A:

Strengths:

* It enables us to include the Instructor dimension in the fact table.
* It allows us to track instructor teams, which can be helpful in understanding course popularity.

Weaknesses:

* It can lead to increased complexity in the Instructor dimension.
* It may not be suitable if there are too many courses with multiple instructors.

Option B:

Strengths:

* It provides a way to include the Instructor dimension in the fact table while maintaining the grain.
* It allows us to allocate enrollments among multiple instructors.

Weaknesses:

* It may not accurately reflect the true enrollment count.
* It may not be suitable if there are too many courses with multiple instructors.

Option C:

Strengths:

* It provides a way to handle the Instructor dimension while maintaining the grain of the fact table.
* It allows for separate fact tables, which can be helpful in optimizing queries.

Weaknesses:

* It requires two fact tables, which may lead to increased complexity.
* It may not be suitable if users frequently need to join the two fact tables.

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

**ANSWER:** I would choose Option C because it provides the most flexibility and allows for separate fact tables optimized for different queries. This approach also avoids the potential complexity of Option A and the accuracy issues of Option B.

**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.)**

**ANSWER:** If the majority of classes had multiple instructors, I would still choose Option C. It can handle the Instructor dimension while maintaining the grain of the fact table, which is important for accurate analysis. However, if only one or two classes had multiple instructors, I might consider using Option A, as it can handle instructor teams in a simpler way.

**Question 4. [OPTIONAL] Can you think of another reasonable alternative**

**ANSWER:** Another alternative design could be to include a separate fact table for instructor teams, similar to Option A, but without modifying the Instructor dimension. This approach would enable us to track instructor teams separately while avoiding the potential complexity of modifying the dimension. However, it would require a separate fact table, which may not be as efficient as Option C.

**Scenario II**

**ANSWER:** Each of the four options has its own advantages and disadvantages, and the best option depends on the specific requirements and constraints of the project. Here are some considerations for each option:

Option A: Scores as Type 1 Slowly Changing Dimension

In this option, the customer scores are stored as attributes of the Customer dimension, and when scores change, the old score is overwritten with the new score. This approach is simple and easy to implement, but it has some limitations:

Advantages:

* Simple and easy to implement
* Minimal storage requirements
* Straightforward to query

Disadvantages:

* Does not keep track of historical scores
* Cannot analyze changes in customer scores over time
* Cannot analyze customer behavior based on previous scores

Option B: Scores as Type 2 Slowly Changing Dimension

In this option, the customer scores are stored as attributes of the Customer dimension, but when scores change, new Customer dimension rows are created using the updated scores. This approach allows for tracking historical scores and analyzing changes in customer behavior over time, but it also has some drawbacks:

Advantages:

* Keeps track of historical scores
* Allows analysis of changes in customer behavior over time
* Provides a complete history of customer scores

Disadvantages:

* Higher storage requirements compared to Option A
* More complex to implement and maintain
* Can lead to data duplication if other attributes of the Customer dimension are also changing frequently

Option C: Scores stored in a separate dimension

In this option, the customer scores are stored in a separate CustomerScores dimension, which contains 45 rows, one for each combination of activity and profitability scores. The Trades fact table includes a foreign key to the CustomerScores dimension. This approach provides a clean separation between customer scores and other customer attributes, but it has some limitations:

Advantages:

* Provides a clean separation between customer scores and other customer attributes
* Simplifies queries that only need to reference customer scores
* Allows for efficient aggregation of trade data based on customer scores

Disadvantages:

* Limited to the 45 combinations of activity and profitability scores
* Cannot track changes in individual customer scores over time
* May require complex joins to include other customer attributes in queries

Option D: Scores stored in a CustomerScores outrigger table

In this option, the customer 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. This approach is similar to Option C but avoids the need for complex joins to include other customer attributes in queries. However, it has some limitations:

Advantages:

* Provides a clean separation between customer scores and other customer attributes
* Simplifies queries that only need to reference customer scores
* Allows for efficient aggregation of trade data based on customer scores
* Avoids complex joins to include other customer attributes in queries

Disadvantages:

* Limited to the 45 combinations of activity and profitability scores
* Cannot track changes in individual customer scores over time
* Requires additional maintenance to update the foreign key column in the Customer table

Overall, the best option depends on the specific requirements and constraints of the project. Options A and B are suitable for projects where historical scores are not important, and storage space is a concern. Options C and D are more suitable for projects where customer scores need to be tracked separately from other customer attributes and efficient aggregation based on customer scores is required. Option B is the best choice if historical customer scores need to be tracked, but it requires more storage and maintenance overhead.

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

**ANSWER:**

Option A (Type 1 Slowly Changing Dimension)

Strengths:

* Simplest to implement and requires the least amount of storage space.
* Allows easy querying and reporting on current scores without having to join to another dimension table.

Weaknesses:

* Historical data is lost when scores change, making it difficult to analyze changes in customer behavior over time.
* Does not allow for analysis of customers' historical profitability or activity level.

Option B (Type 2 Slowly Changing Dimension)

Strengths:

* Preserves historical data and allows for analysis of changes in customer behavior over time.
* Provides a complete picture of a customer's profitability or activity level across time.

Weaknesses:

* Requires additional storage space and increased complexity in maintaining the dimension table.
* Can result in more complex queries due to the need to aggregate over multiple rows to get a complete picture of a customer's profitability or activity level.

Option C (Separate CustomerScores Dimension)

Strengths:

* Allows for easy querying and reporting on the combination of customer activity segment and profitability score.
* Can result in more efficient queries as the fact table only needs to join to a smaller dimension table.

Weaknesses:

* Does not allow for analysis of changes in customer behavior over time.
* Results in additional storage space and increased complexity in maintaining the dimension table.

Option D (CustomerScores Outrigger Table)

Strengths:

* Allows for easy querying and reporting on the combination of customer activity segment and profitability score.
* Preserves historical data and allows for analysis of changes in customer behavior over time.

Weaknesses:

* Requires additional storage space and increased complexity in maintaining the Customer table and outrigger table.
* Can result in more complex queries due to the need to join to both the Customer and outrigger tables.

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

**ANSWER:** I would choose Option B (Type 2 Slowly Changing Dimension) because it provides a complete picture of a customer's profitability or activity level across time while preserving historical data. This option allows for analysis of changes in customer behavior over time, which is important in understanding why customers' profitability or activity level changes. Although this option requires additional storage space and increased complexity in maintaining the dimension table, the benefits of historical analysis outweigh the costs.

**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.)**

**ANSWER:** If the number of customers and/or the time interval between score recalculations was much larger, I would still choose Option B. Although this option requires additional storage space and increased complexity in maintaining the dimension table, it provides a complete picture of a customer's profitability or activity level across time while preserving historical data. However, if the number of customers and/or the time interval between score recalculations was much smaller, Option A (Type 1 Slowly Changing Dimension) may be a reasonable alternative as it is the simplest to implement and requires the least amount of storage space.

**Question 8. [OPTIONAL] Can you think of another reasonable alternative**

**ANSWER:** Another reasonable alternative could be to use a hybrid approach where customer scores are stored in a separate dimension table but with Type 2 Slowly Changing Dimension. This would allow for easy querying and reporting on the combination of customer activity segment and profitability score while preserving historical data and allowing for analysis of changes in customer behavior over time. However, this approach may result in more complex queries due to the need to aggregate over multiple rows to get a complete picture of a customer's profitability or activity level