

Data Warehouse Assignment 2

Scenario I

1st Solution:

Option A Strength: By adding special row “Instructor Team”, it help’s in matching the intended grain. If there are two instructors, we can give them 0.5 each and aggregate will be equal to 1. If there is only one instructor then we can give him 1. And if there are two instructors, here we are representing as ‘Instructor1/Instructor2’ so it will be useful. If there is a single instructor there won’t be any problem because as we are representing instructor individually.

Option A Weakness: As mentioned that less than 5% courses are having multiple instructors. For this 5% courses we are adding “Instructor Team” and in that row we are adding Multiple instructor’s names together and separately. Which increase the size of the table.

Option B Strength: This option is very useful if a course has multiple instructors. As one points to instructor1 and one points to instructor2. However, each of two rows will have a value of 0.5 in the Enrollment Count Field instead of a value of 1.

Option B Weakness: If a course has single instructor. Using two fact rows for each student makes no sense.

Option C Strength: As we are creating two fact tables (Student_course table and Student_Instructor table). If we want course details we only refer to student_course table and if we want instructor details we can refer to student_instructor table.

Option C Weakness: In both the fact tables, Student details are common. Same Student details are used in both tables.

2nd Solution: Among all the three options I would prefer first option because in the first option we are just adding a row. Whereas, in the second option we are adding two fact rows and in the third row we are creating two fact tables. Adding a row is far better than adding two fact rows and creating two fact tables.

3rd Solution: No, Still I would prefer option one. As mentioned that less than 5% courses are having multiple instructors. For this 5% courses we are adding “Instructor Team” and in that row we are adding Multiple instructors names together and separately. Which increase the size of the table. Whereas, in the second option we are adding two fact rows and in the third row we are creating two fact tables. Adding a row is far better than adding two fact rows and creating two fact tables.

Scenario II

1st Solution:

Option A Strength: Without creating additional customer dimension row, old scores are overwritten with new scores. Here, size of the table is not increasing. Which therefore, increase the performance.

Option A Weakness: when we need to perform any analysis or to know loss/profit at that time we'll compare old scores with new scores. But here we'll miss old scores because old scores are overwritten by new scores.

Option B Strength: option 1's disadvantage has overcome in this option. When scores changes, new customer dimension rows are created using the updated scores. Here we'll have both old and updated scores.

Option B Weakness: By adding new customer dimension row, we are increasing the size of the table. Which therefore effect the performance.

Option C Strength: Trade fact table includes a foreign key to the Customer Scores dimension, foreign keys helps to prevent some forms of database pollution.

Option C Weakness: The main disadvantage is increase in size for the fact table records. Increased row size in the fact table would slow down queries.

Option D Strength: Customer dimension includes a foreign key to the outrigger table and Trade fact table includes a foreign key to the Customer Scores dimension, foreign keys helps to prevent some forms of database pollution.

Option D Weakness: Customer dimension includes a foreign key to the outrigger table and Trade fact table includes a foreign key to the Customer Scores dimension, any changes in outrigger table should reflect in customer dimension and Trade fact. Which therefore effect the performance.

2nd Solution: Among all the four options I would prefer third option because in the third option, because in the first option old scores are overwritten. In 2nd option, for every score update we are adding a new customer dimension row, size of the table increases and effect the performance. Whereas in 4th option Customer dimension includes a foreign key to the outrigger table and Trade fact table includes a foreign key to the Customer Scores dimension, any changes in outrigger table should reflect in customer dimension and Trade fact. Which therefore effect the performance.

3rd Solution: No, Still I would prefer option three. In the first option old scores are overwritten, so I won't prefer that option. If the number of customers and/or the time interval between score recalculations are small I would be the best option. If the number of customers and/or the time interval between score recalculations are large I won't prefer this option, as this option effects performance. Whereas in 4th option Customer dimension includes a foreign key to the outrigger table and Trade fact table includes a foreign key to the Customer Scores dimension, any changes in outrigger table should reflect in customer dimension and Trade fact. Which therefore effect the performance

