Data Warehousing Assignment:02

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?

Option A (Instructor teams):

Strengths:

- Simplest option to implement.
- Maintains consistent grain across the model.

Weaknesses:

- Introduces artificiality ("Manning/Raghavan").
- Might not be intuitive for queries about individual instructors.

Option B (Multi-row enrollments):

Strengths:

- Captures all instructor information with accurate granular detail.
- Allows aggregation across individual instructors.

Weaknesses:

- Fact table size doubles.
- Requires special handling of enrollment count due to multiple instructors.
- Might be confusing for users due to non-integer counts.

Option C (Separate fact tables):

Strengths:

- Flexible for different query types (instructor vs. non-instructor).
- Maintains accurate grain and data integrity.

Weaknesses:

- Increases complexity (managing two tables).
- Users need to know which table to use for specific needs.
- Might create performance challenges for complex joins.

Question 2. Which option would you choose and why?

Choosing between B and C depends on the frequency of multi-instructor courses:

Majority of courses with multiple instructors: Opt for Option B for detailed instructor analysis, despite complexity.

Few courses with multiple instructors: Option C offers good balance between flexibility and complexity. If these courses are rarely queried, prioritize simplicity with Option A.

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.)

Majority of classes with multiple instructors:

- **Option B** remains the most suitable choice, even with increased complexity. Its ability to accurately capture individual instructor involvement across most courses outweighs the drawbacks.
- **Option C** becomes less attractive. Managing two tables becomes inconvenient when most queries will involve instructor information.
- **Option A** is even less suitable. The artificial "instructor teams" become increasingly impractical and misleading with so many courses affected.

Few classes with multiple instructors:

- **Option C** becomes a stronger contender. The simplicity of separate tables is valuable when most queries won't involve multi-instructor scenarios.

- **Option B** still offers detailed analysis but introduces complexity for relatively few courses. Its benefit needs careful evaluation against the additional table size and processing overhead.
- **Option A** might be considered if the specific multi-instructor courses are rarely queried. Its simplicity outweighs its limitations in such cases.

One or two classes with multiple instructors:

- **Option C** becomes very attractive. The complexity is minimal, and separate tables clearly handle the rare exceptions without impacting most queries.
- **Option B** is likely overkill for just one or two cases. Its benefits don't justify the complexity.
- **Option A** might be acceptable if the specific courses aren't crucial for analysis. However, if their instructor information is important, even for limited use cases, Option C's clarity might be preferred.

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?
Alternative Design (Optional):

Consider a single fact table with the original grain **(student, course, term, classroom)** and two separate dimensions:

1. **Instructor dimension:** One row per instructor with usual attributes.
2. **Instructor assignment dimension:** One row per course offering, linking to the course, term, and relevant instructors (can handle multiple instructors).

Advantages:

- Simplified fact table with consistent grain.
- Maintains detailed instructor information for analysis.
- Avoids artificial instructor teams or fractional counts.

Disadvantages:

- Requires a new dimension (increased complexity).
- Might require additional joins for some queries

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.

- EachcustomerisassignedaCustomerProfitabilityScorebasedontheprofitsearnedasaresultoftha tcustomer'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:** Simplest, efficient storage.
- **Weaknesses:** No historical analysis, inaccurate results for past score queries.

Option B:

- **Strengths:** Full historical analysis, accurate results.
- **Weaknesses:** Most complex, significant table size growth.

Option C:

- **Strengths:** Efficient storage, simple analysis by score segments.
- **Weaknesses:** Limited flexibility for individual score change analysis, requires additional joins.

Option D:

- **Strengths:** Balances historical analysis and storage efficiency, avoids joins for score lookups.
- **Weaknesses:** Requires managing an additional table, introduces complexity with foreign key updates.

Question 6. Which option would you choose and why?

- **Prioritize historical analysis:** Opt for **Option B**.
- **Prioritize performance and simplicity:** Choose **Option C**.
- **Balance needs:** Consider **Option D** for a middle ground.

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.)

- **Larger customer base or longer recalculation intervals: Option B** becomes less attractive, **Option D** becomes more appealing.
- **Smaller customer base or shorter recalculation intervals: Option B** becomes more feasible, **Option C** and **D** might be less significant differentiators.

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?

- **Hybrid approach:** Store current scores in Customer dimension, track historical changes in a separate CustomerScoreHistory table.
- **Advantages:** Combines benefits of Option A and B, manages size growth more efficiently.
- **Disadvantages:** Introduces complexity of managing another table, requires designing queries with joins.