Data Warehousing Assignment

This problem set consists of two data modeling scenarios. You will be asked to analyze the strengths and weak- nesses 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 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

Modify the Instructor dimension by adding specia lrows representing in structorteams.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?

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

**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.)

**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?

**Solution:**

**Consieration:** student enrollment in Stanford courses

**Dimension Table:**

**Courses ,-Cid,Cname,CDescription**

**Department-Did,Dname,Sid,Iid**

**Instructor -Iid,** **FirstName, LastName, Title, Did, and TenuredFlag.**

**Students(Under Graduation ,Graduation):Sid,Sname,Saddress,Graduate(yes/no)**

**Term-Tid,Tcode,Startdate,Enddate**

**Classrooms-CRid,size,location**

**Fact Table:**

**CourseEnrollement: CRid ,Did, Cid, Sid, crid, Price, Enrollmentcount=1**

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.

**CourseEnrollement: CRid ,Did, Cid, Sid, crid, Price, Enrollmentcount=1**

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

**Ans:** **OptionA:** Modify the Instructor dimension by adding special rows representing instructorteams. 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.

**Dimension Table:**

**Courses ,-Cid,Cname,CDescription**

**Department-Did,Dname,Sid,Iid**

**Instructor -Iid,** **FirstName, LastName, Title, Did, and TenuredFlag,InstructorTeamId(fk).**

**InstructorTeam- InstructorTeamId(pk),InstructorName**

**Students(Under Graduation ,Graduation):Sid,Sname,Saddress,Graduate(yes/no)**

**Term-Tid,Tcode,Startdate,Enddate**

**Classrooms-CRid,size,location**

**Fact Table:**

**CourseEnrollement: CRid ,Did, Cid, Sid,Iid, CRid, Price, Enrollmentcount=1**

**Strengh:**According to Option 1 we have included InstrucrtorTeamId which referes to InstructorTeam in anthor table with ID and Instructor name separately and If there are multiple instructor for a course we can easily Identify the name of the instructor for a particular course through Its Id and easily achive the mentioned grained.

**Weakness**:To find out the Finest level of Information we have included anthor new dimension table called InstructorTeams.So in terms of space and complexity wise to retrieve operation we have to join those table together.

**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.)

**Ans:**

**Fact Table: CourseEnrollement: CRid ,Did, Cid, Sid,Iid(Instructor ID), CRid, Price, Enrollmentcount=0.5**

**According to Option B:**

**Each Student: CRid ,Did, Cid, Sid,Iid(Instructor ID), CRid, Price, Enrollmentcount=0.5**

**1: CRid,Did, Cid, Sid,** Manning**, CRid, Price, Enrollmentcount=0.5**

**2: CRid,Did, Cid, Sid,** Raghavan**, CRid, Price, Enrollmentcount=0.5**

**Strength:**We can easily find the finest level of information tgrough for each instructor for course satisfied with multiple instructor for a single course.

**Weakness:**For a single course there will be 2 rows(if onc registred course has 2 instructor)

For a single course there will be Multiple rows based on no of instructor per course with calculated sum as 1.

Disadvantage is there will be multiple rows created for a single table which will be going Bigger in size and occupy more space .

**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.

Ans:

With out Instructor dimension

FACT Table:

**CourseEnrollement: CRid ,Did, Cid, Sid, crid, Price, Enrollmentcount=1**

**CourseEnrollementInstructor: CRid ,Did, Cid, Sid,Iid, CRid, Price, Enrollmentcount=1**

**Strength:**In this Scenario we are using TWO Fact table to achve the garin.We can use second Fact table CourseEnrollementInstructor rearding information about Instructor else use Fisrt fact table CourseEnrollement.

**Weakness:**Instead of using Once fact table we are using TWO with many duplicate columns combained in both fact table .Memory space will be High .Multiple same columns value except instructor will be save inboth fact table tofind the grain.

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

**Ans.**I will choose OPTION A as

**Dimension Table:**

**Courses ,-Cid,Cname,CDescription**

**Department-Did,Dname,Sid,Iid**

**Instructor -Iid,** **FirstName, LastName, Title, Did, and TenuredFlag,InstructorTeamId(fk).**

**InstructorTeam- InstructorTeamId(pk),InstructorName,Cid(fk)**

**Students(Under Graduation ,Graduation):Sid,Sname,Saddress,Graduate(yes/no)**

**Term-Tid,Tcode,Startdate,Enddate**

**Classrooms-CRid,size,location**

**Fact Table:**

**CourseEnrollement: CRid ,Did, Cid, Sid,Iid, CRid, Price, Enrollmentcount=1**

**Strengh:** According to Option 1 we have included InstrucrtorTeamId which referes to InstructorTeam in anthor table with ID and Instructor name separately and If there are multiple instructor for a course we can easily Identify the name of the instructor for a particular course through Its Id and easily achive the mentioned grained rather than using One fact table with multiple rows per student or TWO fact table with multiple same columns value in the respected database **in terms of better time as well as space and by considering simple design approach which is easy to use and desgin.**

**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.)

**Ans:** to Question 2 answer will be if the majority of classes had multiple instructors the apporach will be same has OPTION A with just include new dimension table represent the grain as student per instructor per classes per course.(multiple instructorid with classroomID can be include) to achive the finest level of grain .

**If** only one or two classes had multiple instructors ->Go with OPTION A because In OPTION A going with a new dimension table with InstructorTEAM(Itid,Iname,Crid) can esilyly achive the grain with out duplication of any rows nd column value.

**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.
  + 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 Option B Option C Option D

The scores areattributes of the Customer dimension.When scores change,the old score is overwritten with the 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?

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

**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.)

**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?

Solution:

**OPTION A:** The scores areattributes of the Customer dimension.When scores change,the old score is overwritten with the new score (Type 1 Slowly Changing Dimension).

Strength:simple and straigt forward

Best for error situation do not need any histroy for correction

Weakness:Meight still want histroy of errors for auditing purpose

**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).

Strength: allows us to accurately keep all historical information.

Weakness:As new row added to the Dimension Table .

In cases where the number of rows for the table is very high to start with, storage and performance can become a concern.

This complicates the ETL Processes

**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.

FactTable🡪CustomersScores(Fk)

Strength:can easily find the grain by using fact table as Fact table include the FK of CustomersScores.

Can apply type 2 scd in Customer scores table to maintain the Histroy.

Weakness:Wecan directly maintain the scores in the fact table rather than creating a anthor dimesnion which leads to the complexity of the design .

**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.

OuttriggerTable(Fk)🡪Customer Table(FK)🡪Fact table

Strength:To maintain the Score we are creating another table and comabining with the Customer table and foreign key of customer table act as forign key for the fact table.We can separtely easily acghive the grain.

Weakness:

Complexity will increase as creating another dimesnion table and connecting to the customer dimension and to the Fact.So,Memory size will be increase.

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

Ans.I will Choose OPTION C because only have to create a new dimension table named customer scores only for scores and stored it’s fK to Fact table .In that fact table its mapped to the customer dimension FK.So we can esily acive the grain.

**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.)

Ans.If Customer are higher and time inbterval is higher->I can choose OPTION D As it is not frequently changed.

If Customer are higher and time inbterval is lower->I can choose OPTION A,B(Based on the requirements),C(most Exactly) As it is frequently changed.