From: Christopher Singleton

Review of Dimensional Modeling Techniques

Questions:

1. What do we mean by "the grain of the fact" table?

   The grain of the fact means what detail level the measure represents on each row in the fact table.

2. Describe each of the following types of fact tables. Describe a scenario where each would be useful.
   a. Transaction
   b. Periodic Snapshot
   c. Accumulating Snapshot

   a. In the Transaction fact table, each row represents the grain with the lowest level details of the measures. Measures are additive. When the sales amount needs to be tracked, the transaction fact table should be built with the sales order item line as the grain of the fact. Each rows describes sales amount of each line item of each product on the sales order.

   b. In the Periodic Snapshot fact table, each row represents a measure captured at a point in time. That means that a measure is recorded at the end of a specific period time by the lowest level dimension attributes. Measures are not additive. They are non-additive. When a company needs to track its inventory of the products, the Periodic Snapshot applies to this scenario. For example, quantity on hand of each product at the end of day, week, month, or year.

   c. In the Accumulating Snapshot fact table, there are multiple units of measure. The type of design has both transaction and periodic snapshot features. It tracks the measures during an operational process. In this process pipeline, each row captures a measure/fact at the end of each process (it is a periodic snapshot). Between each process of the pipeline, measures are transaction facts. For example, when a manufacturer needs to track the quantities of the components for the factory to make the finished products, the stage of parts released to the production line, the stage of assembled parts passed, and the stage of assembled parts failed need to be recorded during each process. In this scenario, the Accumulated Snapshot should be built.

3. What is a centipede? Are centipedes desirable? Why or why not?

   Centipede is a type of snow-flake dimensional model of the data warehouse. This design attempts to normalize the database structure, instead of de-normalizing, and it ends up with too many dimensions relating to the lowest level dimensions. Centipedes are not desirable because too many dimensions would add more rows to the fact table resulting in increasing the size of the fact table. This will require more disk space to store the fact table data. In addition, too many reference foreign keys will cause difficulties on indexing. Lastly, too many dimensions would cause hardship of using the database and query performance since too many joins have to be used.

4. Describe dimension role-playing and how it is used in a dimensional model.

Dimension role-playing refers to an attribute in a dimension table plays multiple roles reflecting in the fact table. It is referenced multiple times in a fact table. For example, Datekey (primary key) in the date dimension references to (links to) OrderDatekey, ShippedDateKey, and DeliveredDateKey in a sales_fact table. The Datekey plays 3 different roles in the fact table. That is, each foreign key in the sales_fact table references 3 independent views to the date dimension. When querying, DateKey can join OrderDateKey, DateKey can join ShippedDateKey, Datekey can join DeliveredDateKey, so 3 different sets of data results can be extracted.

5. What are conformed dimensions? What are the advantages/disadvantages of conformed dimensions?

Conformed dimensions are dimensions that have consistent dimension keys, consistent attribute names, consistent attribute values, and consistent attribute definitions throughout the entire data warehouse bus architecture. Conformed dimensions have the lowest level granularity to all the fact tables in the data warehouse. Confomed dimensions have the same definition to every fact table they link to in the data warehouse.

Advantages of conformed dimensions are that conformed dimensions can be re-usable, which means confirmed dimension can be used to multiple fact tables or business processes in different data marts across the entire data warehouse bus architecture.

Disadvantages of conformed dimensions are that conformed dimensions require more political decision than technique decisions. That means that it requires the executive level management to force and push the decision made to have the conformed dimensions implemented in all possible data marts throughout the organization.

6. Describe the following types of facts (measures) and, for each, describe a scenario that illustrates the use of each fact type.

   a. Additive
   b. Semi-Additive
   c. Not Additive

   a. Additive facts refers to those measures that are quantitive and can be accumulated. For example, sales amount and quantity are quantitive measures in the SalesOrder_Fact table.
   b. Semi-Additive fact are used in the periodic snapshot fact model. The measures are additive by some dimensions, but not by others. For instance, in the inventory periodic snapshot fact table, quantity on hand is additive by product or by store, but it is not additive by dates. Assuming the periodic snapshot is daily snapshot, if you want to know quantity on hand by week, the quantity on hand can be added up or accumulated from Monday to Friday because quantity is captured by the end of each day, and end of the week (Friday)'s quantity on hand reflects the quantity for that week. Therefore, the measure (quantity on hand) is semi-additive.
   c. Not-Additive measures are the measures that cannot be summed up or added up across all dimensions, such as the average amount by product or by customer.

7. Describe dimensional modeling as though you are talking to someone who is familiar only with normalized database design. Include what you consider to be the key differences between dimensional data warehouse design and operational database design.

Dimensional modeling is different from normalized database modeling (3rd normal form design).

First, the OLTP (normalized) database modeling is for a purpose of recording and manipulating the daily operational transactions. Each record or transaction can be inserted, updated, or deleted easily in the database. The OLAP (dimensional) database modeling is for a purpose of a large volumn of data analysis.

Second, normalized modeling is highly normalized, which means that it tries to avoid data redundancy (avoid data duplications in each table). Data can be accessed and manipulated quickly. Data stored can be queried and modified in real time. In other words, data is always live data. However, data warehouse (dimensional modeling) is highly denormalized. Since there will be less joins needed to query the data, query's performance will be much better and the speed will be much faster. The measures stored in the fact table are precaluted aggregations, so that also increases the query speed tremendously to meet the purpose of data analysis. For the normalized database, querying a large set of aggregated data seems impossible because it will give database engine a huge stress by calculating data in real time. Data warehouse can analyzed data fast in a multidimensional way without too many joins, but data can have latency because it is loaded/updated in schedule.

Finally, a normalized database does not store all historical data. Companies usually archive data periodically to give more room on the production disk. From business intelligence view, companies need to analyze historical data to find insights and trends in order to make clear, concise and better decisions. The operational (OLTP) database cannot satisfy this needs for the organizations. On the other hand, a dimensional data warehouse is a data repository, and it stores all current data and historical data of the company. It allows the company to be able to see an entire picture of all dimensional views of the data.

8. Consider the operational data and the data warehouse data shown below. Do you see any issue(s) with the **fact table rows** in the data warehouse Sales fact table If so, how would you correct it/them? Show the fact table as it should be including any corrected data. ? (Do not consider corrections to the operational data or tables.)

OPERATIONAL SYSTEM

SalesOrder Table

Sales Order No.	CustomerID	Date	Freight
201	A1001	3/1/07	40

SalesOrderLine Table

Sales Order No.	Sales Order Line	Qty	ProductID	Price
201	1	2	P345	40
201	2	5	P89	10

DATA WAREHOUSE

Sales Fact Table

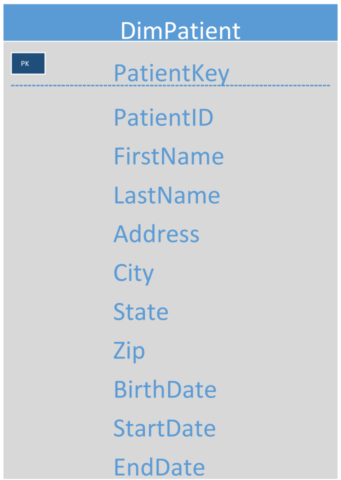
Sales Order No.	SalesOrder Line	Customerkey	DateKey	ProductKey	SalesAmount	**Weighting Factor**	**Freight**
201	1	100	240	432	80	**0.62**	**25**
201	2	100	111	231	50	**0.38**	**15**

Yes. There was a problem in the fact table. The freight for the two products in the same order (201) should not be $40 respectively for each product. From the data shown above in the SalesOrder table of the operational system, $40 spent for the entire order (#201), not for each product on the order. So, there each product should be weighed in ratio according to total sales amount of each product. Total ratio of the weighting factor is 1. Line 1 occupies 0.62 (calculation: 80/(80+50)); Line 2 occupies 0.38 (calculation: 50/(80+50)). Freight calculation for Line 1: 80*0.62 = 25; For Line 2: 50*0.38 = 15

9. Design an abbreviated version of a patient dimension table for a health care billing system and create data to demonstrate how slowly changing dimension data would be accommodated for the address attribute.

   a. The dimension must include:
      i. Attributes for patient name
      ii. Attributes for patient address
      iii. Birth date
      iv. Start and End dates for changes to accommodate a type 2 SCD

   b. You must show data rows in the dimension table to accommodate the following information:
      i. John Chen, born 3/17/67, is a single patient and is registered with the system on 4/2/2003. His address at that time is 3000 Landerholm Circle SE, Bellevue, WA 98007
      ii. John moves to 425 164th Ave SE, Bellevue, WA 98008 on 5/3/2005 and immediately notifies the billing organization.



DimPatient

PK PatientKey

PatientID
FirstName
LastName
Address
City
State
Zip
BirthDate
StartDate
EndDate

Patient Dimension Table

PatientKey	**PatientID**	**FirstName**	**LastName**	**Address**	**City**	**State**	**Zip**	**BirthDate**	**StartDate**	**EndDate**
1	9876	John	Chen	3000 Landerholm Circle SE	Bellevue	WA	98007	3/17/1967	4/2/2003	5/2/2005
2	9876	John	Chen	**425 164th Ave SE**	Bellevue	WA	**98008**	3/17/1967	**5/3/2005**	**Null**

Type 2 slowly changing dimension is reflected on the second row added (Surrogate Key is PatientKey 2). About when was the address changed, it shows on the StartDate (5/3/2005) on the second row.

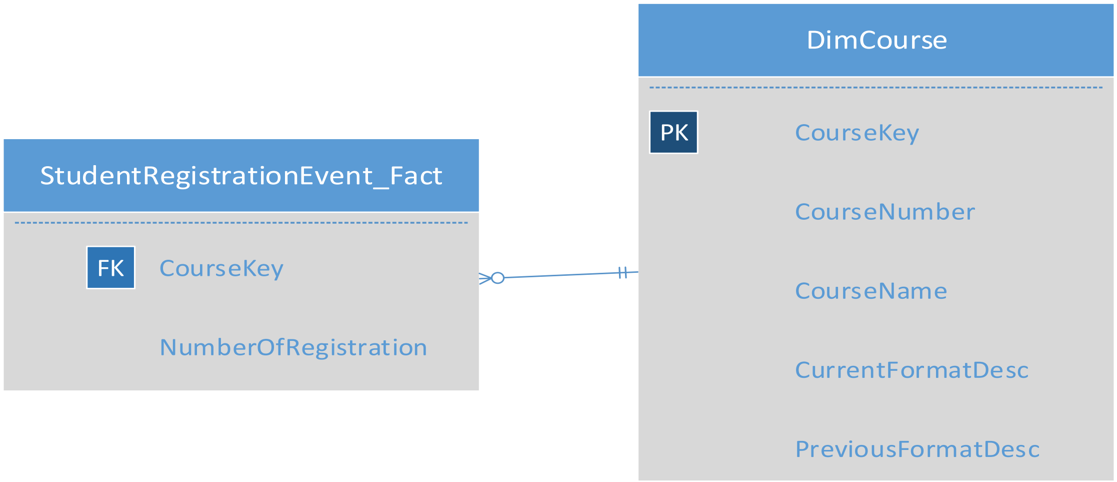
10. What is meant by a type 2 slowly changing dimension? Explain how a type 2 slowly changing dimension differs from a type 1 slowly changing dimension and how it differs from a type 3 slowly changing dimension.

Type 2 slowly changing dimension means that a new row is added to record any attributes changed from the original record. Time stamp should be added to indicate when the change starts and when the change ends (such as Coloum StartDate and column EndDate). Note that the new row's primary key/surrogate key will be different, but natural key/business key remains the same.

Type 1 slowly changing dimension is totally different from Type 2. Type 1 will not keep any historical data. Any attributes changes will overwrite the previous one. There will be no rows or columns to be added to the original model. All you can see from the records is all most current data.

A type 3 slowly changing dimension is different from a type 2. For Type 2 model, a new set of data is added and all historical records will be kept in the dimension table. Unlike Type 2, Type 3 only keeps the previous record as history and a current record. A type 3 model adds a new column to record the previous attribute (the old value) and overwrite the old value of the attribute with the current value. A time stamp may be also necessary to add.

11. Describe a hybrid approach to dimension change. Include an abbreviated dimensional model as an example.



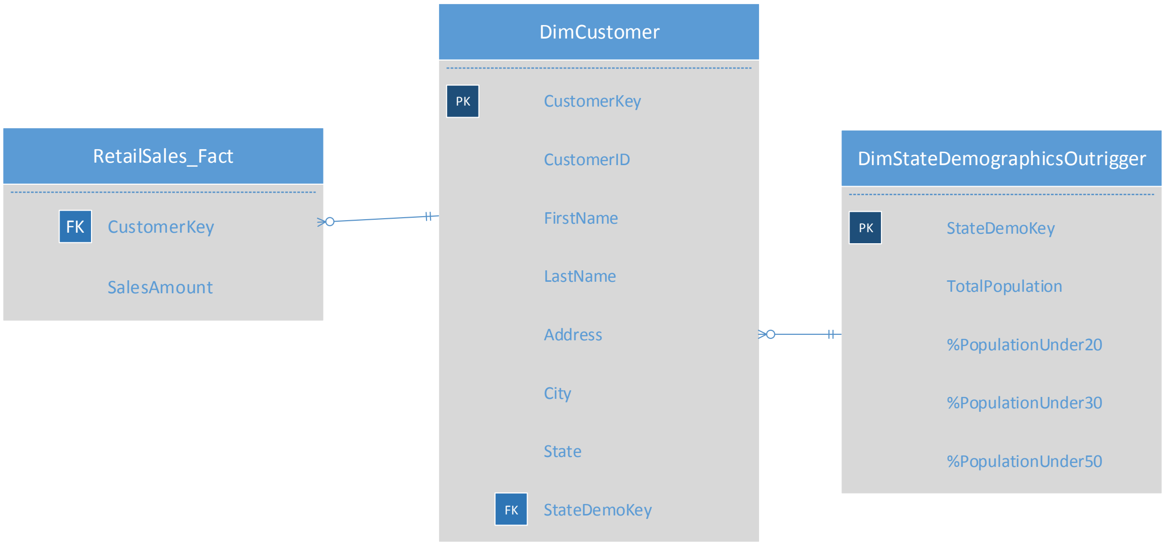
A hybrid approach to dimension changes is a combination of Type 1, Type 2, and Type 3 slowly changing dimension. A hybrid approach adds a new column for the current attribute and keep the prior attribute (This applies to Type 3. See column CurrentFormatDes). A hybrid approach also adds a new row to record all the historical values of the attribute, not just the prior attribute (This applies to Type 2), and a hybrid approach overwrites all the values of the new columns added for the Type 2 to the most updated current values (This applies to Type 1). The example is as above.

12. What is a junk dimension? Describe a situation in which a junk dimension could be used.

    A junk dimension is a dimension created with all unique, discrete values, such as indicators, flags. Each value is unique, but it is significant enough to be added to the fact table. That is going to be overwhelming to the design with too many dimensions if each flag or indicator is added into its own dimension. Creating another dimension can solve this problem with all attributes of the flags and indicators. This dimension is called Junk dimension. For example, for order transaction process, each other has attribute of payment term, payment type, and order type. These attributes are the attributes of the junk dimension which its primay key is referenced in the order transaction fact table.

13. What is an outrigger? Under what circumstances might you use an outrigger? Give an example of the use of an outrigger including an abbreviated dimensional model.

    An outrigger is used to form a snow-flake dimension model that has attribute values that are static, does not change. It is a reference dimension to the primary dimension, and has a different grain from the primary dimension. The purpose of using an outrigger is to reduce the size of the primary dimension because the attribute values that do not change would be duplicated again and again in the primary dimension. When those values of those attributes do not change as the other attributes change, an outrigger should be used, such as total population of the state for a customer and percentage of the total population under a certain age. Please see the dimensional model below. DimStateDemographics Outrigger is an example.



14. What does the term “causal dimension” mean? Give an example of a causal dimension.

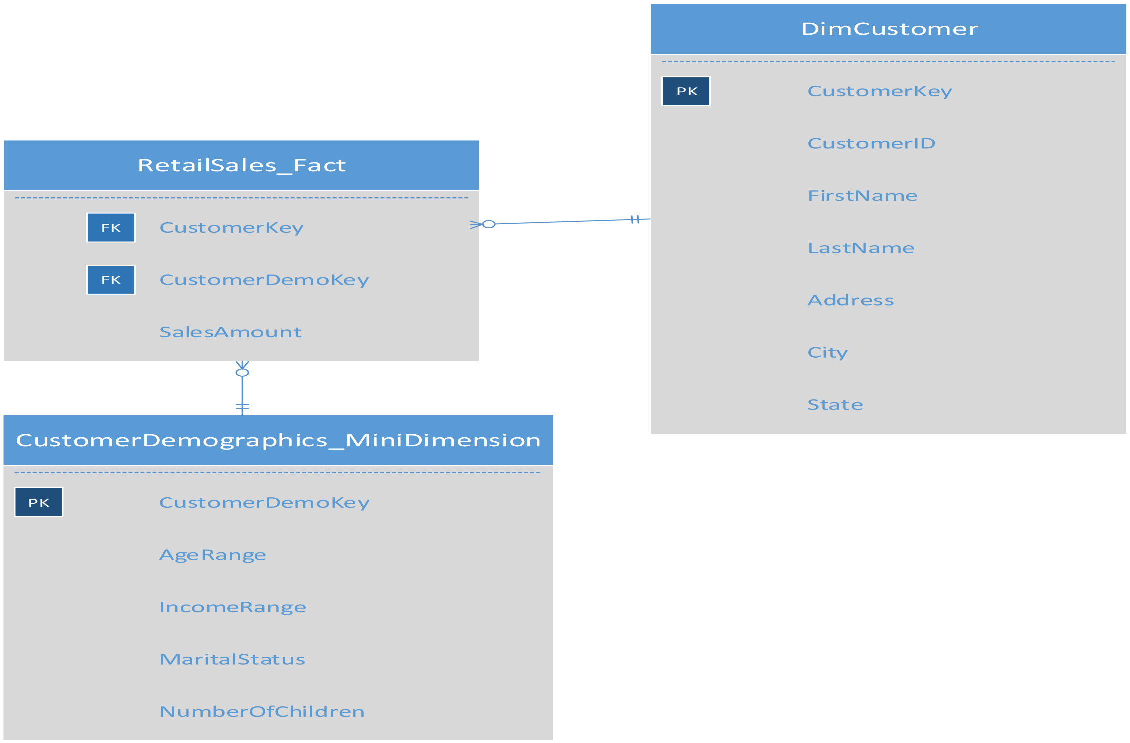
    A causal dimension tells the organization for each transaction recorded about what makes the transaction happen or what triggers the customer to buy the product. It indicates the reason for the customer to make the purchase. For instance, the advertisement, coupon, promotion, and price reduction/rebate are the attributes of a causal dimension.

15. What is allocation? Under what circumstances is allocation used? Give an example.

Allocation refers to assigning the measures to several transactions (the order line items), products, customers, or account. Allocation usually means distributing the overhead. When the organization needs to analyze the net revenue and net profit, the overhead is not included in the price of the product sold, so the allocation should be used. For instace, the shipping charges, supplies, labor cost, customer rebates, or bonuses are the overhead for the organization. The organization needs to assign the shipping charge to each product, and the approach is to force the shipping fee for each order to the lowest level of detail (the order line item), so shipping fees can be calculated by product.

16. What is a mini-dimension? Under what circumstances is a mini dimension used? Give an example including an abbreviated dimensional model.

A mini-dimension is a dimension that has the attributes that are frequently changing, updated, and analyzed. These attributes have the same granularity and are loaded at the same time as the rest of the attributes in a dimension table. These attributes are taken out from this dimension and form a mini-dimension. When the attributes are frequently changing and being updated, mini-dimension should be used. For an example, the customer dimension has attributes such as customer's age, income, marital status, and number of children, which should be the attributes forming a mini-dimension.



17. What is a “late arriving fact”? Give an example of a business case scenario in which late arriving facts may occur.

A “Late arriving fact” is a fact measure that happened in the past, was not recorded in the database, and needs to be added to the fact table at the current time. The new fact is added to the fact table with two time stamps – **occurrence date** and the **load date**. Newly added measure needs to associate with all foreign keys (dimensions’ primary keys). Some dimensions might have type 2 SCD (Slowly Changing Dimension) so relating to the correct primary key (surrogate key) of the dimension table becomes crucial. The way to doing it is to find the date stamp (begin date and end date) in the dimension with the latest begin date less than or equal to the date of the historical fact actually occurring, replacing the natural keys in the fact table for the added row / late arriving data with the proper primary key found in the dimension table. For intstance, the customer purchased a product from a company on 5/1/2015. Transaction was recorded in OLTP then. But, the sales amount was not recorded in the fact table at that time, and needs to be added now, so the “late arriving fact” has already occured.

18. What is a “fact-less” fact table? Give an example of a business case scenario in which a fact-less fact table might be useful.

A “Fact-less” fact table contains one measure that is not measurable. The measure in the factless fact table indicates relative information being collected when there is an event or coverage. For instance, when a student registers a class (this is an event), the demographics about the student, class, faculty’s information are being collected. When a company promotes its products (this is coverage), the details of the product and promotion are collected. In scenarios above, a factless fact table is useful.

19. In data warehouses where both time of day and date are needed, Kimball recommends a dimension for time and a separate dimension for date. Why does he recommend 2 dimensions over the idea of combining date and time into 1 dimension?

The recommendation from Kimball of separating a time dimension from a date dimension is to avoid having too many rows in the date dimension. There are 24 hours a day and 60 minutes per hour which is 1440 minutes per day. According to Kimball, one solution to avoid too many rows in the date dimension is to treat time of the day as a fact, such as number of minutes or seconds if there is no need to analyze the roll up details or filter the time of the day.

However, if there is a need to analyze the time of the day, the solution is to build a separate time dimension with a timeofday Key as the primary key as role-playing dimension referenced roles in the fact table, such as “universal time of day key” and “local time of day key”.

20. What is a “Data Warehouse Bus Matrix”? What are some key elements of a bus matrix? When is a bus matrix primarily used?

A “Data Warehouse Bus Matrix” is the first step to start building a data warehouse from the technical point. It is an architecture defining how a data warehouse can be structured. Some key elements of a bus matrix are business processes (each business process can be considered a data mart), dimensions. In this matrix, the developers or engineers should be able to see which dimension applies to which business processes. Some dimensions will be common dimensions to different business processes/data marts. That means these dimensions can be re-used in different business processes/data marts. From a technical point of view, a bus matrix is primarily used by developers or engineers when logical model design starts. From business point of view, a bus matrix is used to communicate with executives.

Optional Extra Credit + 3 Points

21. Describe ways in which dimensional modeling techniques support resource and cost efficiency in an organization.

    The main purposes of dimensional modeling are usuability and performance. By denormalizing the table, all related attribute values can be extracted from one place (one table). This is efficienct in saving time. From a performance stand point of view, during a dimension load process, joins that are needed become less when a dimensional model has a large size of fact table with simple joins to a few relatively small dimensions. A marjority of the constraints reside in the fact table. A relational optimizer can understand this structure and formulate a query strategy which can greatly improve the speed of most BI queries. That reduces cost in running queries since the measures are pre-aggregated.