1. One of the most important assets of any organization is its data (information). This asset is used for two purposes: operational record keeping and analytical decision making. We capture data in the operational systems; and we analyze data in the data warehousing and business intelligence (DW/BI) systems. According to the Kimball Group, what are the four components of the DW/BI Architecture? Explain each of these four components.

Ans: The four components of Kimball DW/BI architecture are as follows:

a) Operational source systems: These are operational systems that capture the business's transactions. The main priorities of the source systems are processing performance and availability. Operational queries against source systems are narrow, one-record-at-a-time queries.

b) Extract, Transformation, and Load (ETL) System: This part of the DW/BI architecture consists of a work area, instantiated data structures, and a set of processes.
   - Extraction is the first step in the process of getting data into the data warehouse environment. Extracting means reading and understanding the source data and copying the data needed into the ETL system for further manipulation. At this point, the data belongs to the data warehouse.
   - After the data is extracted to the ETL system, there are numerous potential transformations, such as cleansing the data (correcting misspellings, resolving domain conflicts, dealing with missing elements, or parsing into standard formats), combining data from multiple sources, and de-duplicating data. The ETL system adds value to the data with these cleansing and conforming tasks by changing the data and enhancing it.
   - The final step of the ETL process is the physical structuring and loading of data into the presentation area's target dimensional models.

c) Presentation Area to Support Business Intelligence: The DW/BI presentation area is where data is organized, stored, and made available for direct querying by users, report writers, and other analytical BI applications.
   - The data in the presentation area should be presented, stored, and accessed in dimensional (instead of in normalized) schemas, either relational star schemas or OLAP cubes.
   - The presentation area must contain detailed, atomic data. Although the presentation area may contain performance-enhancing aggregated data, it is unacceptable to store only summary data in dimensional models while the atomic data is locked up in normalized models.

d) Business Intelligence Applications: The final major component of the Kimball DW/BI architecture is the business intelligence (BI) application. The term BI application loosely refers to the range of capabilities provided to business users to leverage the presentation area for analytic decision making.
   - A BI application can be as simple as an ad hoc query tool or as complex as a sophisticated data mining or modeling application.
   - Ad hoc query tools, as powerful as they are, can be understood and used effectively by only a small percentage of the potential DW/BI business user population.
   - Most business users will likely access the data via prebuilt parameter-driven applications and templates that do not require users to construct queries directly.

2. The Kimball Group has defined a set of techniques for modeling data in a dimensional way. According to the Kimball Group, what are the four key decisions made during the design of a dimensional model? Explain and give an example for each of these four key decisions.

Ans: Four-Step Dimensional Design Process:

a. Select the business process: Business processes are the operational activities performed by an organization, such as taking an order, processing an insurance claim, registering students for a class, or snapshotting every account each month. Business process events capture performance metrics that translate into facts in a fact table.
b. Declare the grain: Declaring the grain is the pivotal step in a dimensional design. The grain establishes exactly what a single fact table row represents. The grain declaration becomes a binding contract on the design. The grain must be declared before choosing dimensions or facts because every candidate dimension or fact must be consistent with the grain. Atomic grain refers to the lowest level at which data is captured by a given business process. Focus on atomic-grained data because it withstands the assault of unpredictable user queries; rolled-up summary grains are important for performance tuning, but they presuppose the business's common questions. Each proposed fact table grain results in a separate physical table; different grains must not be mixed in the same fact table.
c. Identifying the dimensions: Dimensions provide the "who, what, where, when, why, and how" context surrounding a business process event. Dimension tables contain the descriptive attributes used by BI applications for filtering and grouping the facts. A disproportionate amount of effort is put into the data governance and development of dimension tables because they are the drivers of the user's BI experience.
d. Identifying the facts: Facts are the measurements that result from a business process event and are almost always numeric. A single fact table row has a one-to-one relationship to a measurement event as described by the fact table's grain. Thus a fact table corresponds to a physical observable event. Within a fact table, only facts consistent with the declared grain are allowed. For example, in a retail sales transaction, the quantity of a product sold and its extended price are good facts, whereas the store manager's salary is disallowed.

Examples: Design a dimensional data model to analyze daily quantity-on-hand inventory levels by product and store.

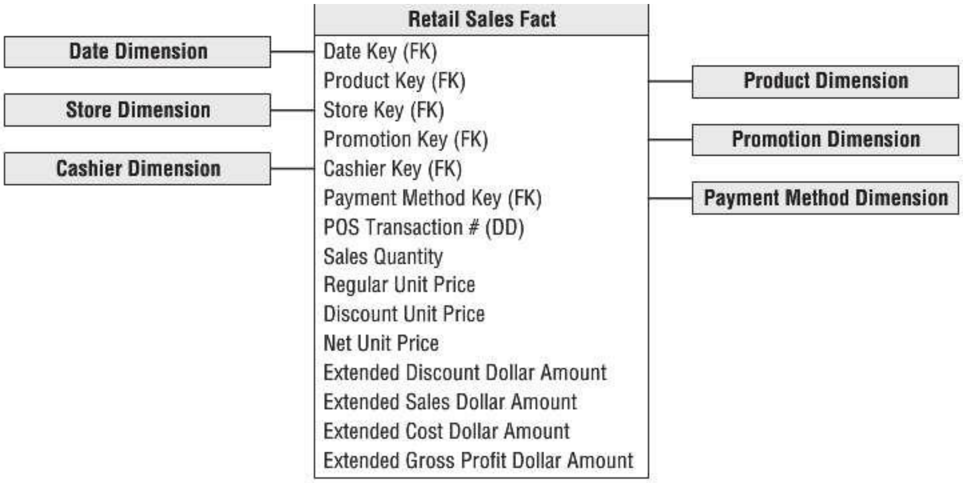
Business Process: Periodic snapshotting of retail store inventory.

Grain: Daily inventory level for each product in each store.

Dimensions:

Facts: Quantity on hand

3. Suppose that you are a dimensional data modeler for a retail store. In a retail business, typically the customers are anonymous. Thus, you have designed and created a retail sales schema without a customer dimension as shown below.



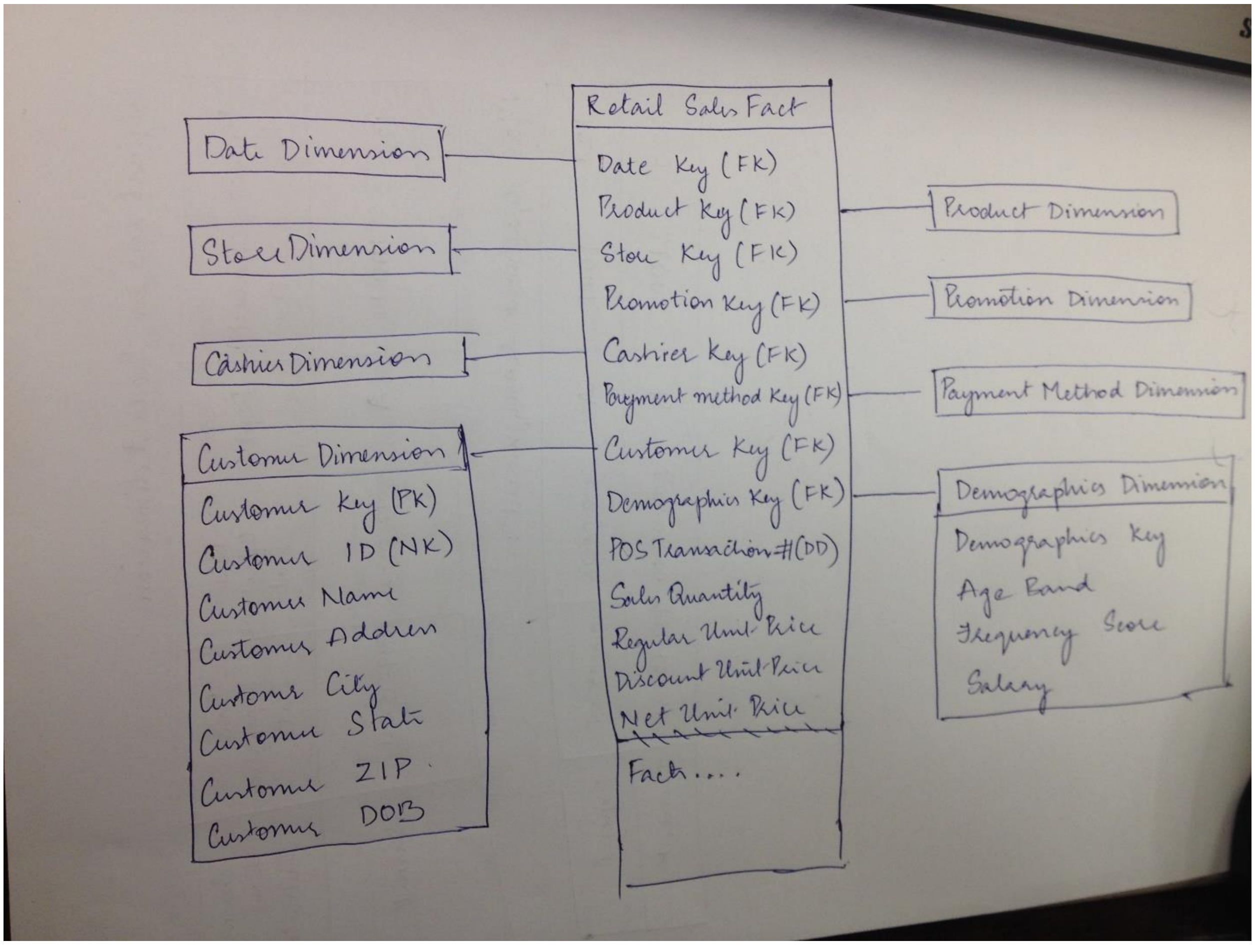
Several years after the rollout of the retail sales schema, the retailer would like to know the demography of its frequent customers. This way, the retailers would be able to analyze shopping patterns by a geographic, demographic, and other differentiating shopper characteristics. Consequently, the retailer would like to implement a frequent shopper program. As a data modeler, please explain the steps to extend the above schema so that you can incorporate the frequent shopper program. Show a revised dimensional schema that illustrates your design.

Ans:

- To implement the frequent shopper program, we need to have customer details as well, and hence we have to include a Customer Dimension.
- This dimension must have the following descriptive attributes: ID, Name, Address, City, State, ZIP, Age, Salary and Frequency score.
- However, observing the attributes, we can infer that a few attributes such as age, salary and frequency score can change quite frequently.
- Hence, we need to use SCD type 4 by splitting these attributes from the Customer Dimension and forming another dimension called Demographic mini dimension. There would be one row in the mini-dimension for each unique combination of age, purchase frequency score, and income level encountered in the data, not one row per customer. With this approach, the mini-dimension becomes a set of demographic profiles.
- Since, there could be many number of combinations possible in the mini-dimension, we could convert these attributes into predefined ranges in order to reduce the number of rows in the mini-dimension table. In other words, the attributes in the mini-dimension are typically forced to take on a relatively small number of discrete values. Below are the sample rows in this mini dimension.

Demographics Key	Age Band	Frequency Score	Salary
1	21 - 25	Low	< $30,000
2	21 – 25	Medium	< $30,000
3	21 – 25	High	< $30,000
4	21 - 25	Low	$30, 000 - $40, 000
5	21 – 25	Medium	$30, 000 - $40, 000
6	21 - 25	High	$30, 000 - $40, 000
…	…	…	…
100	35-40	Low	< $30, 000
101	35-40	Medium	< $30, 000
…	…	…	…

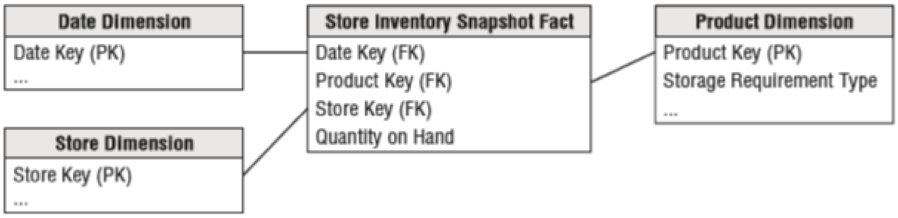
- Revised Dimensional Schema:



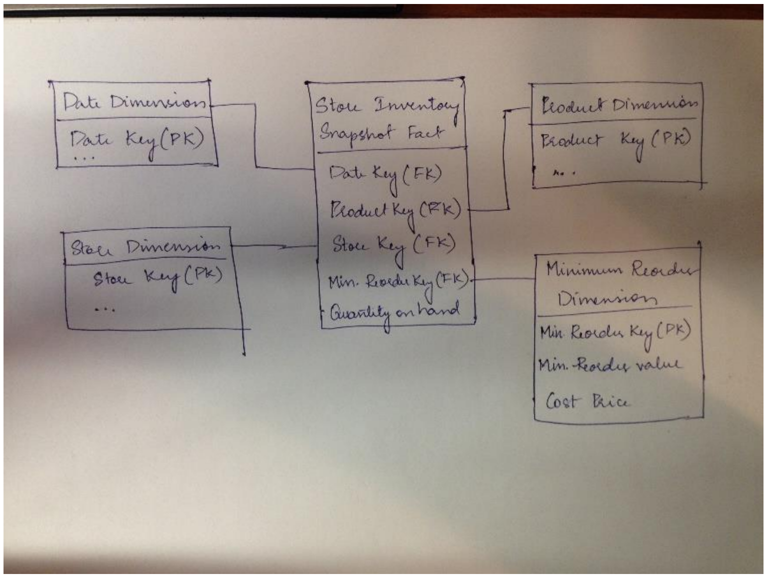
4. Suppose that you work as a dimensional data modeler for a large nation-wide retailer. You are asked to design a dimensional data model to analyze daily quantity-on-hand inventory levels by product and store. You follow Kimball's four-step dimensional process to design the schema.
   a. What is the business process that you are going to model?
   b. What is the grain of your model?
   c. What are the dimensions for this model?
   d. What is the fact (numeric measure) that you are going to capture?
   e. Draw a sample schema to illustrate your dimensional data model.
   f. To enhance the inventory analysis, the retailer wants to store the retail cost and minimum reorder quantity for each product in the data warehouse. Assuming that the cost and minimum reorder quantity varies for a product by store, where would you store the retail cost and minimum reorder quantity data items in your dimensional data model?

Ans:

- Business Process: Periodic snapshotting of retail store inventory.
- Grain: Daily inventory level for each product in each store.
- Dimensions: Product, Store and date
- Facts: Quantity on hand
- Sample Schema:



- If the minimum reorder quantity and retail cost varies for a product by store, then we cannot include them in any of the three dimension tables or even in the inventory fact table. We have to add a new dimension with contains rows with a unique combination of minimum reorder value and Cost price.



5. The dimension table attribute values are relatively static; but, they are not fixed forever. Over time, some attribute values change, although rather slowly. When an attribute value changes in the operational world, how will you respond to the change in the dimensional model? As a dimensional data modeler, you need strategies to deal with slowly changing attributes within dimension tables.
   a. Please explain in what situation you want to use the Slowly Changing Dimension Type 2.
   b. Suppose that a product with product ID 1001 is initially assigned to Department A. However, as of today, a new marketing manager decided to relocate this product to Department B. What changes to the Product dimension table do you need to make to implement the Slowly Changing Dimension Type 2?

Ans: There are several ways for dimensional models to deal with changes in attribute values in the operational world gracefully. Type 1 to Type 7 methods of the slowly changing dimensions provide for this handling. Using these Types, one can deal with business requirements appropriately. For example, if only the latest value of an attribute is needed to be preserved, one could use Type 1 (overwrite) method. If history needs to be preserved methods like Type 2 could be used.

a. Type 2 - Add New Row with updated attribute values is the predominant technique for tracking historically significant attributes. In this approach a new row is inserted in the dimension table when an attribute gets a new value.

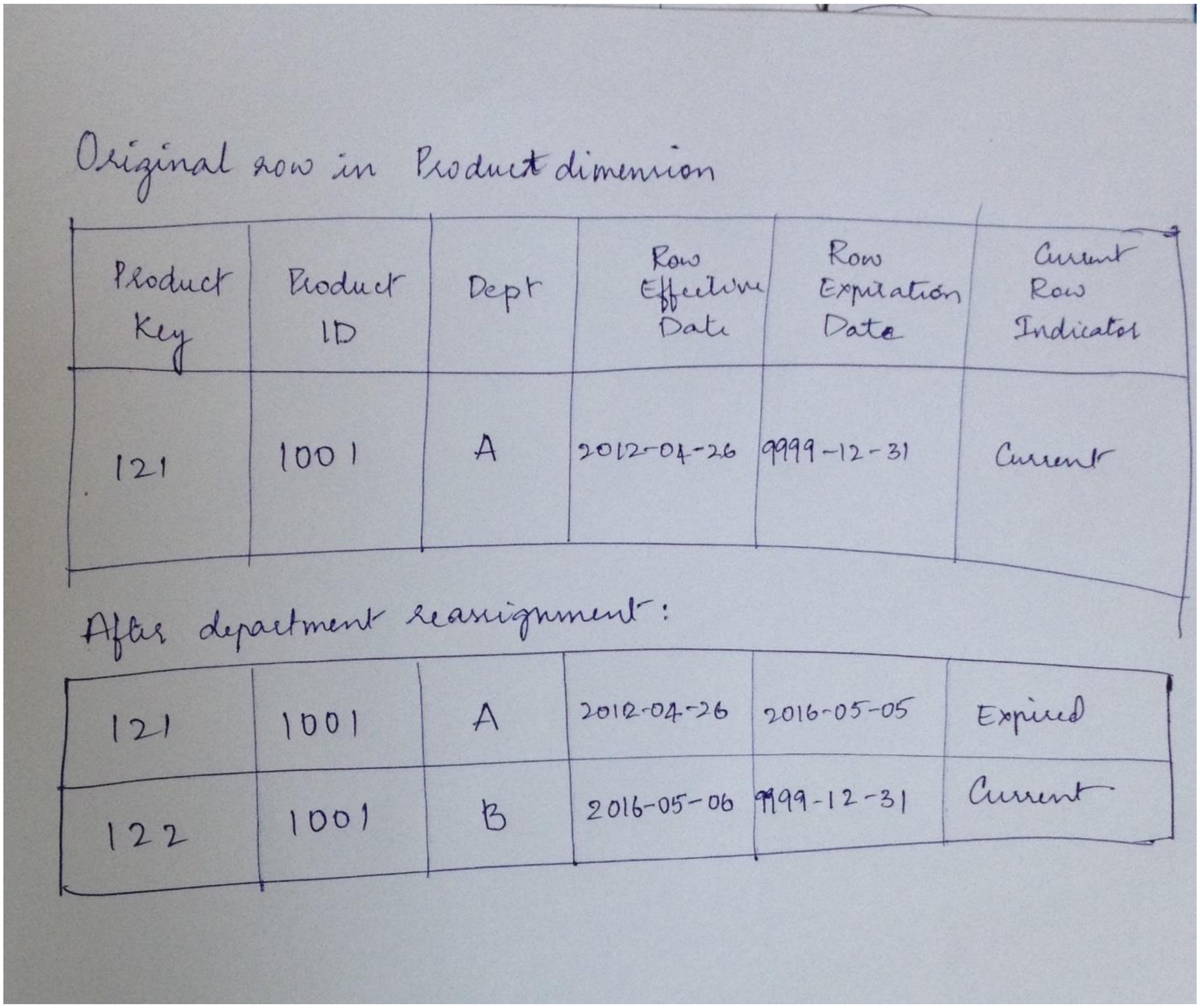
Type2 is used in the following situations:

- Accurately track slowly changing dimension attributes.
- When the business is not absolutely certain regarding the SCD business rules for an attribute.
- We want to support the illusion of Type 1 over-write while still preserving historical information.
- We do not want to modify the Fact tables every time there is a change in dimensional attribute value.
- We do not want to recompute the Aggregation/OLAP cubes every time there is a change in the dimensional attribute value.

Advantages of using Type 2 method:

- Type 2 responses perfectly partition or segment history to account for changes to dimensional attributes. Reports summarizing pre-change facts look identical whether the report is generated before or after the type 2 change.
- If one constrains on the department attribute, the two product profiles are differentiated.
- If one constrains on the product description, the query automatically fetches both product dimension rows and automatically joins to the fact table for the complete product history.
- If you need to count the number of products correctly, then one would just use the SKU natural key attribute as the basis of the distinct count (rather than the surrogate key).
- The effective and expiration dates support precise time slicing of the dimension; however, there is no need to constrain on these dates in the dimension table to get the right answer from the fact table.
- One can reliably use a BETWEEN command to find the dimension rows that were in effect on a certain date.

b. Figure below represents the situation when the Department value is “A” and Figure 2 represents a change to Department value “B” for product ID 1001 is made.



Original row in Product dimension

Product Key	Product ID	Dept	Row Effective Date	Row Expiration Date	Current Row Indicator
121	1001	A	2012-04-26	9999-12-31	Current

After department reassignment:

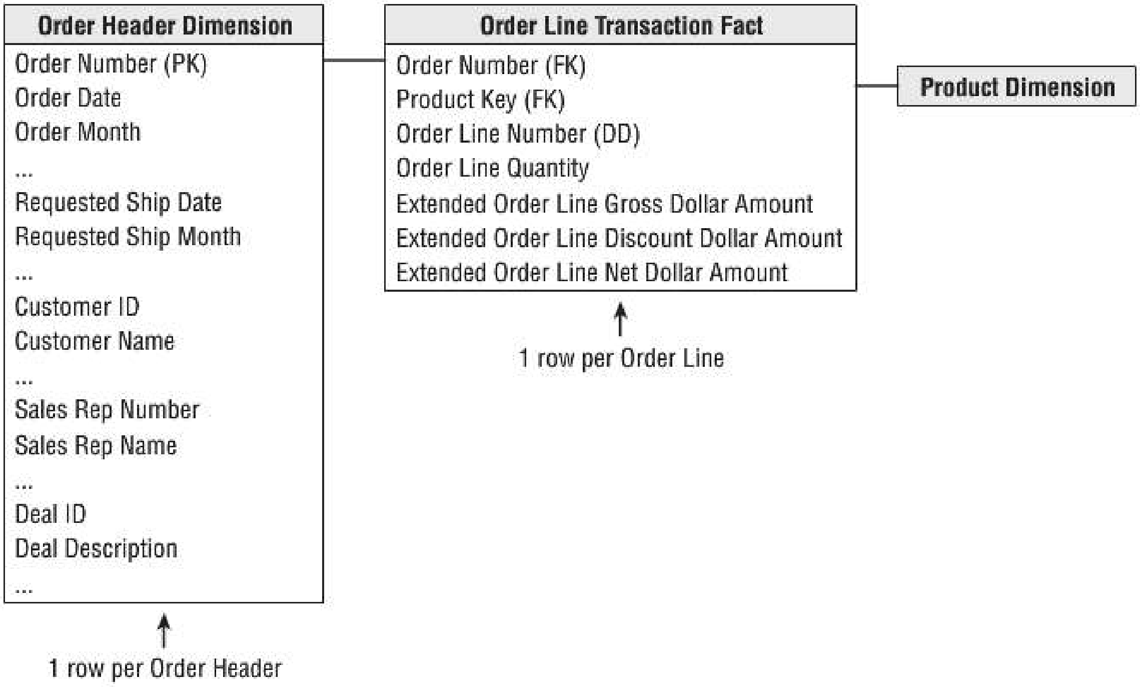
121	1001	A	2012-04-26	2016-05-05	Expired
122	1001	B	2016-05-06	9999-12-31	Current

As can be seen from the diagram, the following changes are made to the product dimension:

- A new row is inserted with a new surrogate key value in the primary key field
- The new row gets most of its data from the original row
- New Row Department is set to the new department “B”
- Row effective date for the new row is set to the date of change
- Row expiration date for the new row is set to 9999-12-31
- Row expiration date for the old row is set to (date of change-1)
- Current Row indicator for the new row is set to “Current”

8 Current Row indicator for the old row is set to “Expired”

6. Consider the following schema. Explain why this schema was not designed properly. Show a better new schema that is equivalent to this schema.

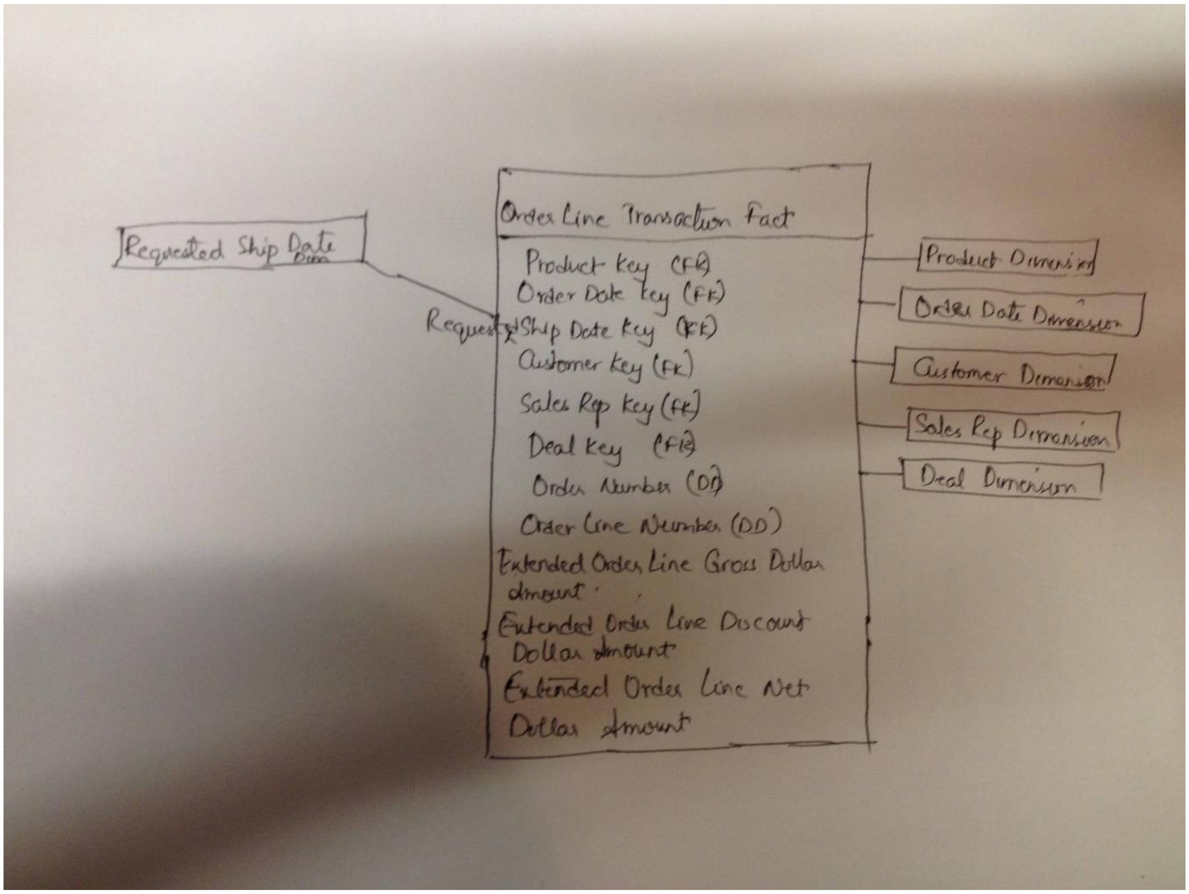


Ans:

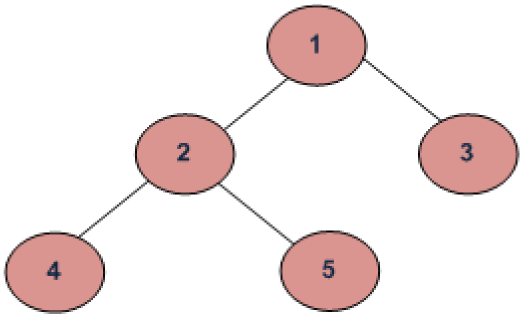
- The operational order header is virtually replicated in the dimensional model as a dimension.
- The header dimension contains all the data from its operational equivalent. The natural key for this dimension is the order number.
- Although this design accurately represents the header/line relationship, there are obvious flaws. The order header dimension is likely very large, especially relative to the fact table itself. If there are typically five line items per order, the dimension is 20 percent as large as the fact table. With this design, you would add one row to the dimension table and an average of five rows to the fact table for every new order. Any analysis of the order's interesting characteristics, such as the customer, sales rep, or deal involved, would need to traverse this large dimension table.

Solution:

- Bring the dimensionality of the order header down to the order line.
- This model represents the data relationships from the order header/line source system. But we've abandoned the operational mentality surrounding a header file. The header's natural key, the order number, is still present in our design, but it's treated as a degenerate dimension.


Requested Ship Date Dim
Order Line Transaction Fact
Product key (FK)
Order Date key (FK)
Requested Ship Date Key (FK)
Customer key (FK)
Sales Rep key (FK)
Deal key (FK)
Order Number (DD)
Order Line Number (DD)
Extended Order Line Gross Dollar Amount
Extended Order Line Discount Dollar Amount
Extended Order Line Net Dollar Amount
Product Dimension
Order Date Dimension
Customer Dimension
Sales Rep Dimension
Deal Dimension

7. Suppose that a large organization has the following rollup structure (see the diagram below). Create a map bridge table for this organization that shows sample rows. The grain of this bridge table is each path in the tree from a parent to all the children below that parent. A row must be constructed from each possible parent to each possible child, including a row that connects the parent to itself.



Ans:

Parent Organization Key	**Child Organization Key**	**Depth From Parent**	**Highest Parent Flag**	**Lowest Child Flag**
1	1	0	TRUE	FALSE
1	2	1	TRUE	FALSE
1	3	1	TRUE	TRUE
1	4	2	TRUE	TRUE
1	5	2	TRUE	TRUE
2	2	0	FALSE	FALSE
2	4	1	FALSE	TRUE
2	5	1	FALSE	TRUE
3	3	0	FALSE	TRUE
4	4	0	FALSE	TRUE
5	5	0	FALSE	TRUE