Advanced Dimensional Modeling Concepts

Time Dimension

- Time is a unique & powerful dimension in every DM & EDW
- Time dimension is very special & should be treated differently from other dimensions
- Example of a Star Schema
    - Fact table records daily orders received by a manufacturing company
    - Time dimension designates calendar days

FAQs About Time Dimension

- **Why can’t I just leave out the time dimension?**
  - Dimension tables serve as the source of constraints and as the source of report row headers
  - A data mart is only as good as its dimension tables
  - SQL provides some minimal assistance in navigating dates
  - SQL certainly doesn't know anything about your corporate calendar, your fiscal periods, or your seasons

FAQs About Time Dimension

- **If I have to have a time dimension, where do I get it?**
  - Build it in a spreadsheet
  - Some data marts additionally track time of day to the nearest minute or even the nearest second. For these cases separate the time of day measure out as a separate "numeric fact." It should not be combined into one key with the calendar day dimension. This would make for an impossibly large time table.

Time Dimension

- **Guard against incompatible rollups like weeks & months**
- **Separate fact tables denominated in weeks and months should be avoided at all costs**
- **Uniform time rollups should be used in all the separate data marts**
- **Daily data rolls up to every possible calendar**

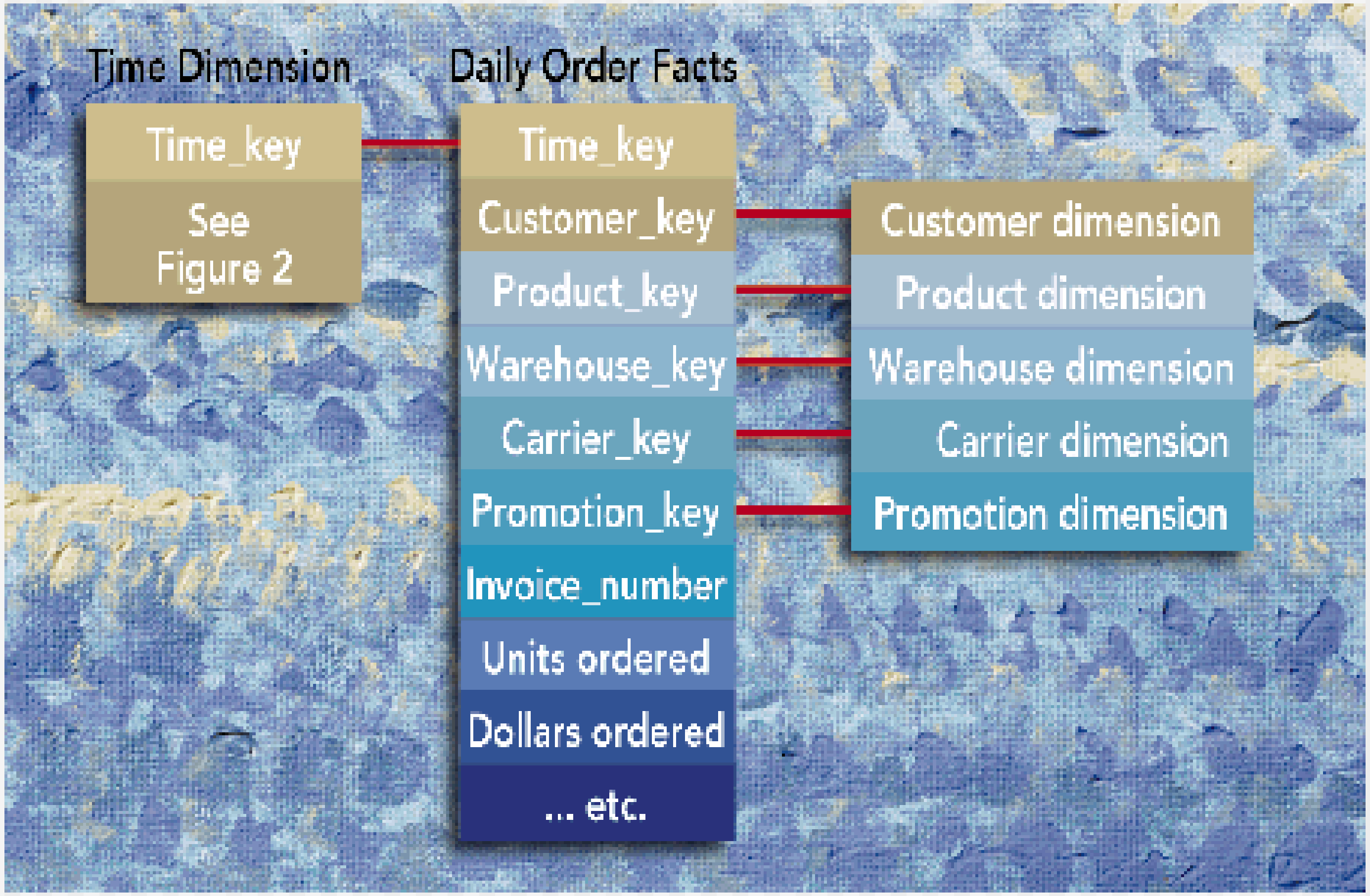
Time Dimension

- **Be careful about aggregating non-additive facts wrt time**
- **Examples: inventory levels & account balances**
- **We need “average over time”**
- **SQL AVG ?**
- **SQL supports no “avgperiodsum” operator**

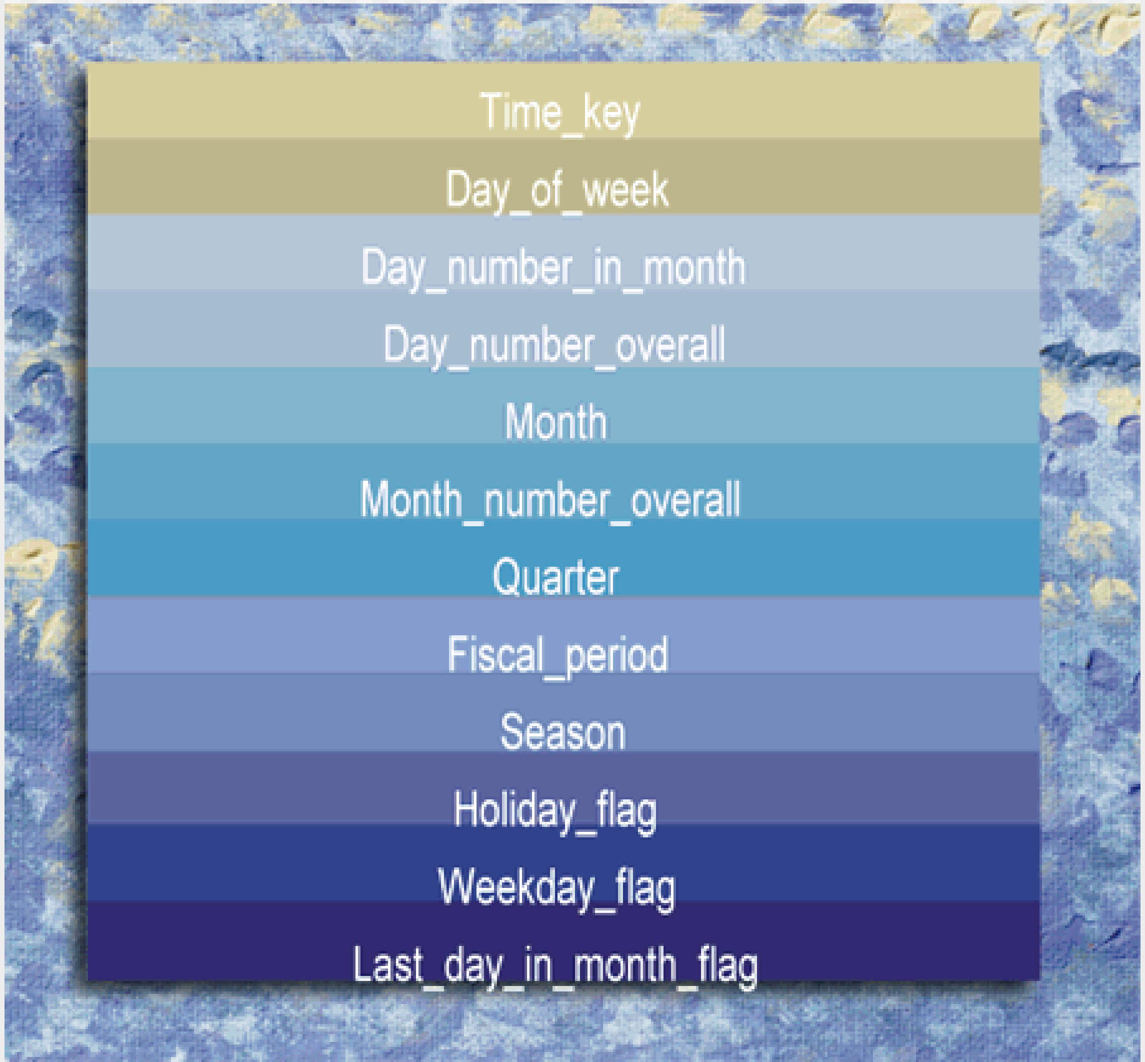
Time Dimension

- **Data mart around individual transactions or around month-end snapshots?**
- **Transaction intensive businesses like insurance**
- **What is the average length of time between the original claim and the first payment to the claimant?**
- **Record of the individual transactions is a poor basis for creating a month-end report for management**
- **Build two versions of the data mart: a tx version & a monthly snapshot version**

Time Dimension

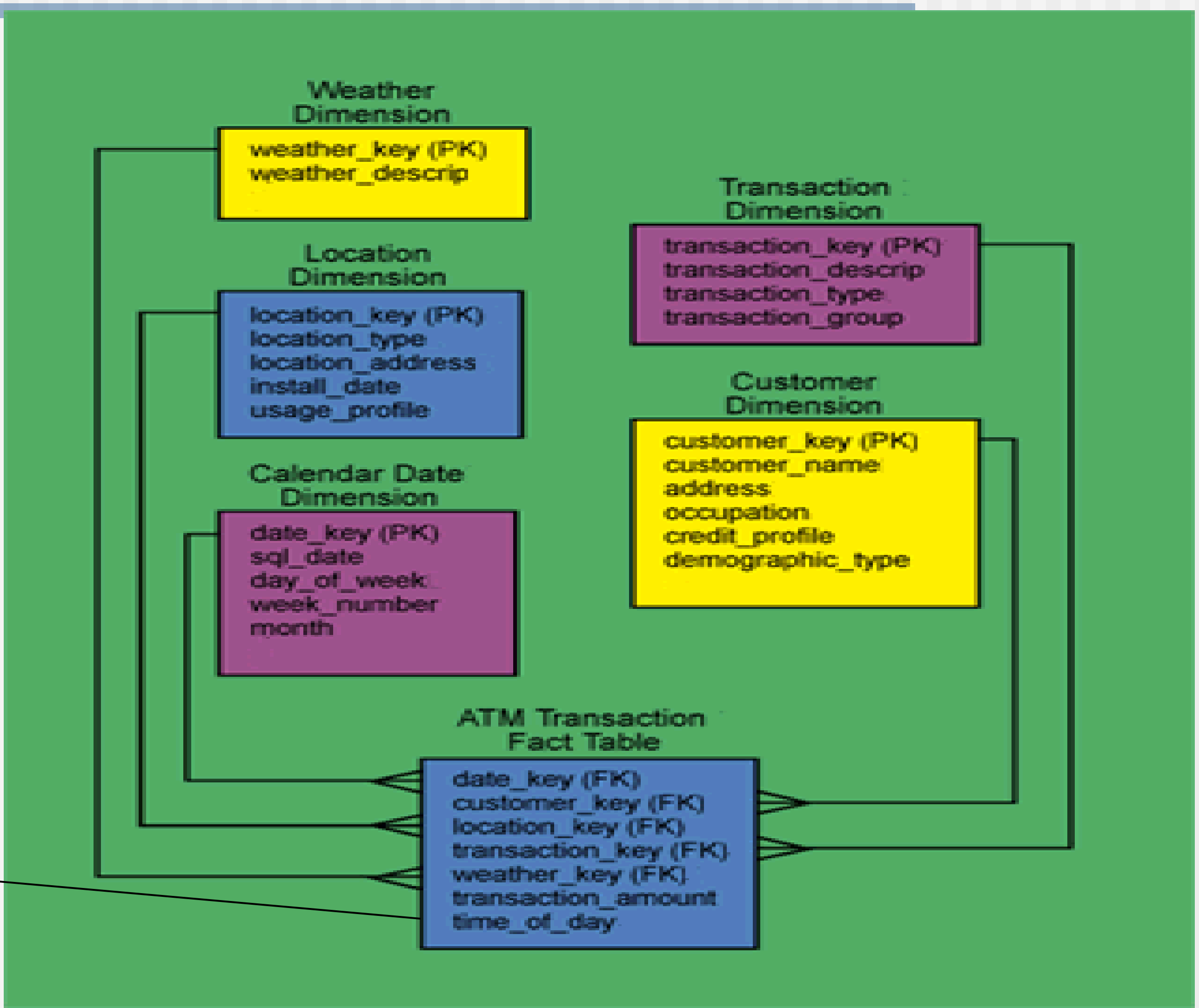


Time Dimension



Time_key
Day_of_week
Day_number_in_month
Day_number_overall
Month
Month_number_overall
Quarter
Fiscal_period
Season
Holiday_flag
Weekday_flag
Last_day_in_month_flag

Time Dimension



Time of Day Dimension

Time_key(PK)
Morning Rush Hour
Mid Morning
Lunch Hour
Mid Afternoon
Afternoon rush Hour

Multi-valued Dimensions

- **Declaring grain of the fact table is one of the important design decisions**
- **Grain declares the exact meaning of a single fact record**
- **If the grain of the FT is clear, choosing Dimensions becomes easy**

Multi-valued Dimensions

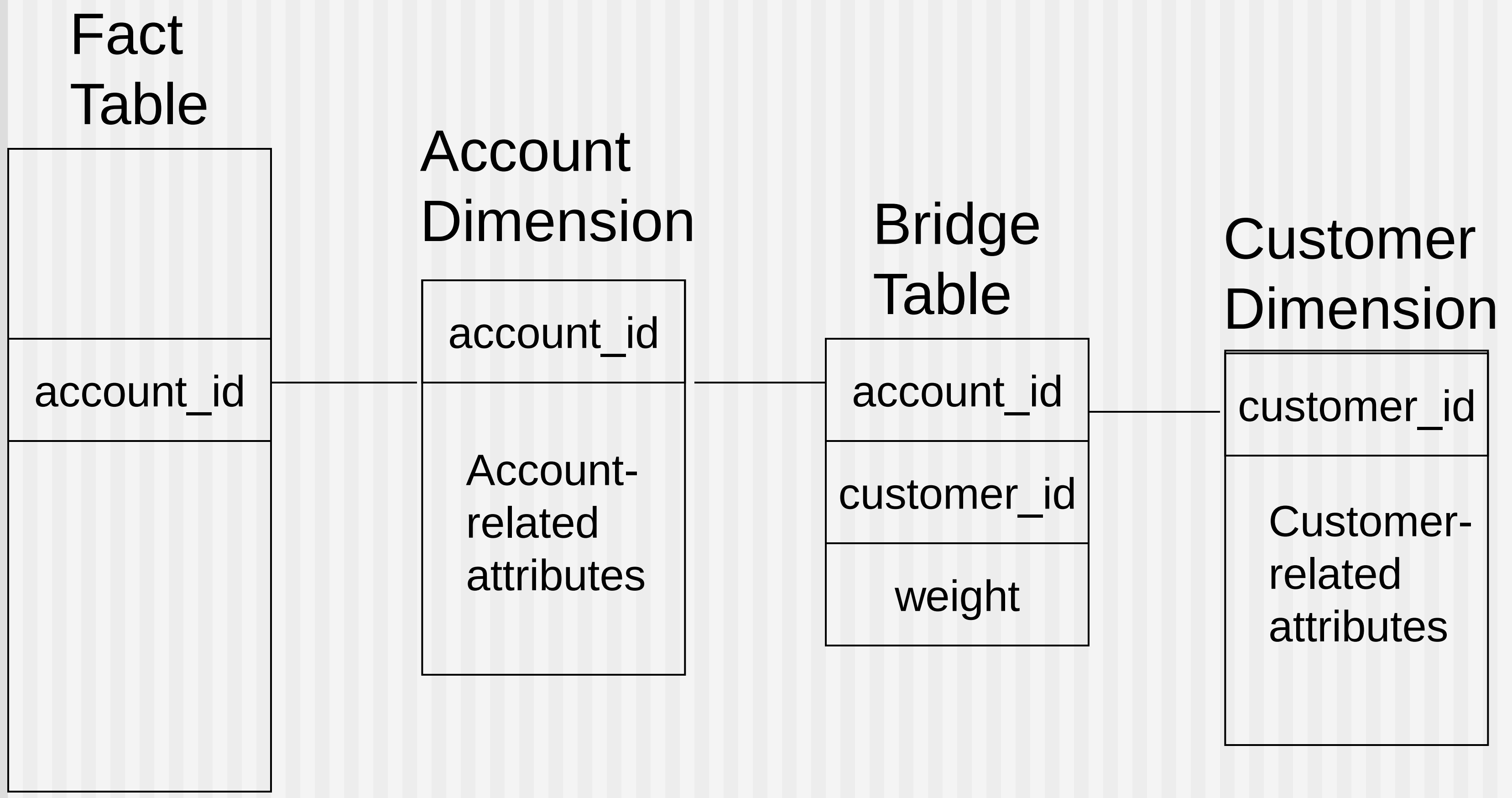
- **John & Mary Smith a single household**
- **John has a current account**
- **Mary has a savings account**
- **John & Mary have a joint current account, & credit card**
- **An account can have one, two or more customers associated with it**

Multi-valued Dimensions

- **Customer as an account dimension attribute?**
- **Doing so violates the granularity of the dimension table as more than one customer could be associated with an account**
- **Customer as an additional dimension in the FT?**
- **Doing so violates the granularity of the FT (one row per account per month)**
- **Classic example of a multi-valued dimension**
- **How to model multi-valued dimensions?**

Bridge Tables

- **Account to Customer BRIDGE table**



Multi-valued Dimensions

- **In the classical Grocery Store Sales Data Mart, the following dimensions are obvious for the daily grain:**
  - **Calendar Date**
  - **Product**
  - **Store**
  - **Promotion (assuming the promotion remains in effect the entire day)**
- **What about the Customer & Check-out clerk dimensions?**
- **Many values at the daily grain!**

Multi-valued Dimensions

- **We disqualified the customer and check-out clerk dimension**
- **Many dimensions can get disqualified if we are dealing with an aggregated FT**
- **The more the FT is summarized, the fewer no. of dimensions we can attach to the fact records**
- **What about the converse?**
- **The more granular the data, the more dimensions make sense!!**

Multi-valued Dimensions

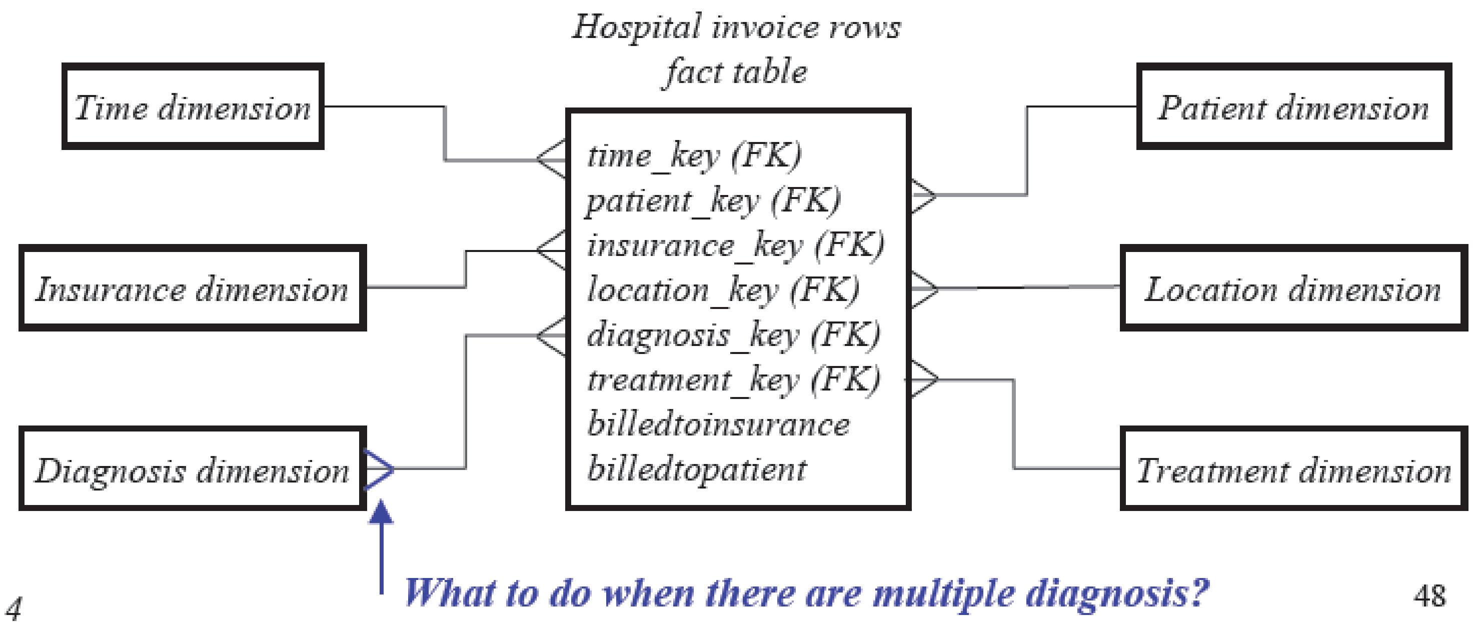
- **Single-valued dimensions are welcome**
- **Any legitimate exceptions?**
- **If yes, how to handle them?**
- **Example: Healthcare Billing**
- **Grain is individual line item on a doctor/hospital bill**
- **The individual line items guide us through to the dimensions**

Multi-valued Dimensions: Healthcare Example

- **Dimensions**
    - **Calendar Date (of incurred charges)**
    - **Patient**
    - **Doctor (usually called ‘provider’)**
    - **Location**
    - **Service Performed**
    - **Diagnosis**
    - **Payer (insurance co., employer, self)**
- **In many healthcare situations, there may be multiple values for diagnosis**
- **Really sick people having 10 different diagnoses!!**
- **How to model the Diagnosis Dimension?**

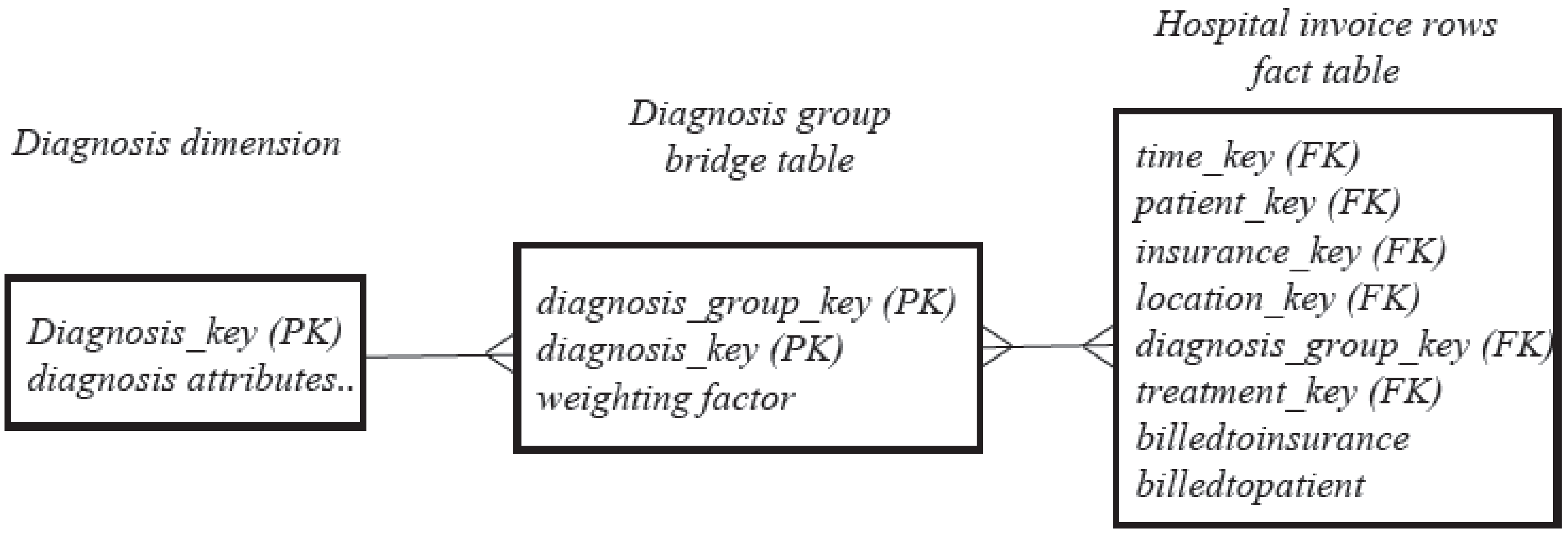
Many-to-many dimensions

- Sometimes the grain of the fact table is clear and easily understood
- Also the dimensions are obvious and not controversial



What to do when there are multiple diagnosis?

Bridge Tables in the logical design



- The fact table now contains a diagnosis group key
- If a patient has three diagnoses the bridge table contains three rows for him; if weighting factors are used, the three factors for this patiént should add to 1
- The diagnosis table contains a complete description of each diagnosis

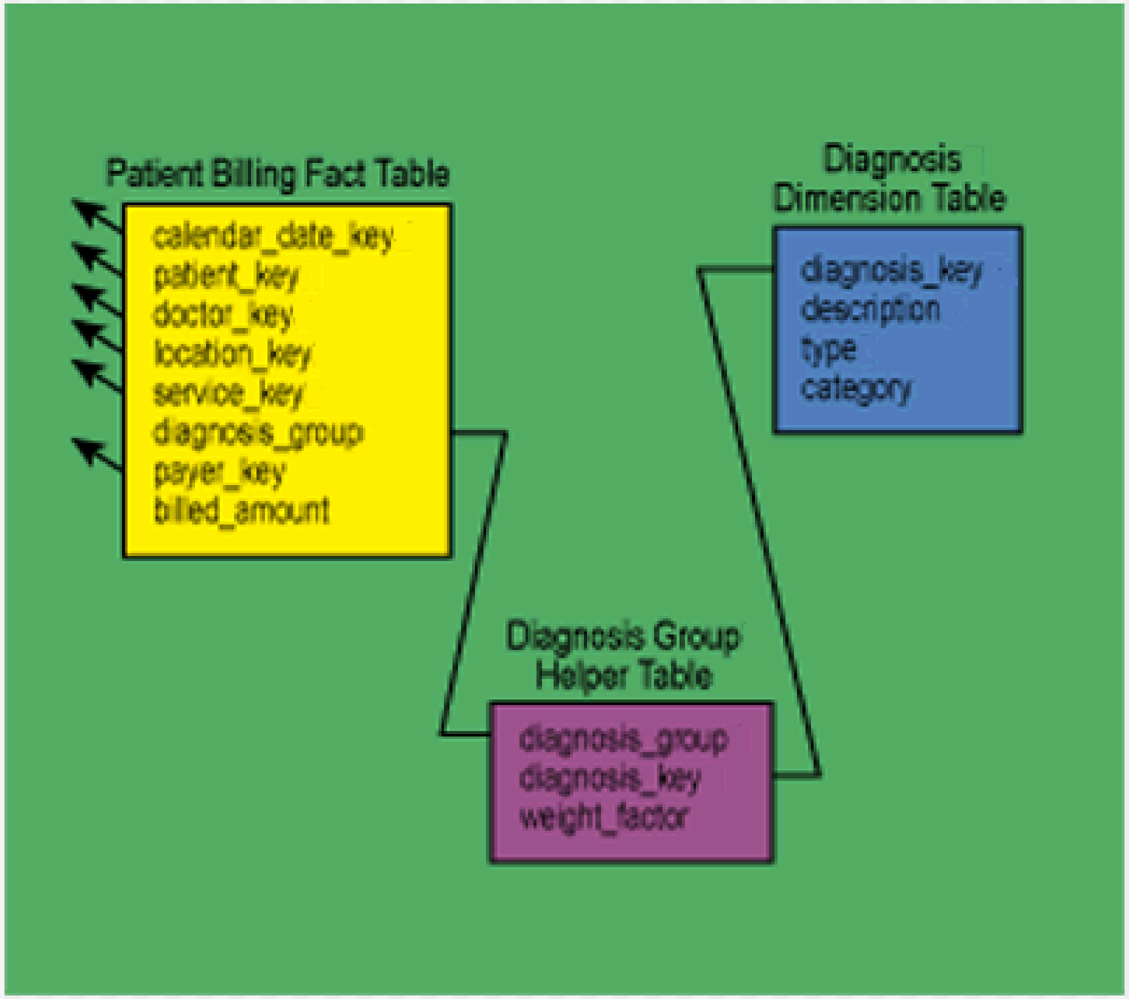
Multi-valued Dimensions: Healthcare Example

1. **Disqualify the Diagnosis Dimension because it is MV**
   - **Easy way out but not recommended**
2. **Choose primary diagnosis & ignore others**
   - **Primary or admitting diagnosis**
   - **Modeling problem taken care of, but is the diagnosis information useful in any way?**
3. **Extend the dimension list to have a fixed number of Diagnosis dimensions Location**
   - **Create a fixed number of additional Diagnosis dimension slots in the fact table key**
   - **There will be some complicated example of a very sick patient who exceeds the number of Diagnosis slots you have allocated**
   - **Multiple separate Diagnosis dimensions can not be queried easily**
   - **If "headache" is a diagnosis, which Diagnosis dimension should be constrained?**
   - **Avoid the multiple dimensions style of design as logic across dimension is notoriously slow on relational databases**

Multi-valued Dimensions: Healthcare Example

4. **Helper Table Approach**
   - Place a "helper" table between the Diagnosis dimension and the fact table
   - The Diagnosis key in the fact table is changed to be a Diagnosis Group key
   - The helper table in the middle is the Diagnosis Group table
   - It has one record for each diagnosis in a group of diagnoses
   - If I walk into the doctor's office with three diagnoses, then I need a Diagnosis Group with three records in it
   - Either build these Diagnosis Groups for each individual or a library of "known" Diagnosis Groups

Multi-valued Dimensions: Healthcare Example



A helper table for an open-ended number of diagnoses

Multi-valued Dimensions: Healthcare Example

4. **Helper Table Approach**
   - The Diagnosis Group table contains a very important numeric attribute: the weighting factor
   - The weighting factor allows reports to be created that don't double count the Billed Amount in the fact table
   - For instance, if you constrain some attribute in the Diagnosis dimension such as "Contagious Indicator" with the values Contagious and Not Contagious, then you can group by the Contagious Indicator and produce a report with the correct totals. To get the correct totals, we must multiply the Billed Amount by the associated weighting factor
   - Assign the weighting factors equally within a Diagnosis Group. If there are three diagnoses, then each gets a weighting factor of 1/3
   - All weight factors in a Diagnosis Group always add up to 1

Multi-valued Dimensions: Healthcare Example

4. **Helper Table Approach**
    - Deliberately omit the weighting factor and deliberately double count the same report grouped by Contagious Indicator
    - An "impact report" is produced, that shows the total Billed Amount implied partially or fully by both values of Contagious Indicator.
    - Correctly weighted report is the most common and makes the most sense
    - Impact report is interesting and is requested from time to time. Such an impact report should be labeled so that the reader is not misled by any summary totals.

Multi-valued Dimensions

4. **Helper Table Approach**
   - Helper table clearly violates the classic star join design where all the dimension tables have a simple one-to-many relationship to the fact table
   - But it is the only viable solution to handling MV dimensions
   - We can preserve the star join illusion in end-user interfaces by creating a view that prejoins the fact table to the helper table
   - Other Applications:
     - Retail Banks
     - Standard Industry Classification

Helper Tables for Hierarchies

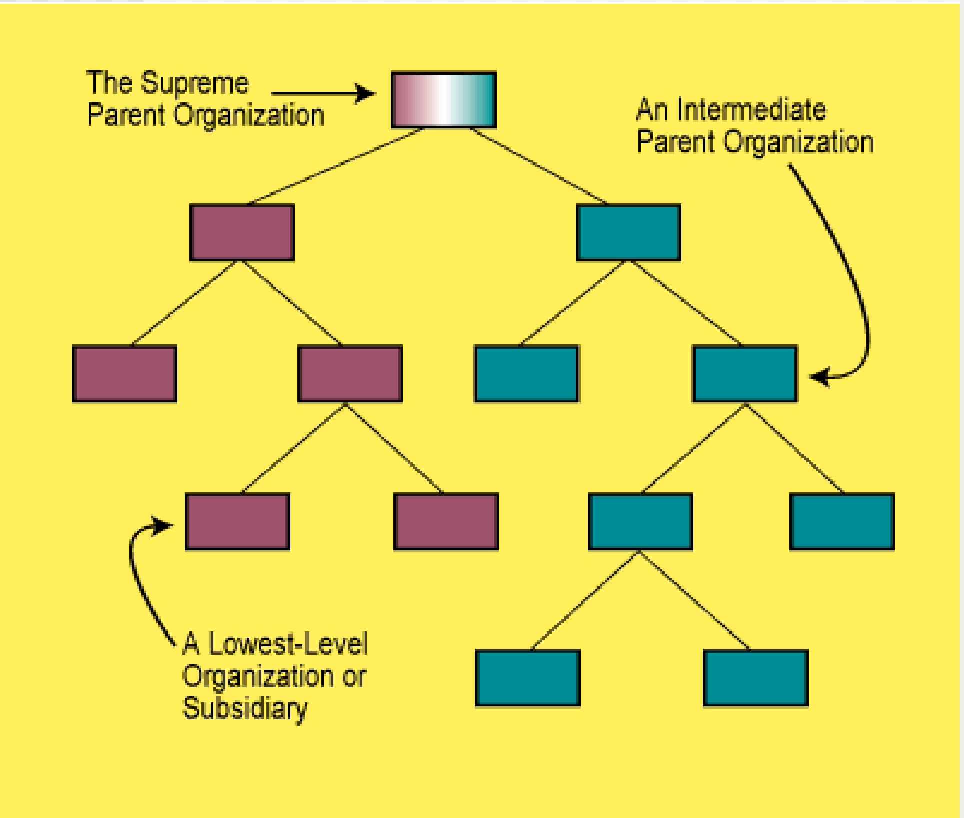
- **Helper Tables are useful for handling M:M relationship between FT & DT**
- **One more real world situation which can be modeled using helper tables**
- **Dimension with complex variable-depth hierarchy**

Helper Tables for Hierarchies

- **Constant depth hierarchies:**
  - **Store all levels of hierarchy in denormalized dimension table**
    - **The preferred solution in almost all cases!**
  - **Create "snowflake" schema with hierarchy captured in separate outrigger table**
    - **Only recommended for huge dimension tables**
    - **Storage savings have negligible impact in most cases**

Variable Depth Hierarchies

- Examples
  - **Corporate organization chart**

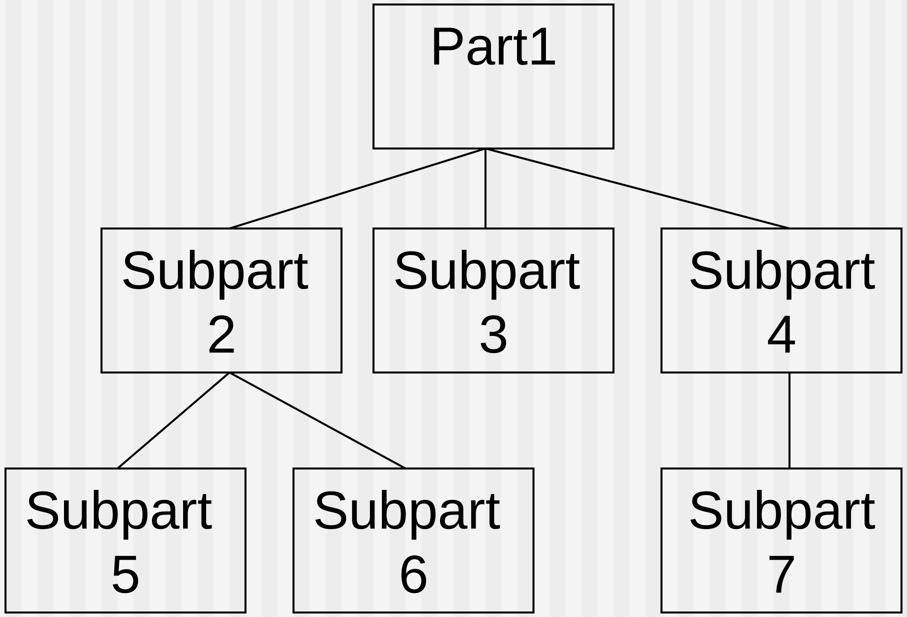

The Supreme Parent Organization
An Intermediate Parent Organization
A Lowest-Level Organization or Subsidiary

Consulting Invoices DM:

- **Consulting services are sold at different organizational levels**
- **Need for reports that show consulting sold not only to ind. Departments, but also to division, subsidiaries and overall enterprise**
- **The report must still add up the separate consulting revenues for each organization structure**

Variable Depth Hierarchies

- Examples
  - **Parts composed of subparts**

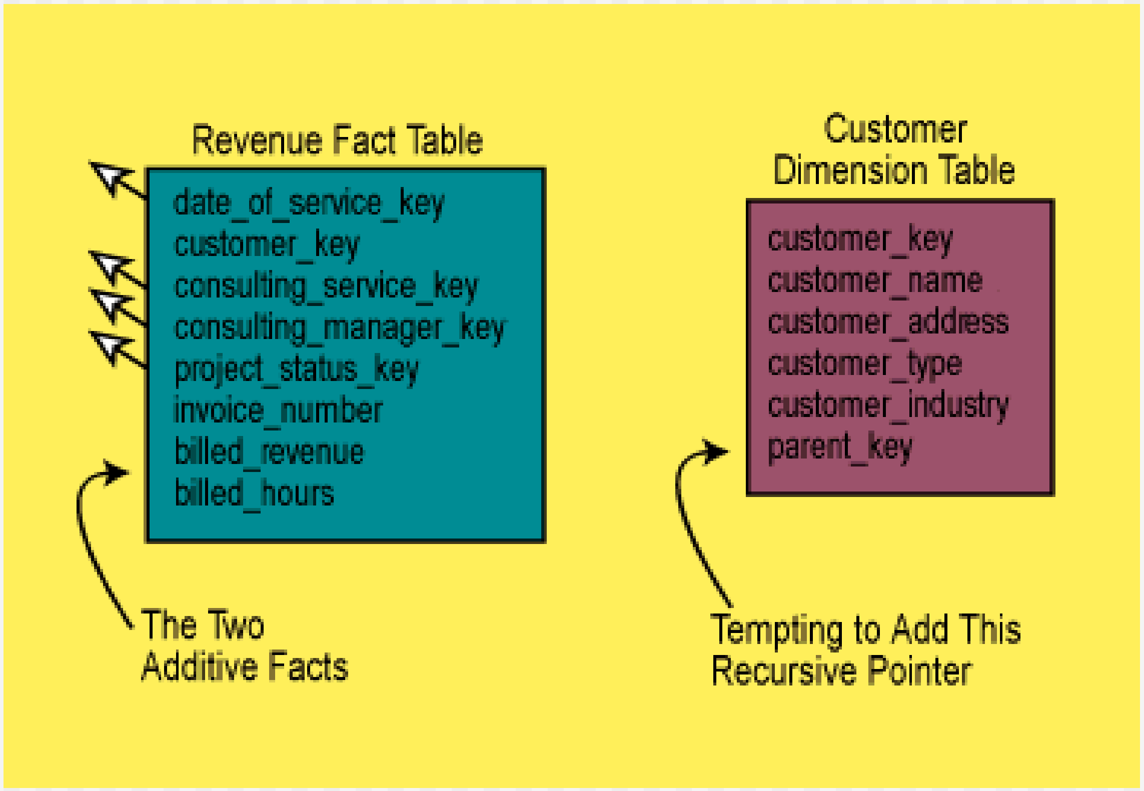


Handling Hierarchies

- Solutions for variable-depth hierarchies?
  - Creating recursive foreign key to parent row is a possibility
    - Employee dimension has “boss” attribute which is FK to Employee
    - The CEO has NULL value for boss
    - This approach is not recommended
    - Cannot be queried effectively using SQL
  - Alternative approach: bridge table

Handling Hierarchies

- **Creating recursive foreign key to parent row is a possibility**



Handling Hierarchies

- **Creating recursive foreign key to parent row is a possibility**
  - **Employee dimension has “boss” attribute which is FK to Employee**
  - **The CEO has NULL value for boss**
  - **This approach is not recommended**
  - **Cannot be queried effectively using SQL**
  - **ORACLE does provide “start with” & “connect by”**

Handling Hierarchies

- **Creating recursive foreign key to parent row is a possibility**
  - **Employee dimension has “boss” attribute which is FK to Employee**
  - **The CEO has NULL value for boss**
  - **This approach is not recommended**
  - **Cannot be queried effectively using SQL**
  - **ORACLE does provide “start with” & “connect by”**

Start With & Connect By

- **The start with connect by clause can be used to select data that has a hierarchical relationship**

```
create table test_connect_by (
parent number,
child number,
constraint uq_tcb unique (child) );
```

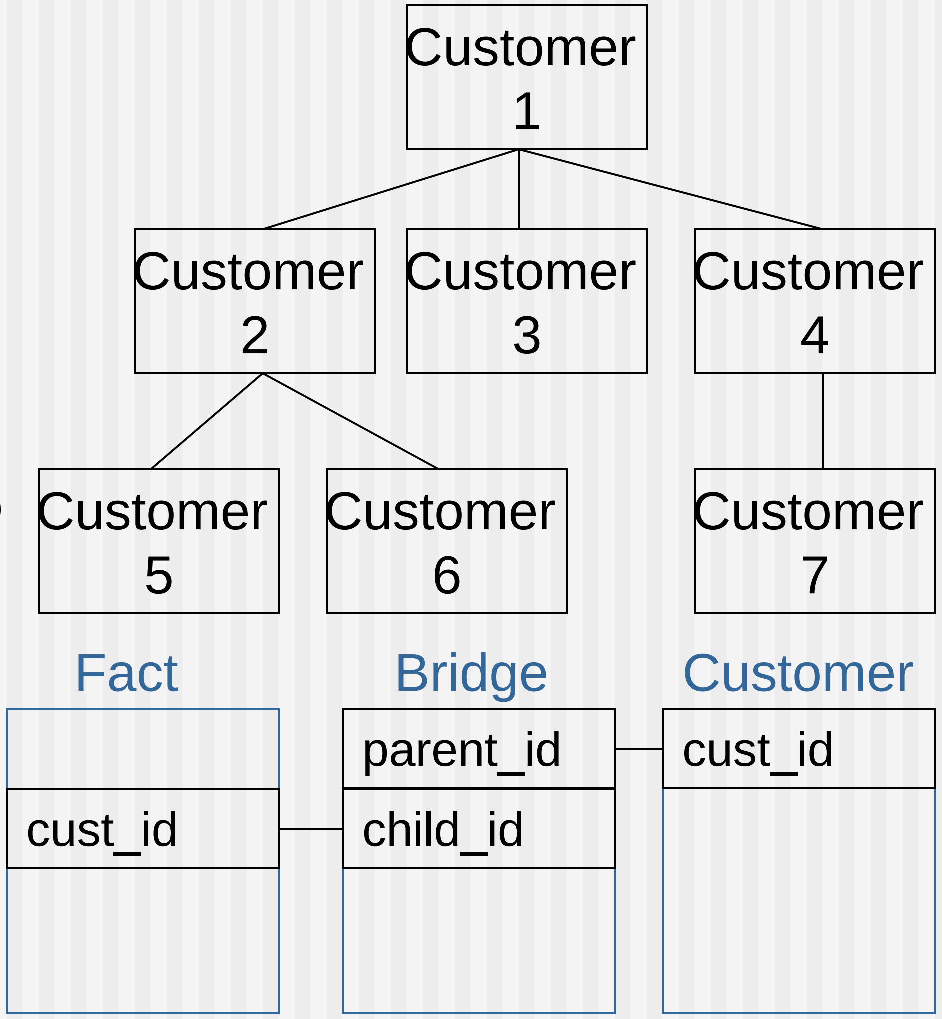
```
insert into test_connect_by values ( 5, 2);
      insert into test_connect_by values ( 5, 3);
insert into test_connect_by values ( 18, 11);
      insert into test_connect_by values ( 18, 7);
insert into test_connect_by values ( 17, 9);
      insert into test_connect_by values ( 17, 8);
insert into test_connect_by values ( 26, 13);
      insert into test_connect_by values ( 26, 1);
      insert into test_connect_by values ( 26, 12);
insert into test_connect_by values ( 15, 10);
      insert into test_connect_by values ( 15, 5);
insert into test_connect_by values ( 38, 15);
      insert into test_connect_by values ( 38, 17);
      insert into test_connect_by values ( 38, 6);
```

38, 26, 16 have no parents

```
insert into test_connect_by values (null,38);
insert into test_connect_by values (null,26);
insert into test_connect_by values (null,16);
```

Bridge Tables

- Customer dimension has one row for each customer entity at any level of the hierarchy
- Separate bridge table has schema:
  - Parent customer key
  - Subsidiary customer key
  - Depth of subsidiary
  - Bottom flag
  - Top flag
- One row in bridge table for every (ancestor, descendant) pair
  - Customer counts as its own Depth-0 ancestor
  - 16 rows for the hierarchy at right
- Fact table can join:
  - Directly to customer dimension
  - Through bridge table to customer dimension



Bridge Table Example

parent_id	child_id	depth	top_flag	bottom_flag
1	1	0	Y	N
1	2	1	Y	N
2	2	0	N	N
1	3	1	Y	Y
3	3	0	N	Y
1	4	1	Y	N
4	4	0	N	N
1	5	2	Y	Y
2	5	1	N	Y
5	5	0	N	Y
1	6	2	Y	Y
2	6	1	N	Y
…	…	…	…	…

Using Bridge Tables in Queries

- **Two join directions**
  - **Navigate up the hierarchy**
    - **Fact joins to subsidiary customer key**
    - **Dimension joins to parent customer key**
  - **Navigate down the hierarchy**
    - **Fact joins to parent customer key**
    - **Dimension joins to subsidiary customer key**
- **Safe uses of the bridge table:**
  - **Filter on customer dimension restricts query to a single customer**
    - **Use bridge table to combine data about that customer's subsidiaries or parents**
  - **Filter on bridge table restricts query to a single level**
    - **Require Top Flag = Y**
    - **Require Depth = 1**
      - **For immediate parent / child organizations**
    - **Require (Depth = 1 OR (Depth < 1 AND Top Flag = Y))**
      - **Generalizes the previous example to properly treat top-level customers**
- **Other uses of the bridge table risk over-counting**
  - **Bridge table is many-to-many between fact and dimension**

Restricting to One Customer

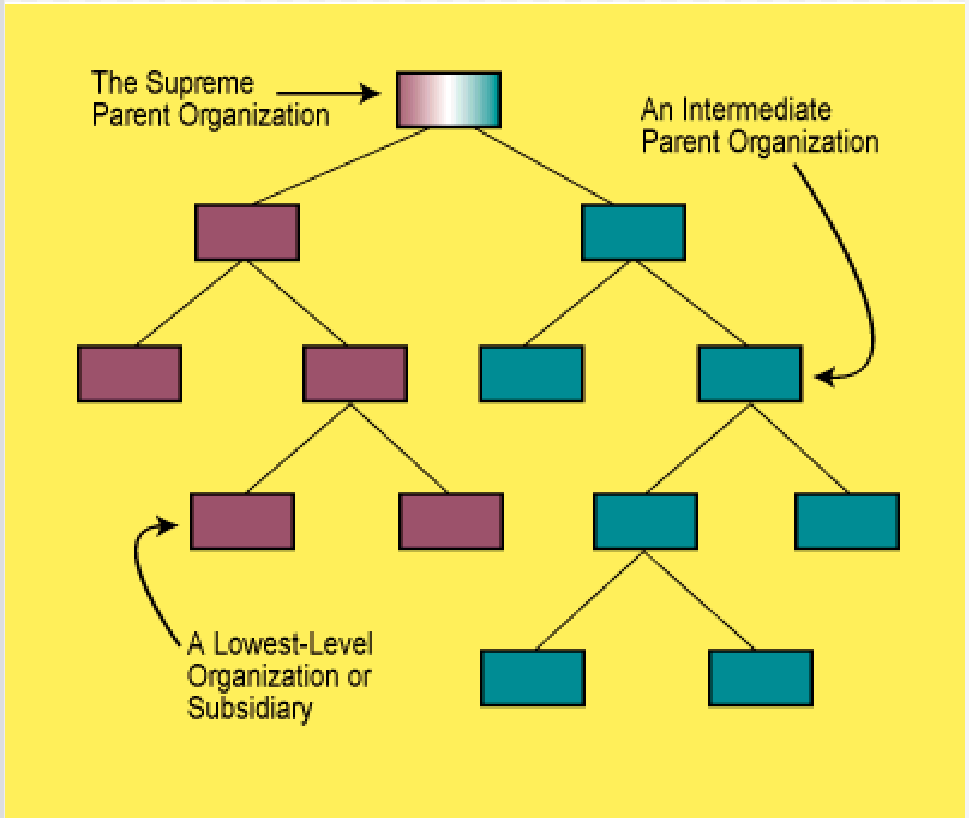
parent_id	child_id	depth	top_flag	bottom_flag
1	1	0	Y	N
1	2	1	Y	N
2	2	0	N	N
1	3	1	Y	Y
3	3	0	N	Y
1	4	1	Y	N
4	4	0	N	N
1	5	2	Y	Y
2	5	1	N	Y
5	5	0	N	Y
1	6	2	Y	Y
2	6	1	N	Y
…	…	…	…	…

Restricting to One Depth

parent_id	child_id	depth	top_flag	bottom_flag
1	1	0	Y	N
1	2	1	Y	N
2	2	0	N	N
1	3	1	Y	Y
3	3	0	N	Y
1	4	1	Y	N
4	4	0	N	N
1	5	2	Y	Y
2	5	1	N	Y
5	5	0	N	Y
1	6	2	Y	Y
2	6	1	N	Y
…	…	…	…	…

Variable Depth Hierarchies

- Examples
  - **Corporate organization chart**


The Supreme Parent Organization
An Intermediate Parent Organization
A Lowest-Level Organization or Subsidiary

How many Records in the Bridge Table?

One record for each separate path from each node in the org. tree to itself & to every node below it

13 nodes (to themselves)

12 nodes below root

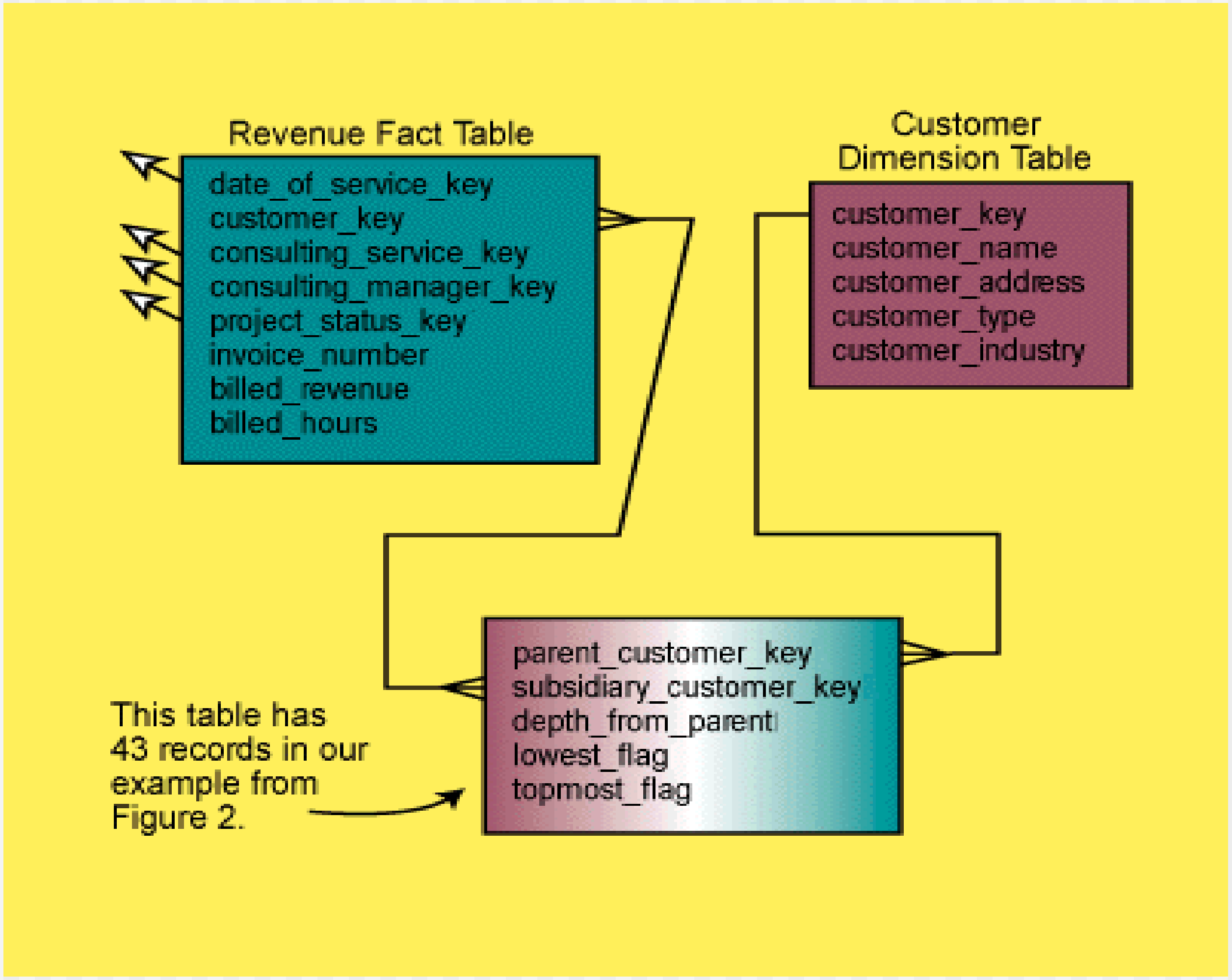
4+6 nodes below 1st level

2+4 nodes below 2nd level

2 nodes below 3rd level

TOTAL = 43

Variable Depth Hierarchies

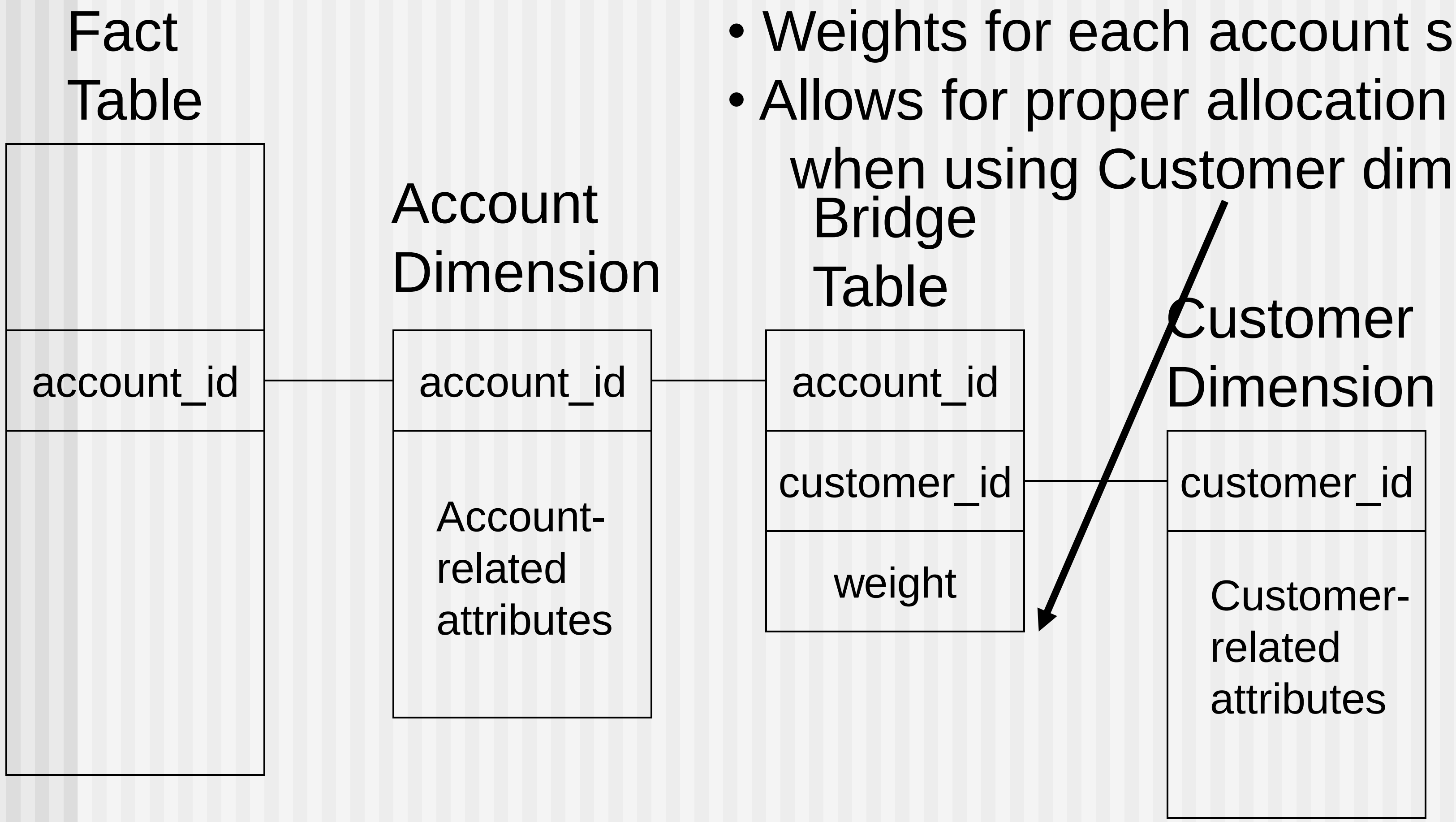


Variable-Depth Hierarchies

- **Advantages of bridge table approach:**
  - **Any normal dimensional constraint against the Customer dimension table and the helper table will cause all the fact table records for the directly constrained customers plus all their subsidiaries to be correctly summarized.**
  - **Standard relational databases and your standard query tools can be used to analyze the hierarchical structure.**

Multi-Valued Dimensions

- Weights for each account sum to 1
- Allows for proper allocation of facts when using Customer dimension



Weighted Report vs. Impact Report

- **Two formulations for customer queries**
- **Weighted report**
  - **Multiply all facts by weight before aggregating**
  - **SUM(DollarAmt * weight)**
  - **Subtotals and totals are meaningful**
- **Impact report**
  - **Don’t use the weight column**
  - **SUM(DollarAmt)**
  - **Some facts are double-counted in totals**
  - **Each customer is fully credited for his/her activity**
  - **Most useful when grouping by customer**