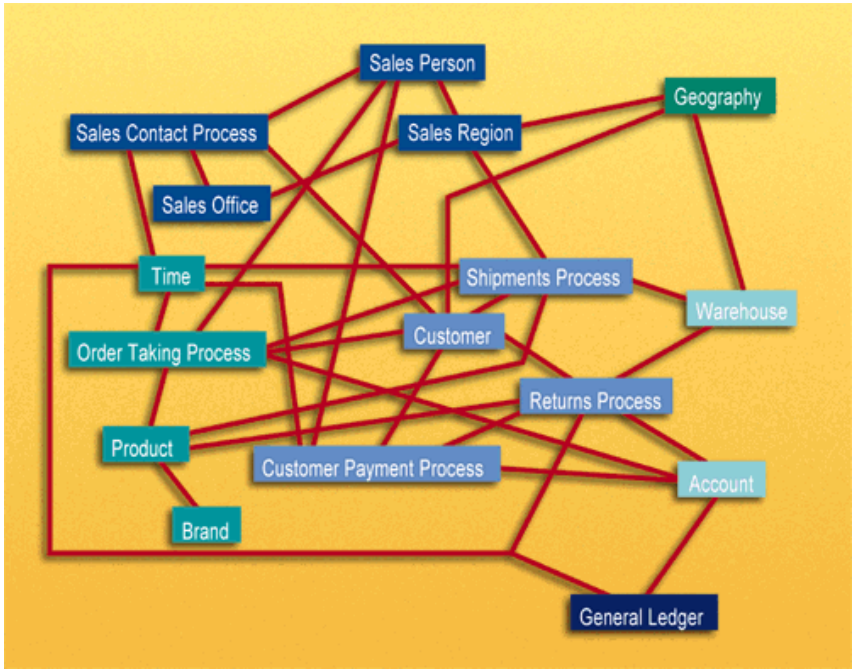
SS G515 - Data Warehousing

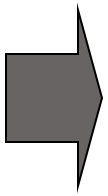
Dr. Yashvardhan Sharma
Assistant Professor, CS & IS Dept.
BITS-Pilani

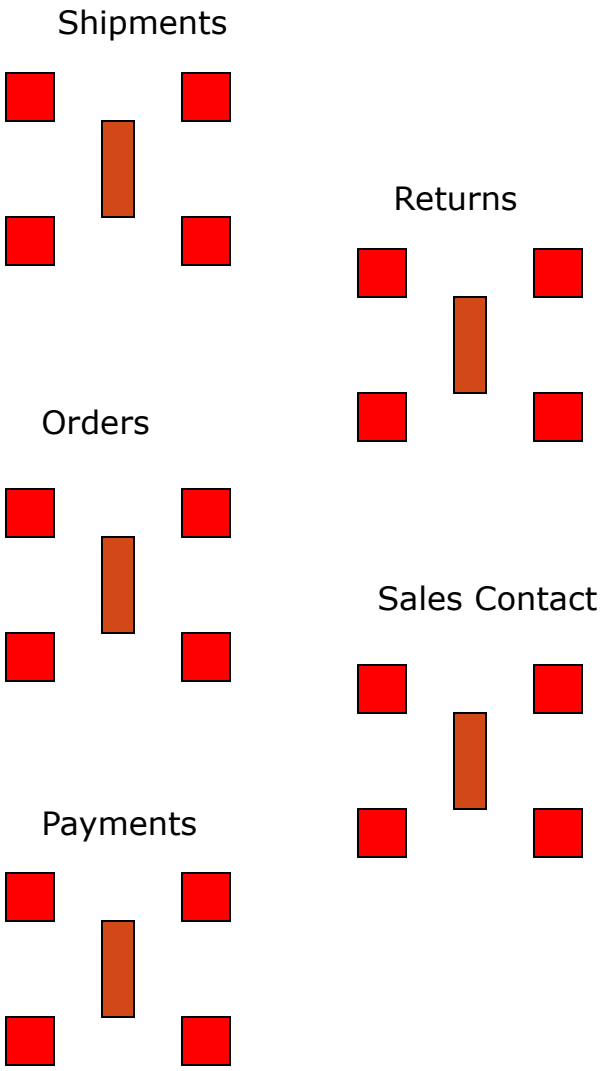
Dimensional Model vs ER model

- The key to understanding the relationship between DM and ER is that a single ER diagram breaks down into multiple DM diagrams, or 'stars'.
- Think of a large ER diagram as representing every possible business process within an application. The ER diagram may have Sales Calls, Order Entries, Shipment Invoices, Customer Payments, and Product Returns, all on the same diagram.

Dimensional Model vs ER model







Dimensional Model vs ER model

- To create the individual 'stars' that exist within an application:
  - Look for many-to-many relationships in the ER model containing numeric and additive facts and to designate them as fact tables.
  - Alternatively, look for 'events' or 'transactions' – these may also be facts
  - Denormalize all of the remaining tables into flat tables with single-part keys that connect directly to the fact tables. These tables become the dimension tables.
  - In cases where a dimension table connects to more than one fact table, we represent this same dimension table in both schemas, and we refer to the dimension tables as "conformed" between the two dimensional models.

DM Strengths

- The dimensional model has a number of important data warehouse advantages that the ER model lacks.
  - First, the dimensional model is a predictable, standard framework. Report writers, query tools, and user interfaces can all make strong assumptions about the dimensional model to make the user interfaces more understandable and to make processing more efficient.
  - Rather than using a cost-based optimizer, a database engine can make very strong assumptions about first constraining the dimension tables and then "attacking" the fact table all at once with the Cartesian product of those dimension table keys satisfying the user's constraints.

DM Strengths

- A second strength of the dimensional model is that the predictable framework of the star join schema withstands unexpected changes in user behavior. Every dimension is equivalent. All dimensions can be thought of as symmetrically equal entry points into the fact table. The logical design can be done **independent of expected query patterns**. The user interfaces are symmetrical, the query strategies are symmetrical, and the SQL generated against the dimensional model is symmetrical.

DM Strengths

- A third strength of the dimensional model is that it is 'gracefully extensible' to accommodate unexpected new data elements and new design decisions.
- Gracefully extensible:
  - all existing tables (both fact and dimension) can be changed in place by simply adding new data rows in the table, or the table can be changed in place with a SQL alter table command.
  - Data should not have to be reloaded.
  - No query tool or reporting tool needs to be reprogrammed to accommodate the change.
  - Old applications continue to run without yielding different results. Adding new unanticipated facts (that is, new additive numeric fields in the fact table), as long as they are consistent with the fundamental grain of the existing fact table

DM Strengths

- A fourth strength of the dimensional model is that there is a body of standard approaches for handling common modeling situations in the business world. These modeling situations include:
  - Slowly changing dimensions, where a "constant" dimension such as Product or Customer actually evolves slowly and asynchronously.
  - Event-handling databases, where the fact table usually turns out to be "factless."
  - Many others

DM Strengths

- A final strength of the dimensional model is the management of aggregates.
- Aggregates are summary records that are logically redundant with base data already in the data warehouse, but they are used to enhance query performance.
- A comprehensive aggregate strategy is required in every medium- and large-sized data warehouse implementation.
- All of the aggregate management software packages and aggregate navigation utilities depend on a very specific single structure of fact and dimension tables that is absolutely dependent on the dimensional model.

ER vs DM – Final Points

- ER models are not appropriate for Data Warehouses. ER modeling does not really model a business; rather, it models the micro relationships among data elements.
- ER models are wildly variable in structure. As such, it is extremely difficult to optimize query performance.

Types of Facts

- Fully-additive-all dimensions
  - Units_sold, Sales_amt
- Semi-additive-some dimensions
  - Account_balance, Customer_count

    28/3,tissue paper,store1, 25, 250,20

    28/3,paper towel,store1, 35, 350,30

    Is no. of customers who bought either tissue paper or paper towel is 50? **NO**
- Non-additive-none
  - Gross margin=Gross profit/amount
  - Note that GP and Amount are fully additive
  - Ratio of the sums and not sum of the ratios

Data Warehouse: Design Steps

Step 1: Identify the Business Process

Step 2: Declare the *Grain*

Step 3: Identify the Dimensions

Step 4: Identify the Facts

Modeling Design Process

1. Identify the Business Process
   - Source of "measurements"
2. Identify the Grain
   - What does 1 row in the fact table represent or mean?
3. Identify the Dimensions
   - Descriptive context, true to the grain
4. Identify the Facts
   - Numeric additive measurements, true to the grain

Step 1 - Identify the Business Process

- This is a business activity typically tied to a source system.
- Not to be confused with a business department or function. An Orders dimensional model should support the activities of both Sales and Marketing.
- "If we establish departmentally bound dimensional models, we'll inevitably duplicate data with different labels and terminology."

Step 2 - Identify the Grain

- The level of detail associated with the fact table measurements.
- A critical step necessary before steps 3 and 4.
- Preferably it should be at the most atomic level possible.
- “How do you describe a single row in the fact table?”

Step 3 - Identify the Dimensions

- The list of all the discrete, text-like attributes that emanate from the fact table.
- They are the "by" words used to describe the requirements.
- Each dimension could be though of as an analytical "entry point" to the facts.
- "How do business people describe the data that results from the business process?"

Step 4 - Identify the Facts

- Must be true to the grain defined in step 2.
- Typical facts are numeric additive figures.
- Facts that belong to a different grain belong in a separate fact table.
- Facts are determined by answering the question, "What are we measuring?"
- Percentages and ratios, such as gross margin, are non-additive. The numerator and denominator should be stored in the fact table.

Grocery Store: The Universal Example

The Scenario:

- Chain of 100 Grocery Stores
- 60000 individual products in each store

- 10000 of these products sold on any given day(average)
- 3 year data

Grocery Store DW

- Step 1: Sales Business Process
- Step 2: Daily Grain
- A word about GRANULARITY
  - Temp sensor data: per ms, sec, min, hr?
  - Size of the DW is governed by granularity
  - Daily grain (club products sold on a day for each store) Aggregated data
  - Receipt line Grain (each line in the receipt is recorded – finest grain data)

Grocery Store: DW Size Estimate

- Daily Grain
- Size of Fact Table

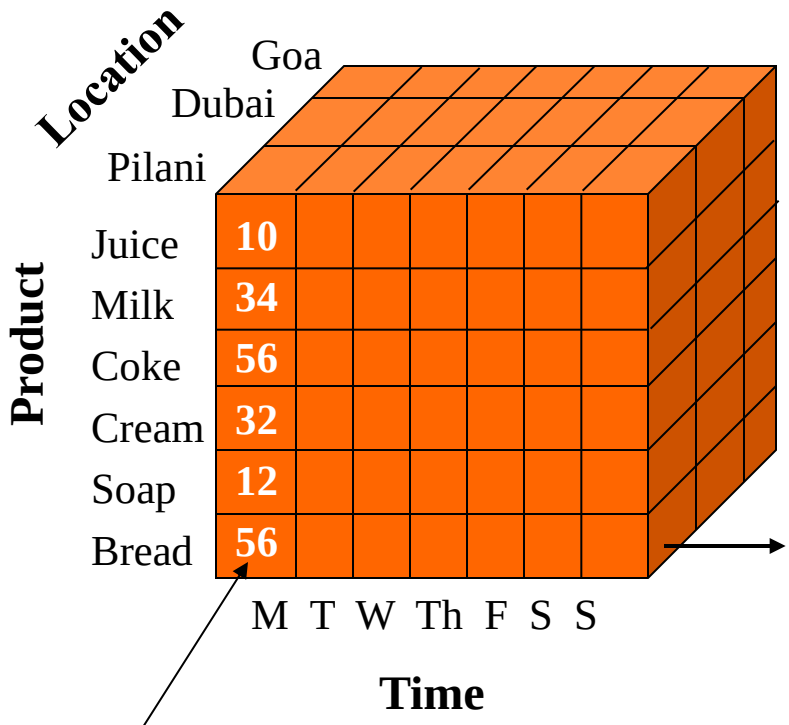
  = 100*10000*3*365

  = 1095 million records
- 4 facts & 4 dimensions (48 bytes)

Fact size: 8 bytes and dimension size: 4 bytes

- 1095 m * 48 bytes = 52560 m bytes
- i.e. ~ 50 GB

Data Cube



Dimensions:

Time, Product, Location

Attributes:

Product (upc, price, …)

Location…

…

Hierarchies:

Product → Brand → …

Day → Week → Quarter

City → Region → Country