Data-warehousing & Business Intelligence Concepts

TABLE OF CONTENTS

What is a data-warehouse

According to Bill Inmon (father of data-warehousing), it is a

- ***Subject-oriented***
- ***Integrated***
- ***Time-variant***
- ***Non-volatile***

collection of data used for decision-making by the management

Purpose of a data-warehouse

- Store enterprise data which is clean and pre-formatted;
- Helps organization to understand the ‘health’ of the business;
- Helps in the generation of reports and perform analysis of the data;

OLTP vs OLAP Systems

OLTP system

- Used for online transaction processing.
- Primary function is to ensure that business processes are up and running.
- The entities (tables) are highly normalized to eliminate data redundancy.
- The system allows easy insert, delete and update of data.

OLAP system

- Used for online analysis of data.
- Primary function to enhance the business processes and provide insight into the 'health' of the business.
- The entities are usually de-normalized.
- The system allows easy retrieval of data for generating reports and performing analysis.

Data-warehouse Architectures

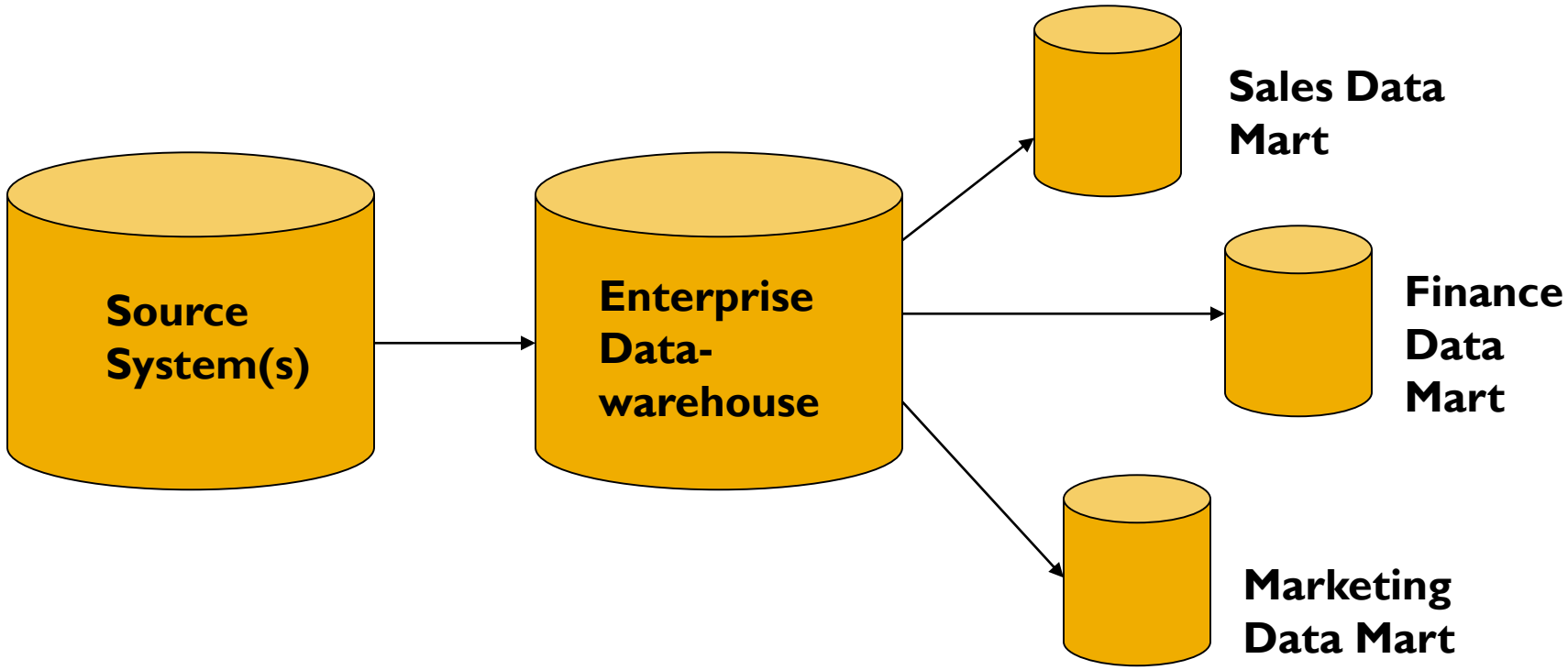
Two primary schools of thought:

- ***Corporate Information Factory (CIF)*** by Bill Inmon
- ***Bus Architecture*** by Ralph Kimball

- Other architectures:
  - ***DW 2.0***

CIF Architecture

Enterprise Data-warehouse & Data Marts



CIF Architecture – some terms & concepts

Data Producer	• Enterprise Data-warehouse • Operational Data Store
Information Consumer	• Data Marts
Process	• 'Getting data in' • 'Getting information out'

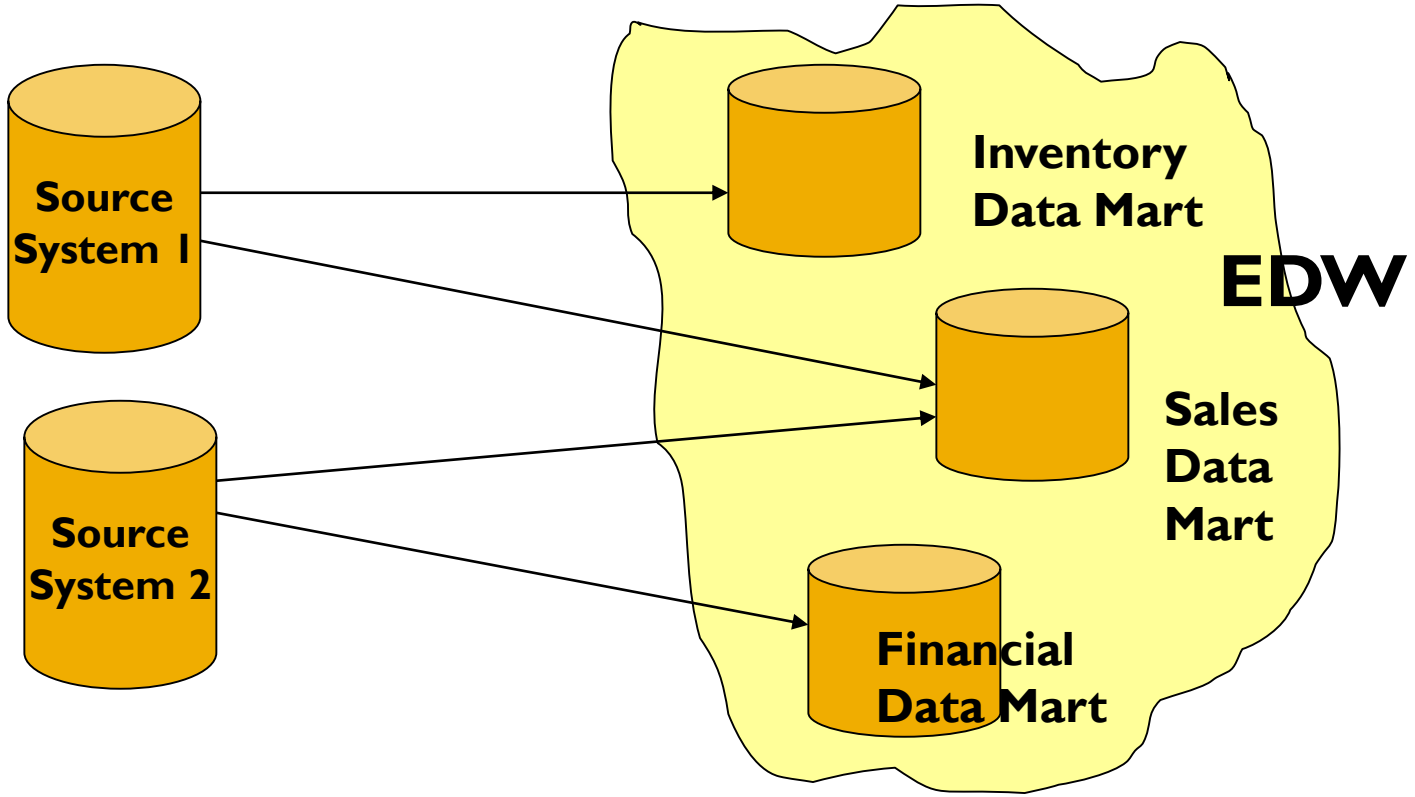
Bus Architecture - some terms & concepts

- Enterprise Data-warehouse
- Data Marts
- Dimensions
- Facts
- Types of facts
- Types of fact tables

- Multi-dimensional
- Conformed dimension
- Star-schema
- Snowflake-schema
- Surrogate keys
- Degenerate dimension
- Junk dimension

Bus Architecture

Enterprise Data-warehouse

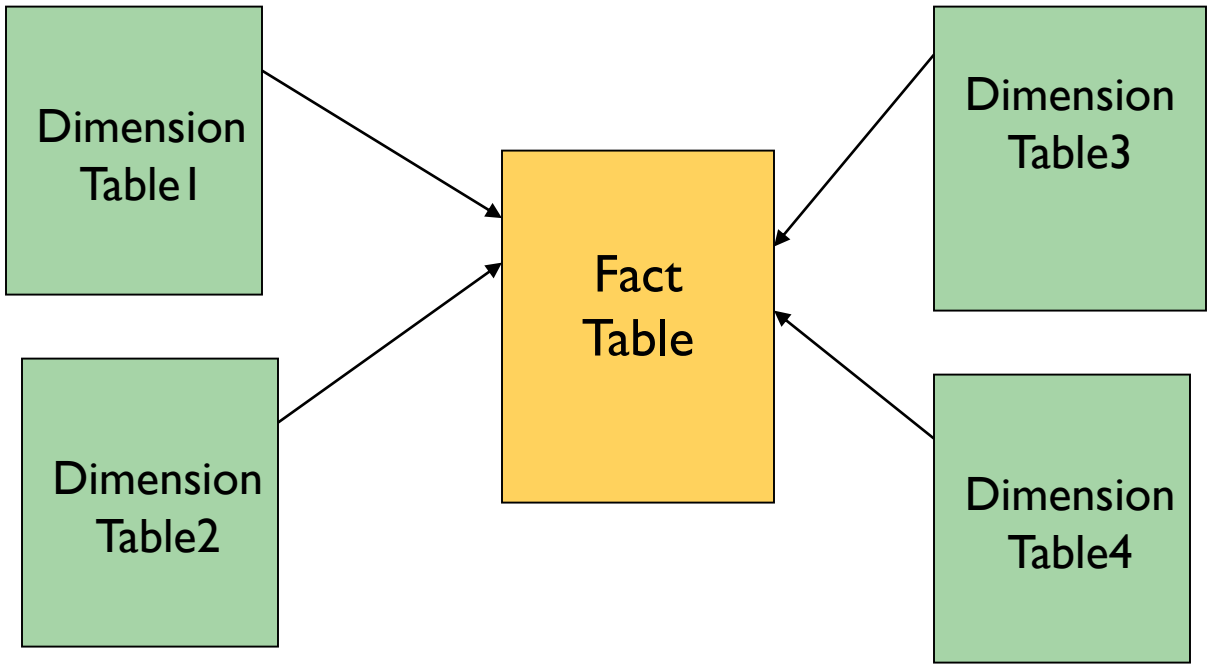


Bus Architecture Data Model

Star Schema

- *Comprises of fact and dimension tables*
- *Multiple dimension tables joined to a fact table*
- *Key joining dimension and fact tables – surrogate key*
- *Also known as multi-dimensional data model*

Star Schema


Dimension Table1
Dimension Table2
Fact Table
Dimension Table3
Dimension Table4

Dimension and Fact table Properties

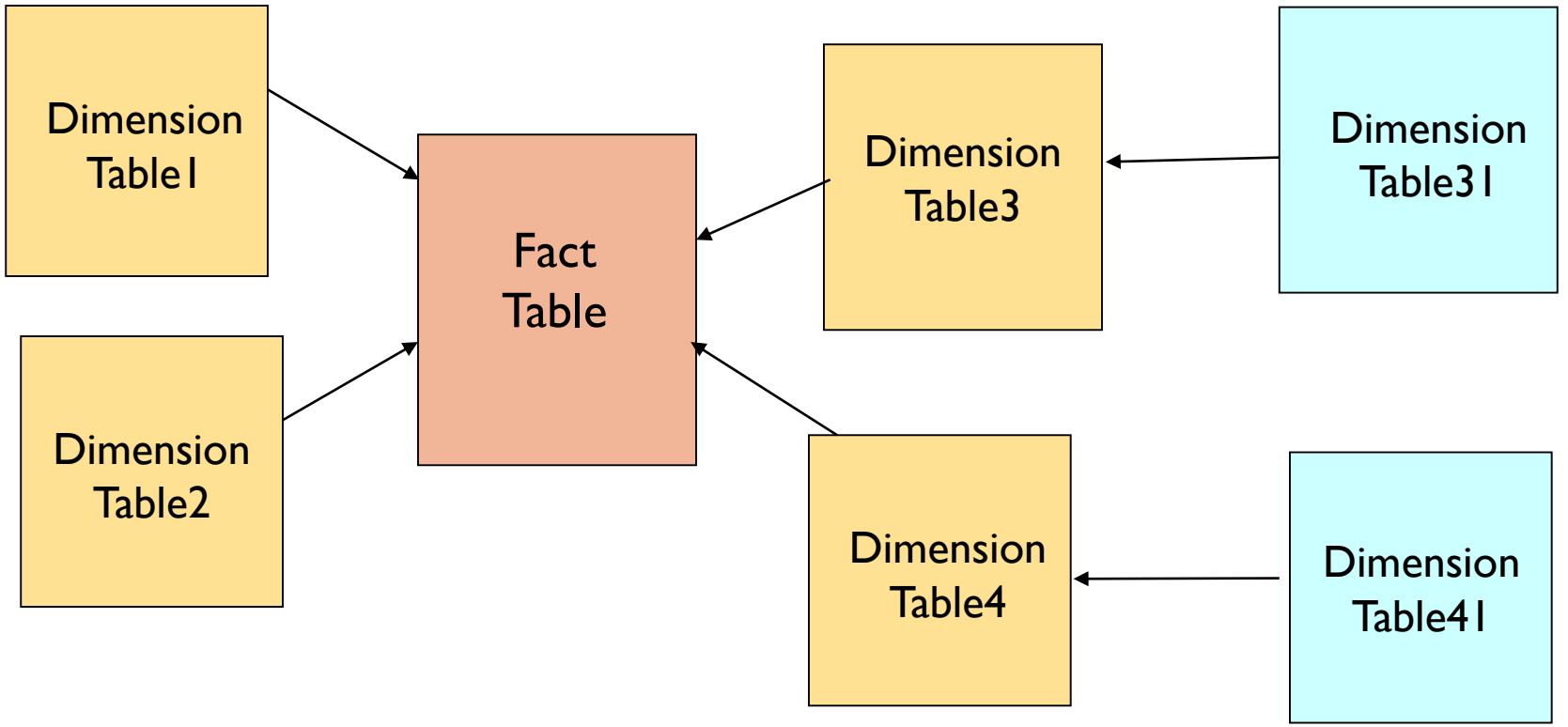
Dimension table

- De-normalised.
- Contains surrogate key as well as the OLTP key.
- Contains data from the master table of the OLTP systems.
- Attributes are descriptive in nature.
- Attributes are usually used in the ‘where’ clause of the reporting queries.

Fact table

- Contains data from the transaction tables of the OLTP system.
- Usually contains ‘facts’ or information that is numerical in nature.
- The ‘facts’ are usually additive across multiple dimensions
- Has a one-to-many relationship with the dimension tables.
- Attributes are usually used in the ‘select’ clause of reporting queries.

Snow-flake Schema



These dimension tables are *normalized*

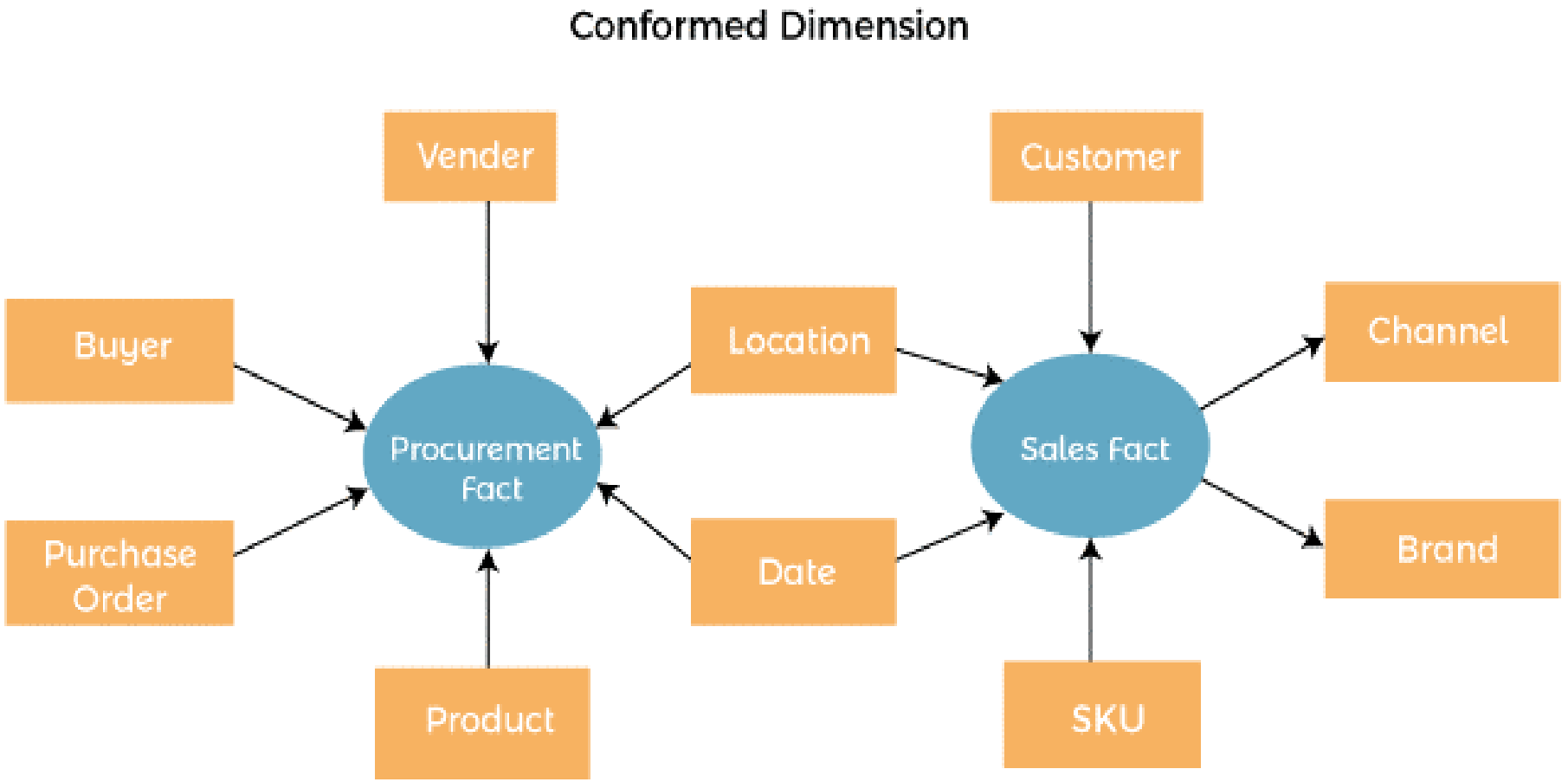
When to Snowflake

- When certain attributes of the dimension table changes more often than the rest of the attributes.
- When most of the queries use a specific set of attributes of a dimension table, then the table can be snow-flaked to reduce the volume of data that needs to participate in the report generation process.
- Reduce storage requirements of dimension data-model.

Conformed Dimension

- A conformed dimension is a dimension which is
  - shared across multiple fact tables;
  - has the same meaning to all the fact tables that it has been joined to.
- Conformed dimensions are used to *join multiple data marts* to build an enterprise data-warehouse.
- Conformed dimensions (for that matter, all dimension tables) should have anonymous or ***surrogate*** key, which is not a production system key.

Conformed Dimension - example



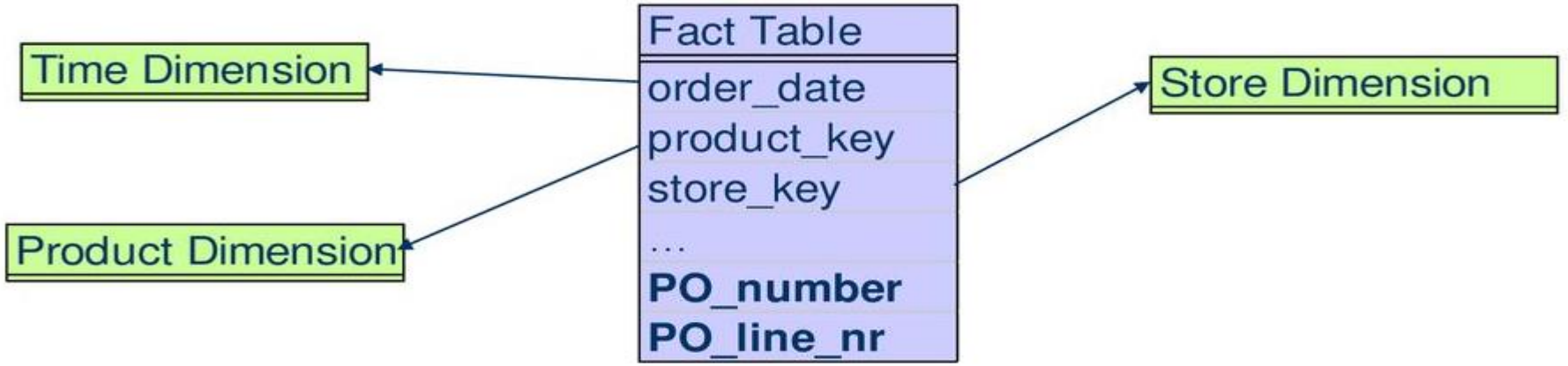
Surrogate Key

- A surrogate key is a key which does not have any contextual or business meaning.
- It is manufactured “artificially” and only for the purposes of data analysis.
- Usually denoted as an integer.
- Used to ensure that:
  - If production keys are re-cycled, the data in the warehouse is still retrievable.
  - In case of a merger between organizations, the nature of the production key may change.

Degenerate & Junk dimensions

Degenerate Dimension

- A degenerate dimension is a dimensional attribute which is embedded in the fact table.
- They usually occur in line-item-oriented fact table design.
- Examples of these are – order number, invoice number, etc.



Degenerate & Junk dimensions

Junk Dimension

- Junk dimensions are usually flags and text attributes that do not belong to the fact table or any of the existing dimension tables.
- These indicator fields are combined into a single dimension.
- Examples: yes/no indicator fields in the source system.

DIM_JUNK

JUNK_ID	TXN_CODE	COUPON_IND	PREPAY_IND
1	1	Y	Y
2	2	Y	Y
3	3	Y	Y
4	1	Y	N
5	2	Y	N
6	3	Y	N
7	1	N	Y
8	2	N	Y
9	3	N	Y
10	1	N	N
11	2	N	N
12	3	N	N

Slowly Changing Dimensions

Dimensions whose attributes change over time are called *slowly changing dimensions* **(SCD)**.

Types of SCD:

- ***Type 1***
  - This type of dimension table *does not hold* any historical value. The latest value of an attribute overwrites the previous value of the same attribute.
- ***Type 2***
  - This type of dimension table *holds the complete* historical value of the attributes. A new record is inserted in the table every time the attribute value changes.
- ***Type 3***
  - This type of dimension table stores *'partial'* historical values. For e.g., depending on the business need, last two historical values may be stored.

Slowly Changing Dimensions

Type 1: Update Changes

Supplier_Key	Supplier_Code	Supplier_Name	Supplier_State
123	ABC	Acme Supply Co	CA

Supplier_Key	Supplier_Code	Supplier_Name	Supplier_State
123	ABC	Acme Supply Co	IL

Type 2: Keep Historical

Supplier_Key	Supplier_Code	Supplier_Name	Supplier_State	Start_Date	End_Date
123	ABC	Acme Supply Co	CA	01-Jan-2000	21-Dec-2004
124	ABC	Acme Supply Co	IL	22-Dec-2004	

Type 3: Preserve Limited History

Supplier_Key	Supplier_Code	Supplier_Name	Original_Supplier_State	Effective_Date	Current_Supplier_State
123	ABC	Acme Supply Co	CA	22-Dec-2004	IL

Types of Facts

Additive

- Additive facts are facts that can be summed up through all of the dimensions in the fact table.
- **Example**: ***sales*** (purchases from a store). Can add hourly sales to get the sales for a day, week, month, quarter, or year.

Semi-additive

- Semi-additive facts are facts that can be summed up for some of the dimensions in the fact table, but not the others.
- **Example**: ***Daily balances*** fact can be summed up through the customer dimension but not through the time dimension.

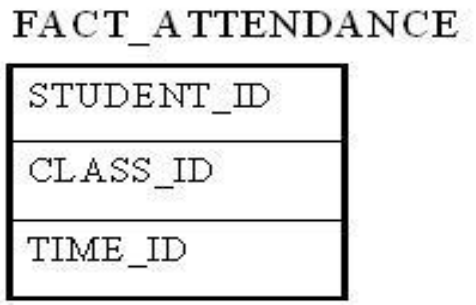
Non-additive

- Non-additive facts are facts that cannot be summed up for any of the dimensions present in the fact table.
- **Example:** ***ratios***, they cannot be added across any dimension.

Types of Fact Tables – Factless Fact Table

A ***factless*** fact table is a fact table that does not have any measures. It is essentially an intersection of dimensions. On the surface, a factless fact table does not make sense, since a fact table is, after all, about facts. However, there are situations where having this kind of relationship makes sense in data warehousing. They primarily capture *events*.

For example, think about a record of student attendance in classes. In this case, the fact table would consist of 3 dimensions: the *student dimension*, the *time dimension*, and the *class dimension*. This factless fact table would look like the following:



Factless fact tables offer the a lot of flexibility in data-warehouse design. For *example*, one can easily answer the following questions with this table:

How many students attended a particular class on particular day?

How many classes on average does a student attend on a given day?

Types of Fact Tables – Transaction Fact Table

A ***Transaction*** table is the most basic and fundamental view of business operations. These fact tables represent an event that occurred at an instantaneous point in time. A row exists in the fact table for a given customer or product only if a transaction has occurred.

A given customer or product is likely linked to multiple rows in the fact table because the customer or product is involved in more than one transaction. Transaction data often is structured quite easily into a dimensional framework. The lowest-level data is the most natural dimensional data, supporting analyses that cannot be done on summarized data.

Types of Fact Tables – Snapshot Fact Table

This type of fact table describes the state of things in a particular instance of time, and usually includes more semi-additive and non-additive facts. The example presented here is a snapshot fact table.

Periodic snapshots are needed to see the cumulative performance of the business at regular, predictable time intervals. Unlike the transaction fact table, where we load a row for each event occurrence, with the periodic snapshot, we take a picture of the activity at the end of a day, week, or month, then another picture at the end of the next period, and so on.

Example: A performance summary of a salesman over the previous month.

Table relationships

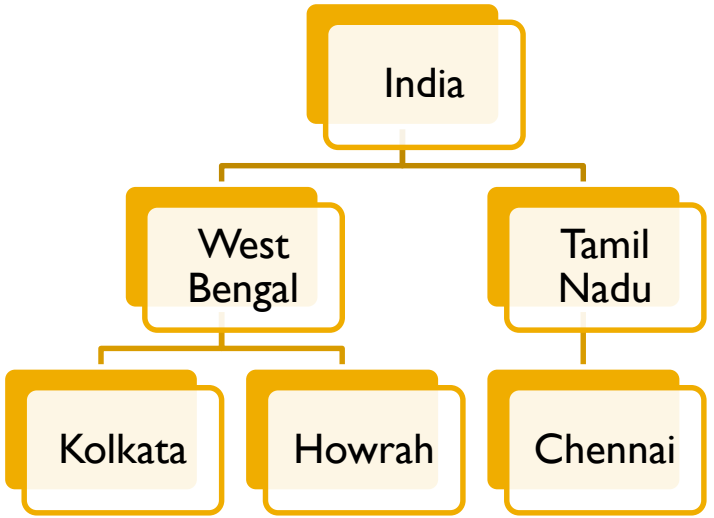
- **Dimension & Fact tables**
  - ✓ One to many :: dimension : fact

- **Snow-flaked dimension tables**
  - ✓ One to many :: dimension : secondary dimension

Hierarchy and Granularity

Hierarchy

- It represents the '***order***' in which information is arranged in the dimension table;
- Example:



Granularity

- It is the ***lowest level of data*** that is stored in the fact table;
- Example: The line items in a sales invoice.

Aggregation

What is Aggregation

Aggregation is the process of summarizing fact data based on one or more dimensions.

Purpose of Aggregation

The primary reason for data aggregation is to improve performance of queries in a large data-warehouse.

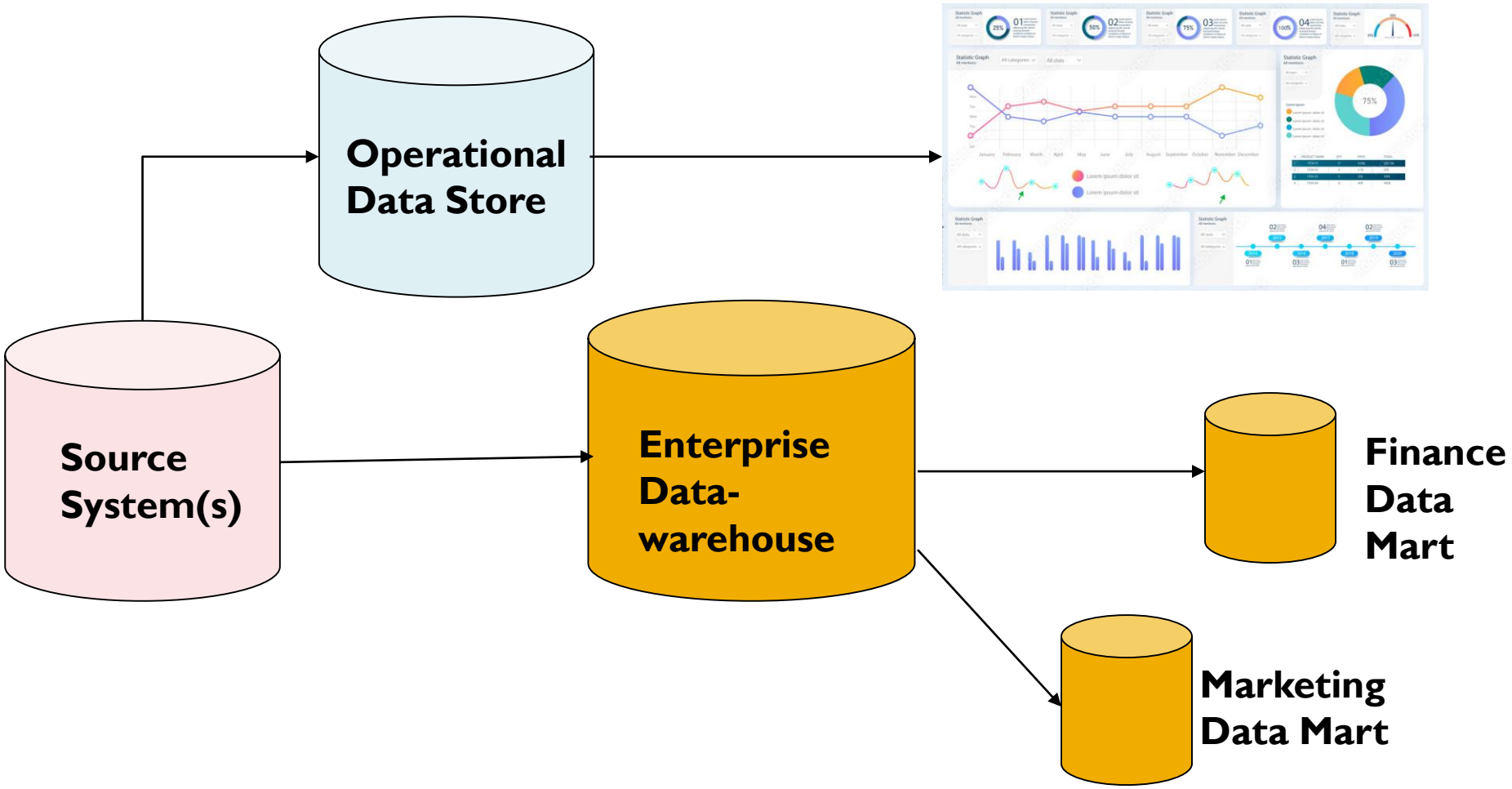
Data Marts

- A logical subset of a complete data-warehouse.
- Contains data from a single business process or a set of related business processes.
- Primarily used for analysis of data for the business process for which it was built.
- The data contained in a data mart can be at the most granular level of transaction or can be aggregated to support specific reporting needs.
- Data marts which contain the data at the most granular level of transaction are combined together to create the Enterprise Data warehouse as per the Bus Architecture.

Operational Data Store

- An operational data store (ODS) is a data repository that contains online transaction data.
- The data in an ODS is stored for a short period of time, say 60 days, and hence historical data is not available in an ODS.
- An ODS is basically used as a reporting system to remove the burden of querying data from an OLTP system directly.
- This is done to ensure that the OLTP system is not ‘choked’ running complex reporting queries.

Operational Data Store



Reporting and Data Analysis

Types of Reports

- **Batch**
  - Standard reports that are generated on a regular basis
  - Users can usually just view the data
- **Ad-hoc**
  - These reports are created by users according to their unique business needs
  - Users can perform different types of analysis on these reports
- **Canned**
  - Pre-defined reports but are usually parameterized
  - Users can view data based on the value of the parameter

Reporting and Data Analysis – OLAP Cubes

- The OLAP cube provides the multidimensional way to look at the data.
- The specific design of an OLAP cube ensures report optimization.
- The storage of OLAP cube data is in such a way as to make easy and efficient reporting.
- OLAP cubes are built from categories of data called dimensions and measures.

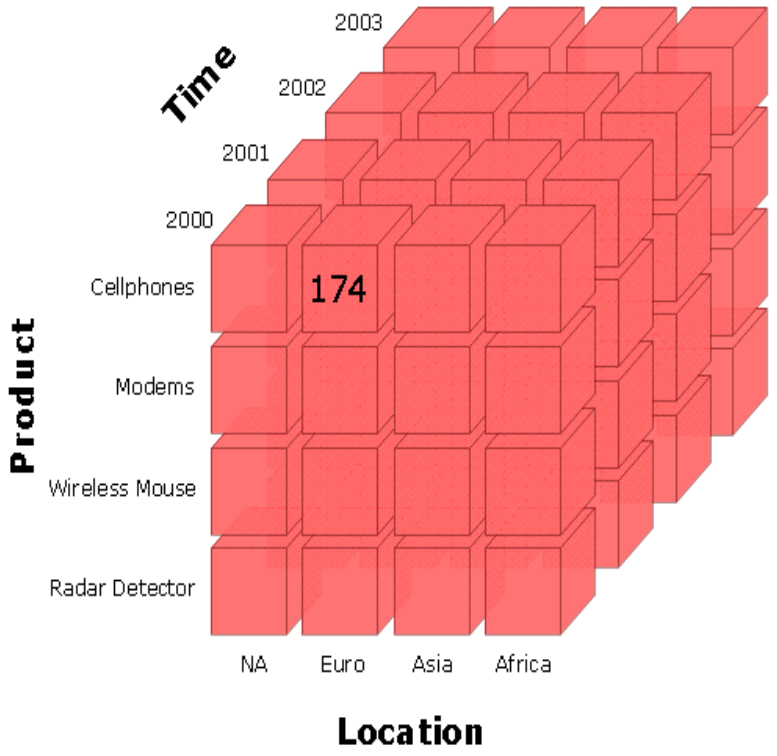
OLAP Cubes

- The OLAP cube provides the multidimensional way to look at the data.
- The specific design of an OLAP cube ensures report optimization.
- The storage of OLAP cube data is in such a way as to make easy and efficient reporting.
- OLAP cubes are built from categories of data called dimensions and measures.

OLAP Cubes

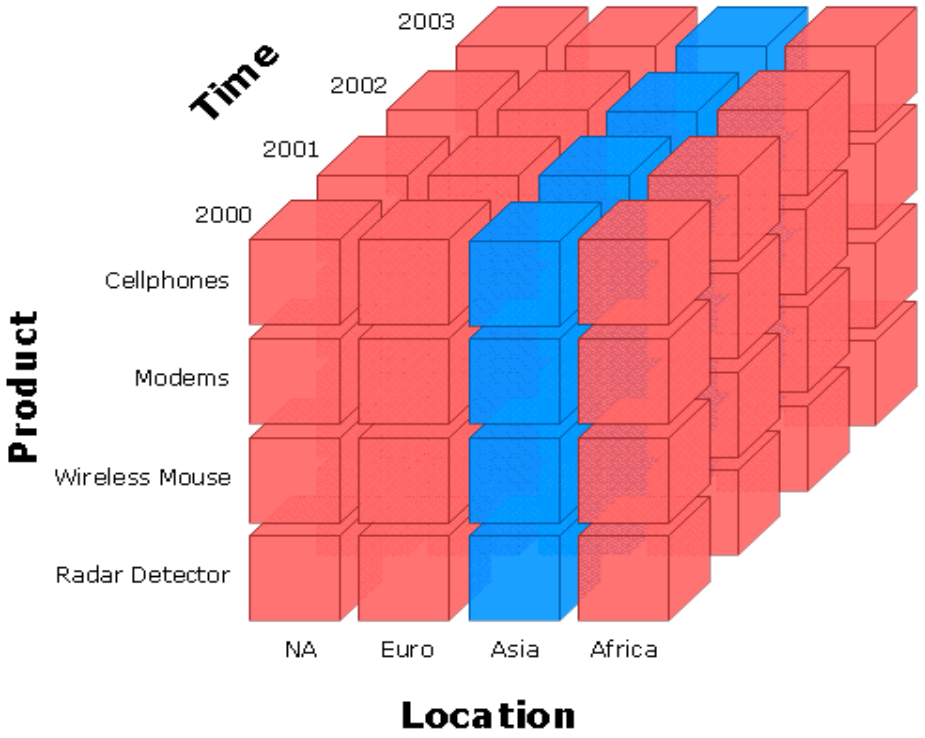
- The term cube comes from the geometric object and implies three dimensions; but in actual use, the cube may have more than three dimensions (usually called a hypercube).

- The figure here shows an OLAP cube in which Time, Product, Location represent the dimensions of the cube, while 174 represents the measure (fact).



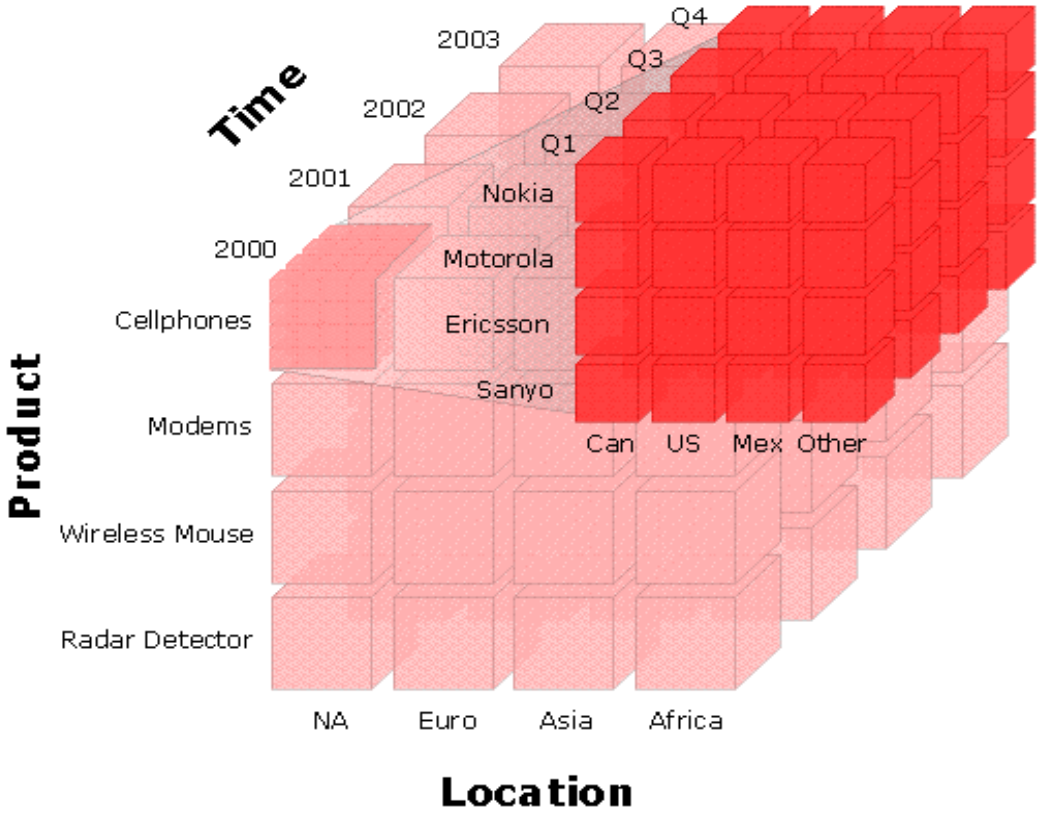
Slicing

This figure illustrates slicing the location Asia. When data is sliced as shown in this example, it results in the product and years data for Asia. Thus the data is effectively filtered to display the measures associated with the Asia location.



Dicing

A related operation to slicing is dicing. In the case of dicing, a sub-cube of the original space is defined. The data that is seen is that of one cell from the cube. Dicing provides the smallest available slice.



Drilling

- Users can perform two different kinds of ‘drilling’ operations on the data.
- One is drilling-down; the other is drilling-up.
- Drilling down allows the user to move from summarized (aggregate) data to more detailed (granular) data.
- Drilling up is the reverse operation in which an user will be able to view summarized data from detailed/granular data.

Questions…???

THANK YOU...!!!

NEVER STOP LEARNING