BITS Pilani
Pilani Campus

Data Warehousing
SS ZG515

PC Reddy
Guest Faculty – WILP, BITS Pilani

BITS Pilani
Pilani Campus

Data Warehousing – Lecture 1

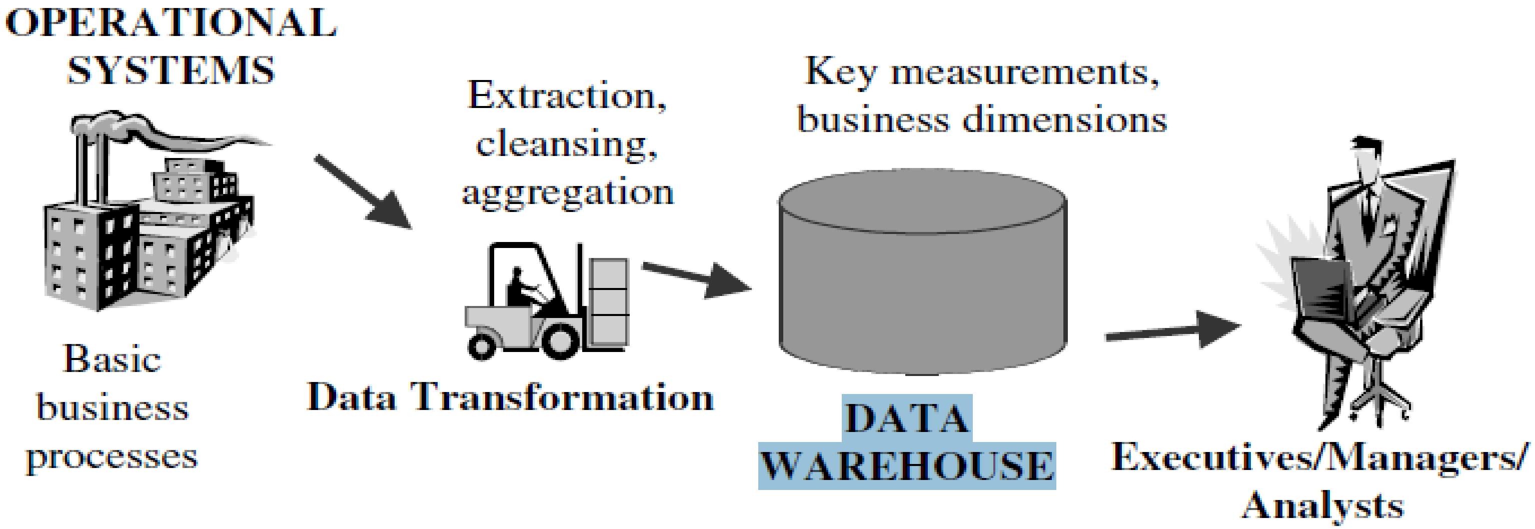
Introduction to Data Warehouses, Definitions, Concepts, Architecture

Operational vs DW

- Operational system
  - OLTP
  - Systems that support day to day operations
  - These systems “get data **into** DB”
    - Ex:Take an order, process a claim, make a shipment, generate an invoice etc.

- Data Warehouse system
  - OLAP
  - Systems that support strategic decisions
  - These systems “get data **out** of DB”
    - Ex: Show top selling products, show problem regions, show the highest margins, alerts on thresholds.

Operational vs DW


OPERATIONAL SYSTEMS
Basic business processes
Extraction, cleansing, aggregation
Data Transformation
Key measurements, business dimensions
DATA WAREHOUSE
Executives/Managers/ Analysts

Data Warehouse is an environment that →

- Provides an integrated and total view of the enterprise
- Makes the enterprise's current and historical information easily available for decision making
- Makes decision-support transactions possible without hindering operational systems
- Renders the organization's information consistent
- Presents a flexible and interactive source of strategic information

R. Kimball's definition of a DW

- A **data warehouse** is a copy of transactional data specifically structured for querying and analysis.

- According to this definition:
  The form of the stored data (RDBMS, flat file) has nothing to do with whether something is a data warehouse.

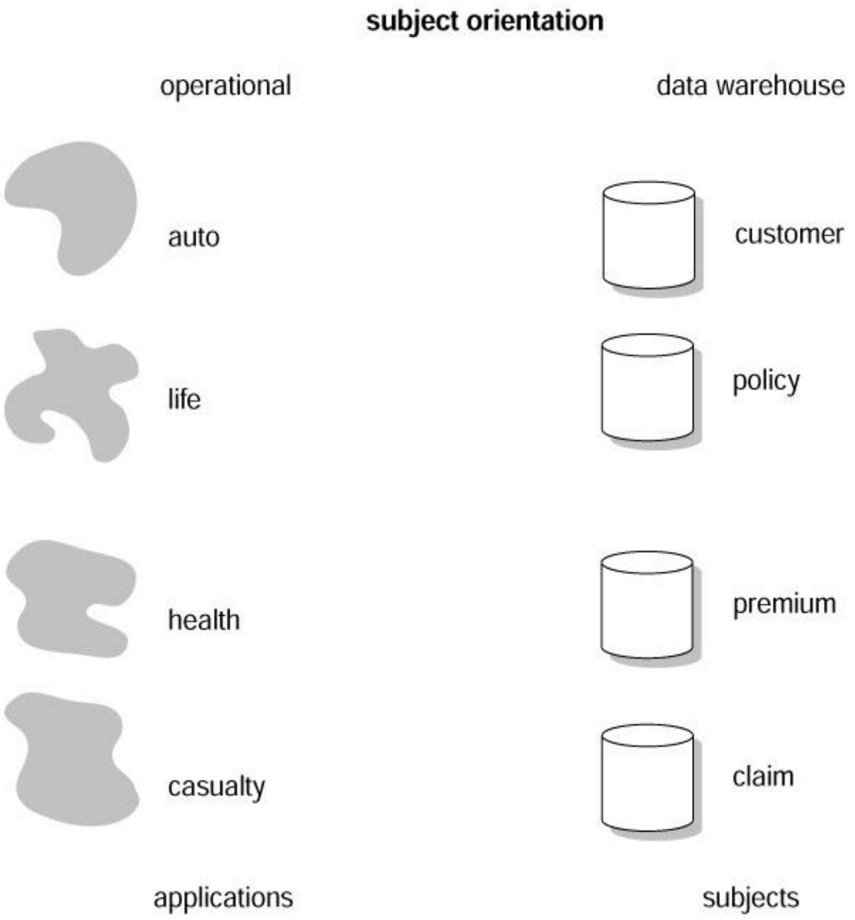
  Data warehousing is not necessarily for the needs of "decision makers" or used in the process of decision making.

Inmon’s Definition of a DW

- A **data warehouse** is a
  subject-oriented,
  integrated,
  nonvolatile, and
  time-variant

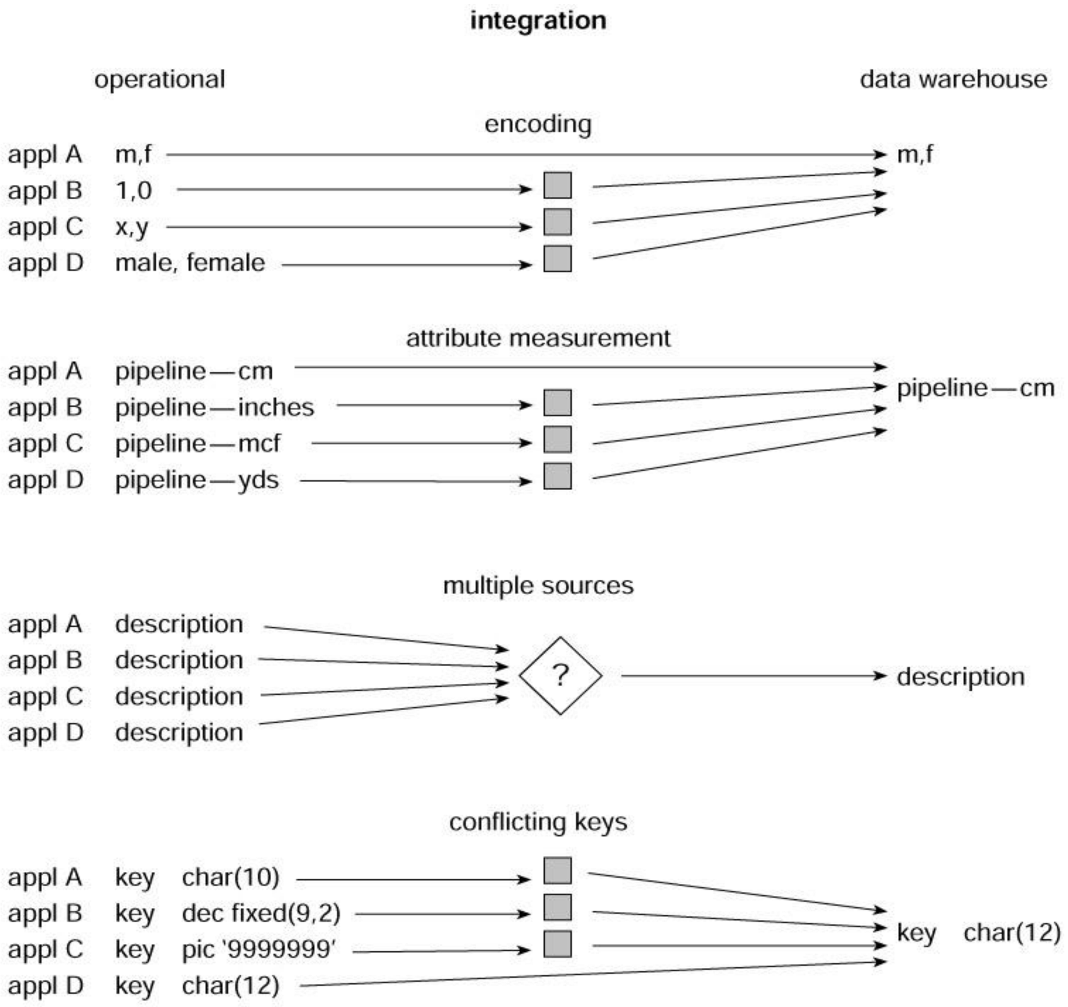
collection of data in support of management’s decisions.
The data warehouse contains granular corporate data.

Subject-Oriented Data Collections



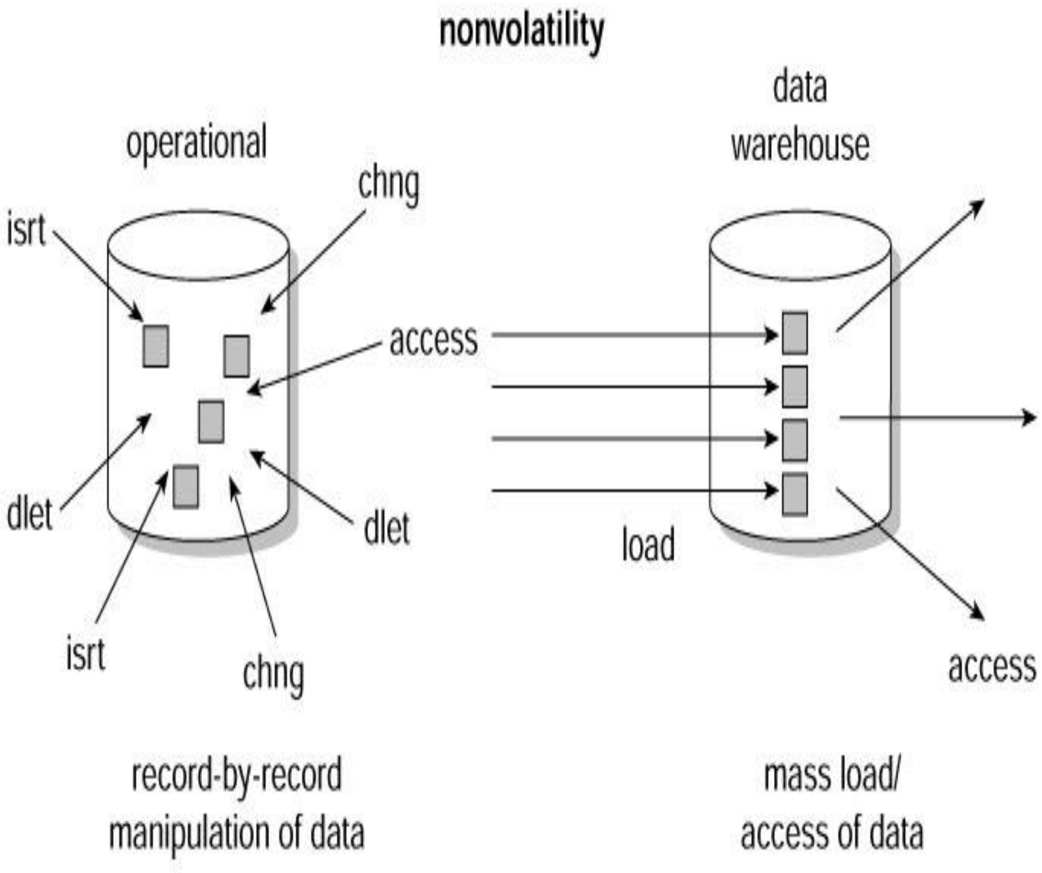
Classical operations systems are organized around the applications of the company. For an insurance company, the applications may be auto, health, life, and casualty. The major subject areas of the insurance corporation might be customer, policy, premium, and claim. For a manufacturer, the major subject areas might be product, order, vendor, bill of material, and raw goods. For a retailer, the major subject areas may be product, SKU, sale, vendor, and so forth. Each
type of company has its own unique set of subjects

Integrated Data Collections



Of all the aspects of a data warehouse, integration is the most important. Data is fed from multiple disparate sources into the data warehouse. As the data is fed it is
converted, reformatted, resequenced, summarized, and so forth. The result is that data—once it resides in the data warehouse—has a single physical corporate image.

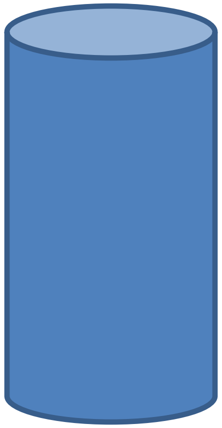
Non-volatile Data Collections



Data is updated in the operational environment as a regular matter of course, but warehouse data exhibits a very different set of characteristics. Data warehouse data is loaded (usually en masse) and accessed, but it is not updated (in the general sense). Instead, when data in the data warehouse is loaded, it is loaded in a snapshot, static format. When subsequent changes occur, a new snapshot record is written. In doing so a history of data is kept in the data warehouse.

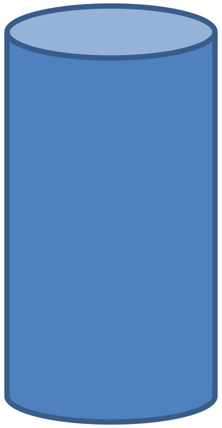
Time-variant Data Collections

Operational



- Time horizon – 1-2 years.
- Update of records
- Key structure may/may not contain an element of time

Data Warehouse



- Time horizon – 5-15 years.
- Sophisticated snapshots of data
- Key structure contains an element of time

Time variancy implies that every unit of data in the data warehouse is accurate as of some one moment in time. In some cases, a record is time stamped. In other cases, a record has a date of transaction. But in every case, there is some form of time marking to show the moment in time during which the record is accurate. A 1 to 2 year time horizon is normal for operational systems; a 5-to-15-year time horizon is normal for the data warehouse. As a result of this difference in time horizons, the data warehouse contains *much* more history than any other environment.

Operational Data Store (ODS)

The Operational Data Store is used for tactical decision making while the DW supports strategic decisions. It contains transaction data, at the lowest level of detail for the subject area

- subject-oriented, just like a DW
- integrated, just like a DW
- volatile (or updateable) , **unlike** a DW
  - an ODS is like a transaction processing system
  - information gets overwritten with updated data
  - no history is maintained (other than audit trail) or operational history
- current, i.e., not time-variant, **unlike** a DW
  - current data, up to a few years
  - no history is maintained (other than audit trail) or operational history

Data Warehouses and Data Marts

A **data warehouse** is a central repository for all or significant parts of the data that an enterprise's various business systems collect. Enables strategic decision making.

A **data mart** is a repository of data gathered from operational data and other sources that is designed to serve a particular community of knowledge workers. In scope, the data may derive from an enterprise-wide database or data warehouse or be more specialized. The emphasis of a data mart is on meeting the specific demands of a particular group of knowledge users in terms of analysis, content, presentation, and ease-of-use. Users of a data mart can expect to have data presented in terms that are familiar.

In practice, the terms *data mart* and *data warehouse* each tend to imply the presence of the other in some form. However, most writers using the term seem to agree that the design of a data mart tends to start from an analysis of user needs and that a data warehouse tends to start from an analysis of what data already exists and how it can be collected in such a way that the data can later be used. A data warehouse is a central aggregation of data (which can be distributed physically); a data mart is a data repository that may derive from a data warehouse or not and that emphasizes ease of access and usability for a particular designed purpose.

A data warehouse tends to be a strategic but somewhat unfinished concept; a data mart tends to be tactical and aimed at meeting an immediate need.

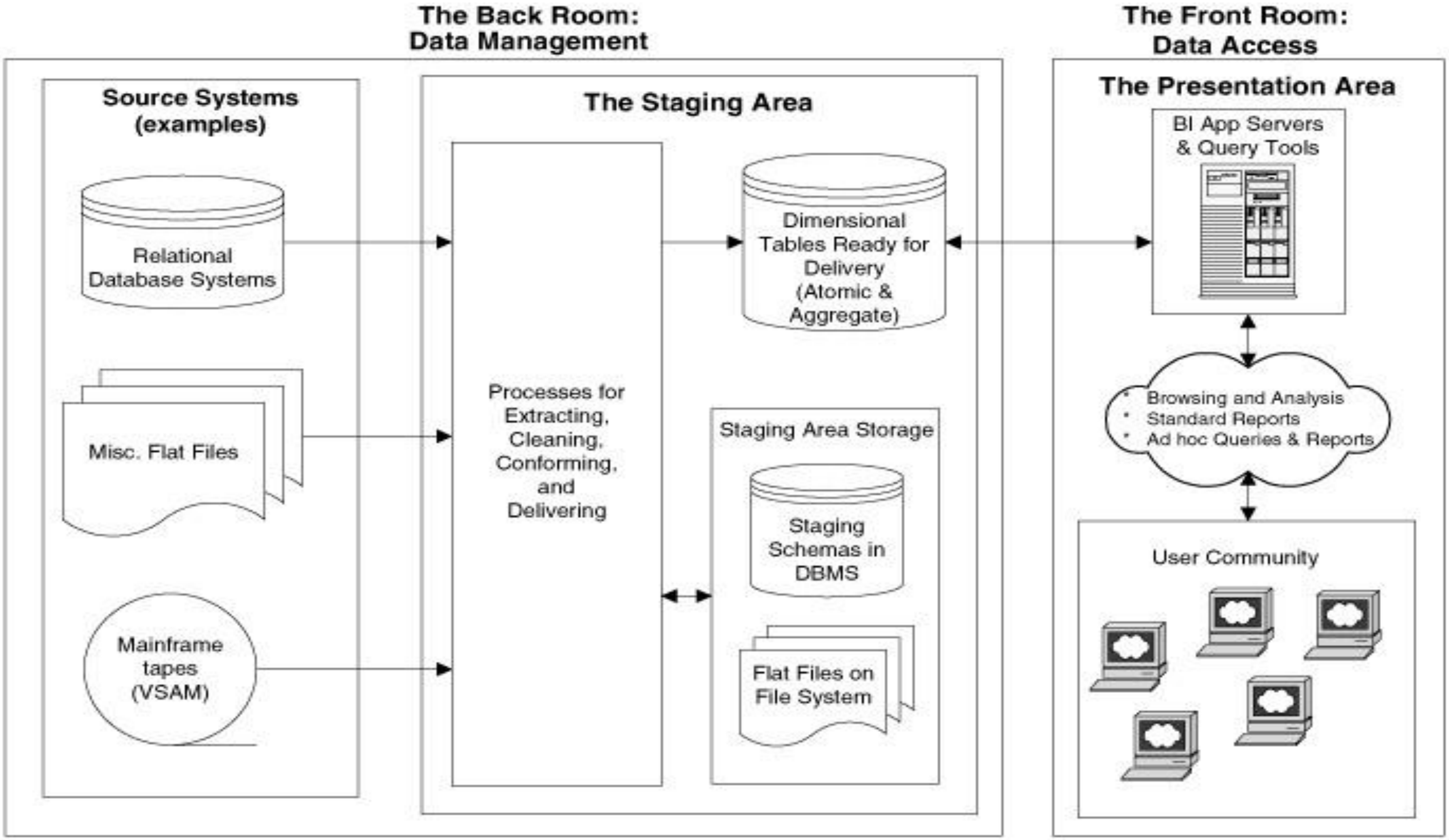
The goals of a Data Warehouse

- We have mountains of data in this company, but we can't access it."
- "We need to slice and dice the data every which way."
- "You've got to make it easy for business people to get at the data directly."
- "Just show me what is important."
- "It drives me crazy to have two people present the same business metrics at a meeting, but with different numbers."
- "We want people to use information to support more fact-based decision making."

The goals of a Data Warehouse

- The data warehouse must make an organization's information easily accessible.
- The data warehouse must present the organization's information consistently.
- The data warehouse must be adaptive and resilient to change.
- The data warehouse must be a secure bastion that protects our information assets.
- The data warehouse must serve as the foundation for improved decision making.
- The business community must accept the data warehouse if it is to be deemed successful.

Data Warehouse Architecture



Requirements for DW

Security Requirements

- a paradox:
  - Data Warehouse: publish data widely
  - Security: restrict data to those with a need to know
- role-based security at the final applications (not `grant` or `revoke` at the DBMS level)
- security for developers (separate subnet), backups (tapes, disks)

Requirements for DW(cont'd)

Data Integration

- at the core of the IT business, aka, “the 360 degree view of the business”
- specific to Data Warehouses: establishing common attributes (conforming dimensions), agreeing on common business metrics (conforming facts) so that one can perform mathematical calculations (differences, ratios, etc)

Requirements for DW (cont'd)

Data Latency

- how quickly to deliver data to the end user
- improvements with algorithms, parallel processing, streaming

Archiving

- change calculations
- legal compliance lineage requirements

End User

- reports, OLAP, data handoff

Technical Requirements for DW

Architecture

- ETL tool versus hand coding
- batch updates versus data streaming
- horizontal (orders/shipments) versus vertical (customers/orders) task dependency
- scheduler automation
- quality handling/data cleansing
- metadata
- security
- staging

Data Warehouse Design

Start with the ER-Diagram that represents the corporate data model or with one ore more operational data models to be integrated

Remove data used purely in the operational environment

Enhance key structures with an element of time

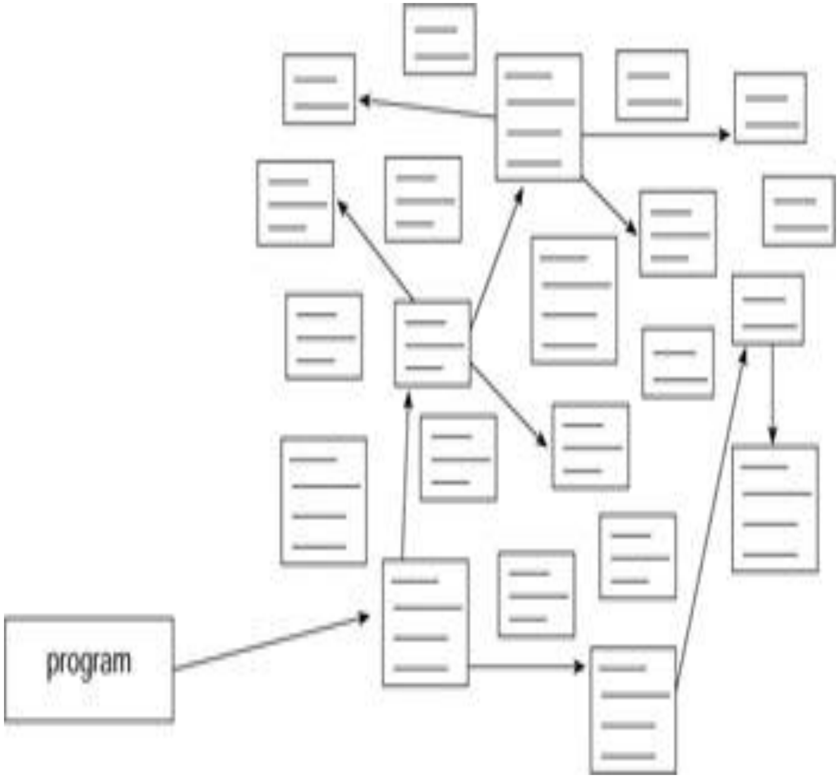
Add derived (calculated) data (i.e., summaries)

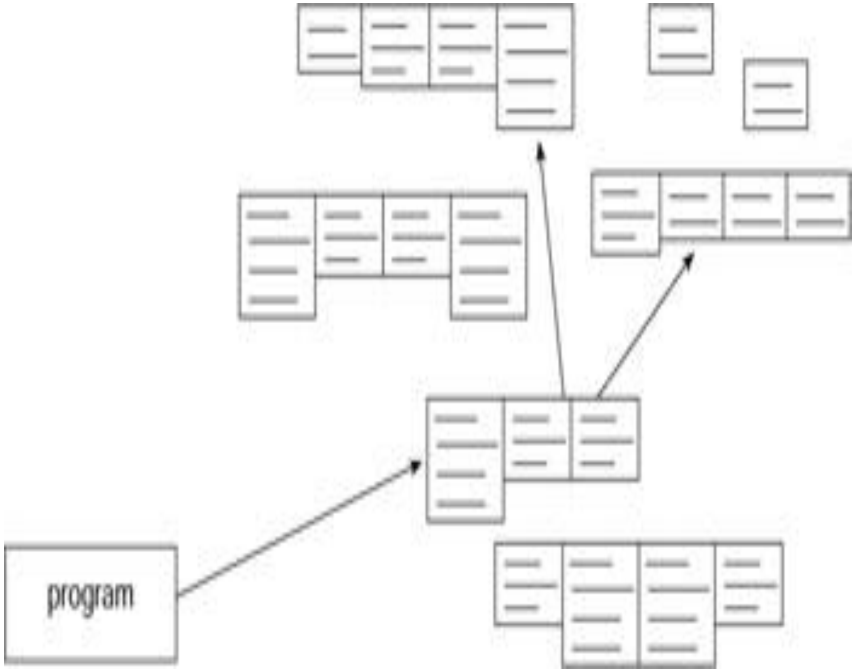
Turn “relationships” of the ER model into “artifacts” in the data warehouse

Design Techniques: Merging Tables

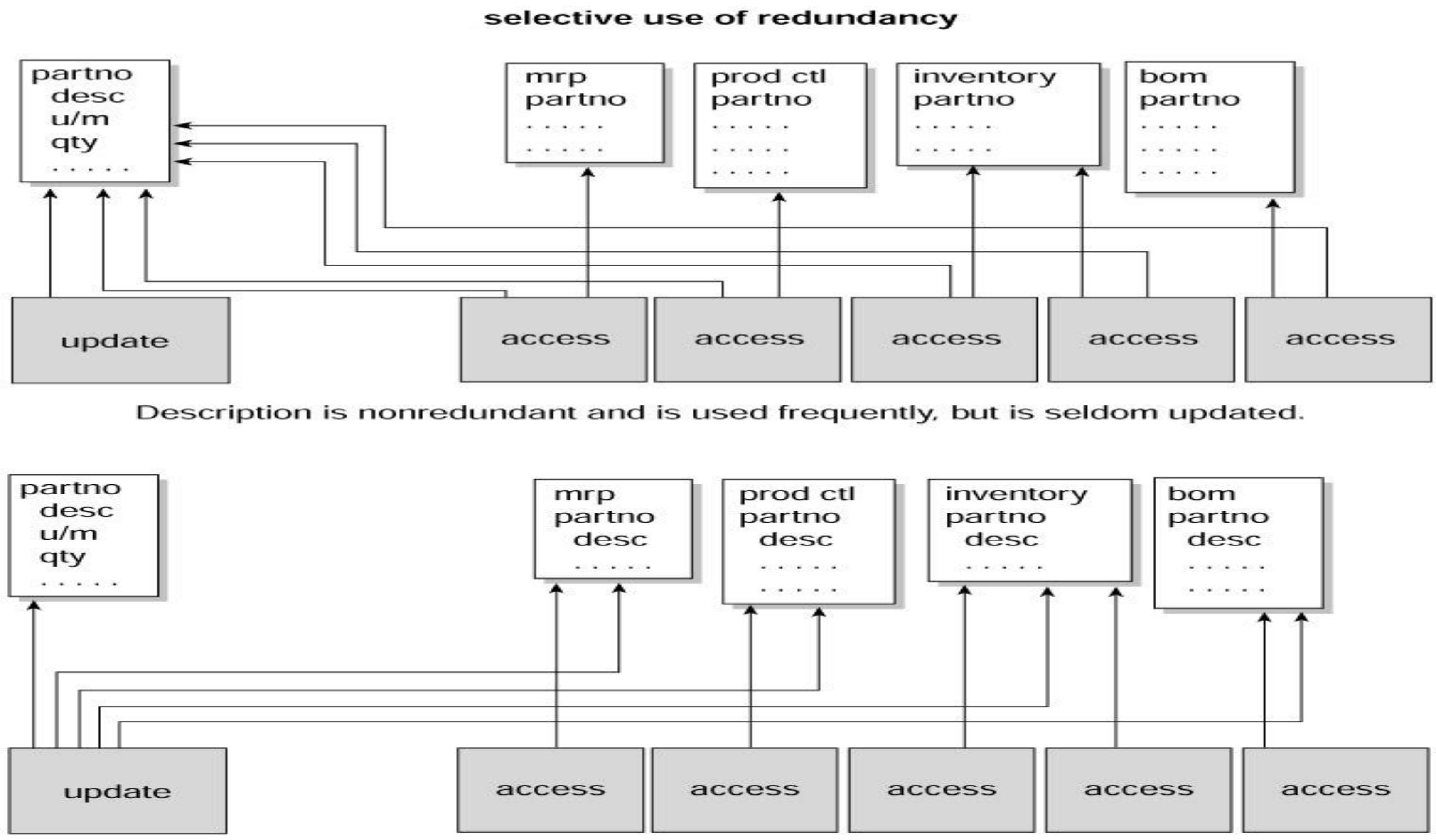
Many tables imply much dynamic I/O

Merging many tables together makes access faster



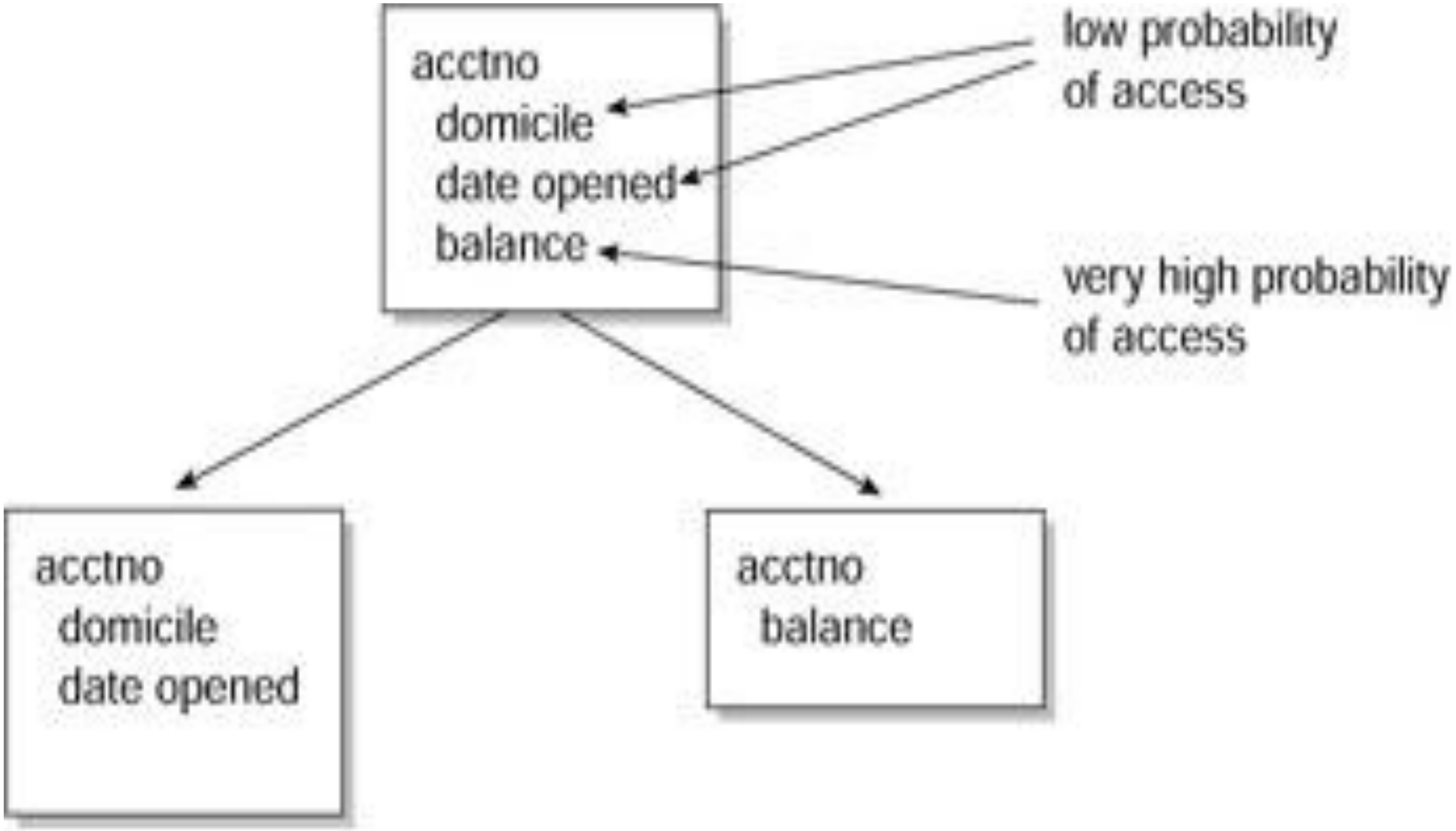


Design Techniques: Introduction of Redundant Data



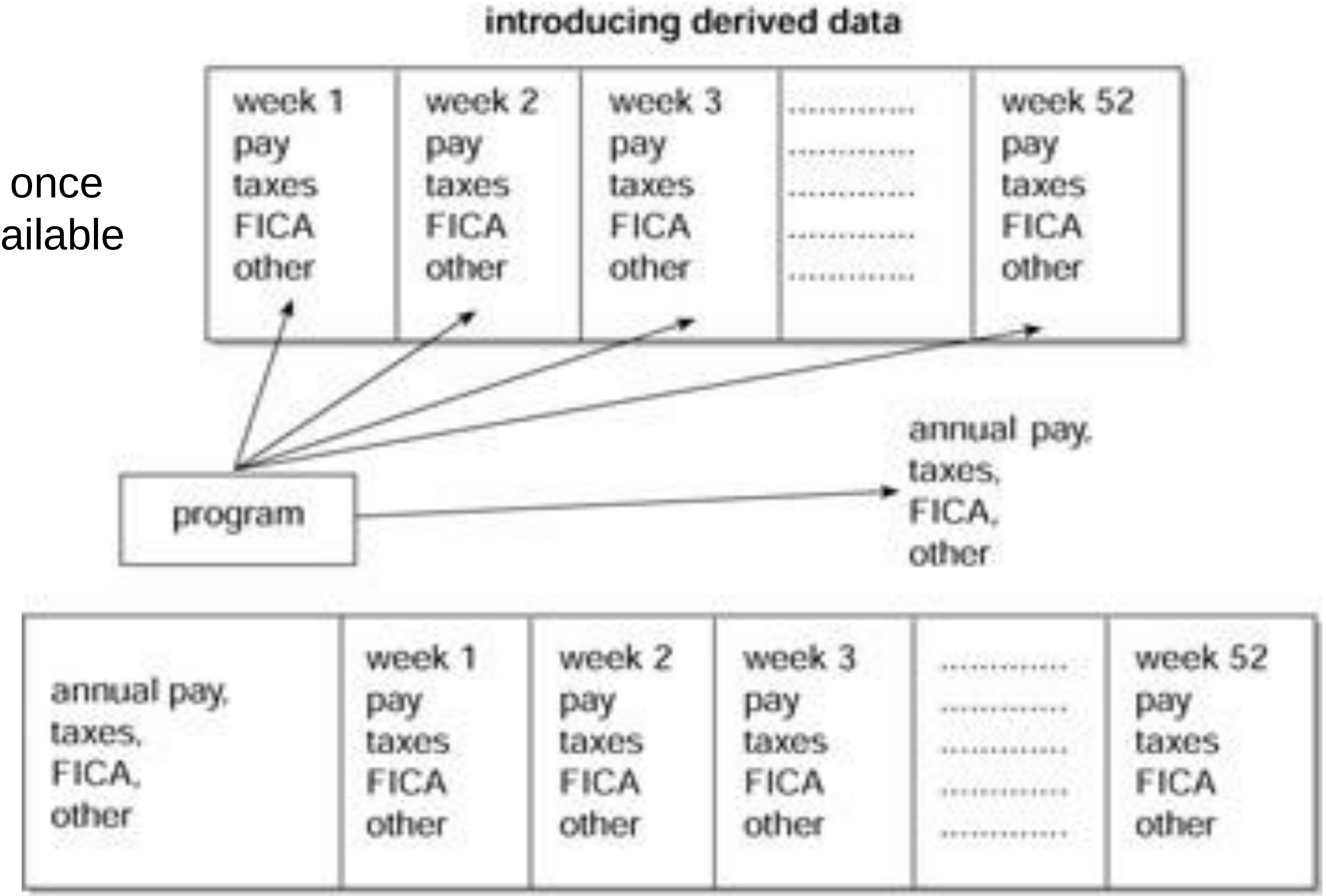
Design Technique:

Separation of Data when there is a disparity of probability of access



Design Technique: Introduce Derived Data

Calculated once
Forever available



introducing derived data

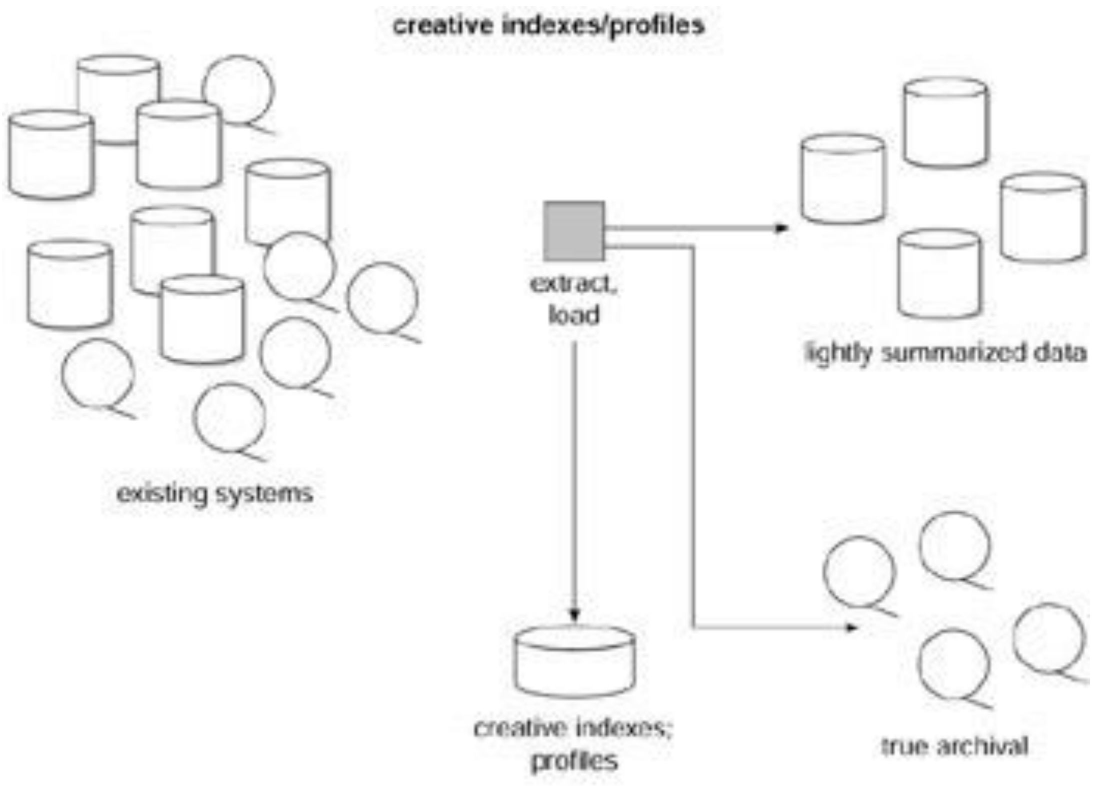
week 1	week 2	week 3	....	week 52
pay	pay	pay	....	pay
taxes	taxes	taxes	....	taxes
FICA	FICA	FICA	....	FICA
other	other	other	....	other

program

annual pay,
taxes,
FICA,
other

annual pay, taxes, FICA, other	week 1	week 2	week 3	....	week 52
	pay	pay	pay	....	pay
	taxes	taxes	taxes	....	taxes
	FICA	FICA	FICA	....	FICA
	other	other	other	....	other

Design Techniques: Creative Indexes



- The top 10 customers
- The average transaction value for this extract
- The largest transaction
- The number of customers who showed activity without purchasing.

Design Technique: Forget Referential Integrity

In the operational environment, referential integrity appears as a dynamic link among tables of data.

Not in a data warehouse because

- volume of data is too large
- the data warehouse is not updated, just appended to
- the warehouse represents data over time and relationships do not remain static

relationships of data are represented by an artifact in the data warehouse environment. Therefore, some data will be duplicated, and some data will be deleted when other data is still in the warehouse. In any case, trying to replicate referential integrity in the data warehouse environment is a **patently incorrect approach**.

Dimensional Modeling

Data modeling for Data Warehouses

Based on

- fact tables
- dimension tables

Fact Tables

- Represent a business process, i.e., models the business process as an artifact in the data model
- contain the **measurements** or metrics or facts of business processes
  - "monthly sales number" in the **Sales** business process
  - most are additive (sales this month), some are semi-additive (balance as of), some are not additive (unit price)
- the level of detail is called the "grain" of the table
- contain foreign keys for the dimension tables

Fact Tables

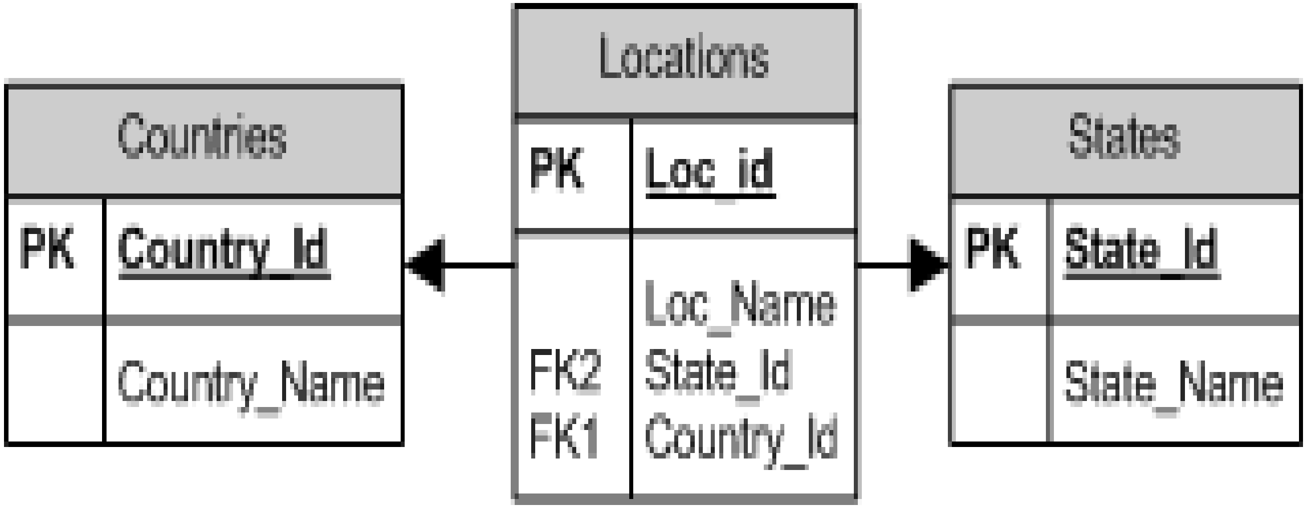
- Represent a business process, i.e., models the business process as an artifact in the data model
- contain the **measurements** or metrics or facts of business processes
  - "monthly sales number" in the **Sales** business process
  - most are additive (sales this month), some are semi-additive (balance as of), some are not additive (unit price)
- the level of detail is called the “grain” of the table
- contain foreign keys for the dimension tables

Dimension Tables

- Represent the *who, what, where, when* and *how* of a measurement/artifact
- Represent real-world entities not business processes
- Give the context of a measurement (subject)
- For example for the **Sales fact table**, the characteristics of the 'monthly sales number' measurement can be a Location (*Where*), Time (*When*), Product Sold (*What*).
- The **Dimension Attributes** are the various columns in a dimension table. In the Location dimension, the attributes can be Location Code, State, Country, Zip code. Generally the Dimension Attributes are used in report labels, and query constraints such as *where Country='USA'.* The dimension attributes also contain one or more hierarchical relationships.
- Before designing your data warehouse, you need to decide what this data warehouse contains. Say if you want to build a data warehouse containing monthly sales numbers across multiple store locations, across time and across products then your dimensions are:

  Location
  Time
  Product

Possible OLTP Location Design



Countries	
PK	Country_Id
	Country_Name

Locations	
PK	Loc_id
	Loc_Name
FK2	State_Id
FK1	Country_Id

States	
PK	State_Id
	State_Name

In order to query for all locations that are in country 'USA' we will have to join these three tables:

*SELECT **
FROM Locations, States, Countries
where Locations.State_Id = States.State_Id
AND Locations.Country_id=Countries.Country_Id
AND Country_Name='USA'

Location Dimension

Dim_id	Loc_cd	Name	State_NM	Country_NM
1001	IL01	Chicago Loop	Illinois	USA
1002	IL02	Arlington	Illinois	USA
1003	NY01	Brooklyn	New York	USA
1004	TO01	Toronto	Ontario	Canada
1005	MX01	Mexico City	Distrito Federal	Mexico

In order to query for all locations that are in country 'USA'

```
SELECT *
FROM Location_dim
where Country_Name='USA'
```

Note the redundancy

Time Dimension

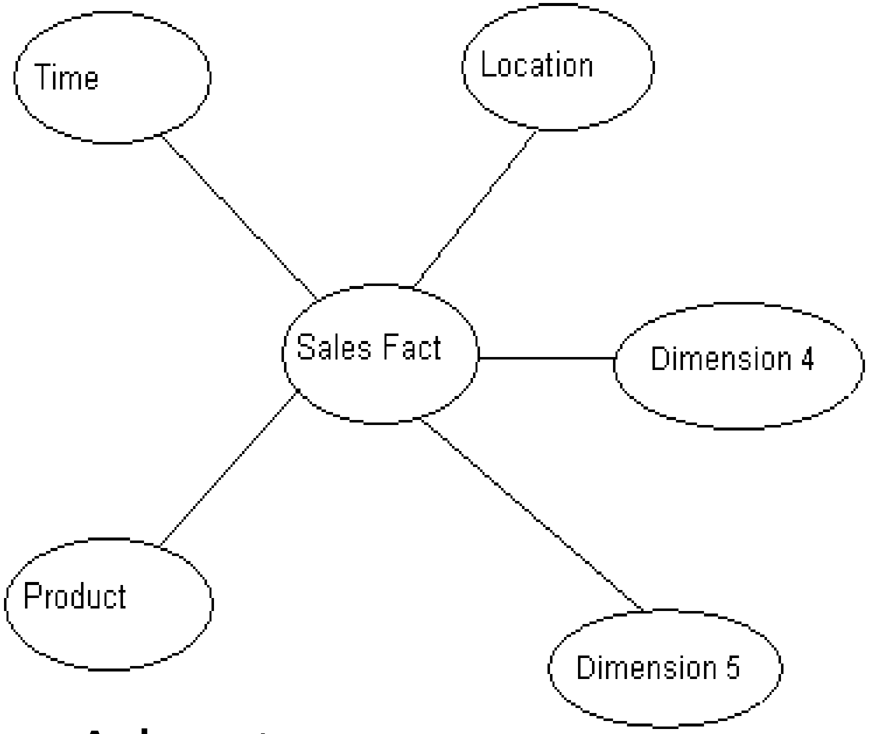
Dim_id	Month	Month-Name	Quarter	Quarter-Name	Year
1001	1	Jan	1	Q1	2005
1002	2	Feb	1	Q1	2005
1003	3	Mar	1	Q1	2005
1004	4	Apr	2	Q2	2005
1005	5	May	2	Q2	2005

Not as trivial at it seems ;-)

Product Dimension

Prod_id	Prod_cd	Name	Category
1001	STD	Short-Term-Disability	Disability
1002	LTD	Long-Term Disability	Disability
1003	GUL	Group Universal Life	Life
1004	PA	Personal Accident	Accident
1005	VADD	Voluntary Accident	Accident

Star Schemas



Select the measurements
SELECT P.Name, SUM(F.Sales)

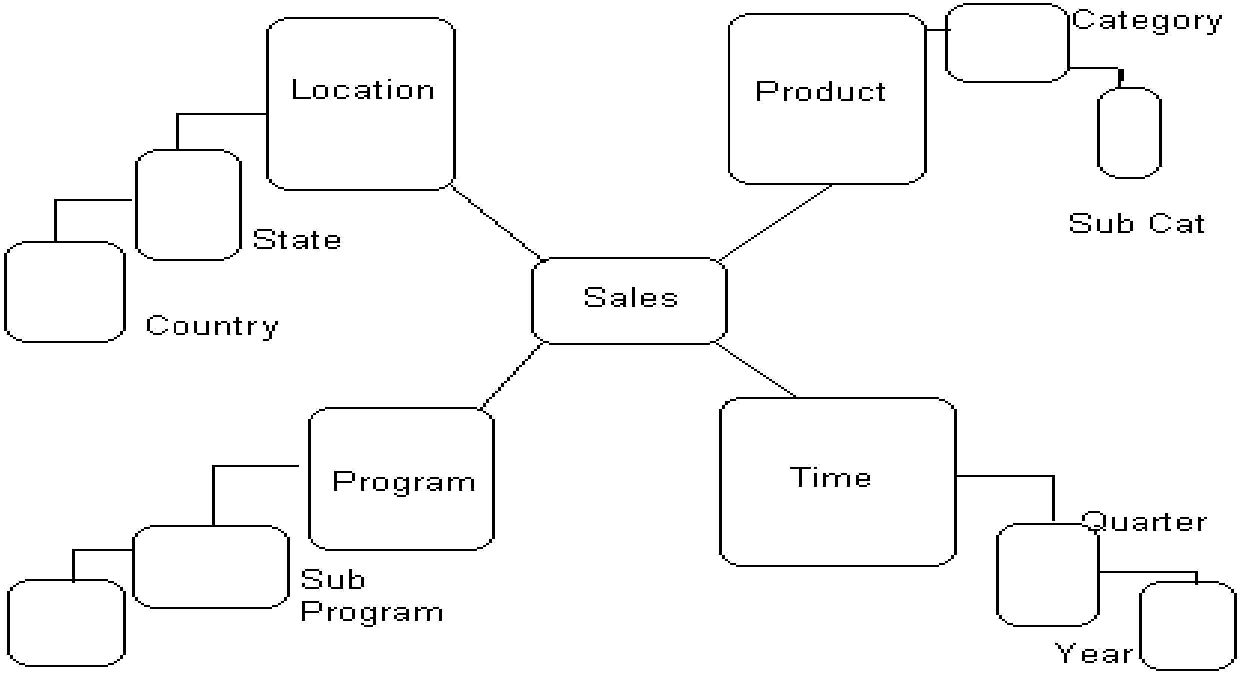
JOIN the FACT table with Dimensions
FROM Sales F, Time T, Product P,
Location L
WHERE F.TM_Dim_Id = T.Dim_Id
AND F.PR_Dim_Id = P.Dim_Id
AND F.LOC_Dim_Id = L.Dim_Id

Constrain the Dimensions
AND T.Month='Jan' AND T.Year='2003' AND L.Country_Name='USA'

'Group by' for the aggregation level
GROUP BY P.Category

Advantages:
-easy to understand
-better performance
-extensible

Snow-flake Schemas



If we did not de-normalize the dimensions

Rule of thumb:
don't use them

Dimensional Modeling Steps

- Identify the business process
- Identify the level of detail needed (grain)
- Identify the dimensions
- Identify the facts