Database Management Systems

- What is a Data Warehouse?
- Dimensional Modeling

Doug Shook

Data Integration Issues

- Data Redundancy
- Heterogeneous sources
  - DBMS, OLTP
  - Documents
  - Legacy
- Data designed with operational systems in mind
- Quality
- Volatility

Query Based Data Integration

- Data is integrated on demand
- Pros:
  - Access to up-to-date data
  - No data duplication
- Cons:
  - Delay in query processing
  - Competes with existing processes
  - Data losses cannot be recovered

Warehouse Based Data Integration

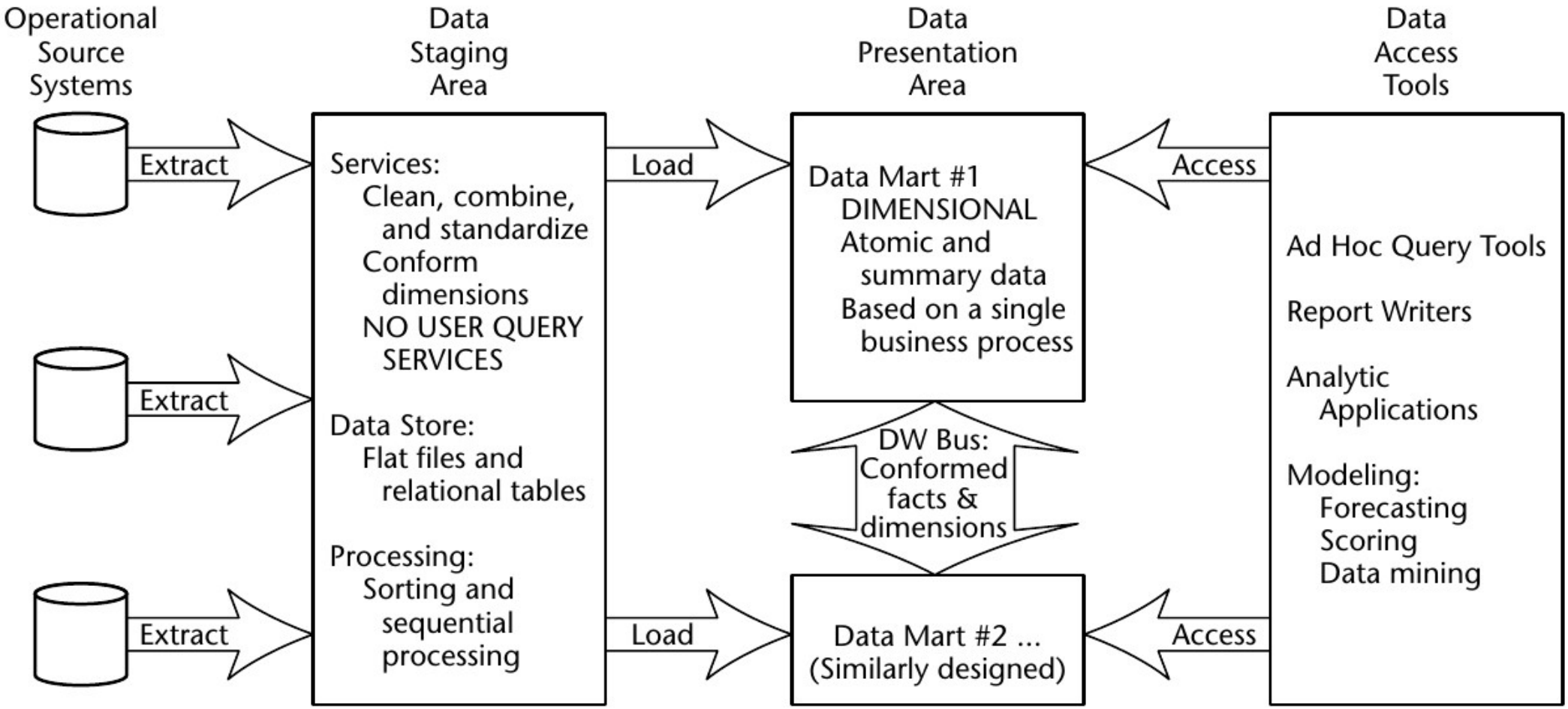
- Data is integrated in advance
- Pros:
  - Better performance
  - Does not interfere with business processes
- Cons:
  - Duplication of data
  - Updates not immediate

Data Warehouses

- Data Analysis Environment
  - Subject oriented
  - Integrated
  - Time variant
  - Stable

- Data warehouses store information in an organized, unified way
  - Assist in decision making

DW Structure



Elements of a Data Warehouse

- Operational Systems
  - Captures business transactions
  - Technically outside of DW
  - Primary data source

- Data Staging Area
  - Responsible for Extract-Transform-Load
    - Extract from operational systems
    - Transform data to make it suitable for presentation
    - Perform bulk loading into data marts

Elements of a Data Warehouse

- Data Presentation
  - Data Marts
  - Broken down by departments
  - Accessible by business users
  - Contains raw data as well as metadata (aggregates, summaries)
  - Relies on dimensional modeling
- Data Access Tools
  - Ad-hoc querying
  - Data mining

OLTP vs. OLAP

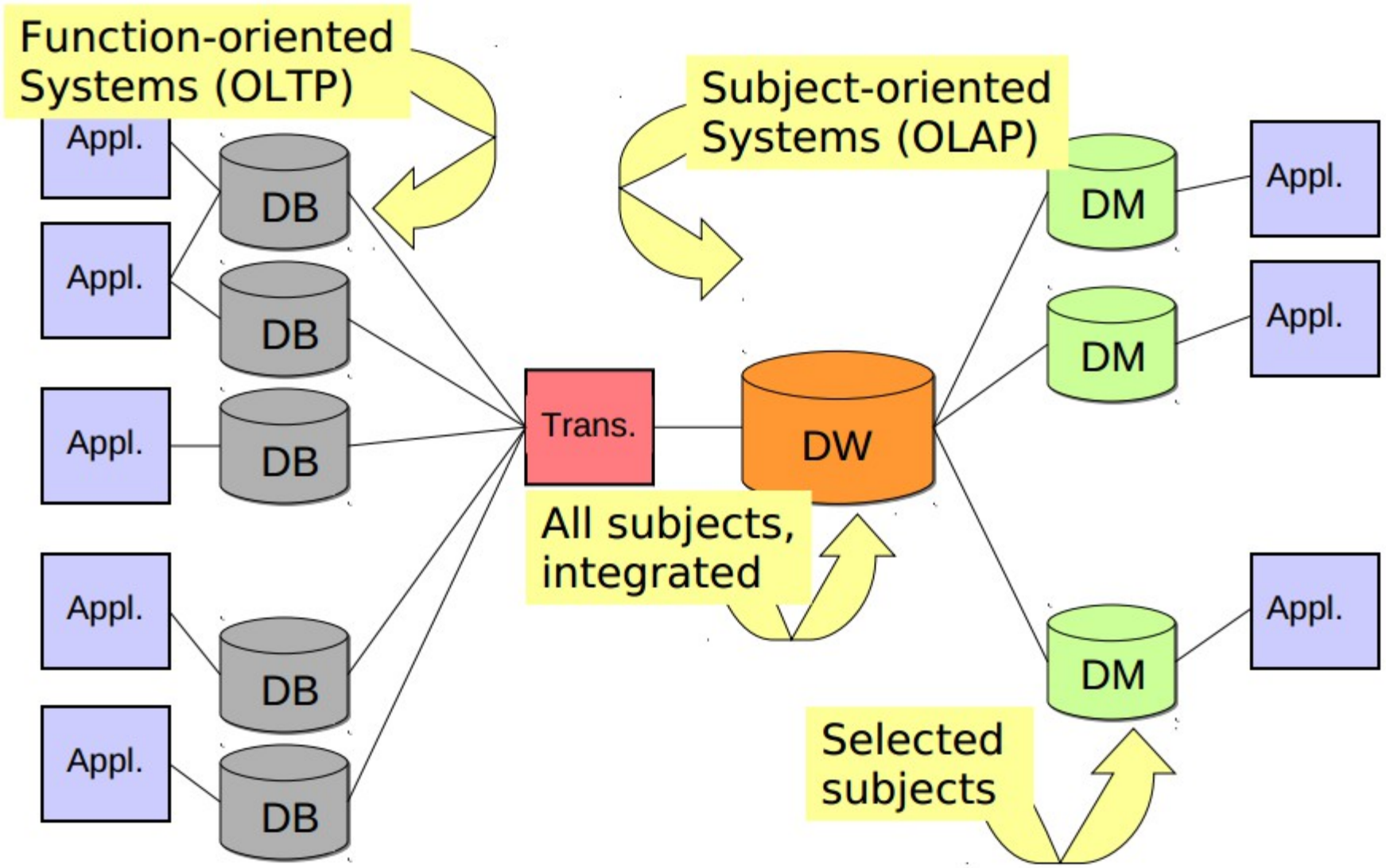
- On-line Transaction Processing
  - Many small, fast queries
  - Geared towards business processes (functions)
  - Less historical data
  - Frequent updates

- On-line Analytical Processing
  - Fewer, larger queries
  - Fewer, larger updates
  - More historical data
  - Dimensional / subject oriented

Example Queries

- Which product has been the most profitable in the past ten years?
- Which week of the year do we have the most sales?
- Do the sales of product X increase over time?
- These queries are difficult for OLTP systems to answer. Why?

DW Structure



Multidimensional Modeling

- Relational Databases typically use normalized data
  - Eliminates redundancies
  - Provides for fast insertions
  - Modeled through ER diagrams

- This approach works well for operational systems, but poorly for data warehouses
  - Why?

ER vs. Multidimensional

- ER
  - One table per entity
  - Minimize data redundancy
  - Optimize update
  - OTLP
- Multidimensional
  - One fact table per process
  - Maximize understandibility
  - Optimized for retrieval
  - OLAP

Multidimensional Modeling

- Design Concepts
  - Facts
  - Measures
  - Dimensions
  - Hierarchies
- Logical Design Types
  - Star Schemas (Cubes)
  - Snowflake Schemas

Fact Tables

- Purpose: store performance measurements from business process events
  - Sales, shipments, purchases, etc.
  - Events have a 1 to 1 relationship with rows in a fact table

Daily Sales Fact Table
Date Key (FK)
Product Key (FK)
Store Key (FK)
Quantity Sold
Dollar Sales Amount

Fact Tables

- A measure is a numerical property of a fact
  - Sales Dollars, units, etc.
  - Most useful measures are also additive
    - I.e. Sales Dollars vs. Unit Price
- Fact tables contain true data
  - No filler
  - Typically quite sparse
- Foreign Keys link facts with dimensions

Types of Facts

- Transaction
  - A fact for every business event
- Snapshot
  - A fact for every dimension combination at a given time interval
  - Captures current status
- Accumulating Snapshot
  - Like regular snapshot, except cumulative

Granularity

- What does a single fact mean?
  - What does each row in a fact table represent?
  - Related to the primary key
- Small grain – lots of detail, lots of data
  - Every item scanned at check out
  - Every transaction of a bank account
- Large Grain – more efficient at loss of detail
  - Every sale made
  - Monthly statement data for a bank account

Dimension Tables

- Describe textual properties of facts
  - Who, what, when, where, how, and why
  - Typically discrete values
  - Limited range of values
  - Typically very large

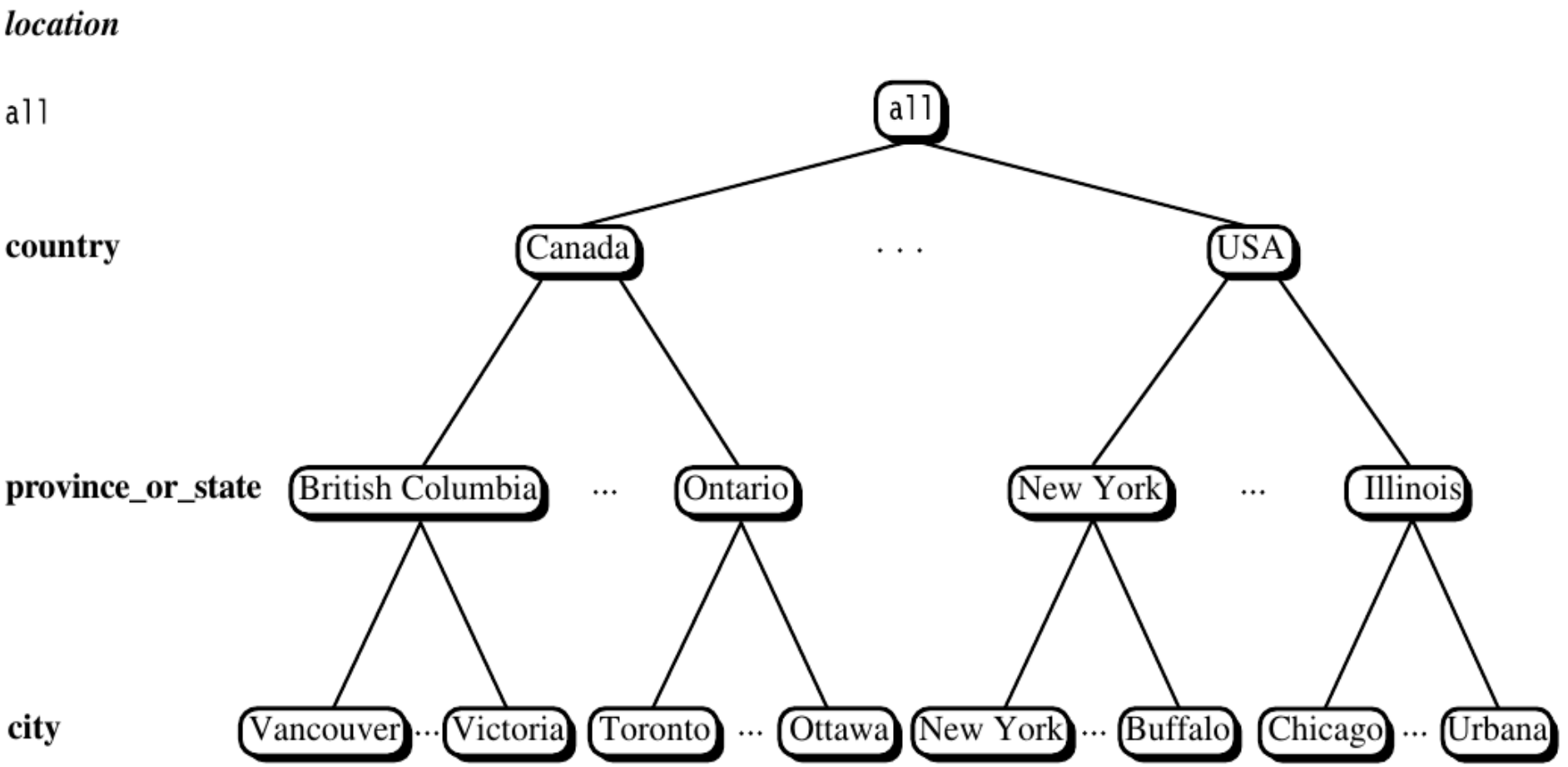
Product Dimension Table
Product Key (PK)
Product Description
SKU Number (Natural Key)
Brand Description
Category Description
Department Description
Package Type Description
Package Size
Fat Content Description
Diet Type Description
Weight
Weight Units of Measure
Storage Type
Shelf Life Type
Shelf Width
Shelf Height
Shelf Depth
... and many more

Dimension Tables

- Vital role within the data warehouse
  - Grouping
  - Constraints
  - Labels
- Avoid using cryptic descriptions
  - Spell it out
  - Use real language
  - If you must use codes, include another column that describes the encoded data

Multidimensional Modeling

- Dimension tables often represent hierarchies:

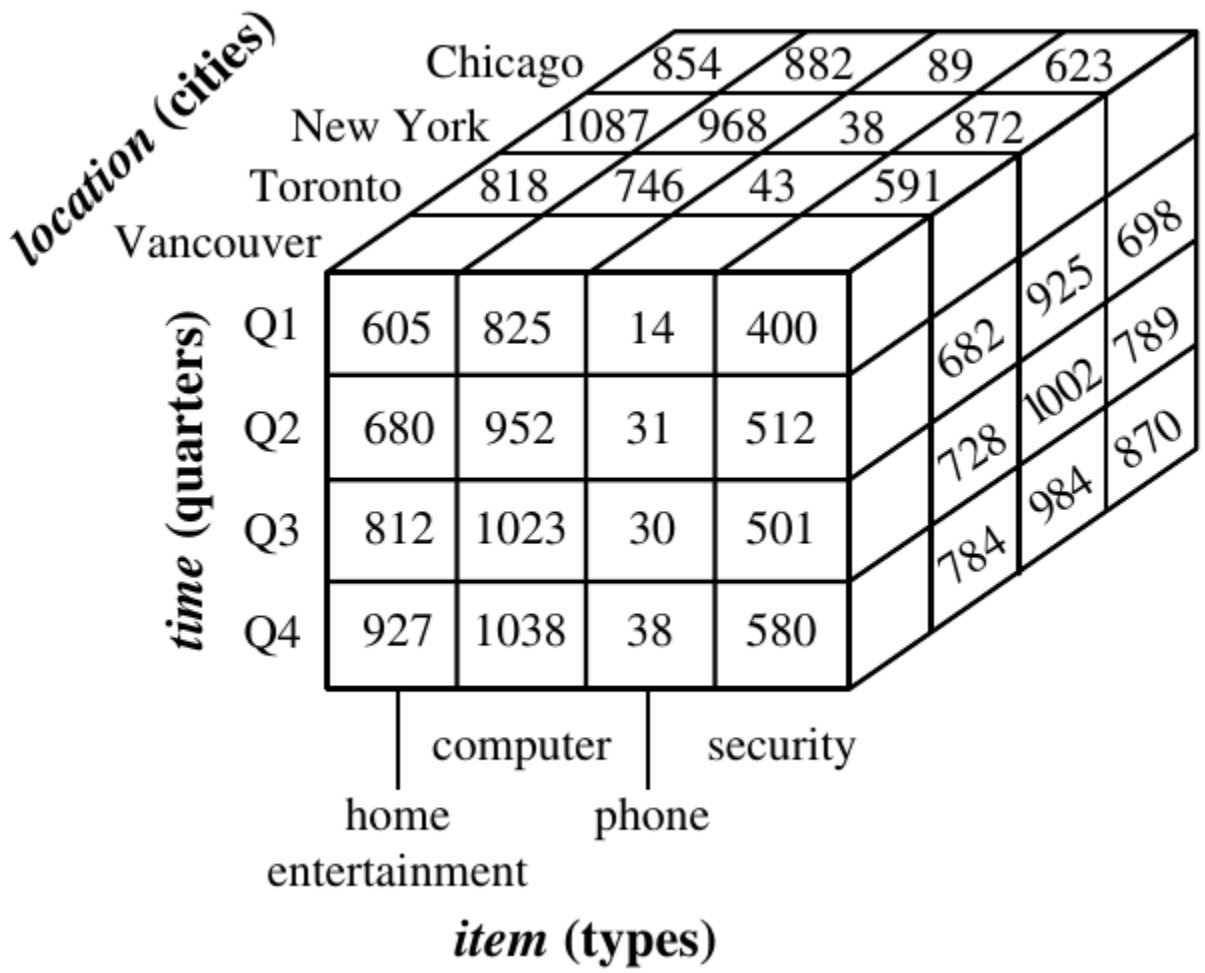


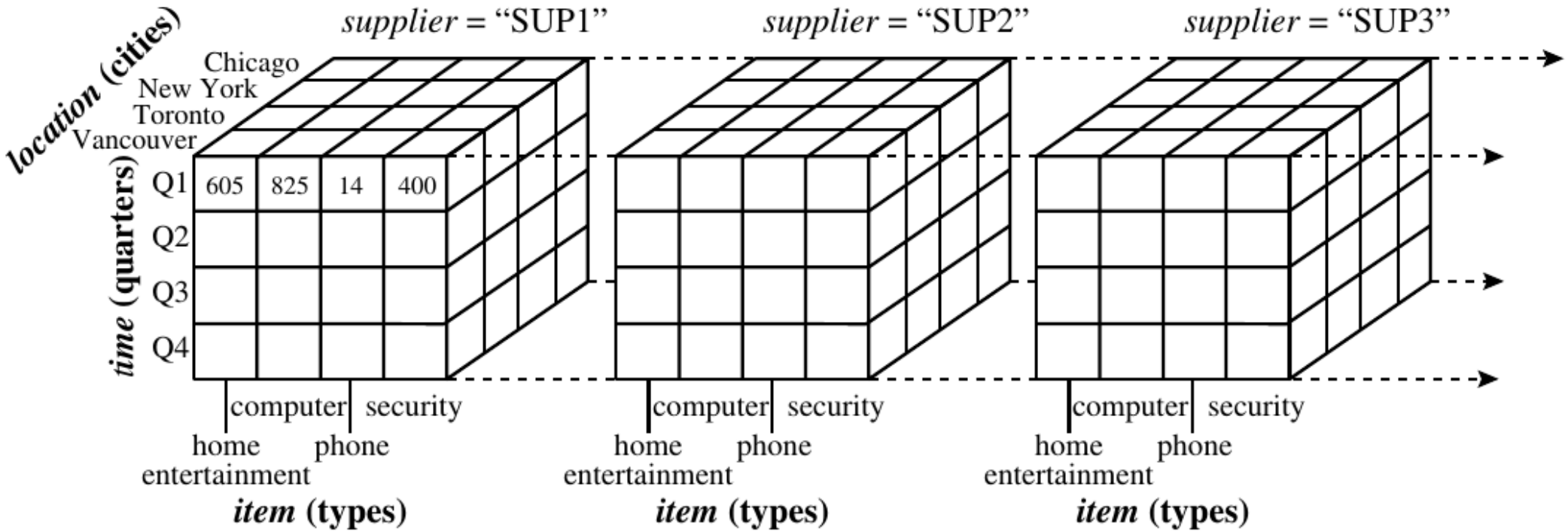
- Data in hierarchies is redundant, but that's OK!

Data Cubes

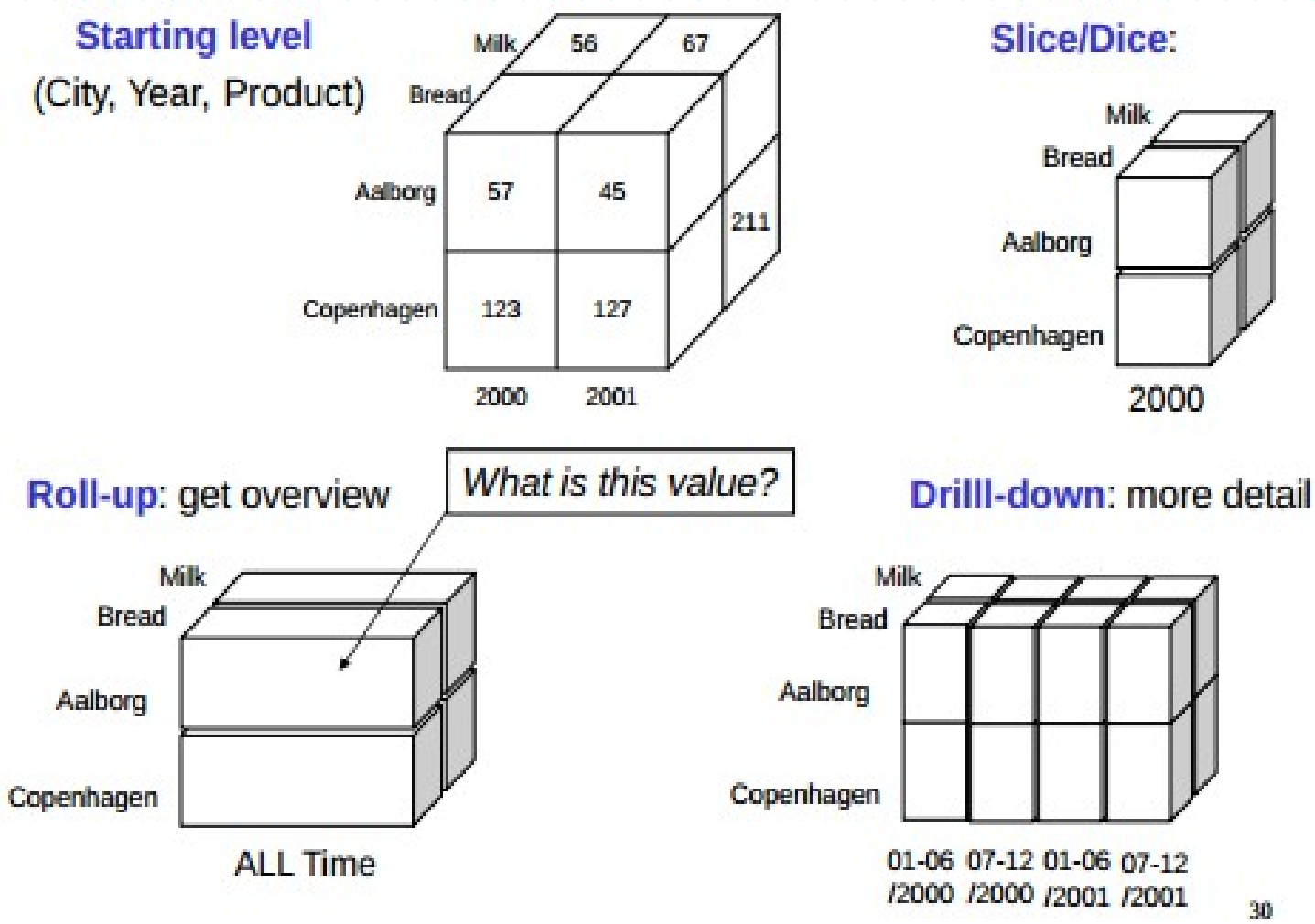
- Defined by a combination of dimensions and facts
  - Can have many dimensions, usually > 4
  - Only 2-3 can be viewed at one time

- Cubes are represented by cells
  - Combination of dimension values
  - Can be empty


location (cities)
Chicago
New York
Toronto
Vancouver
854
882
89
623
1087
968
38
872
818
746
43
591
time (quarters)
Q1
Q2
Q3
Q4
605
825
14
400
680
952
31
512
812
1023
30
501
927
1038
38
580
682
925
698
728
1002
789
784
984
870
computer
security
home
entertainment
phone
item (types)


supplier = "SUP1"
supplier = "SUP2"
supplier = "SUP3"
location (cities)
Chicago
New York
Toronto
Vancouver
time (quarters)
Q1
Q2
Q3
Q4
605
825
14
400
computer
security
home entertainment
phone
item (types)

Data Cube Operations



Data Cube Operations

- Slice
  - Creating a cube with one fewer dimension
- Dice
  - Creating a sub-cube
- Drill down / up
  - Change level of detail of a dimension
- Roll up
  - Summarize data within a dimension

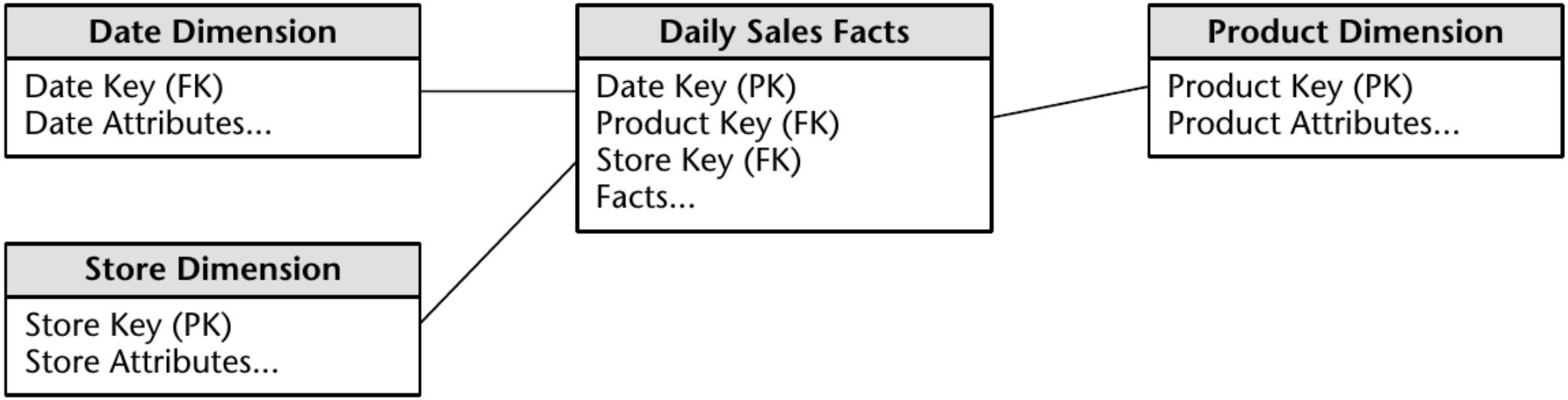
Data Cubes

- True building block of Dimensional Modeling
  - Certain DB platforms include syntax for dimensions, cubes, etc.
- Most of the world is built around RDBs
  - Enter star schemas: Data cubes modeled using ER like diagrams

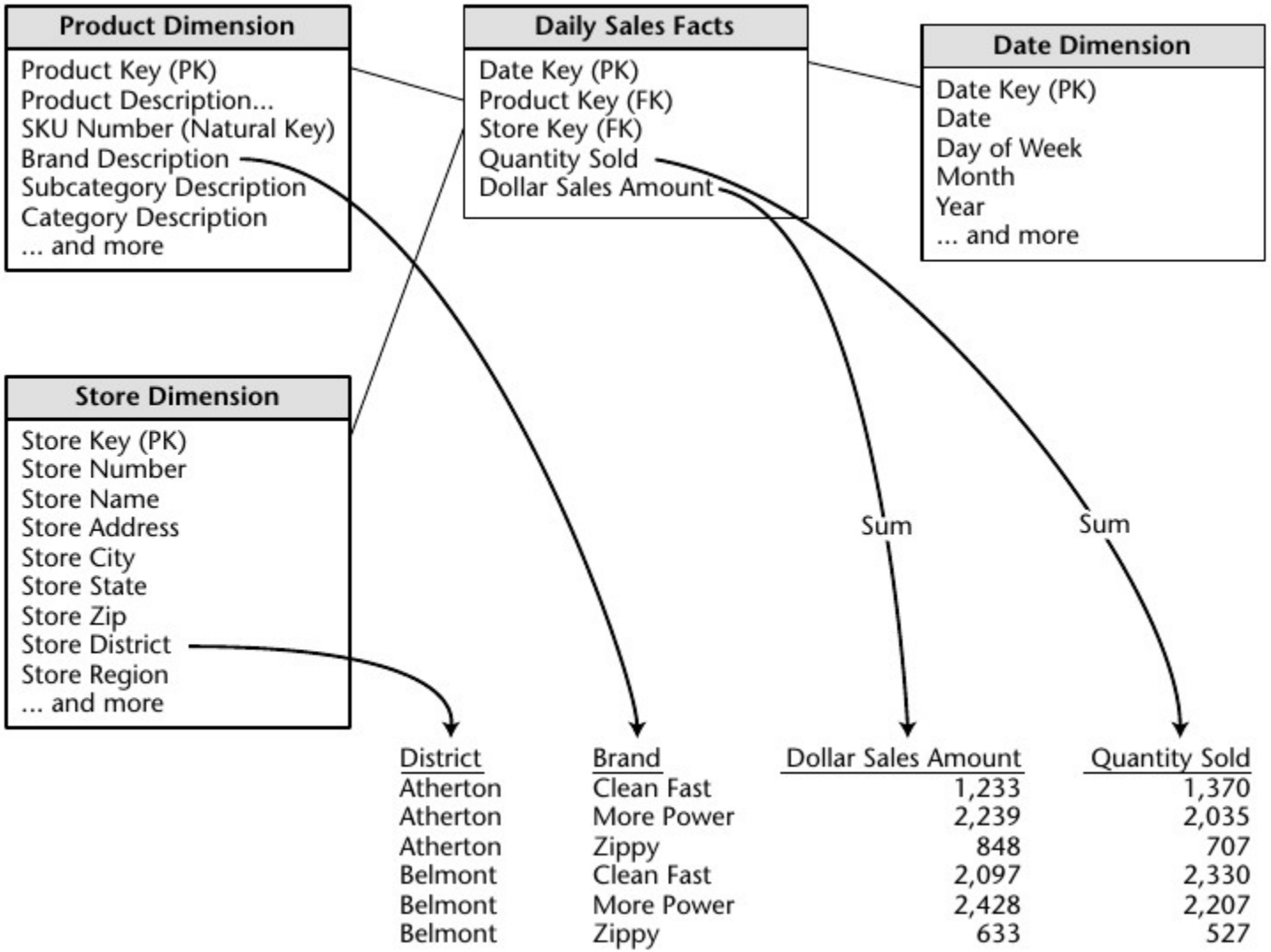
Star Schemas

- Simple way of modeling dimensional data cubes
  - Easy enough for non-technical users to understand
  - Fewer tables, increased performance
  - Easily extensible
    - New facts and dimensions can be added at any time

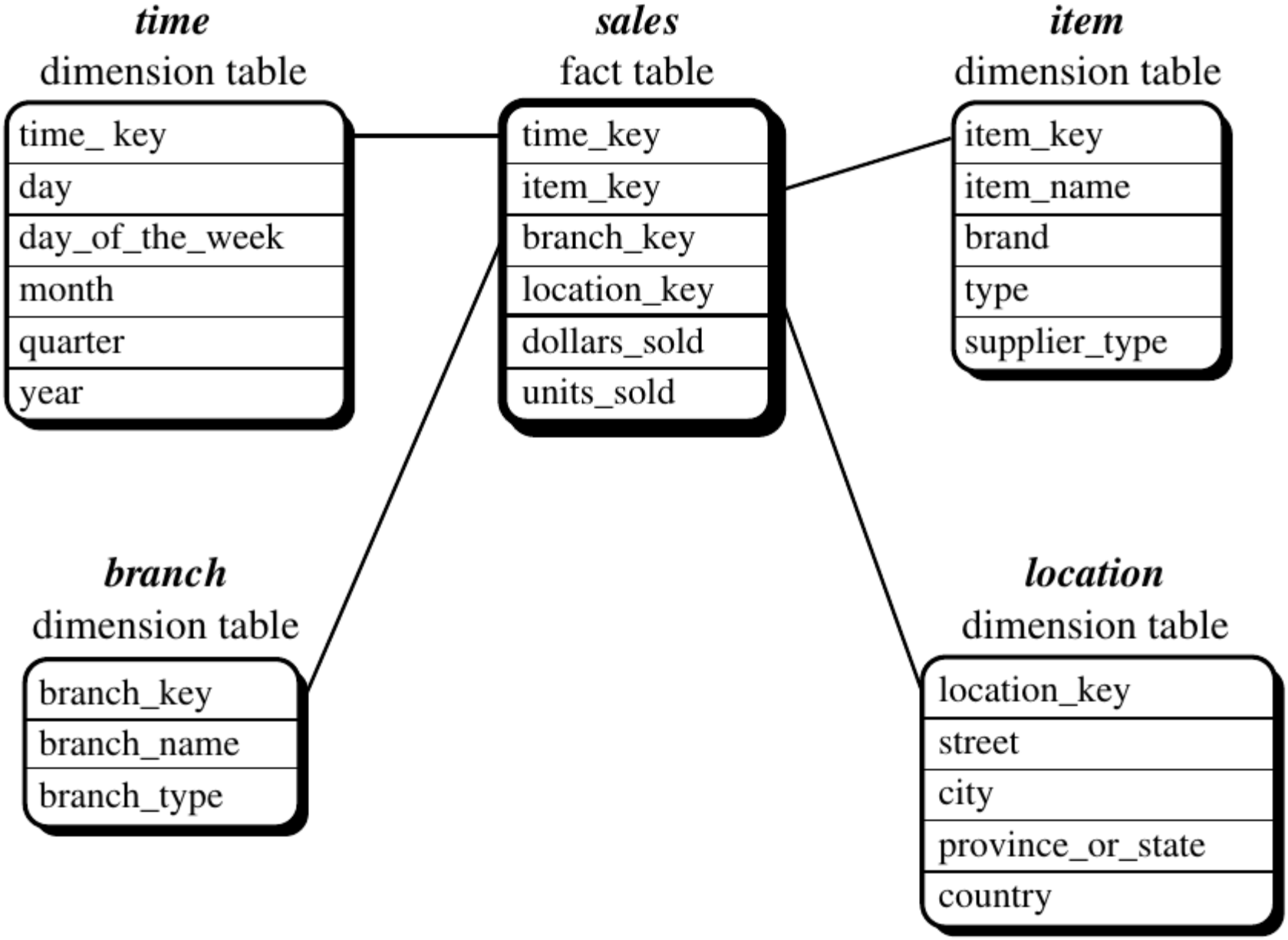
Star Schemas



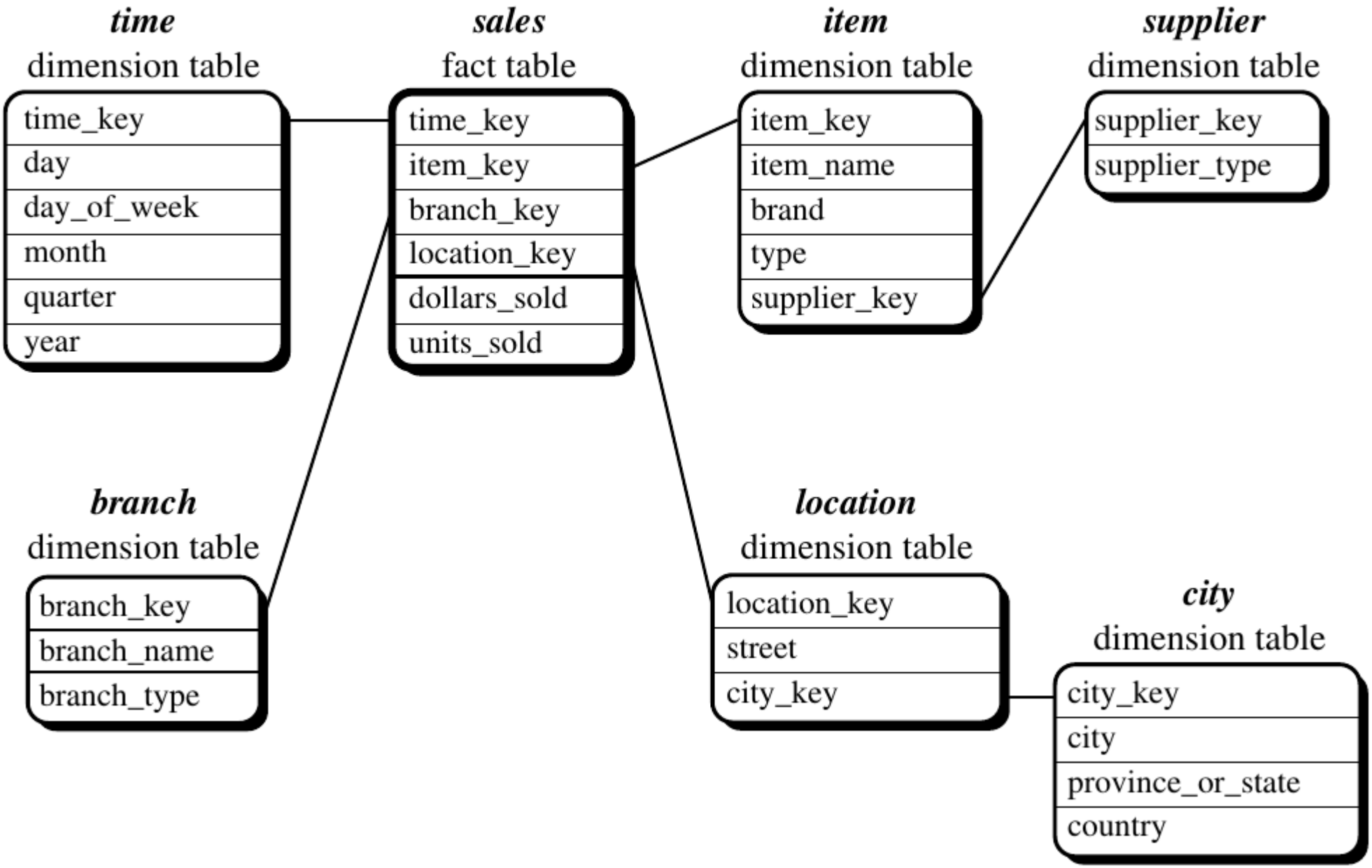
Generating Reports



Star vs. Snowflake



Star vs. Snowflake



Star vs. Snowflake

- Star Schemas
  - Simple and easy overview
  - Relatively flexible
  - Dimension tables often relatively small
  - Recognized by many RDBMSes
  - Hierarchies are ”hidden” in the columns
  - Dimension tables are de-normalized
- Snowflake schemas
  - Hierarchies are made explicit/visible
  - Very flexible
  - Dimension tables use less space
  - Harder to use due to many joins
  - Worse performance

Questions

- Which will typically be larger: a dimension table or a fact table? Why?

- Explain granularity, and the difference between small grained and large grained data. Do you think that one is preferable over the other? Why?

- What is the relationship between dimensions and facts (1-1, 1-M, M-M)?

- Why is it necessary for measures to be numerical? Can you think of an example of a non-numerical measure?

Questions

Would you expect to see more tables in an ER diagram or a star schema? Why?

Would you expect to see more columns in a fact table or a table in a RDBMS? Why?

Questions

Suppose you have a data warehouse for a university. This warehouse contains *student, course, semester,* and, *instructor* dimensions as well as measures *count,* and *avg_grade*.

- What possible hierarchies exist in this example? List at least one for each dimension.
- Draw a star schema for this example.
- Starting with a [*student, course, semester, instructor*] cube, which cube operations would be needed to display the average grade of computer science courses for each student?