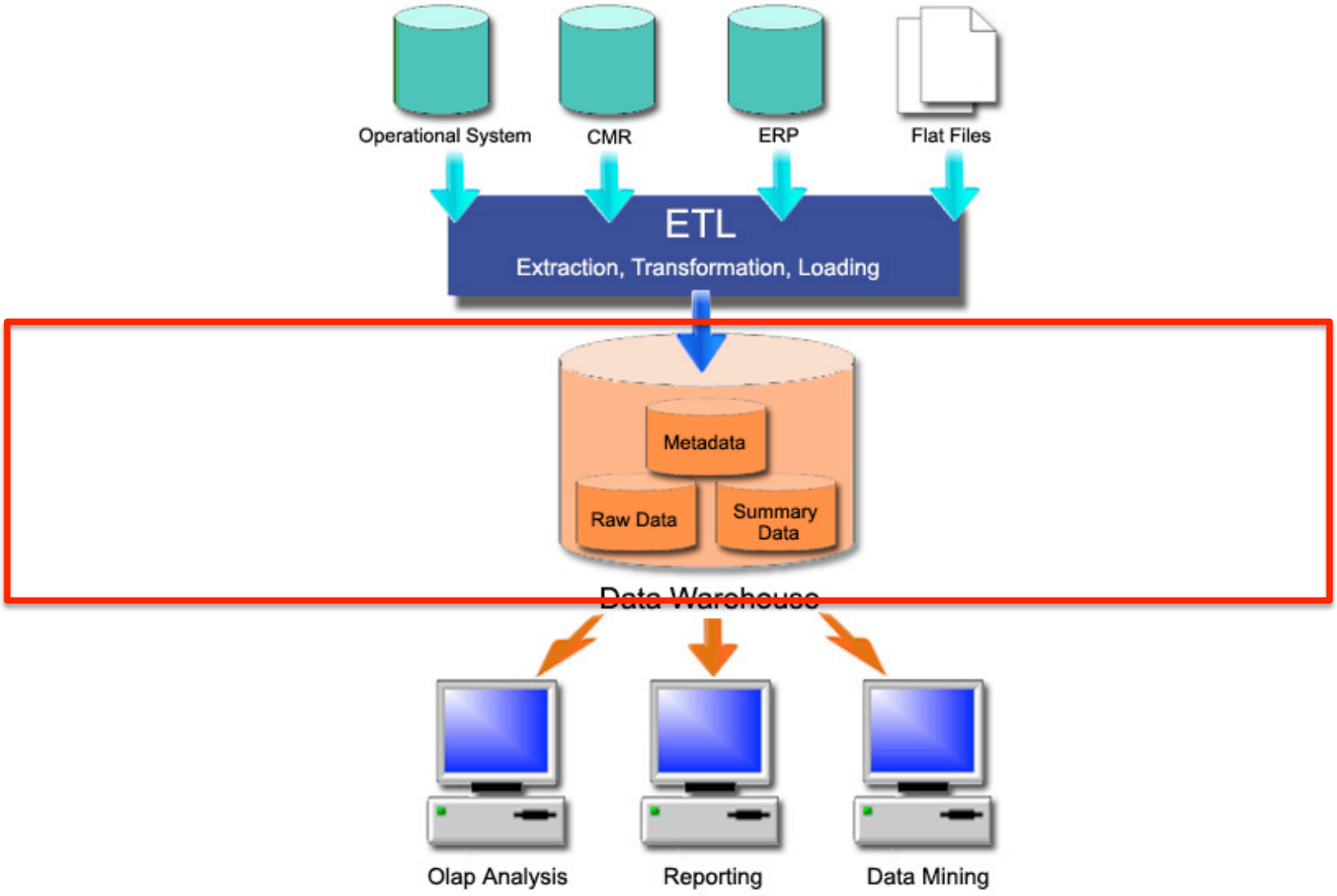
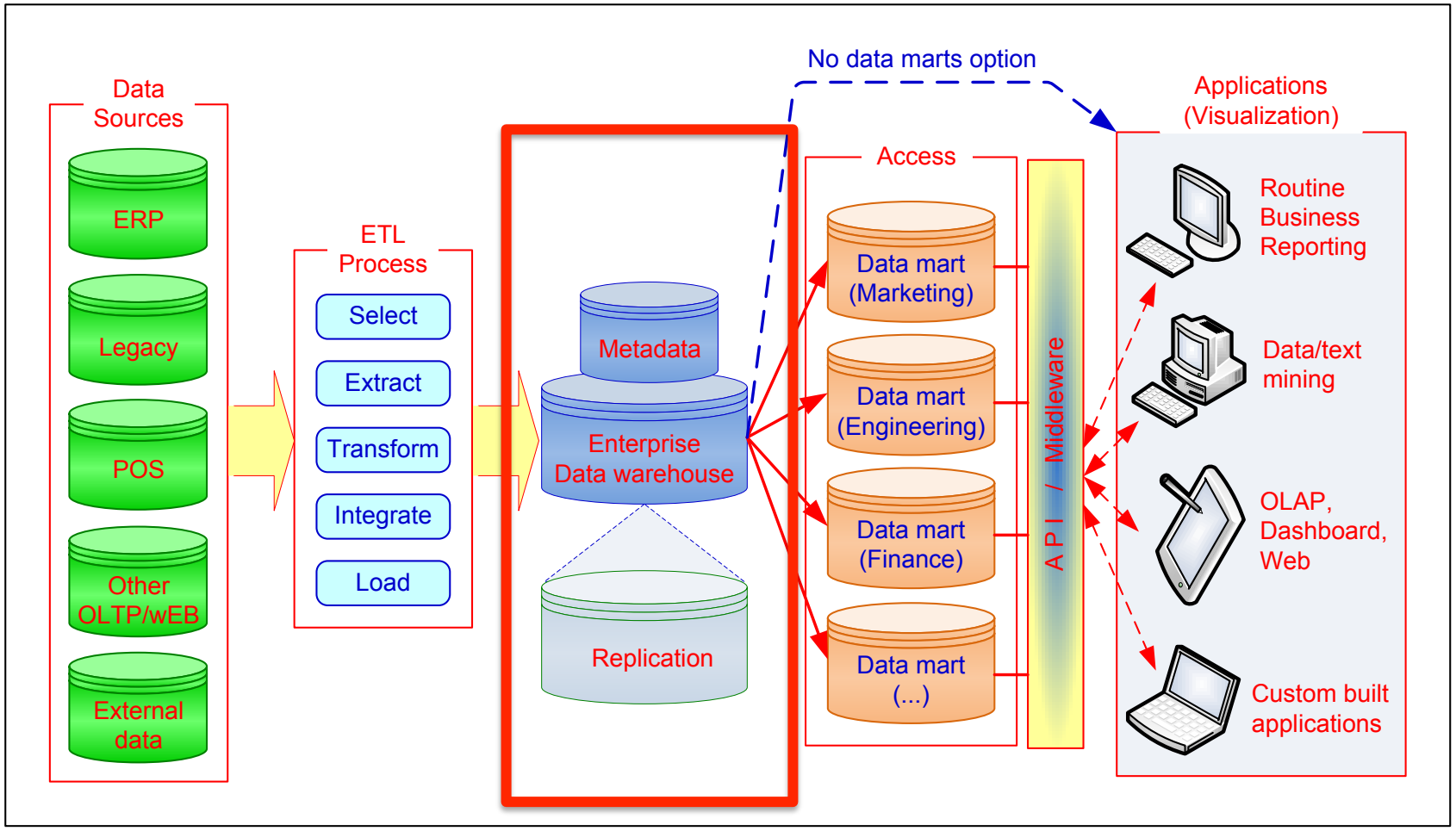
3. Data storage and data structures in Warehouses

Architecture of a DW



Data storage: metadata and replicated data



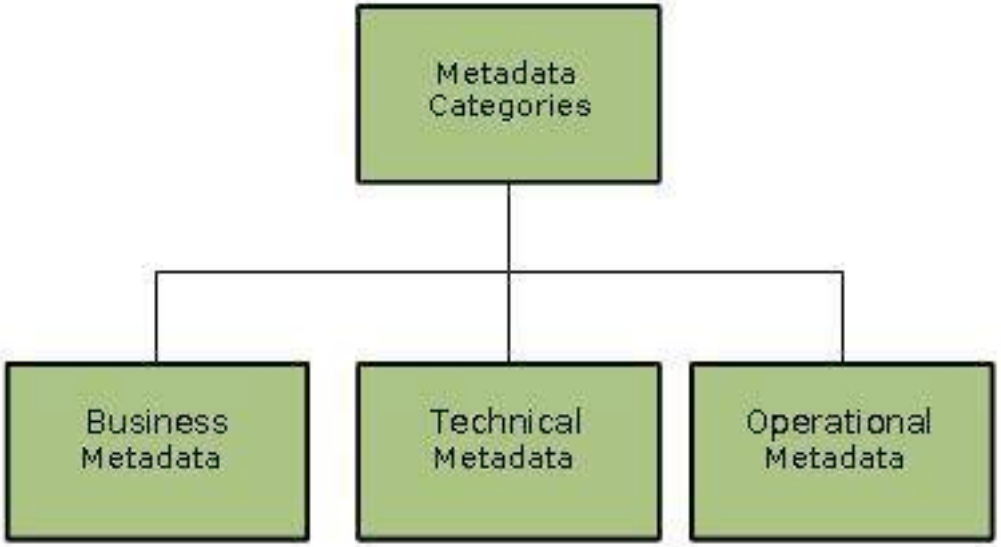
Adding Meta Data

- What is metadata? “Data about data”
- Additional data added to data for the purpose of *maintenance, retrieval, and documentation*
- Needed by both information technology personnel and business users
- IT personnel need to know data sources and targets; database, table and column names; refresh schedules; data usage measures; etc.
- Business Users need to know entity/attribute definitions and reports/query tools available

Metadata

- The Meta data functional element is responsible for maintaining information (meta data) about the operation and rules of the data warehouse.
- It also documents the data mappings used during the transformation.

Metadata categories



- Technical metadata describes the information required to access the data, such as where the data resides or the structure of the data in its native environment.
- Business metadata details other information about the data, such as keywords related to the meta object or notes about the meta object.
- Operational metadata: information about data movement, source and target systems, rules of usage, backups, recovery, last updates..

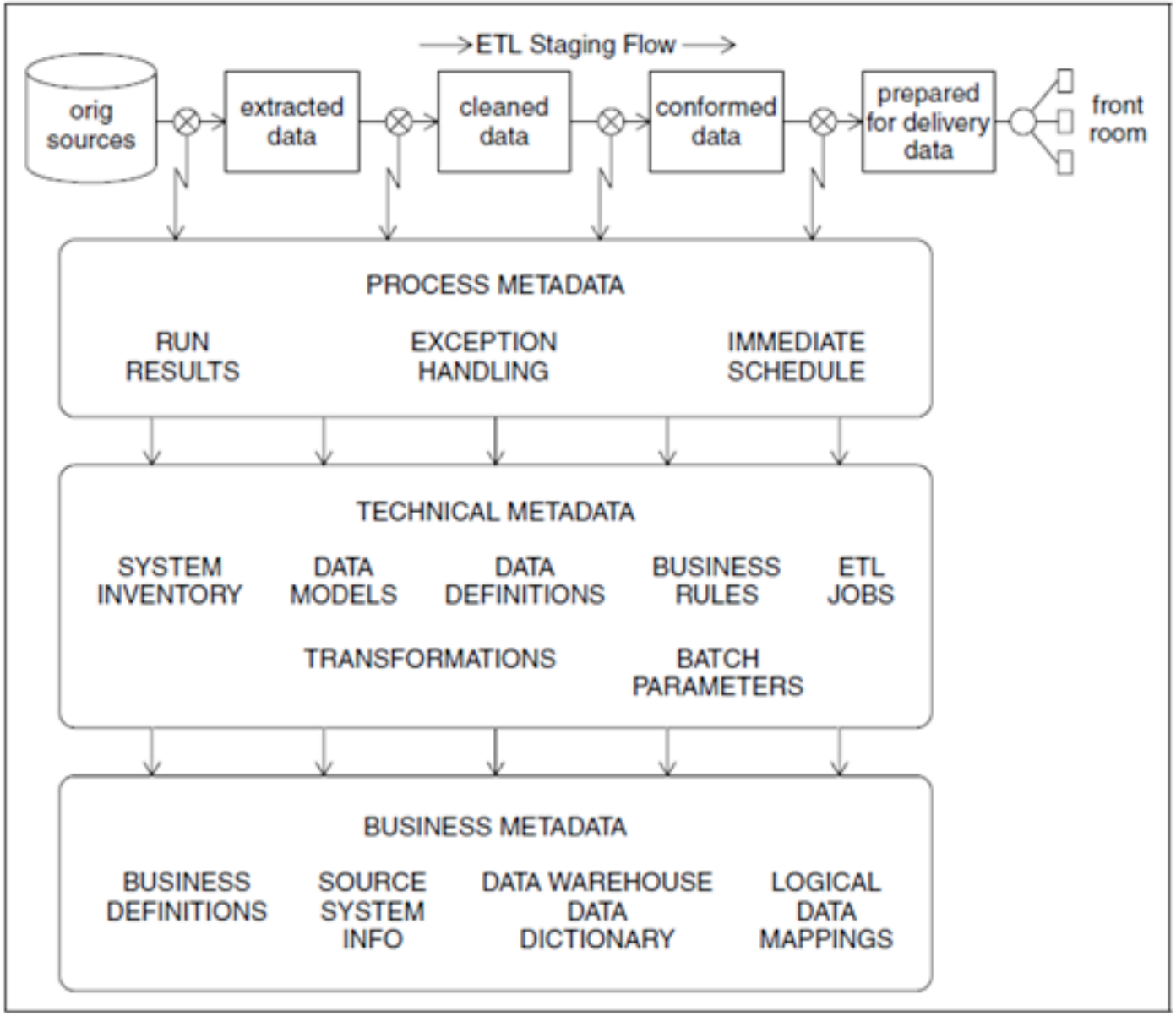


Figure 9.1 Metadata sources in the back room of the data warehouse.

Example of Business metadata

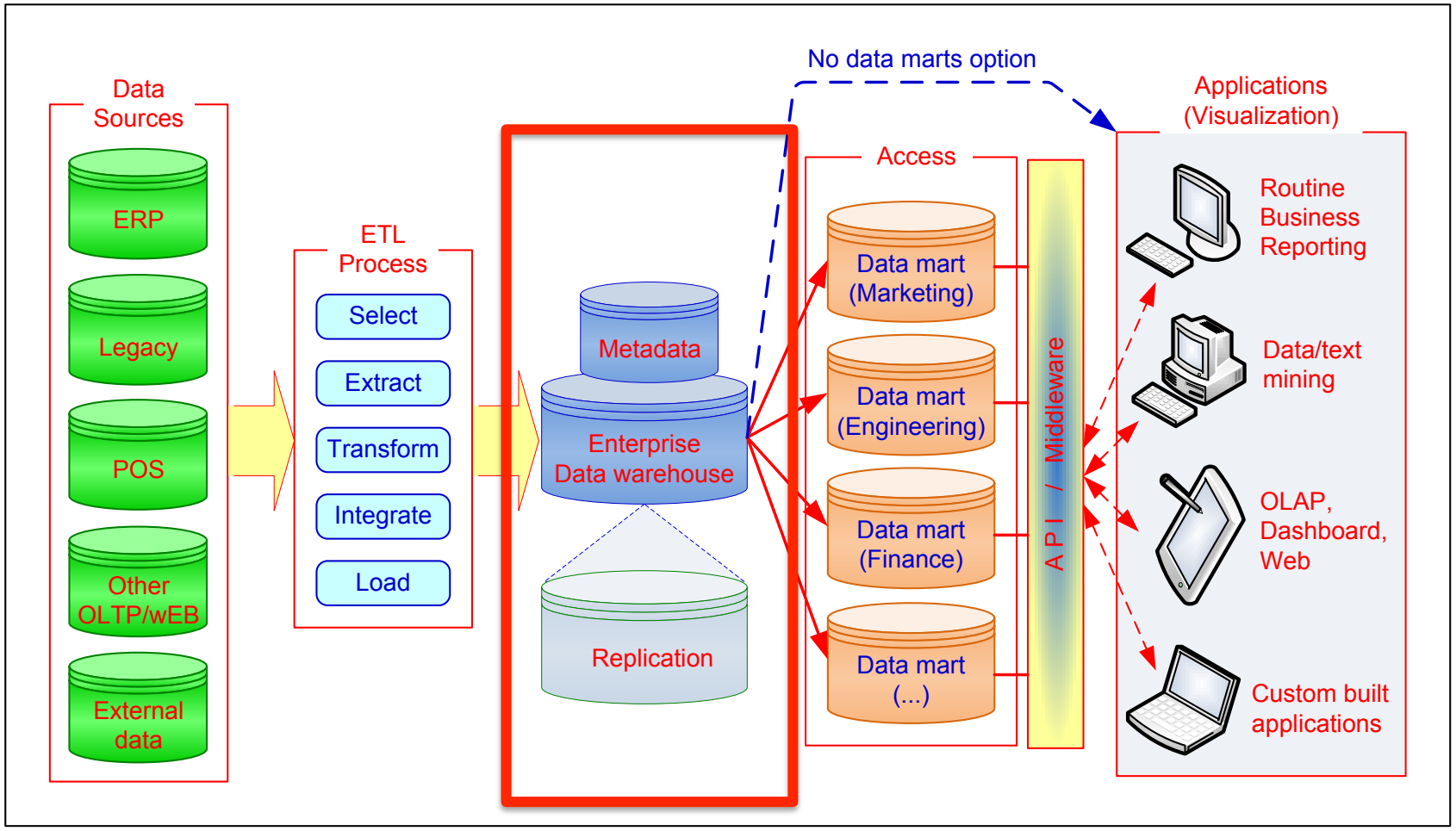
- Additional text or statement around a particular word (field name, value, etc) that adds value to the data

Entity (Table) Name	Attribute (Column)	Attribute Definition
TARGET AUTO LOAN BY WEB	Auto Loan Identifier	The number that uniquely identifies an AUTO LOAN.
	Auto Loan Amount	The amount of auto loan that has been approved. Mapping: SOURCE_AUTO_LOAN_BY_WEB.AUTO_LOAN_AMOUNT
	Auto Loan Broker Commission Amount	The commission amount that has to be paid to AUTO loan broker. Note: This column is a derived column and not found in the source system.

Example of Technical metadata

Table Name	Column Name	Column Null	Column Datatype	Primary Key	Foreign Key
SOURCE_AUTO_LOAN_BY_WEB	AUTO_LN_ID	NOT NULL	VARCHAR2(20)	Yes	No
	AUTO_LN_AMT	NOT NULL	NUMBER(11)	No	No
	AUTO_VIN_ID	NOT NULL	VARCHAR2(16)	No	No
	BOR_FST_NAME	NOT NULL	VARCHAR2(20)	No	No
	BOR_LAST_NAME	NOT NULL	VARCHAR2(20)	No	No
	DATETIMESTMP	NOT NULL	DATE	No	No
TARGET_AUTO_LOAN_BY_WEB	AUTO_LN_ID	NOT NULL	VARCHAR2(20)	Yes	No
	AUTO_LN_AMT	NOT NULL	NUMBER(11)	No	No
	AUTO_LN_BRK_COMIS_AMT	NOT NULL	NUMBER	No	No
	AUTO_VIN_ID	NOT NULL	VARCHAR2(20)	No	No
	BOR_FULL_NAME	NOT NULL	VARCHAR2(40)	No	No
	DATETIMESTMP	NOT NULL	DATE	No	No

Data storage: metadata and replicated data



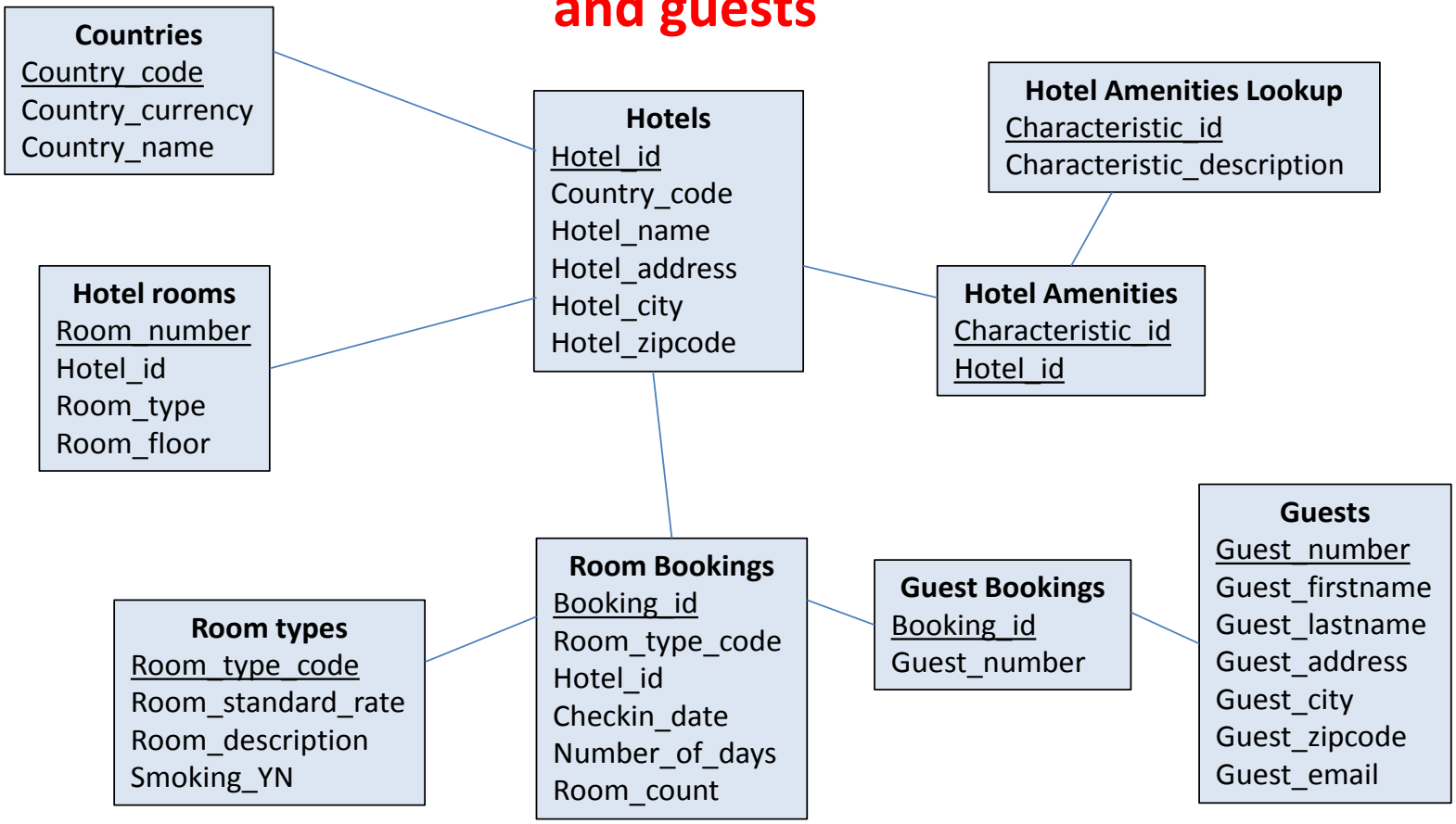
Replication

- It is not always possible for an organization to consolidate all of its data into a single database.
- The data might be spread (replicated) over several geographic locations, and some remote locations might not have good connectivity with a primary site.
- These are only a few reasons why organizations might need to share information between locations or applications.
- Replication is often used in data warehousing and reporting applications to:
  - Consolidate data so it can be moved into other databases.
  - Distribute data to **read-only databases** for reporting.
  - Distribute data to an **online analytical processing** (OLAP) database.

Example: TU hotel chains

Hotel Reservation Database

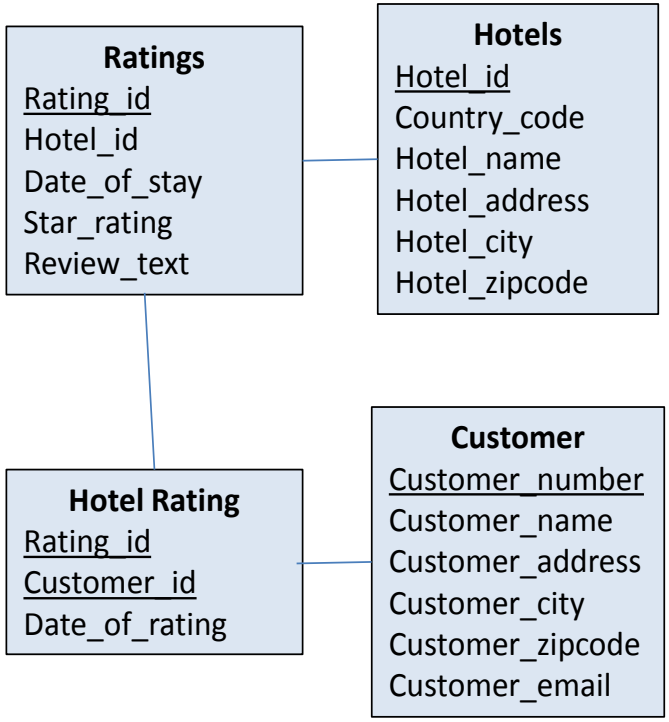
Store data on each hotel, countries, rooms, amenities, reservations and guests



Two additional databases:

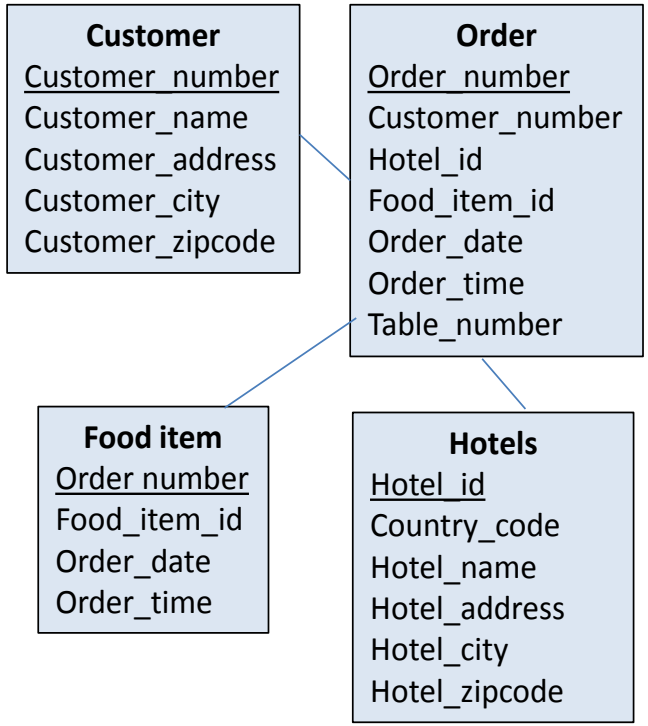
HotelComplainer Ratings Database

(totally external company)



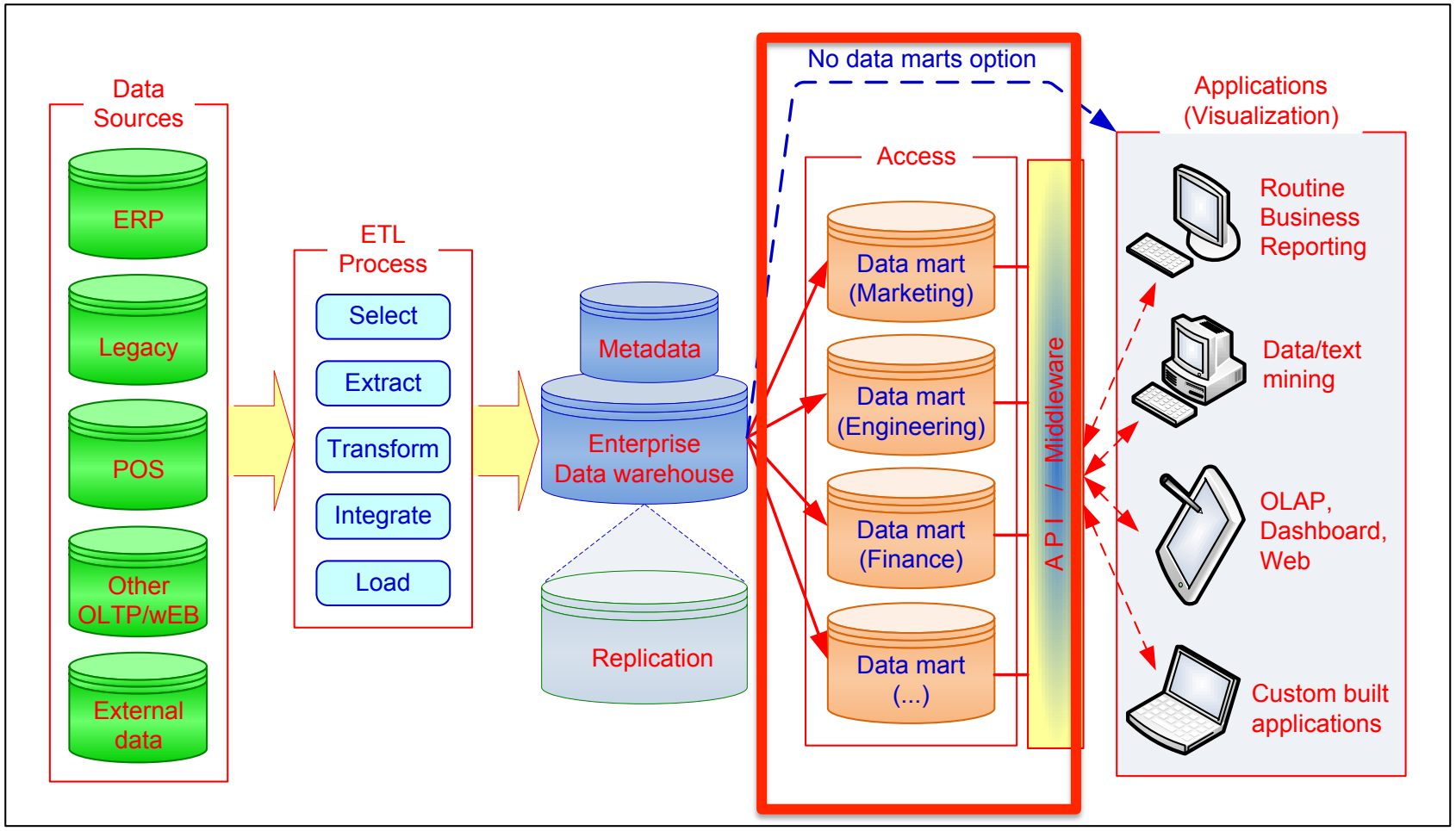
Café in the Hotel Database

(same company but database is not connected to the hotel)



Note that several data need **replication**: e.g., hotel, customer (guest)

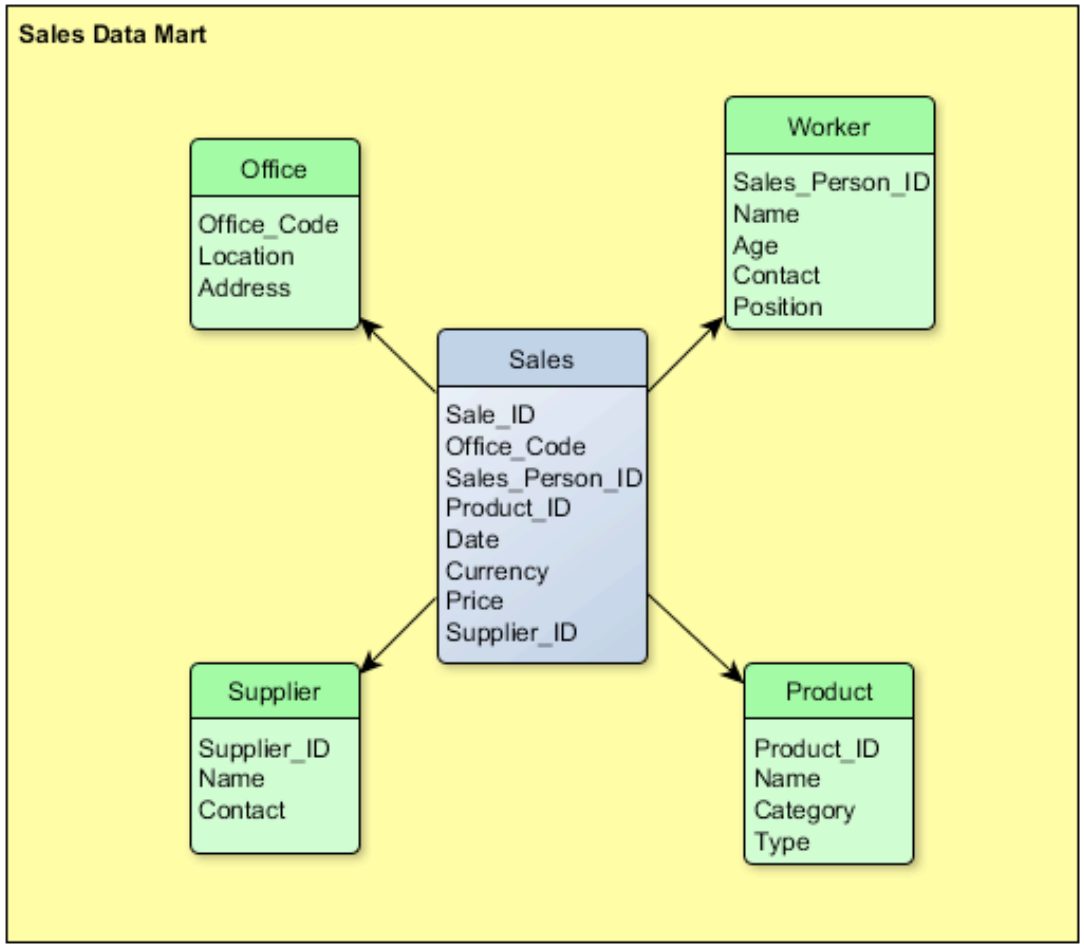
Data storage: Data Marts and data structures



Data Mart

- A **data mart is one piece of a data warehouse** where all the information is related to a **specific business area**.
- Therefore it is considered a **subset** of all the data stored in that particular database, since all data marts together create a data warehouse.
- Data marts are a specific VIEW of the data, tailored for specific types of analyses (e.g. business management, assistence and repairs, customer care..).

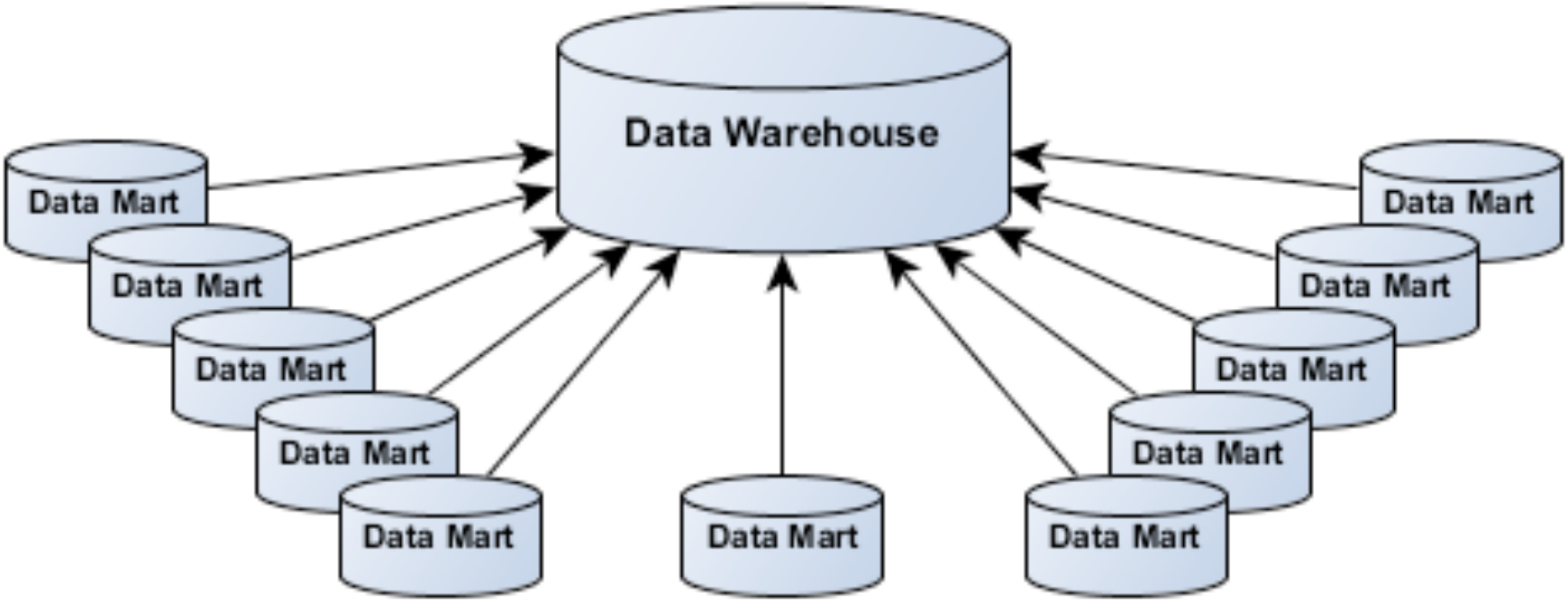
Example of data mart



Sales data mart

- Since this is a data mart, all the information contained in this data structure is only relative to **sales** and its dependencies.

Why Data Marts?

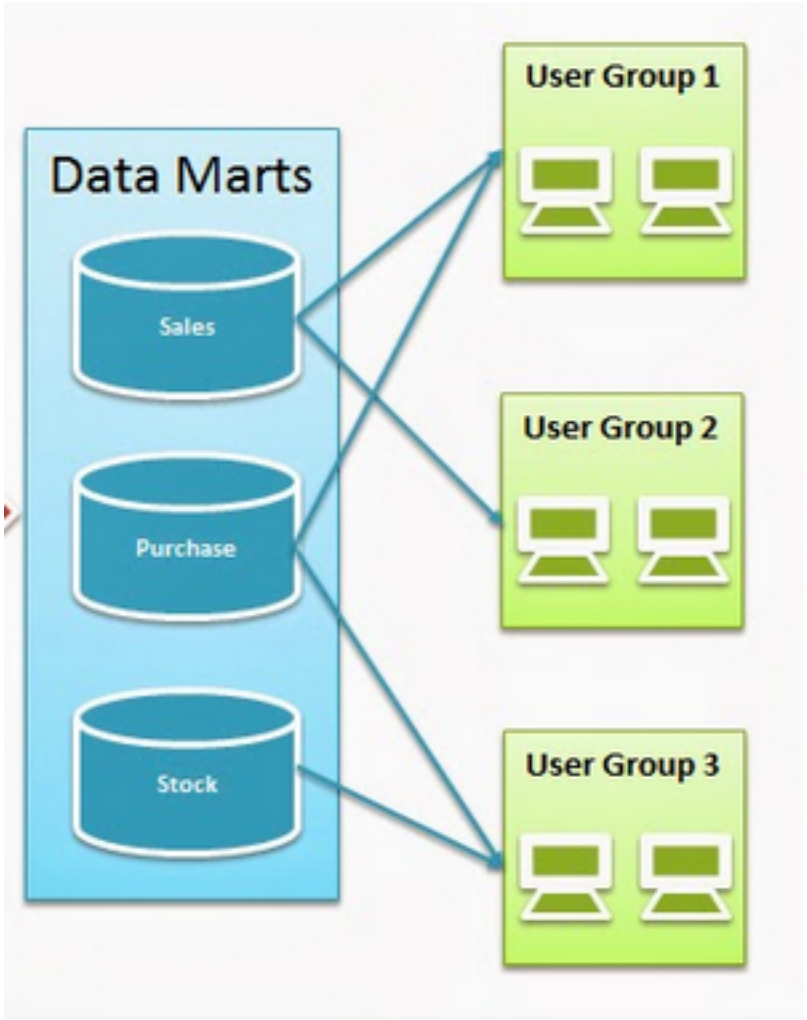


Why data Marts?

Data Scope:

- On one hand, **data warehouses save all kinds of data related to system**.
- On the other hand, **data marts** just store **specific subject information**, becoming **much more focused** on these functionalities.

Data Marts serve specific user groups



Why data Marts?

1. Size

- A **data warehouse** is usually **much bigger than data marts**, because it keeps a lot more data.

2. Integration

- A **data warehouse** integrates **several sources of data** in order to feed its database and the system's needs.
- In opposite, a **data mart has a lot less integration** to do, since its data is very specific.

Why data Marts?

3. Creation

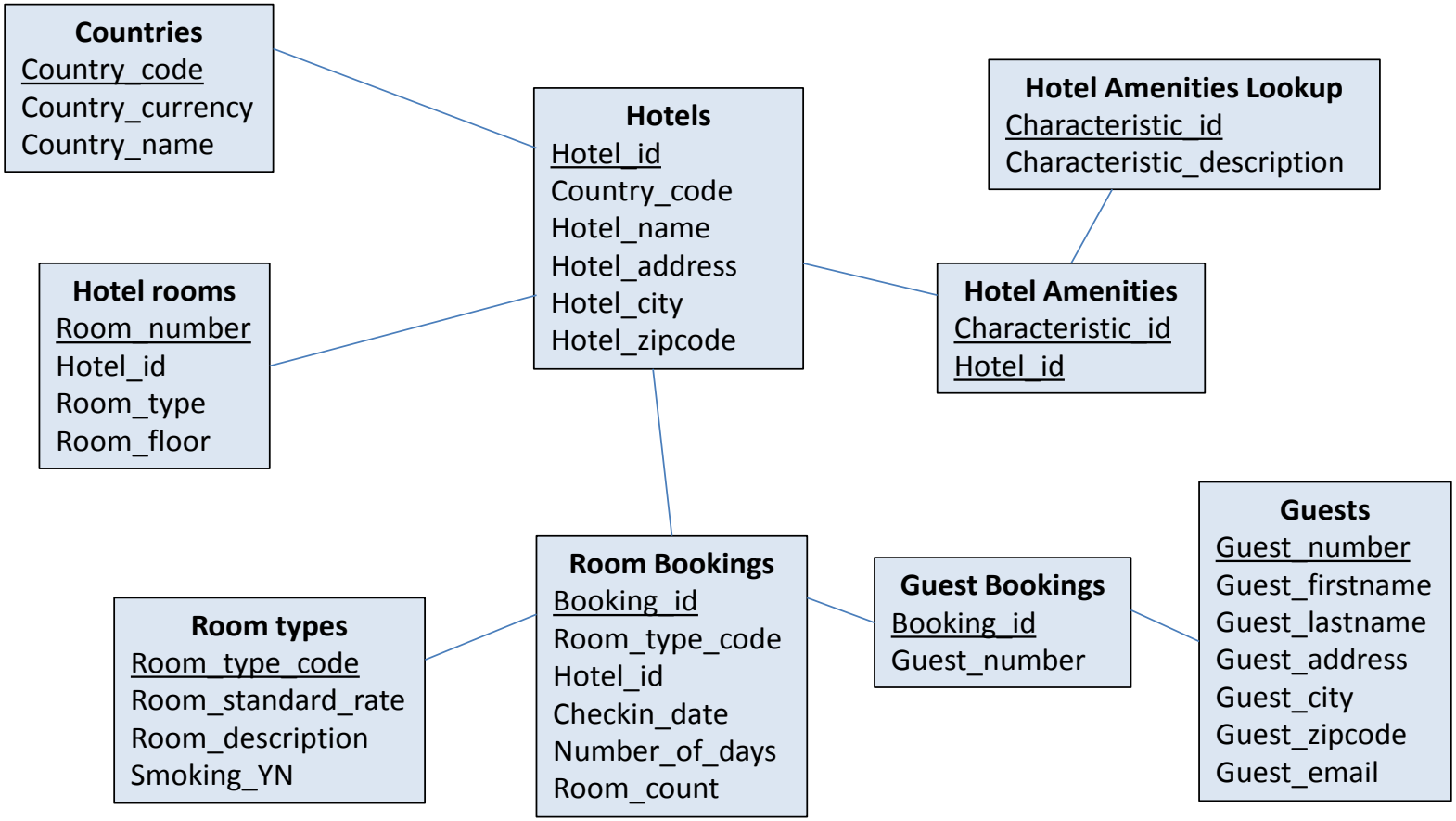
- Creating a **data warehouse** is way **more difficult and time consuming** than building a **data mart**.
- Since **data marts** are smaller and **subject oriented**, these actions tend to be much **simpler**.
- However a well built data warehouse can support large systems for the long run. In the other hand a good data mart is only limited to its activity area.

Cons of Data Marts

- Datamarts also create problems with **inconsistency**. It takes tight discipline to keep data and calculation definitions consistent across data marts.
- This problem has been widely recognized, so data marts exist in two styles:
  - **Independe**nt data marts are those which are fed directly from source data. They can turn into islands of inconsistent information.
  - **Dependent** data marts are fed from an existing data warehouse. Dependent data marts can avoid the problems of inconsistency, but they require that an enterprise-level data warehouse already exist

Back to the example of TU hotel chains

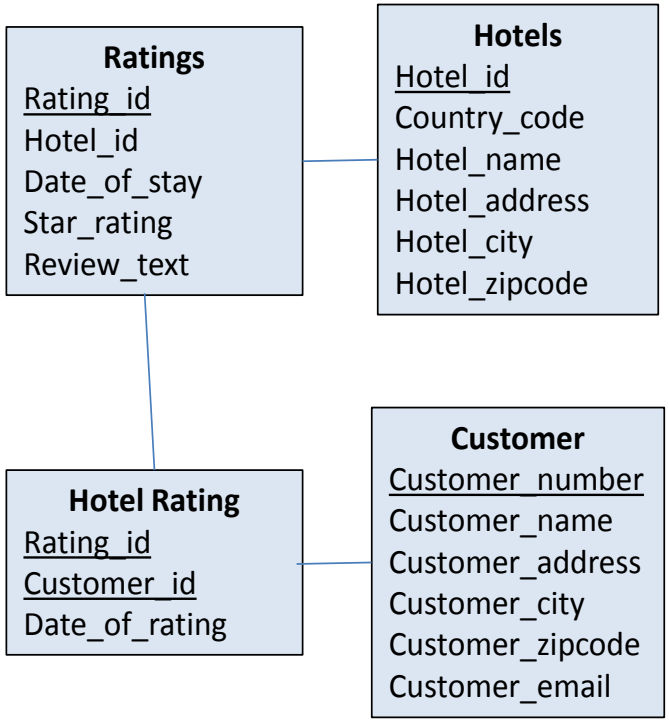
Hotel Reservation Database



Ratings and Café are possible Marts

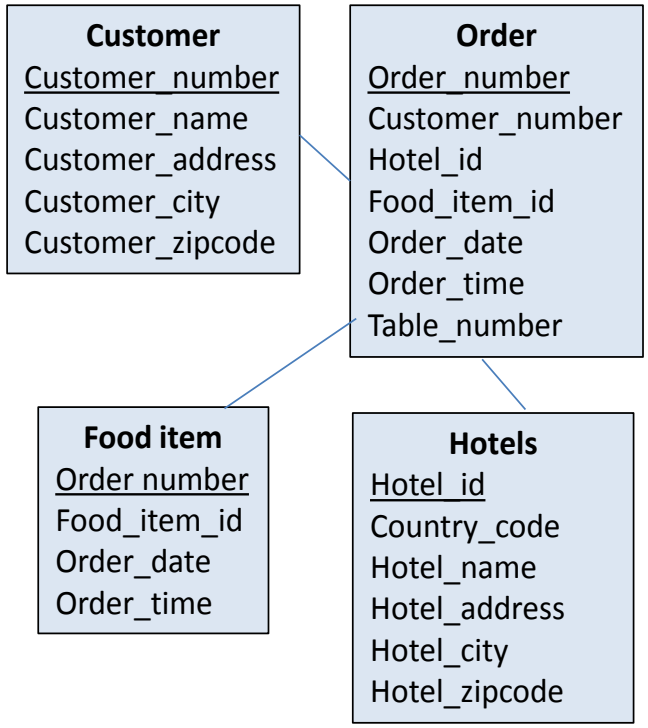
HotelComplainer Ratings Database

(totally external company)



Café in the Hotel Database

(same company but database is not connected to the hotel)

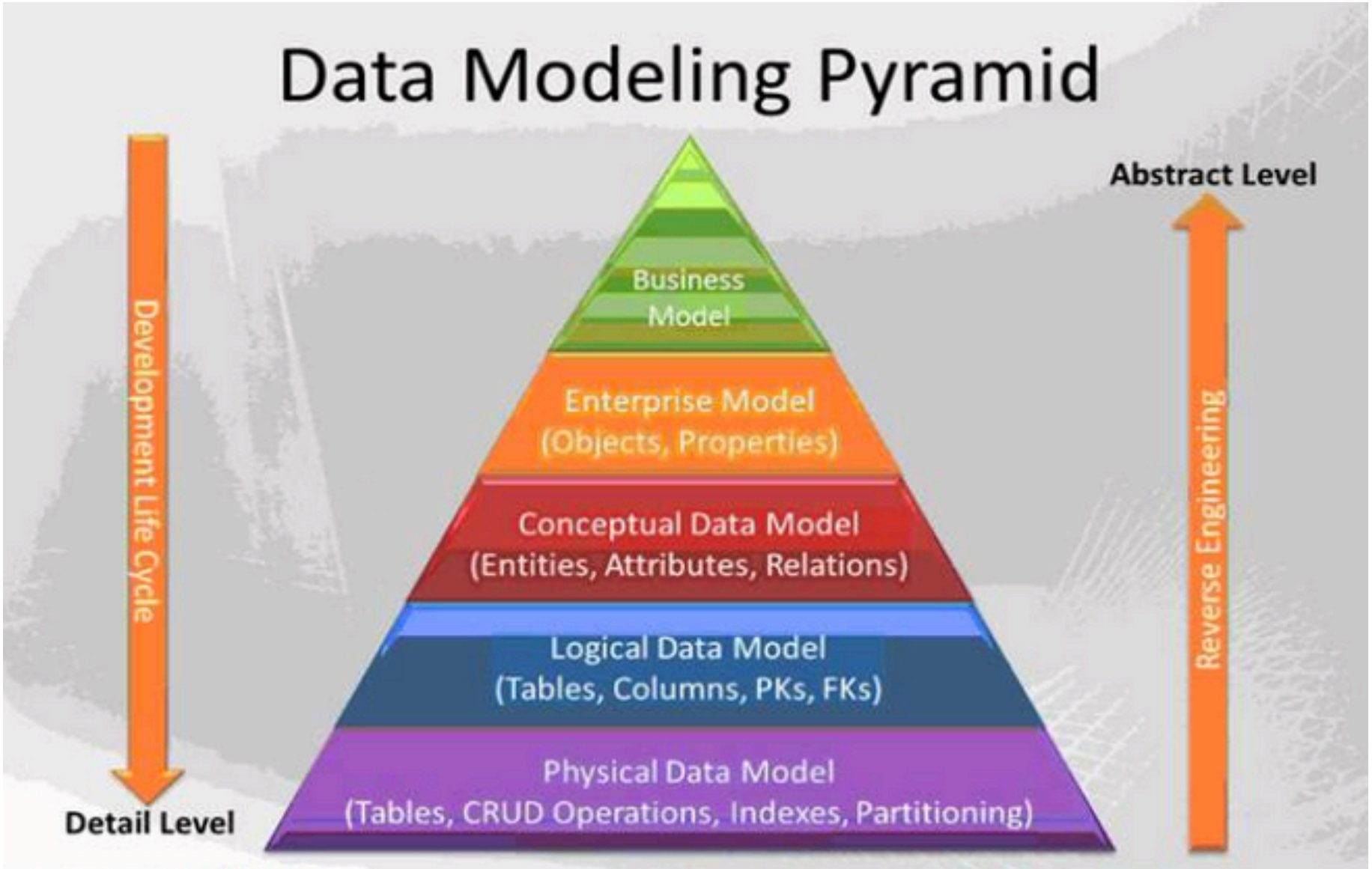


In-Class Work: show the schema of the integrated DW and two marts

- Can you identify data types that need harmonization during ETL processing?
- How would you organize the data marts to answer the following questions (i.e.: each Mart needs to store all information needed to answer each query)

Data Mart 1: Hotel Performance	Data Mart 2: Restaurant Performance
• During which month are the most rooms rented? ○ Identify the "off season" (if any) for our hotels in Arizona, Florida, Pennsylvania, and New York • Which hotel generates the most (non-restaurant) revenue? • What is the average length of stay in hotels with 4.5 or more stars? • Do smokers stay longer than non-smokers? • For a given hotel, how many customers come from out of state?	• Which hotel restaurant generates the most revenue? • Do the best rated hotels generate more restaurant revenue? • What is the most frequently ordered item in the Philadelphia metropolitan area?

Process the Warehouse: data models and operations



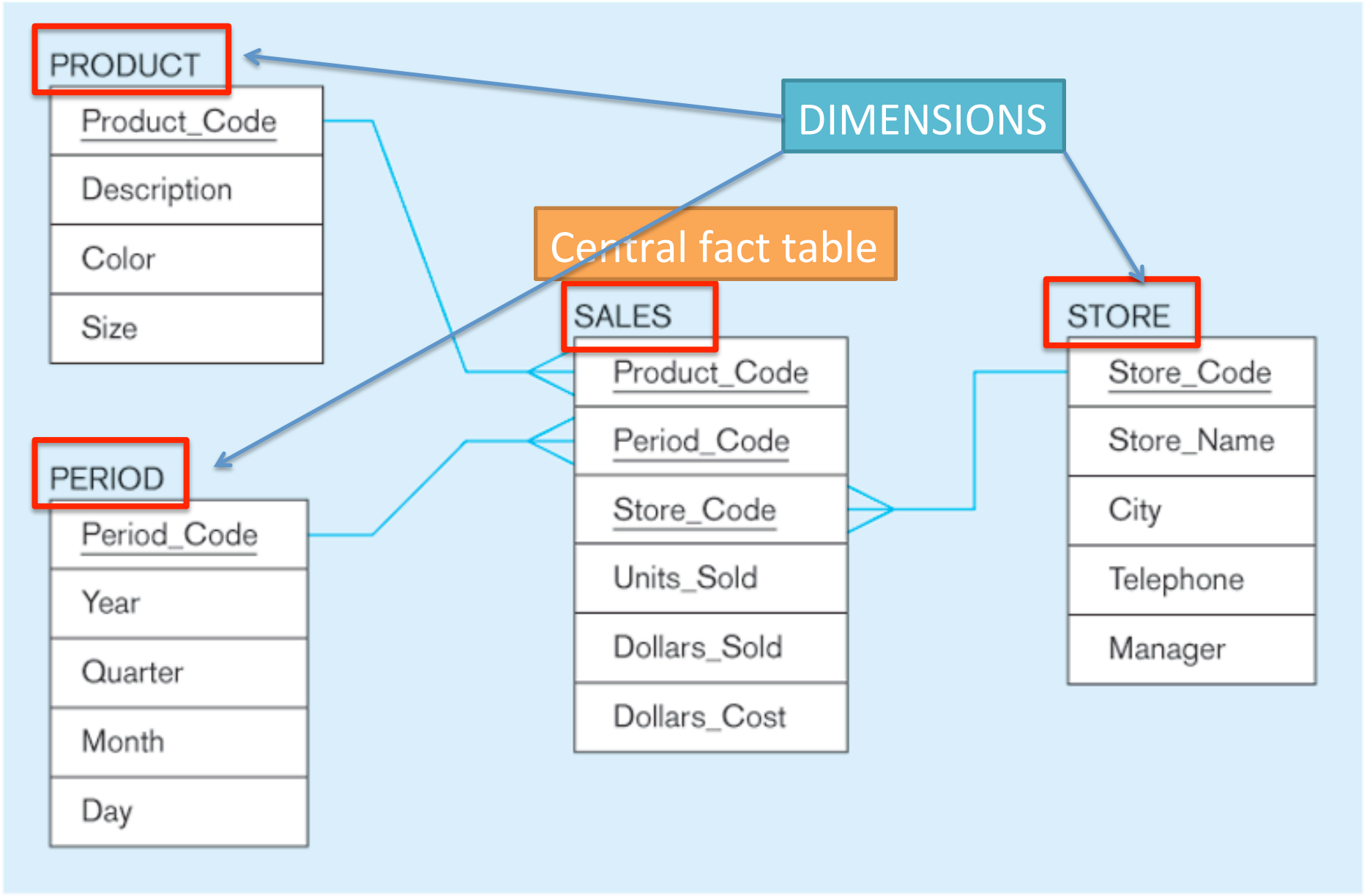
Process the warehouse: models and operations

- **Data Schema:** In which data structures we organize the loaded data?
  - Star
  - Snowflake
  - Multidimensional data (cubes)
- **Operators:** Which operations we can perform on the data?
  - slice & dice
  - roll-up, drill down
  - pivoting
  - other

Schemas: Star schema

- The star schema architecture is the simplest data warehouse schema.
- It is called a star schema because the diagram resembles a star, with points radiating from a center.
- The center of the star consists of fact table and the points of the star are the dimension tables.Has attributes (fields) that represent **dimensions (=ways of aggregating data)**, plus other dependent attributes

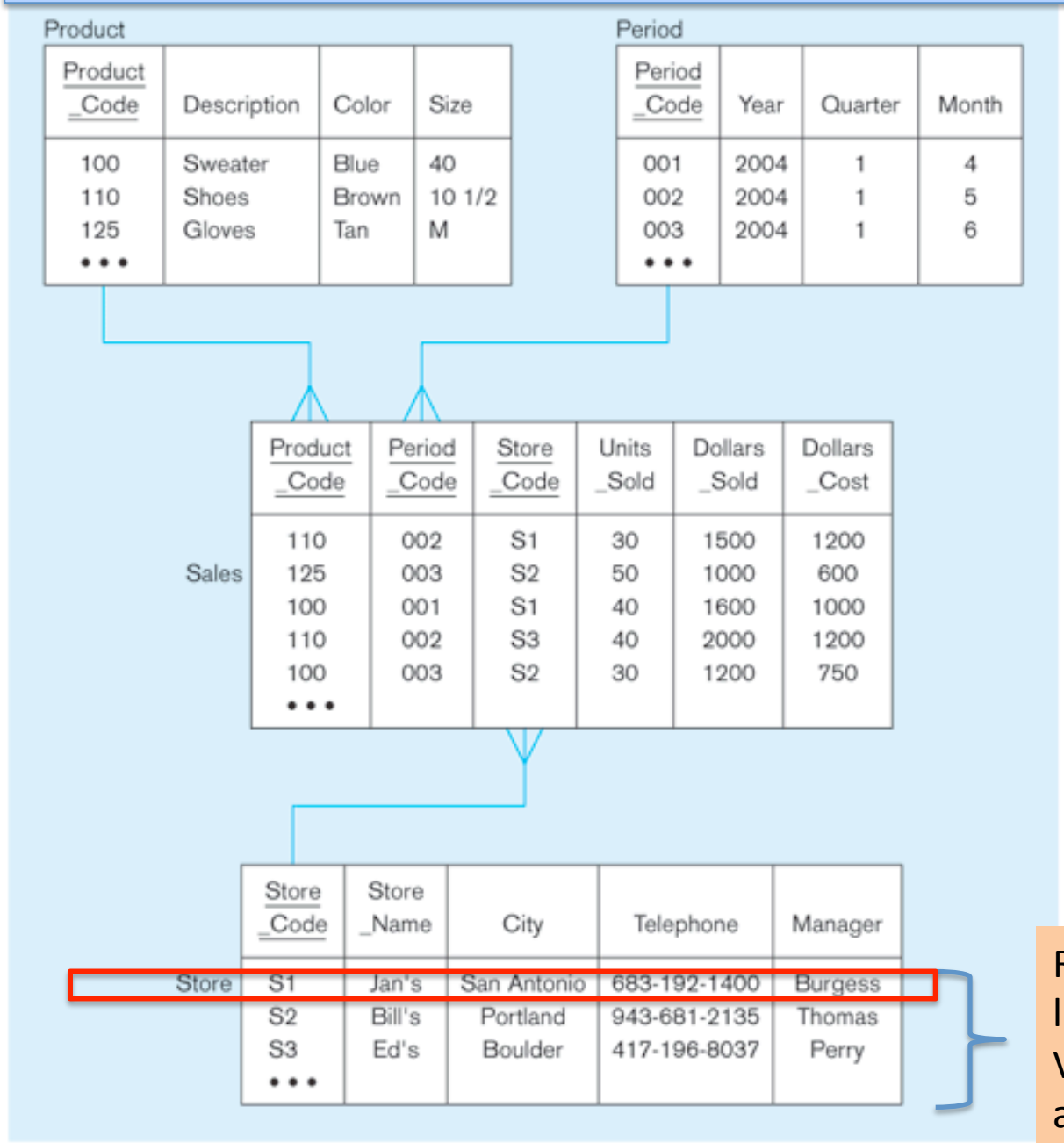
STAR SCHEMA EXAMPLE: 4 dimensions



Star schema: dimensions

- **Dimensions** provide **structured labeling** information to otherwise unordered numeric measures.
- The primary functions of dimensions are threefold: to provide
  - filtering,
  - grouping and
  - labelling.
- Common dimensions are *people, products, places and time*.
- Each dimension is described by its own dimension table
- Dimension tables have corresponding dimension *attributes*

Star Schema with example data (= values of attributes)



Product

Product_Code	Description	Color	Size
100	Sweater	Blue	40
110	Shoes	Brown	10 1/2
125	Gloves	Tan	M
• • •			

Period

Period_Code	Year	Quarter	Month
001	2004	1	4
002	2004	1	5
003	2004	1	6
• • •			

Sales

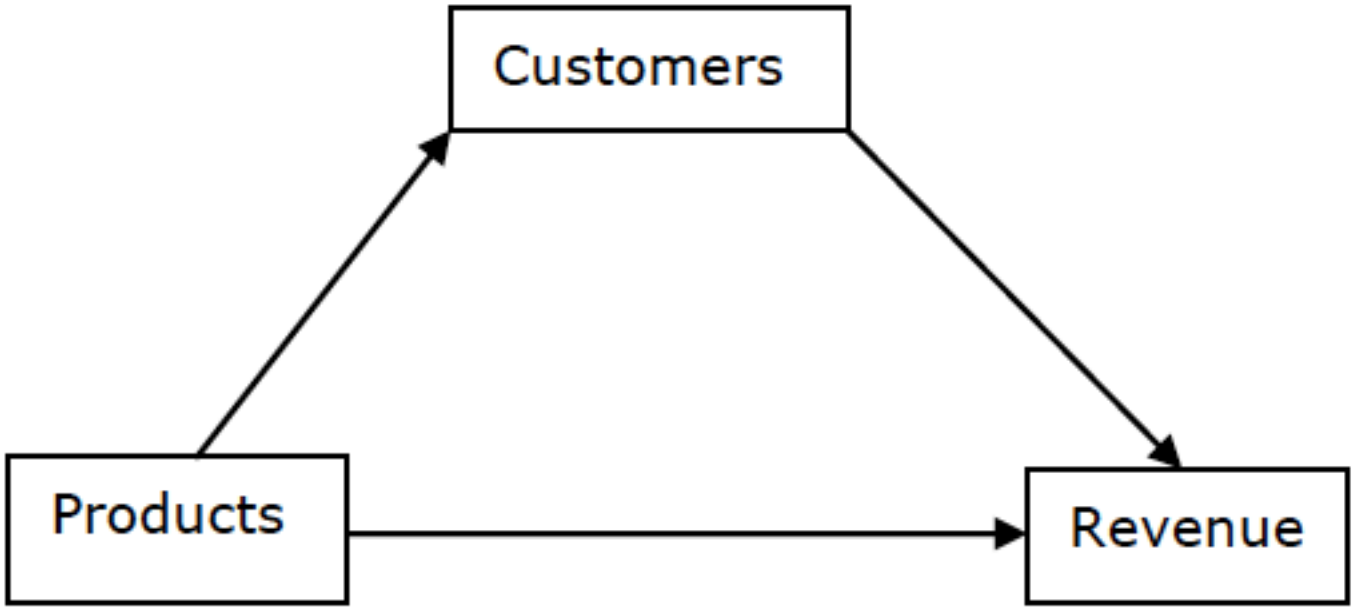
Product_Code	Period_Code	Store_Code	Units_Sold	Dollars_Sold	Dollars_Cost
110	002	S1	30	1500	1200
125	003	S2	50	1000	600
100	001	S1	40	1600	1000
110	002	S3	40	2000	1200
100	003	S2	30	1200	750
• • •					

Store

Store_Code	Store_Name	City	Telephone	Manager
S1	Jan's	San Antonio	683-192-1400	Burgess
S2	Bill's	Portland	943-681-2135	Thomas
S3	Ed's	Boulder	417-196-8037	Perry
• • •				

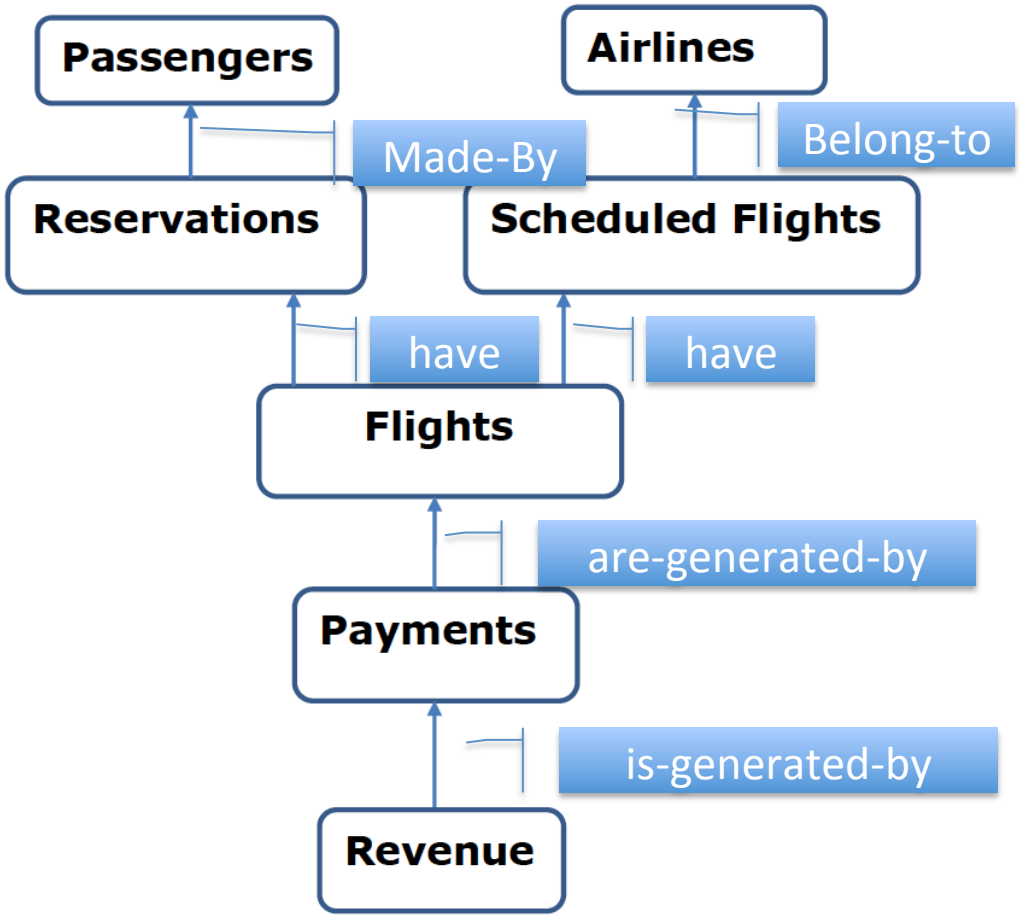
RECORDS: each line shows the VALUES of each attribute

All business cases reflect 3 main subject areas (however other dimensions are possible)



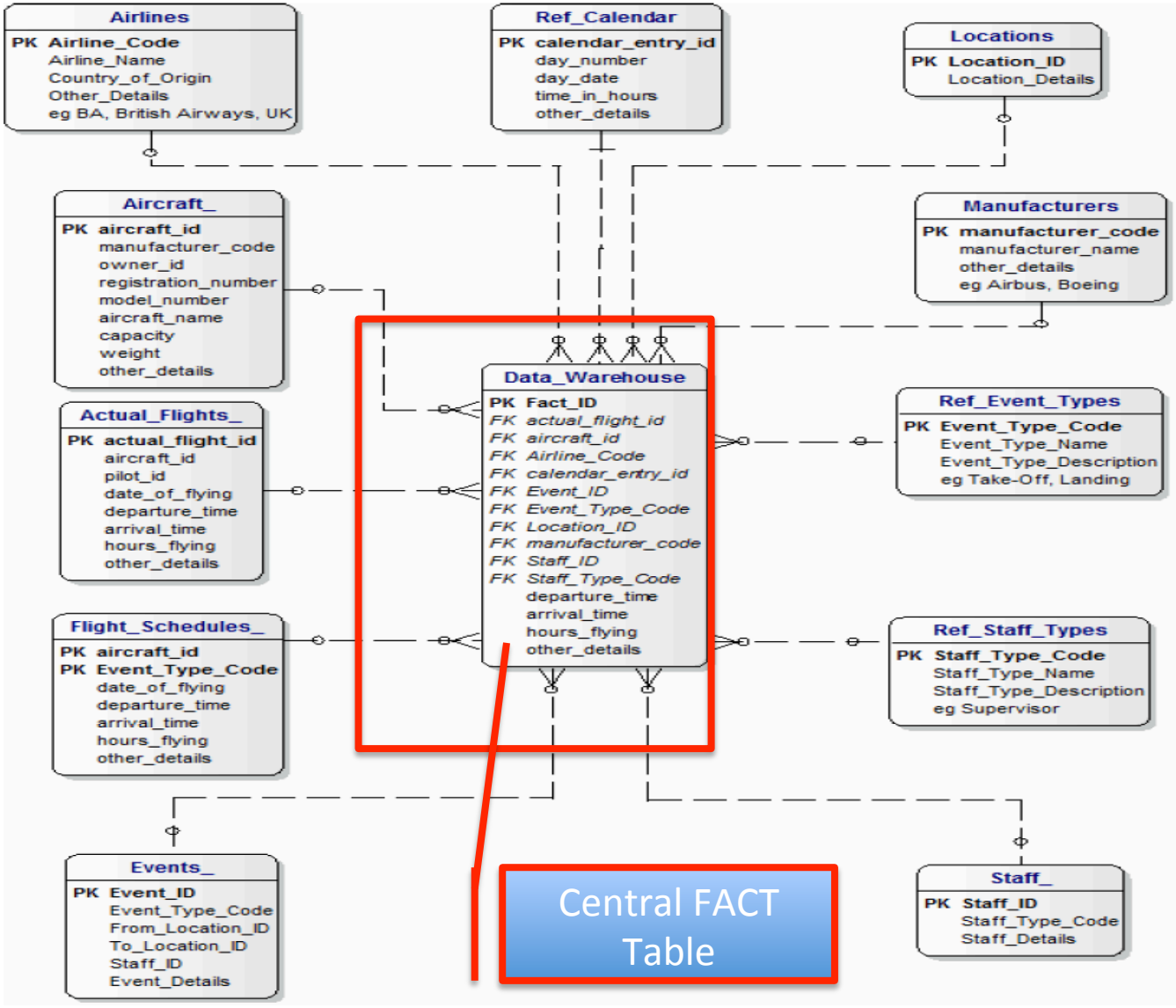
Another example: Airline company

A Simplified Model (with relevant entities)

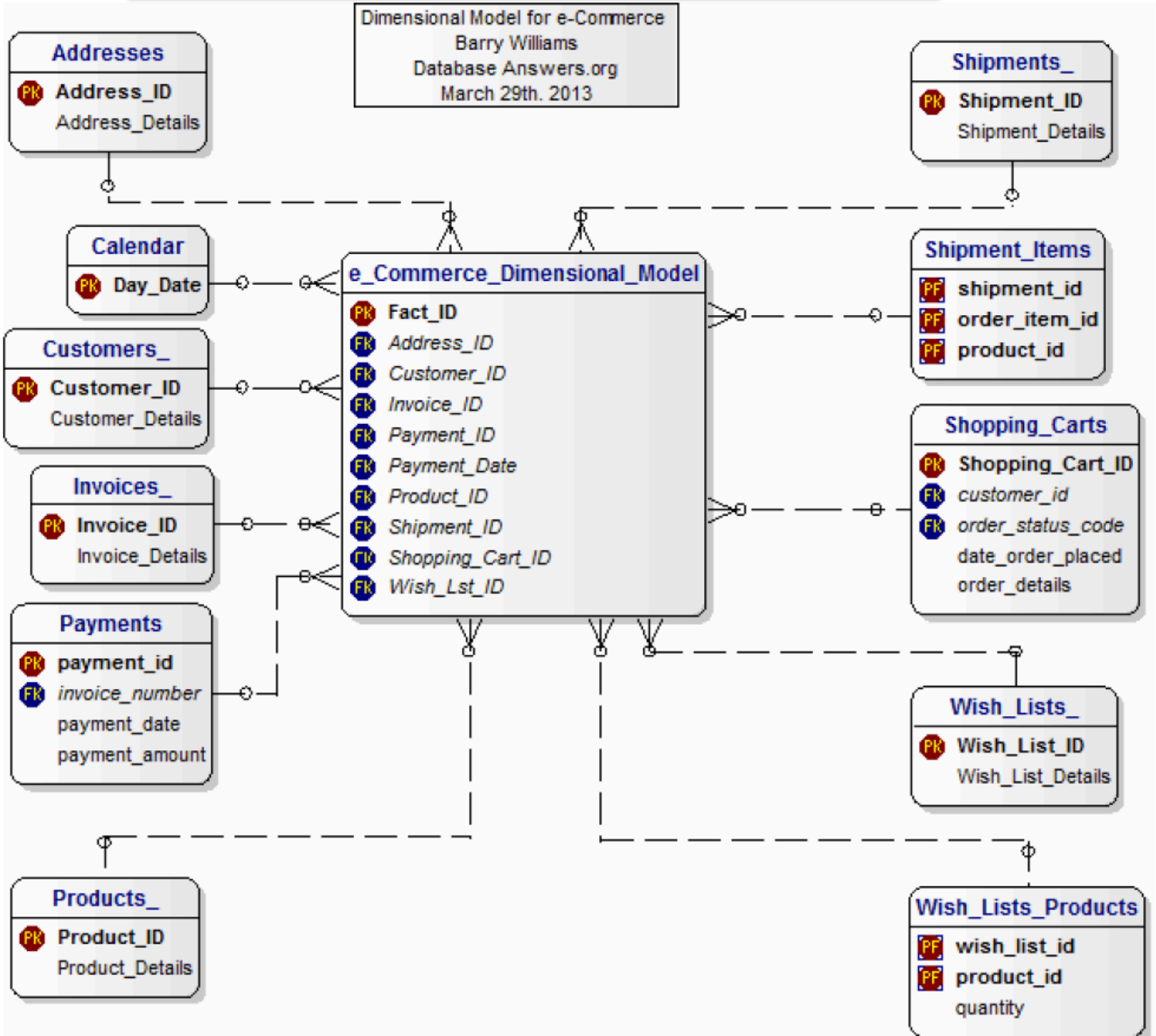


- **Passengers** make **Reservations** for **Scheduled Flights** operated by **Airlines**.
- They take **Flights** and make **Payments** which generate **Revenue**.
- Note that this “reading” of the schema assign a semantics to relationships!!!

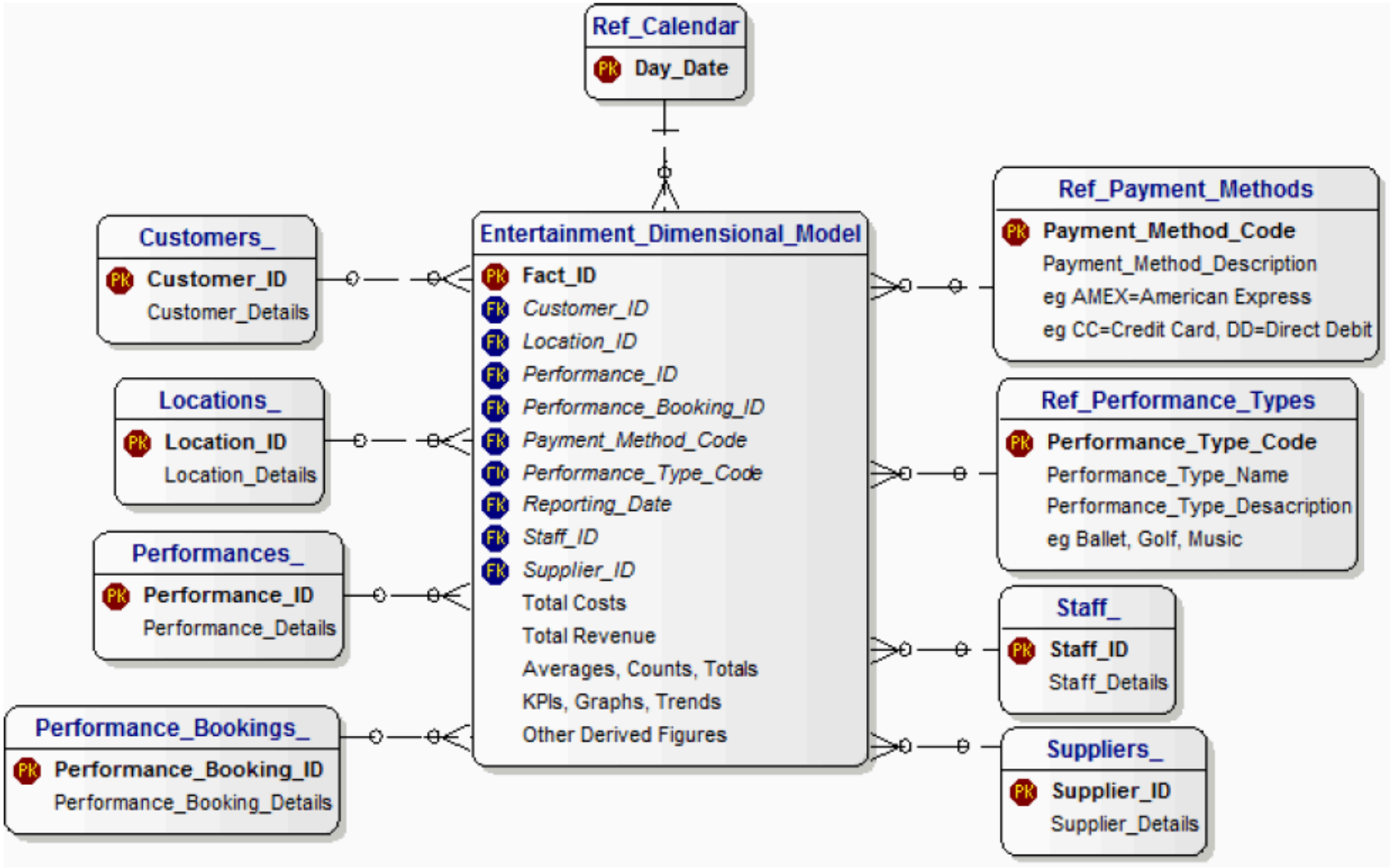
Example 2: A real-case Airline star-schema



Example 3: e-commerce star schema



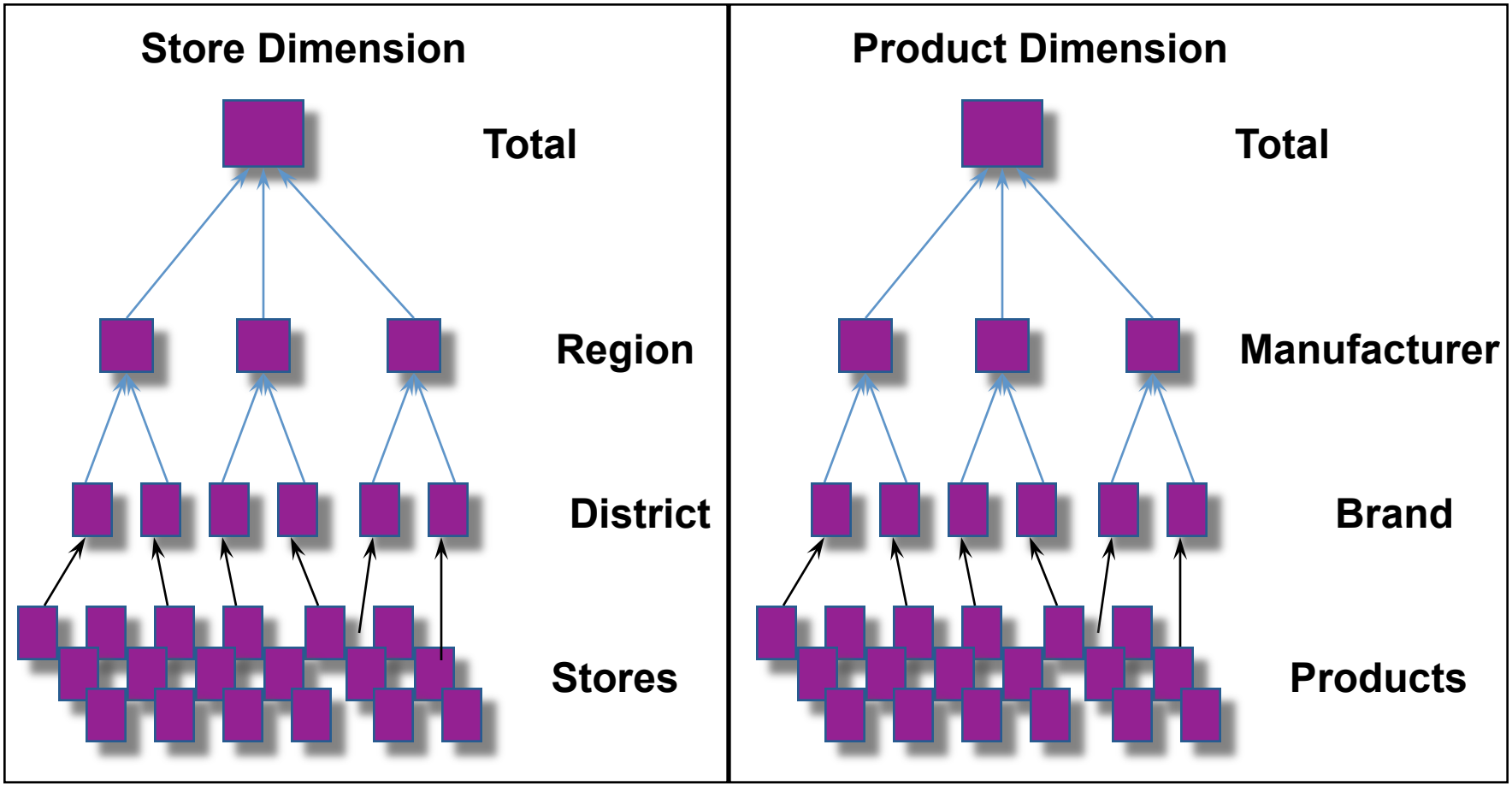
Example 4: Entertainment star schema



More on Dimensional Modeling

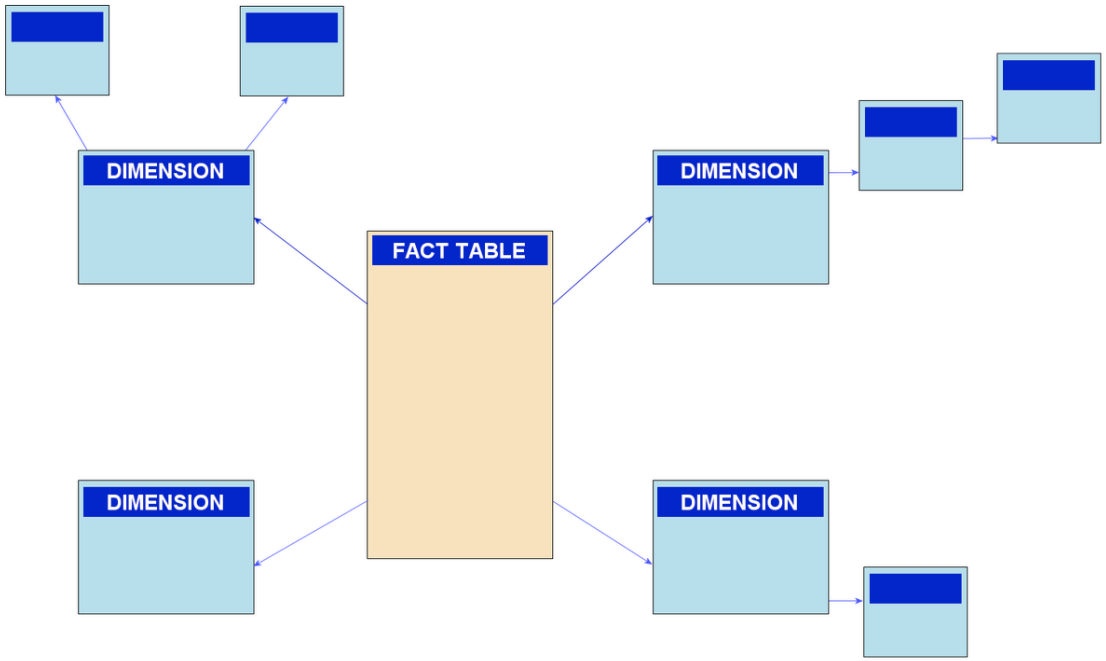
- Dimensions can be further organized into **hierarchies**
  - E.g., Time dimension: days → weeks → quarters
  - E.g., Product dimension: product → product line → brand
  - E.g. Location dimension: Continent→ nation → city→ store

Dimension Hierarchies Examples

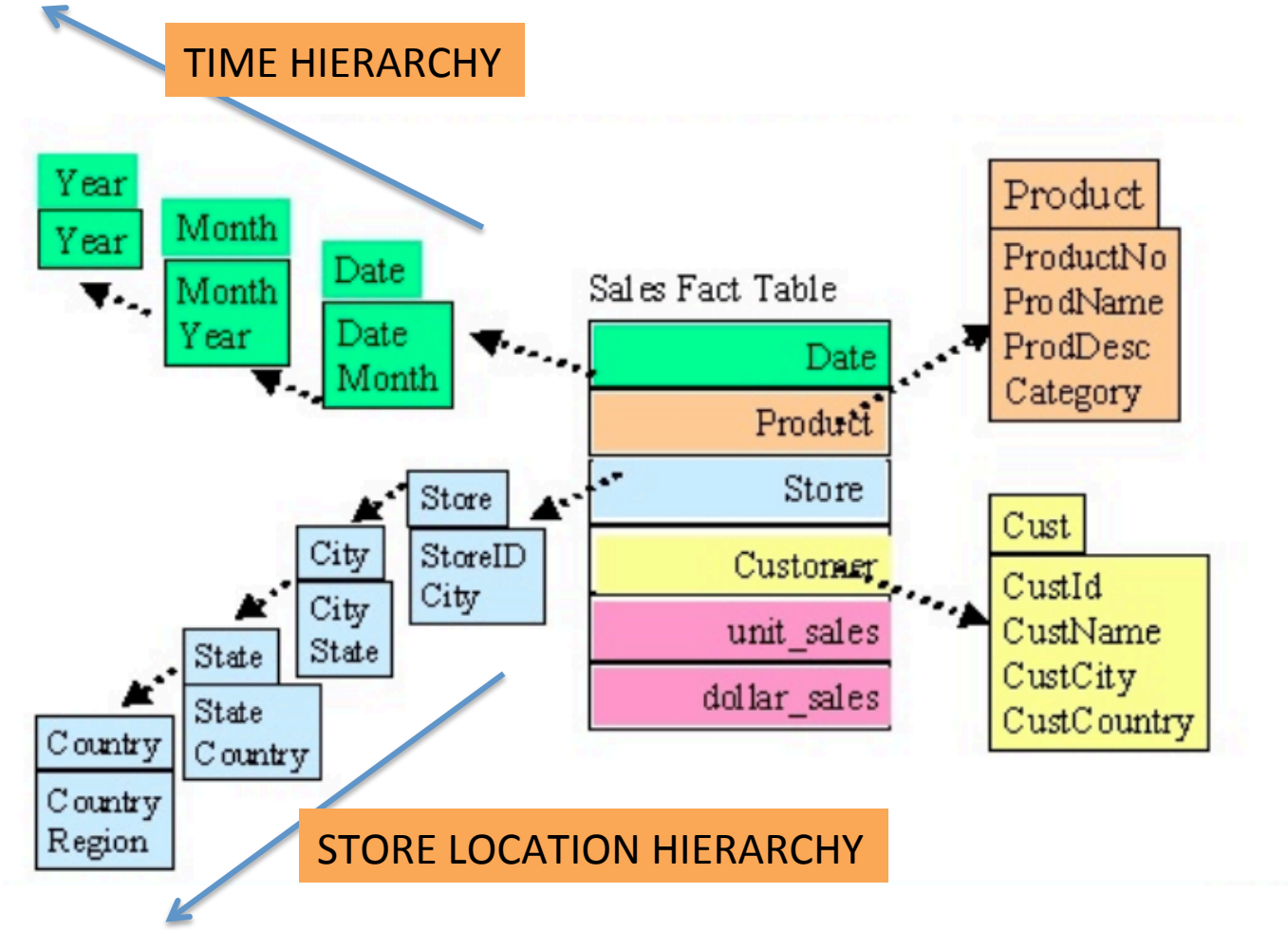


Other schemas types: Snowflake

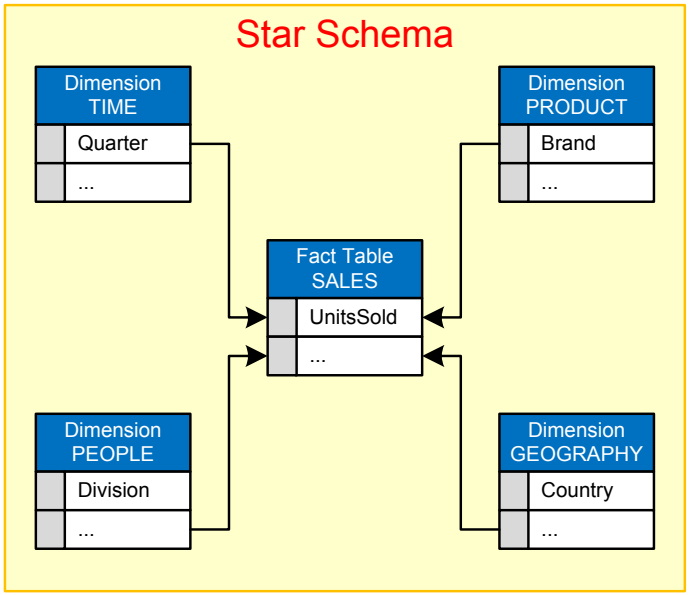
- Variation of star schema, in which normalized **dimension tables** have dimensions themselves
- More suitable to represent **hierarchical** dimensions

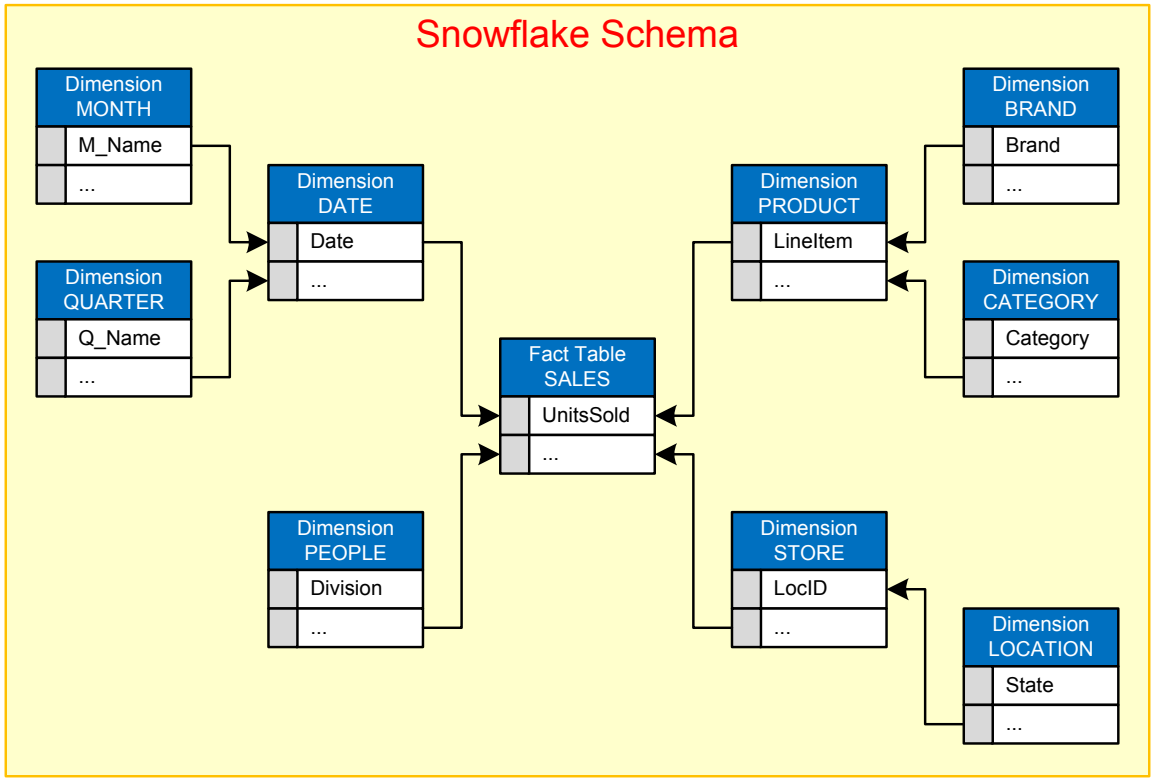


Example

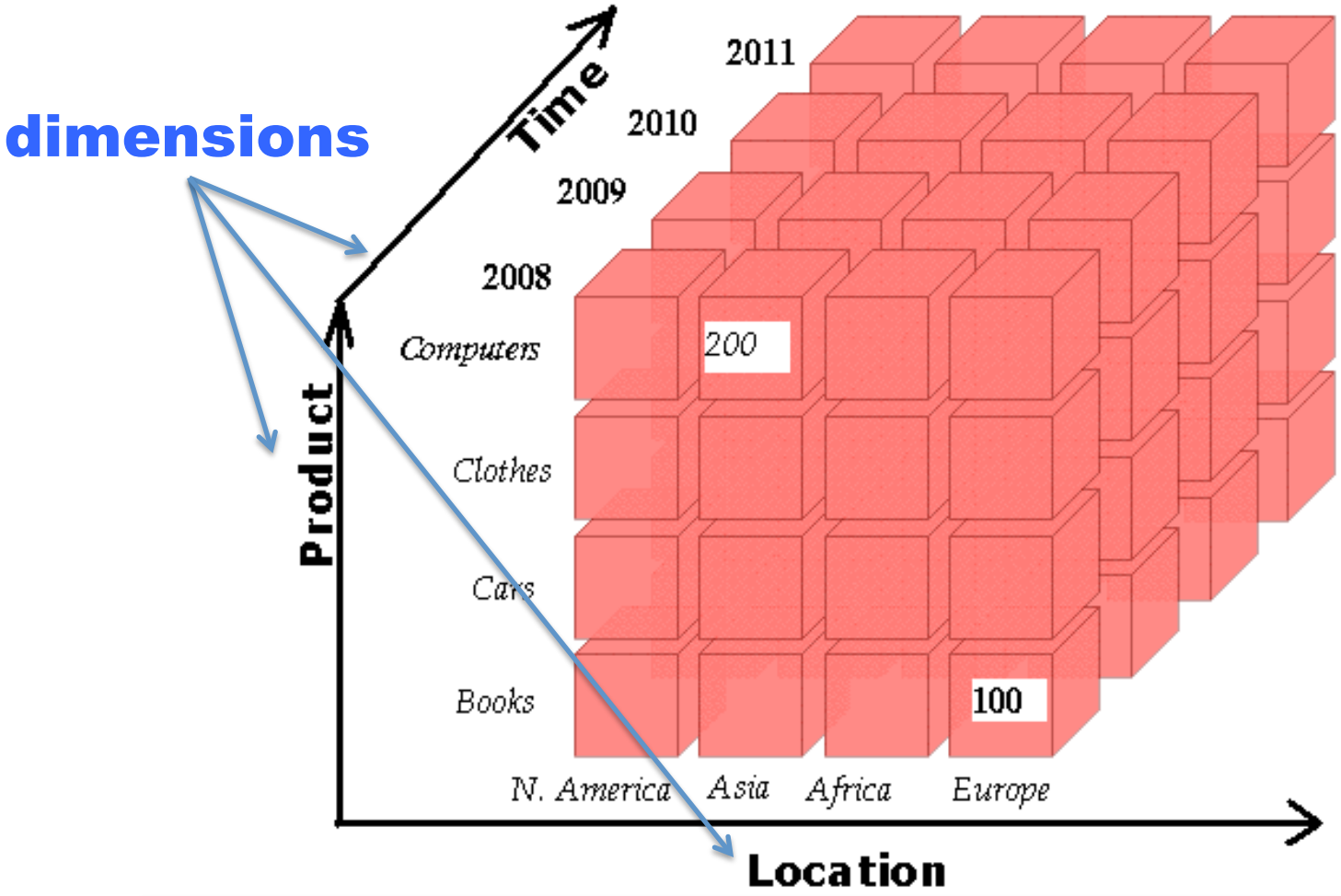

TIME HIERARCHY
Year
Year
Month
Month
Year
Date
Date
Month
Sales Fact Table
Date
Product
Store
Customer
unit_sales
dollar_sales
Product
ProductNo
ProdName
ProdDesc
Category
Store
StoreID
City
City
City
State
State
State
Country
Country
Region
Cust
CustId
CustName
CustCity
CustCountry
STORE LOCATION HIERARCHY

Star versus Snowflake Schema



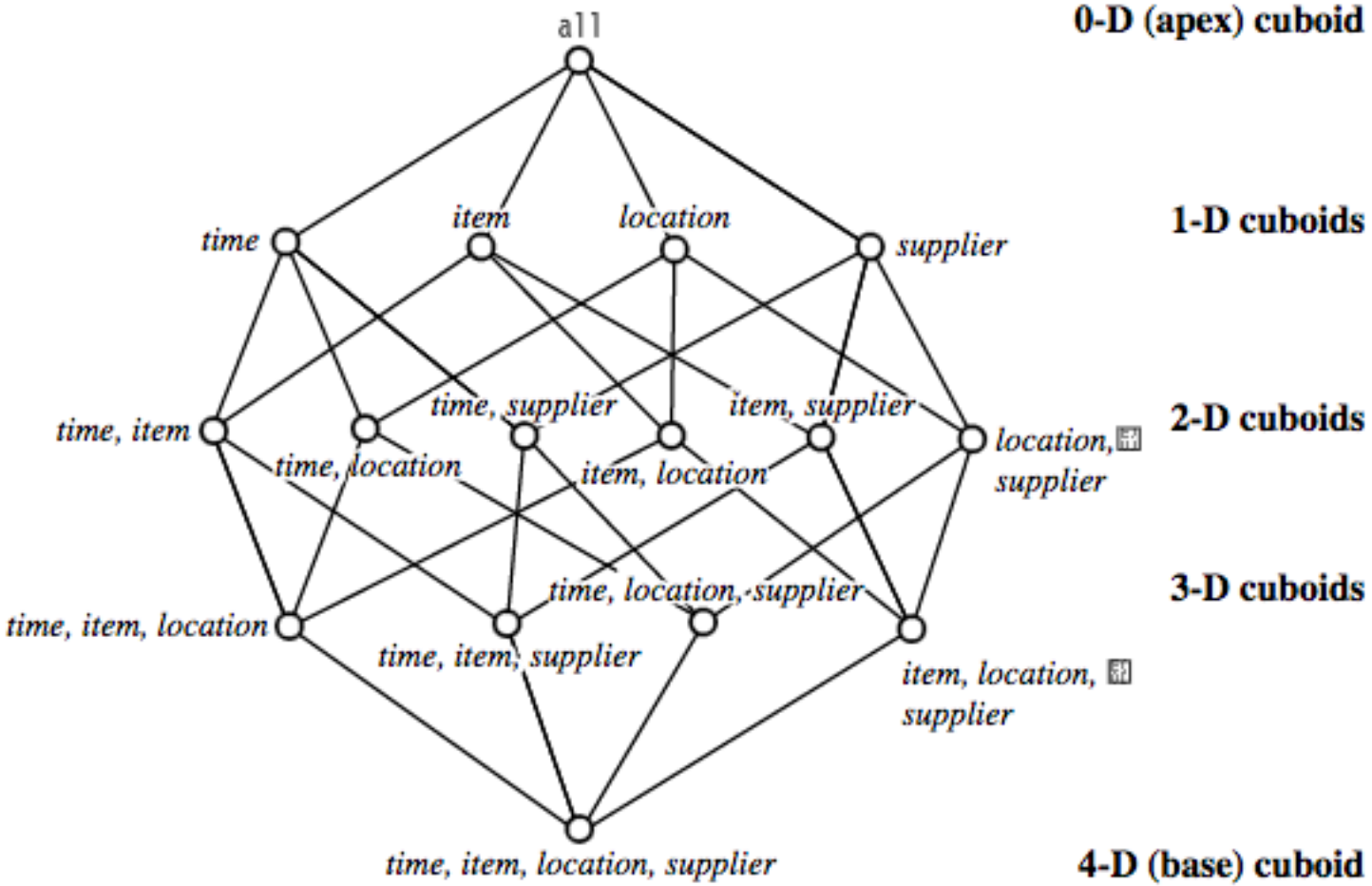


Another very powerful schema: Cubes (multi-dimensional tables)

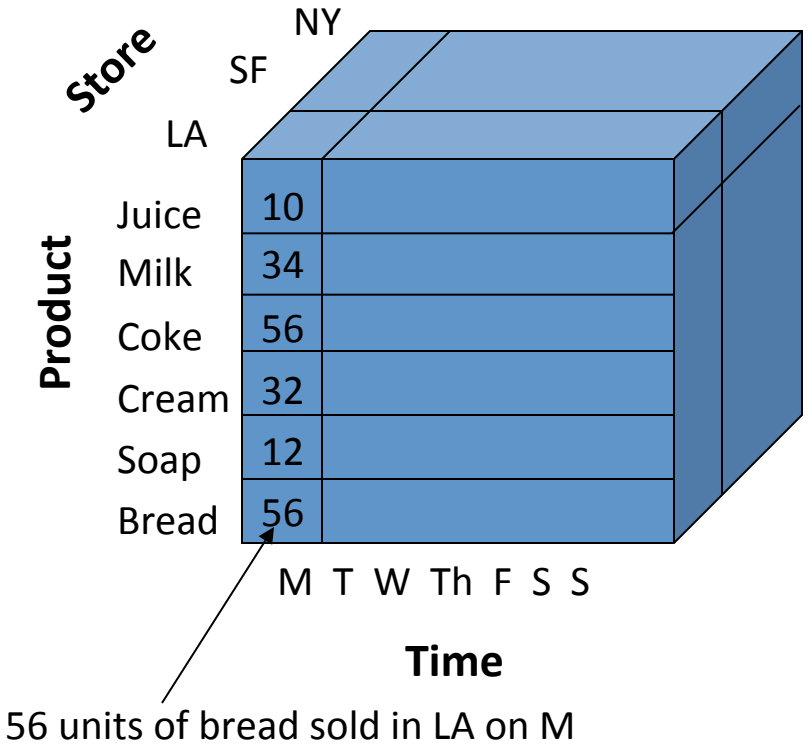


Note if dimensions >3, this would be a **hypercube!**)

>3 dimensions: cuboid



Another example of cube



Dimensions:

Time, Product, Store

Attributes:

Product (orders, price, …)

Store …

…

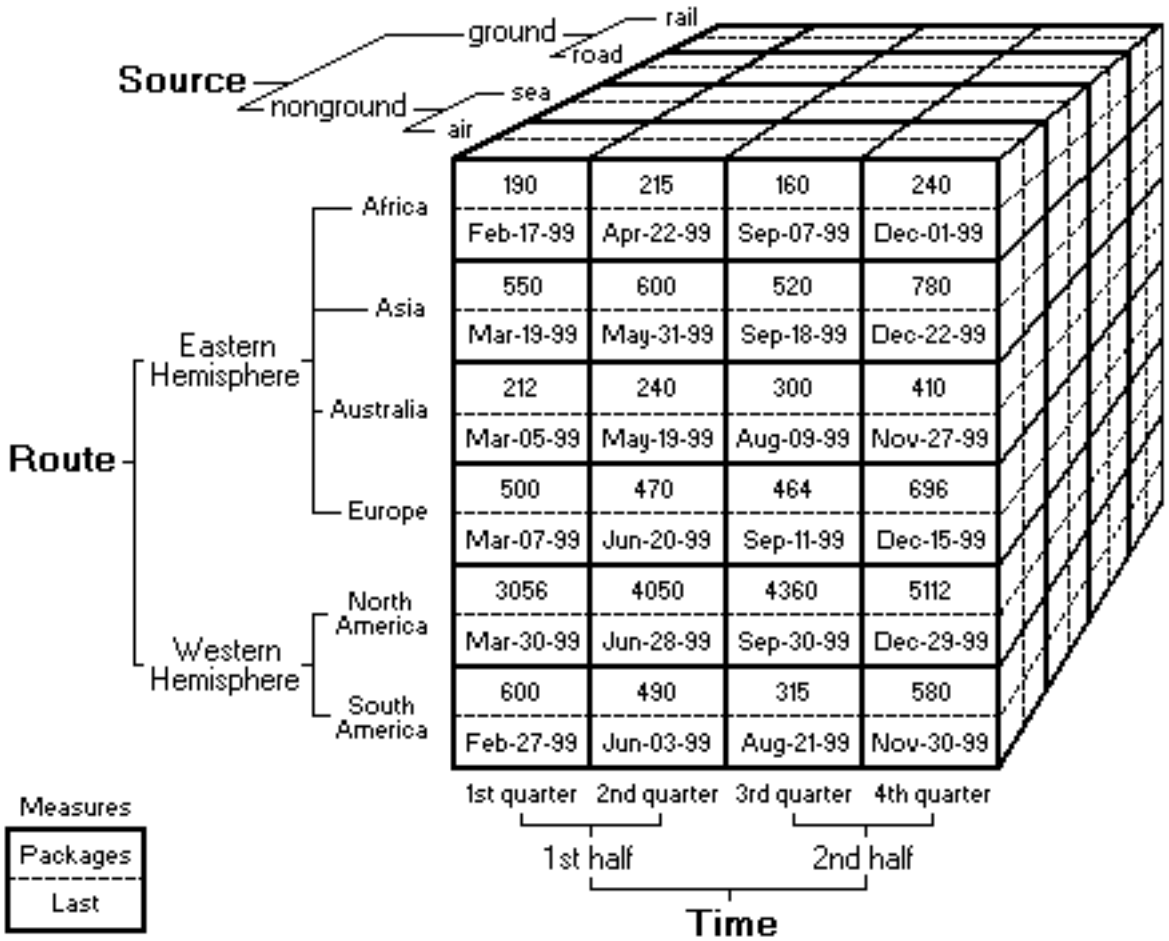
Hierarchies:

Product → Brand → …

Day → Week → Quarter

Store → Region → Country

A real-life example of cube: *SQL Server Books Online*

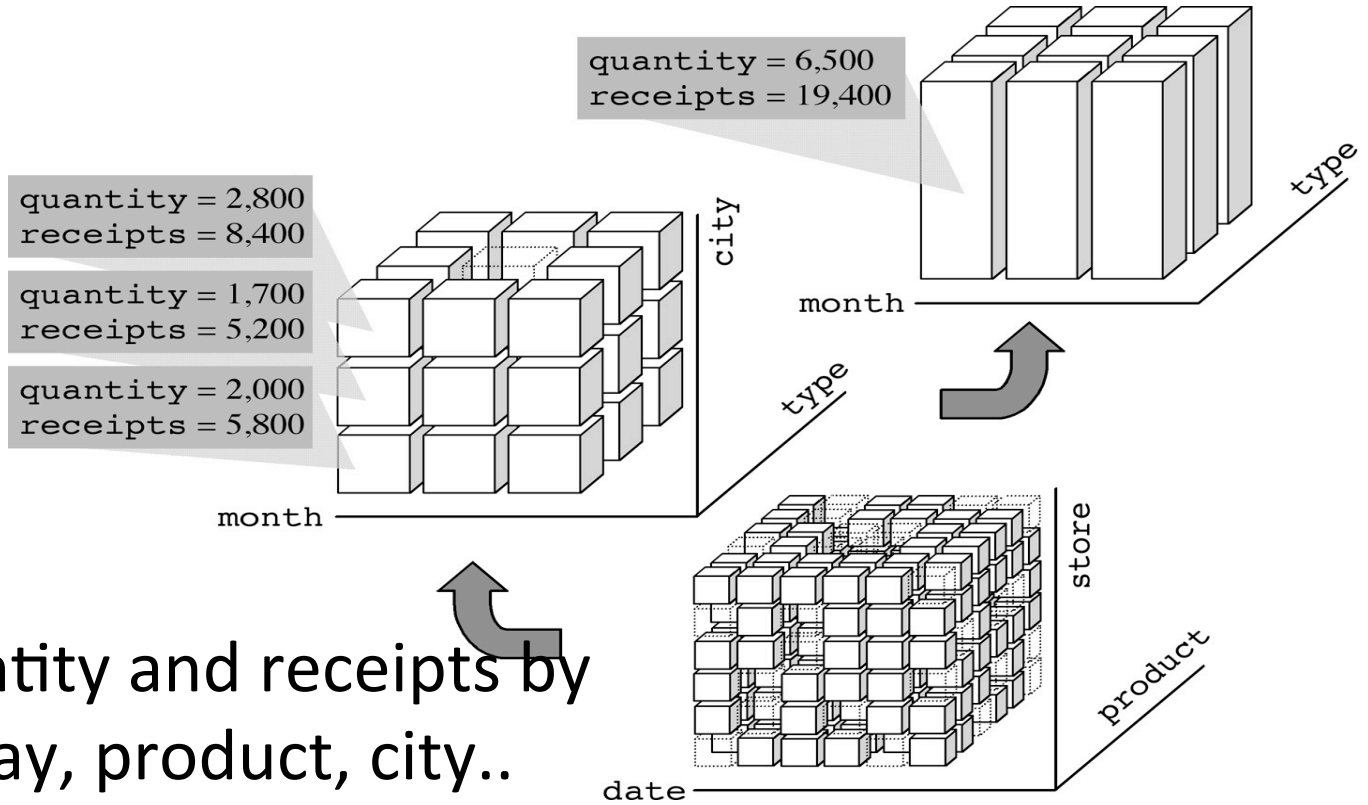


Process the warehouse: models and operations

- **Data Schema:** In which data structures we organize the loaded data?
  - Star
  - Snowflake
  - Multidimensional data (cubes)
- **Operators:** Which operations we can perform on the data?
  - slice & dice
  - roll-up, drill down
  - pivoting
  - other

Operations on data cubes

- The main purpose of DW is being able to aggregate/disaggregate/combine data using different perspectives and dimensions



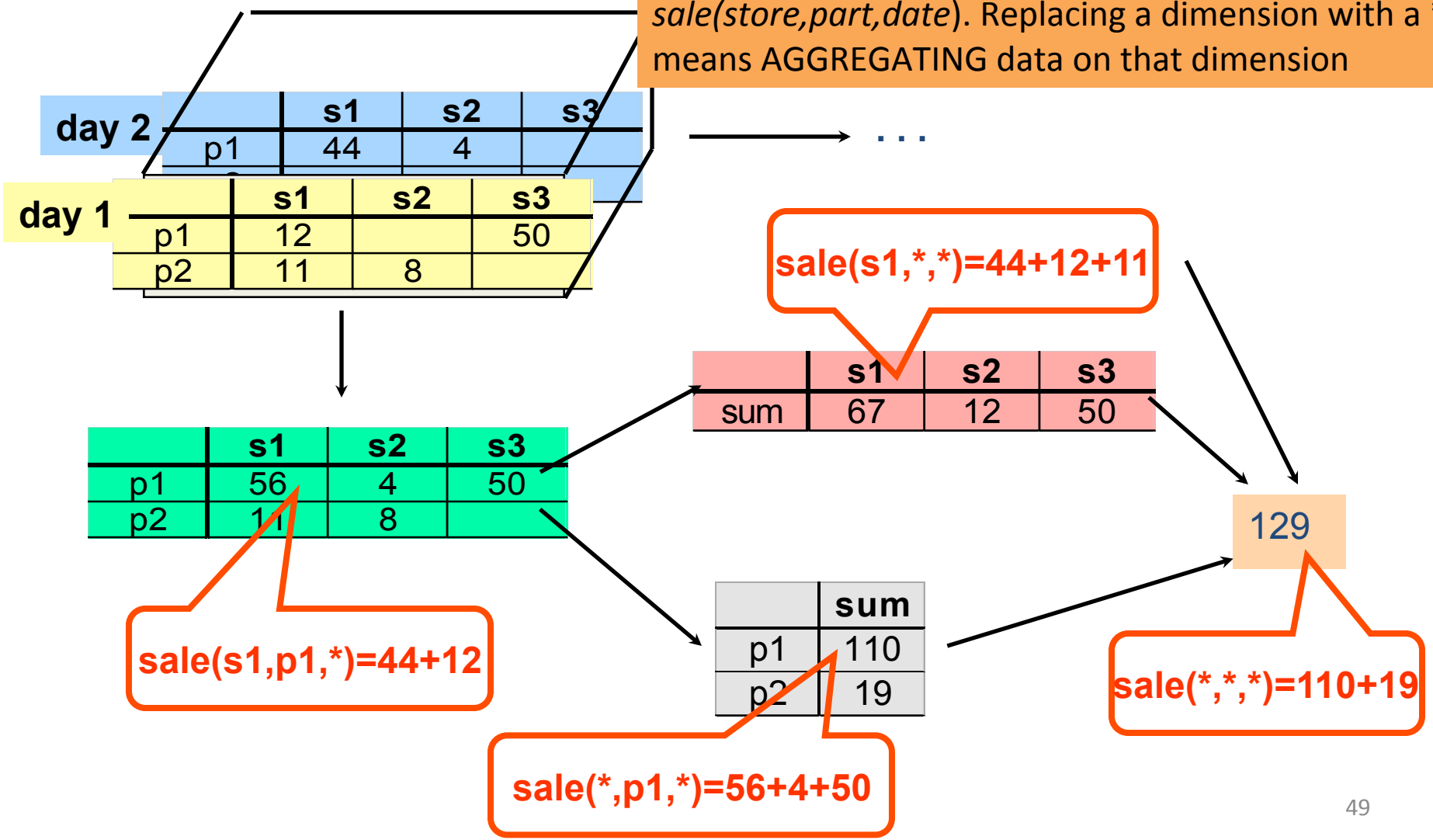
e.g., quantity and receipts by month, day, product, city..

Operations on cubes: (1) rolls up/ down

- **Roll-up**: (also called drill-up or **aggregation** operation) performs aggregation on a data cube, *climbing up a concept hierarchy for a dimension*
- **Roll-down** : *climbing down a concept hierarchy*, i.e. **dimension reduction**.

An example of Roll-up

Table stores sales by **store**, **part sold**, **date:** *sale(store,part,date)*. Replacing a dimension with a * means AGGREGATING data on that dimension

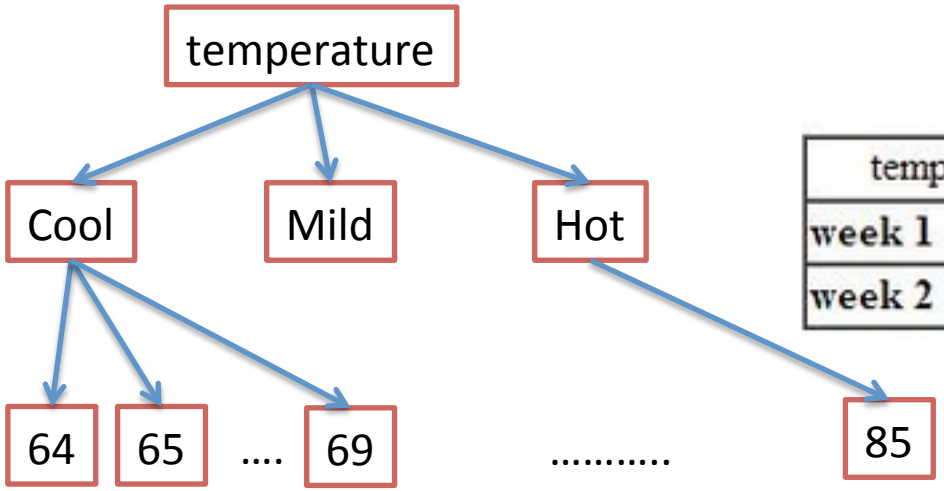


Roll-up using hierarchies (more complex)

This cube provides the number of days in each week a given temperature was reached
Dimensions are: temperature, week, day

temperature	64	65	68	69	70	71	72	75	80	81	83	85
week 1	1	0	1	0	1	0	0	0	0	0	1	0
week 2	0	0	0	1	0	0	1	2	0	1	0	0

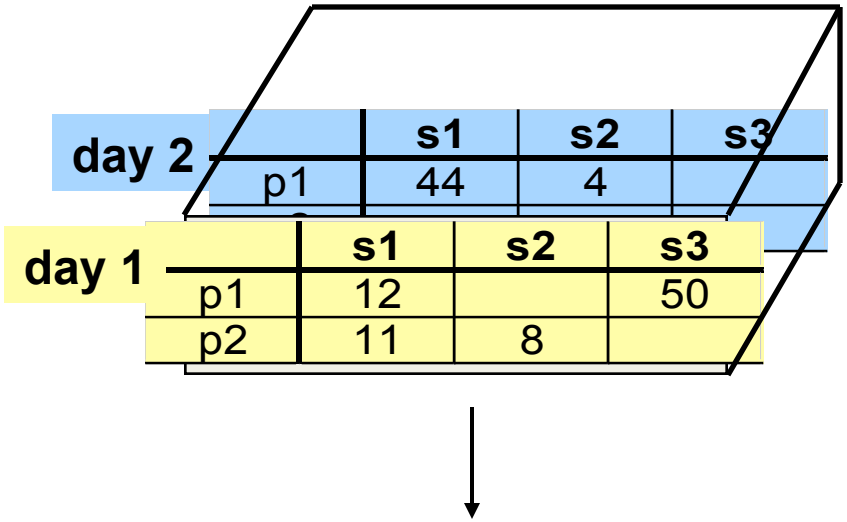
Assume we want to set up 3 levels (hot(80-85), mild(70-75), cool(64-69))in temperature from the above cube. To do this we have to group columns and add up the values according to the concept hierarchy.



temperature	cool	mild	hot
week 1	2	1	1
week 2	1	3	1

Note we need some code to do so, since levels are not explicitly present in the cube!!

Roll-up Using Hierarchies (2)



	region A	region B
p1	56	54
p2	11	8

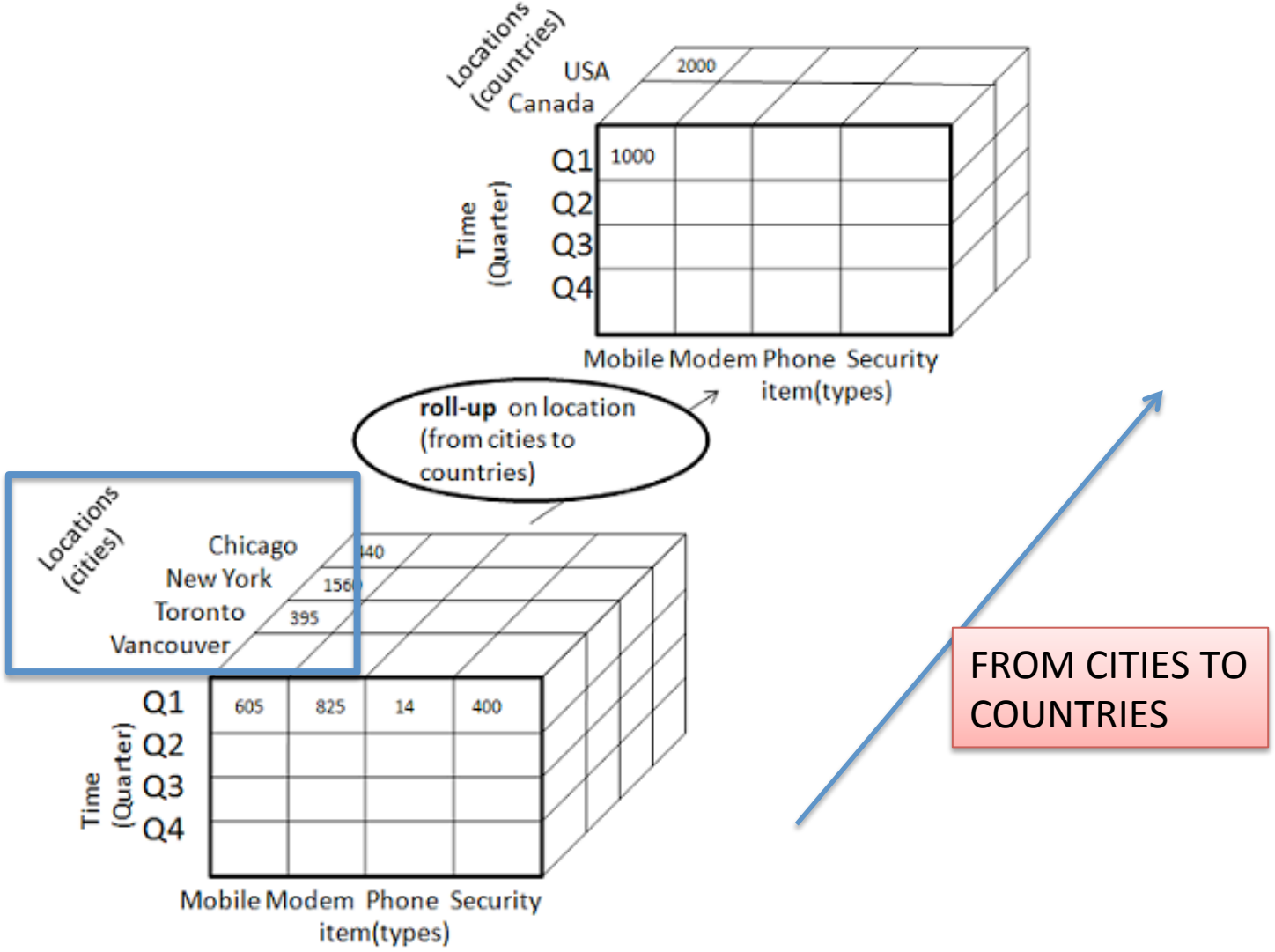
store

region

country

(store s1 in Region A;
stores s2, s3 in Region B)

Another example of roll up with hierarchies

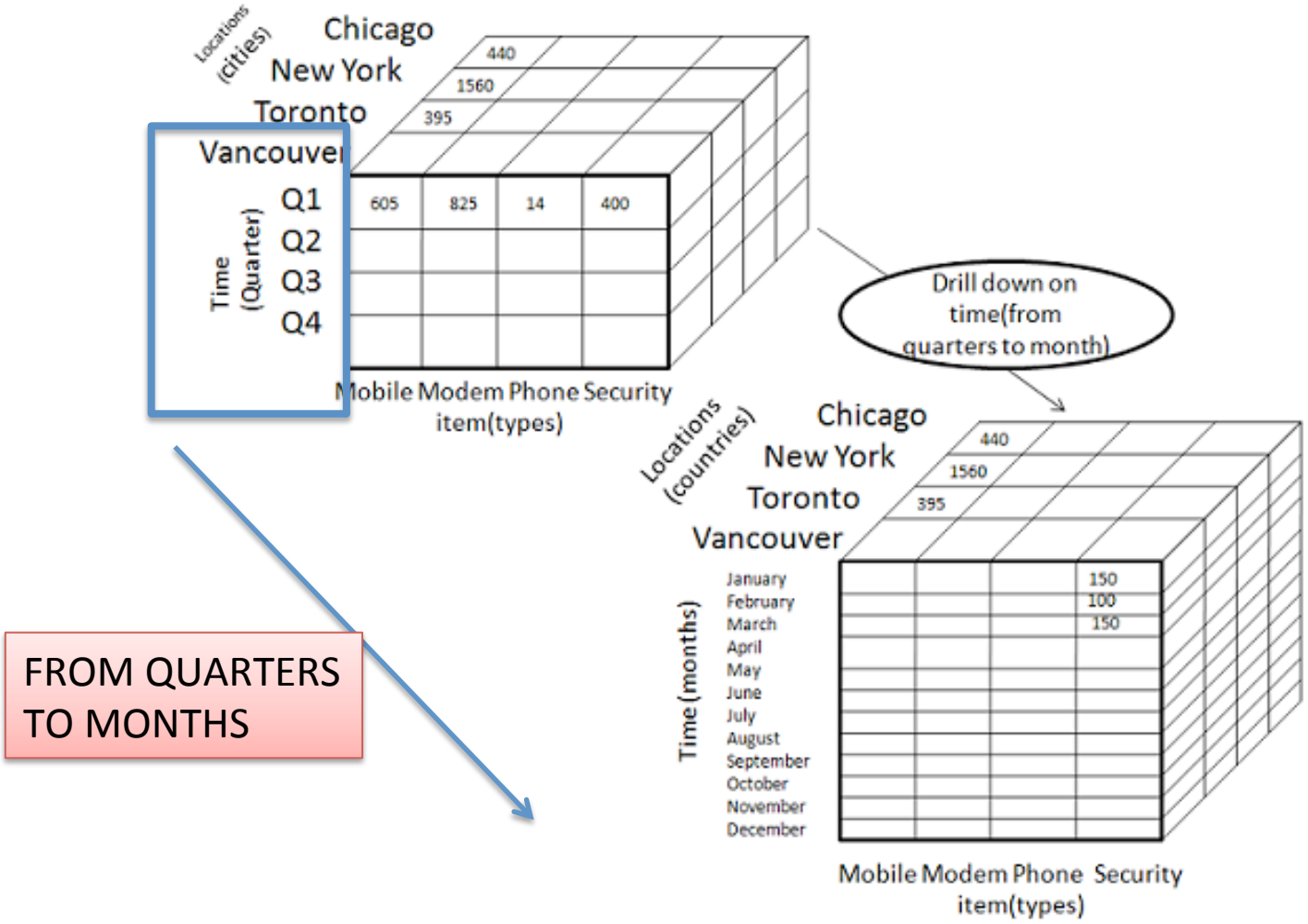


Roll-down (or drill-down)

	cool	mild	hot
day 1	0	0	0
day 2	0	0	0
day 3	0	0	1
day 4	0	1	0
day 5	1	0	0
day 6	0	0	0
day 7	1	0	0
day 8	0	0	0
day 9	1	0	0
day 10	0	1	0
day 11	0	1	0
day 12	0	1	0
day 13	0	0	1
day 14	0	0	0

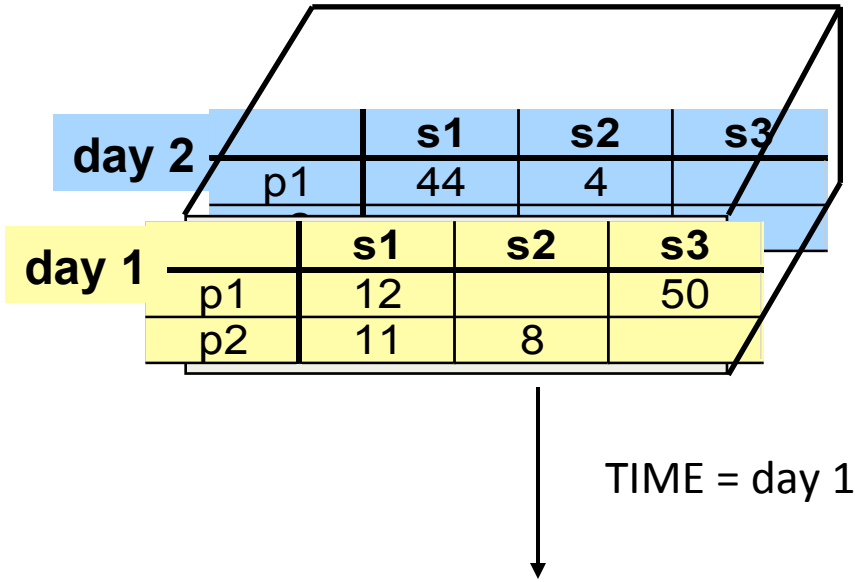
- Here we are “stepping down” the concept (from general to specific)
- E.g.: week→day
- We are going from the level of week to the more detailed level of day.

Roll-down example



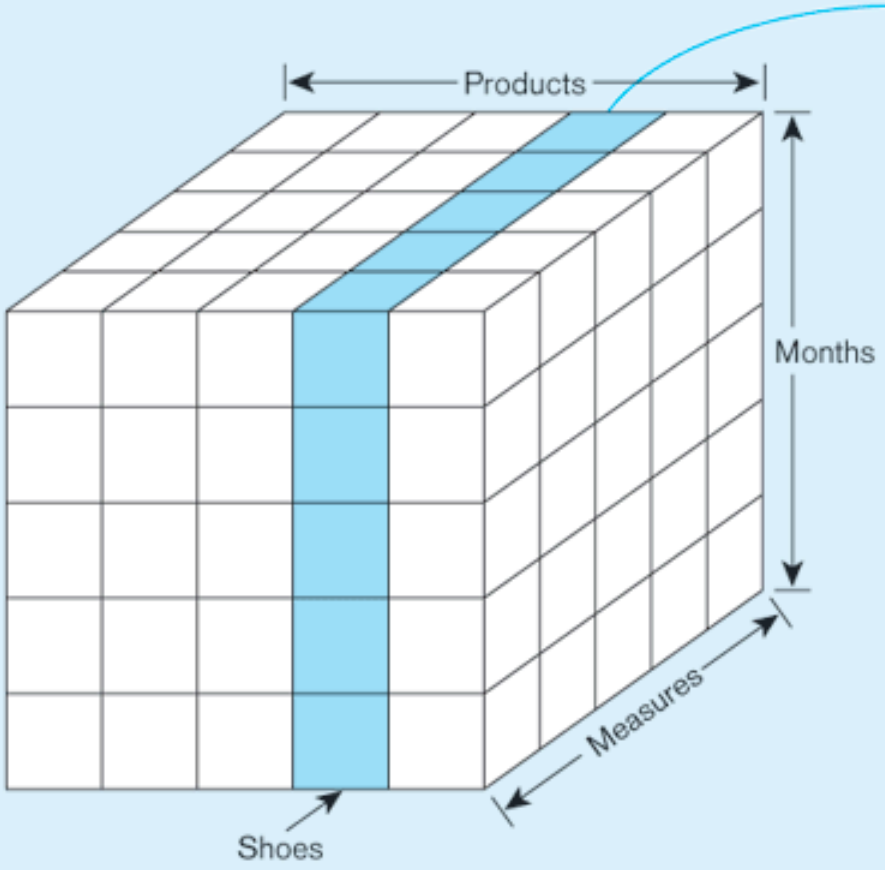
Operations on cube: 2) Slicing

- Slice performs a selection on one dimension of the given cube, thus resulting in a *subcube*.



	s1	s2	s3
p1	12		50
p2	11	8	

Another example of slicing



	Measure		
	Units	Revenue	Cost
January	250	1564	1020
February	200	1275	875
March	350	1800	1275
April	400	1935	1500
May	485	2000	1560

Product: Shoes

Operations on cube: 3) Dicing

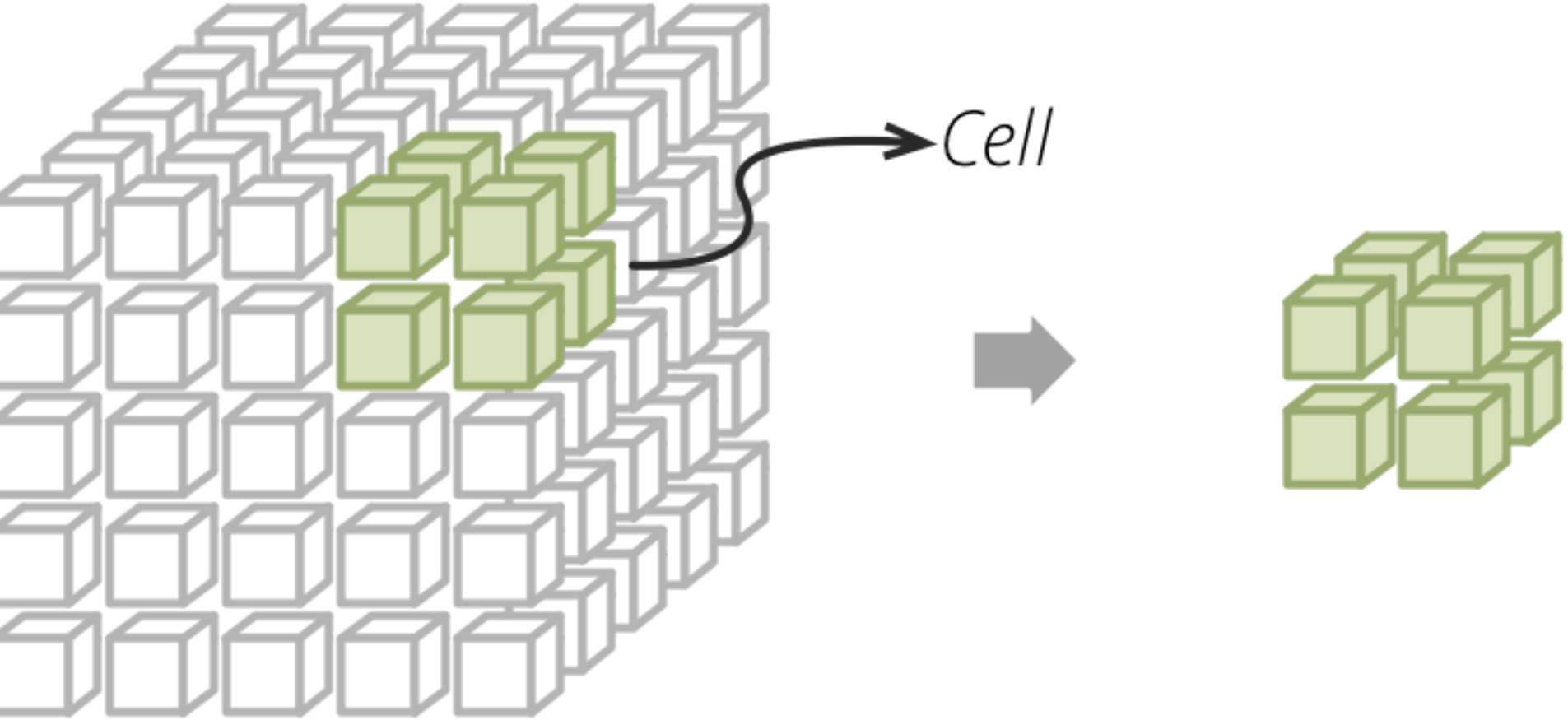
- The dice operation defines a **subcube** by performing a selection on **two or more** dimensions - so you extract a smaller cube (dice) from the cube.

	cool
day 1	0
day 2	0
day 3	0
day 4	0
day 5	1
day 6	0
day 7	1
day 8	0
day 9	1
day 10	0
day 11	0
day 12	0
day 13	0
day 14	0

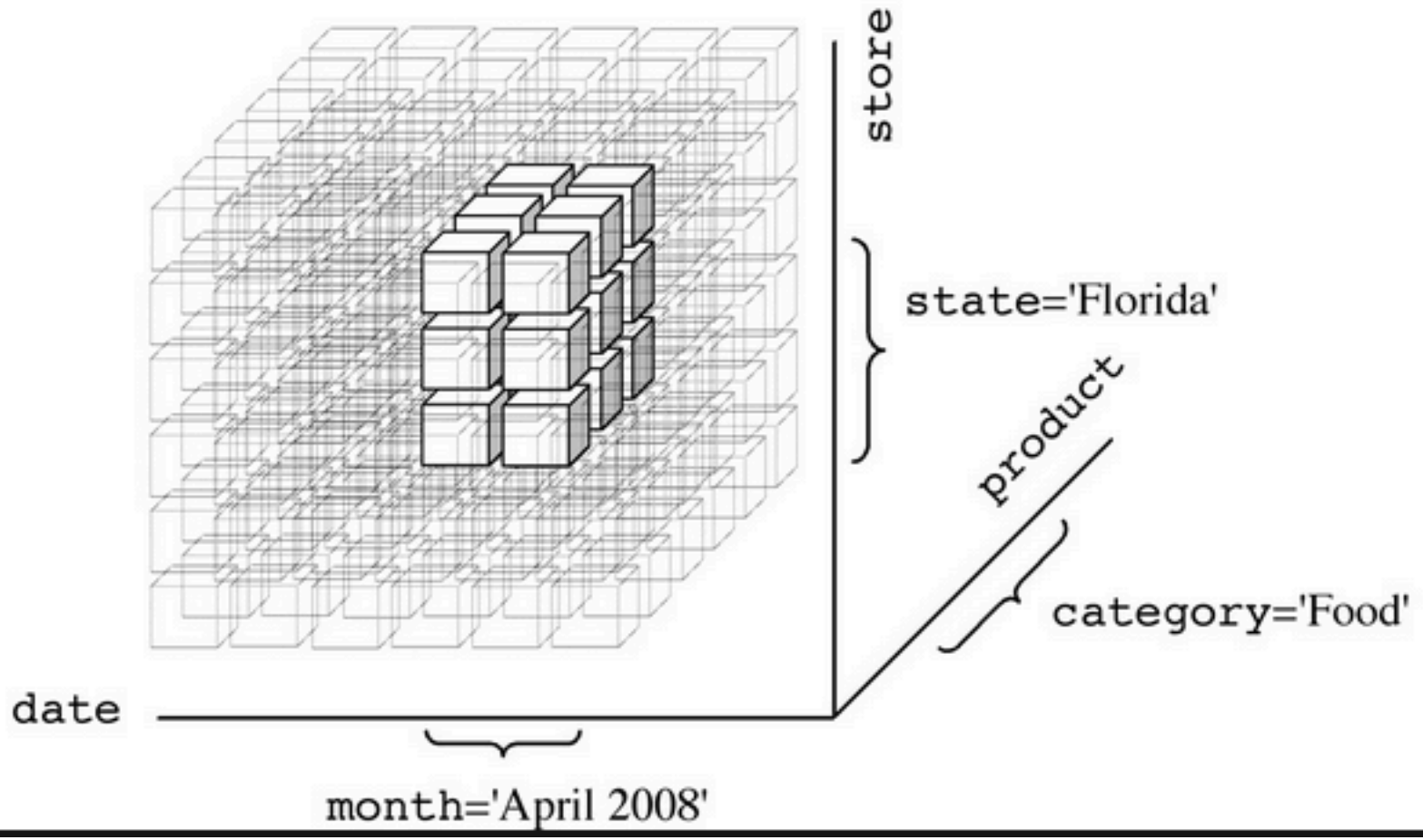
Time=(day 3)OR(day 4)

	cool	hot
day 3	0	1
day 4	0	0

Dicing on cubes

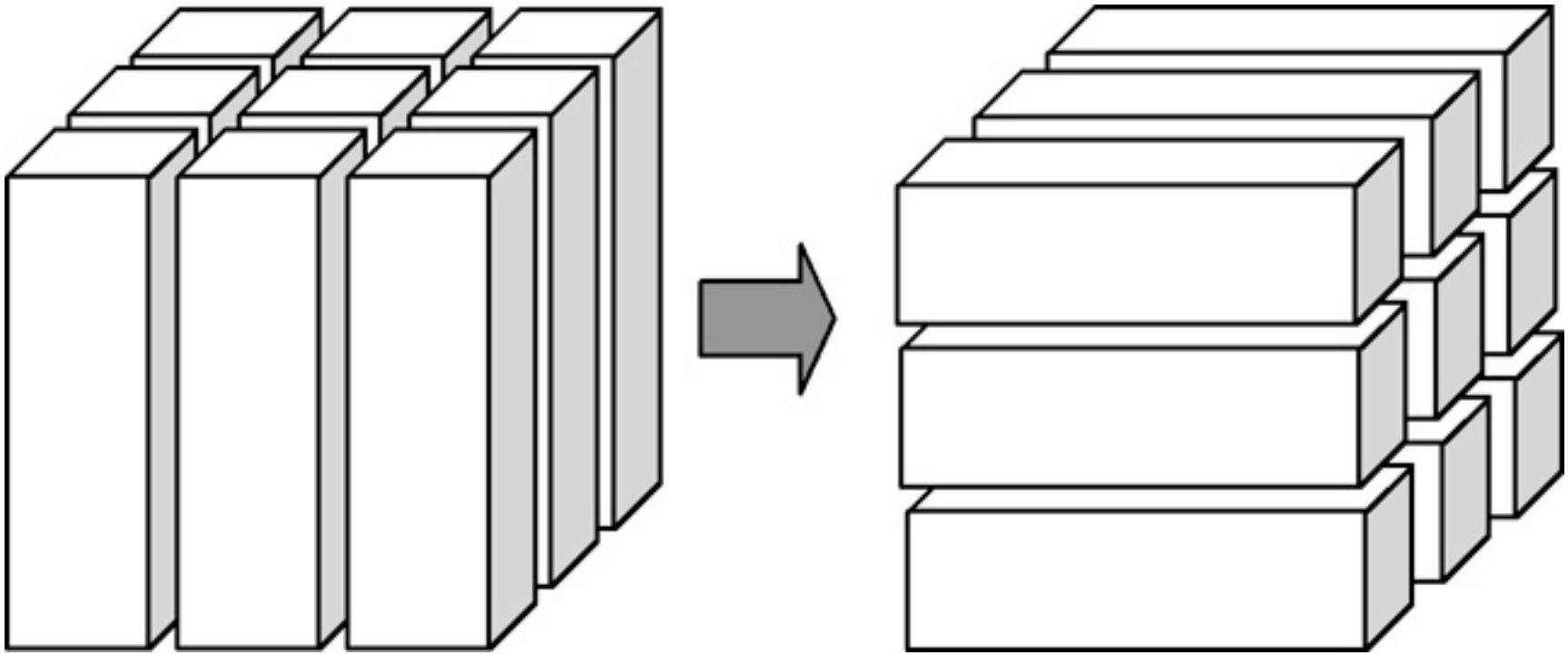


An example of dicing on cubes

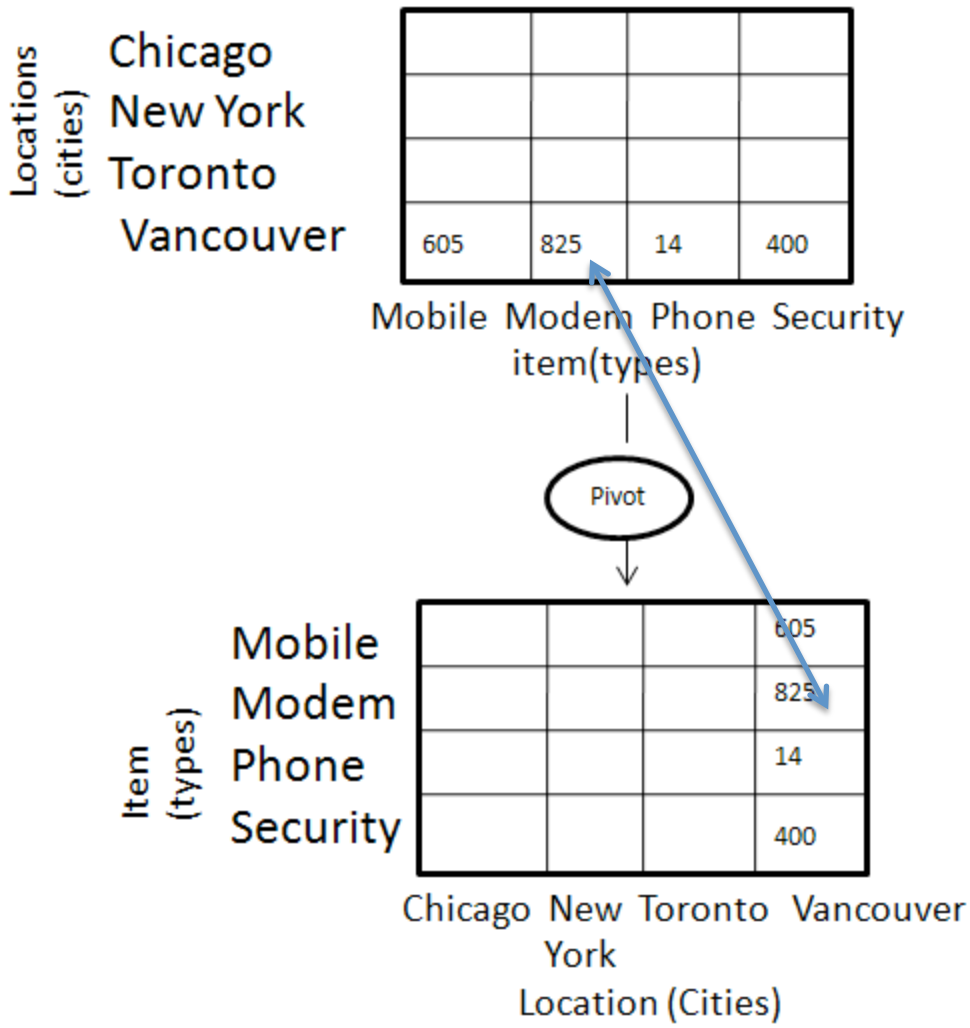


Operations on cubes: 4)Pivoting

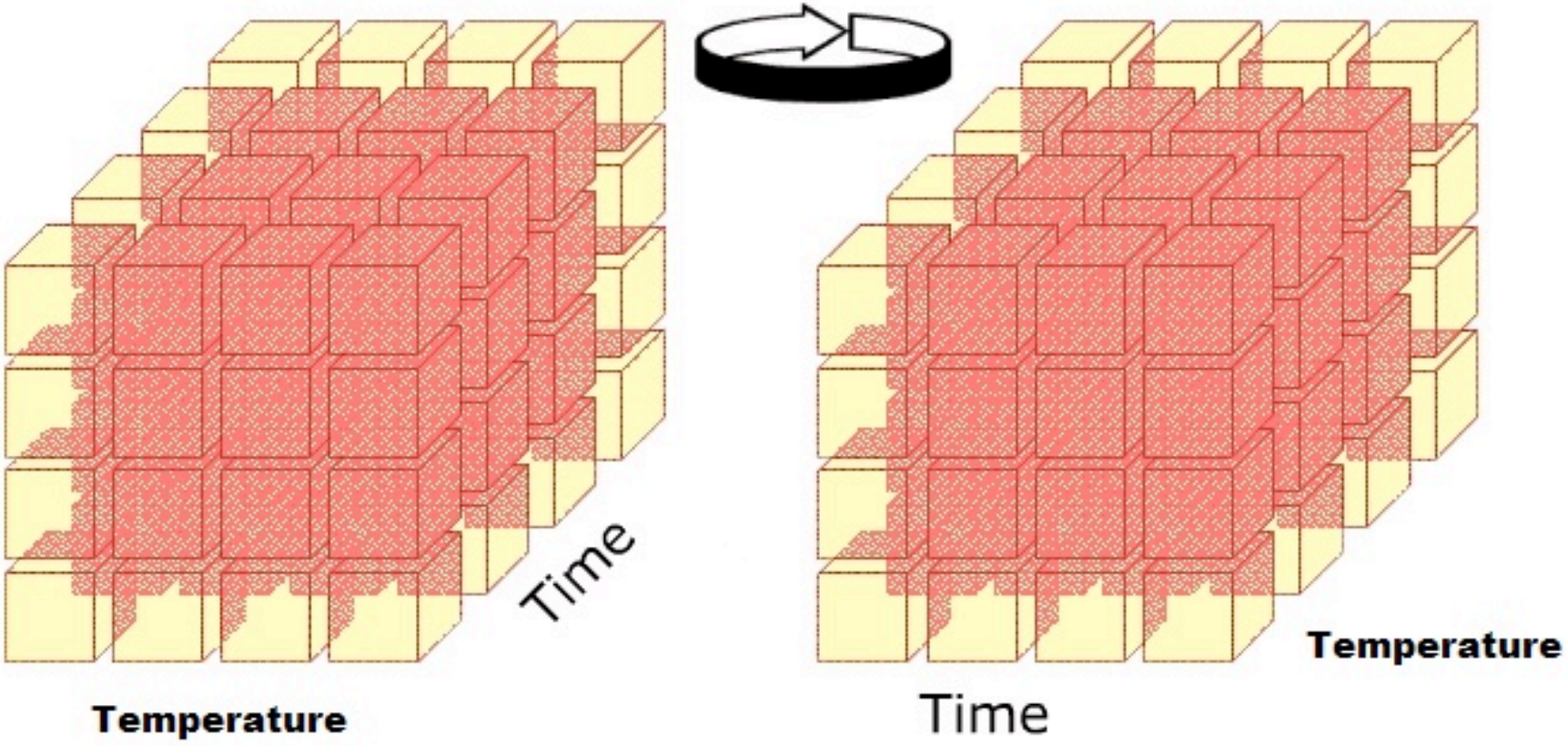
- Pivot otheriwise known as **Rotate** changes the dimensional orientation of the cube, i.e. rotates the data axes to view the data from different perspectives.



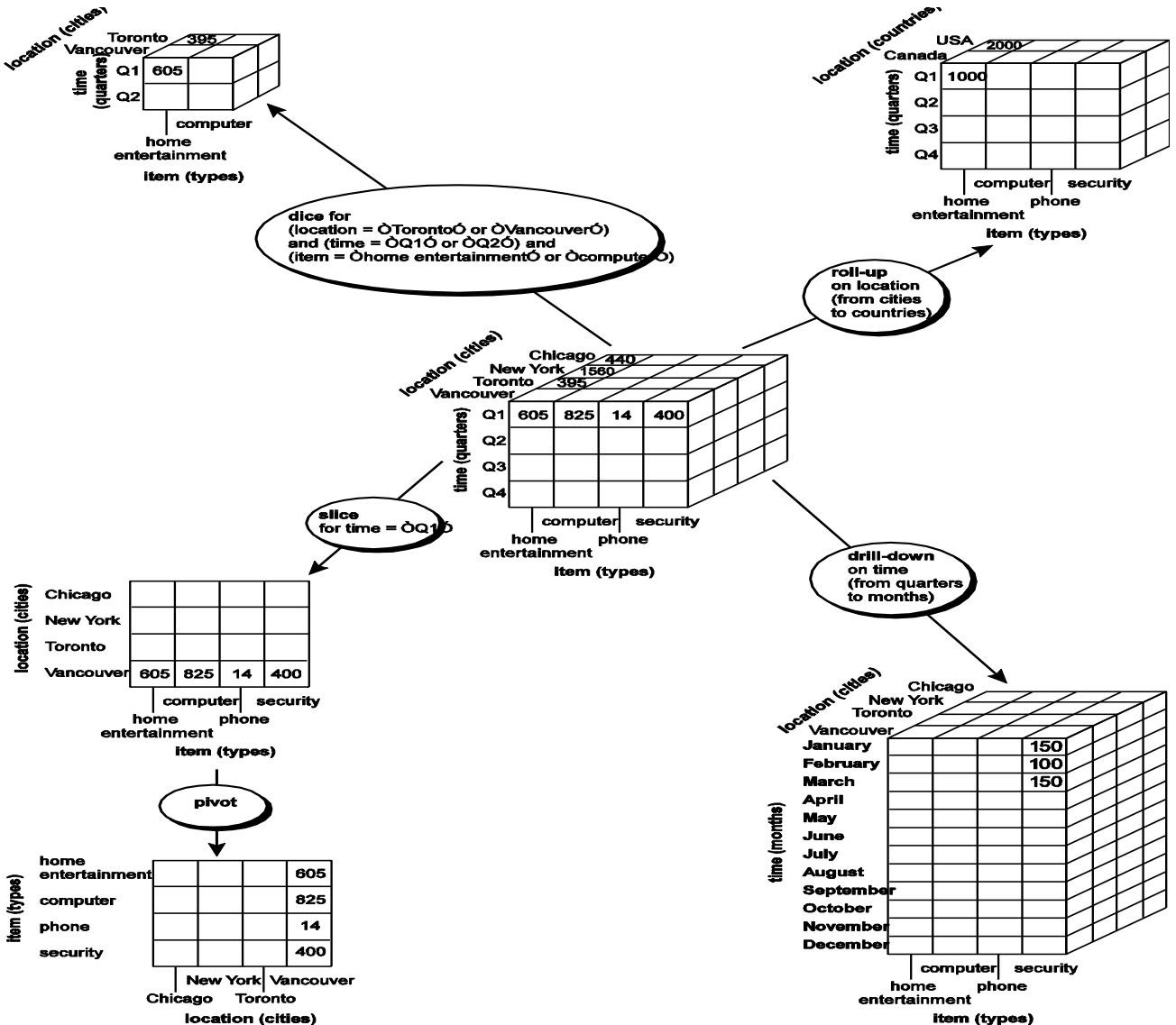
Example pivoting



Example of pivoting on cubes



Summary of operations on multidimensional data structures



DW vendors (some)

- IBM
  - http://www-306.ibm.com/software/data/informix/redbrick/
- Microsoft
  - http://www.microsoft.com/sql/solutions/bi/default.mspx
- Oracle
  - http://www.oracle.com/siebel/index.html
- Business Objects
  - http://www.businessobjects.com/

DW vendors (more)

- Microstrategy
  - http://www.microstrategy.com/
- Cognos
  - http://www.cognos.com/
- Informatica
  - http://www.informatica.com/
- Actuate
  - http://www.actuate.com/home/index.asp

Property Type	City	Time	Total Revenue
Flat	Glasgow	Q1	15056
House	Glasgow	Q1	14670
Flat	Glasgow	Q2	14555
House	Glasgow	Q2	15888
Flat	Glasgow	Q3	14578
House	Glasgow	Q3	16004
Flat	Glasgow	Q4	15890
House	Glasgow	Q4	15500
Flat	London	Q1	19678
House	London	Q1	23877
Flat	London	Q2	19567
House	London	Q2	28677
.....	.....	.....	.....
.....	.....	.....	.....

In class Exercise

- Organize the data in a cube
- Slice (based on shown data) on City=Glasgow
- Roll-up on Property Type

Next Topics

- Data Analytics:
  - Decision Support Systems (data mining and machine learning)
  - Unstructured data analytics (social and web data)

- Visualization interfaces