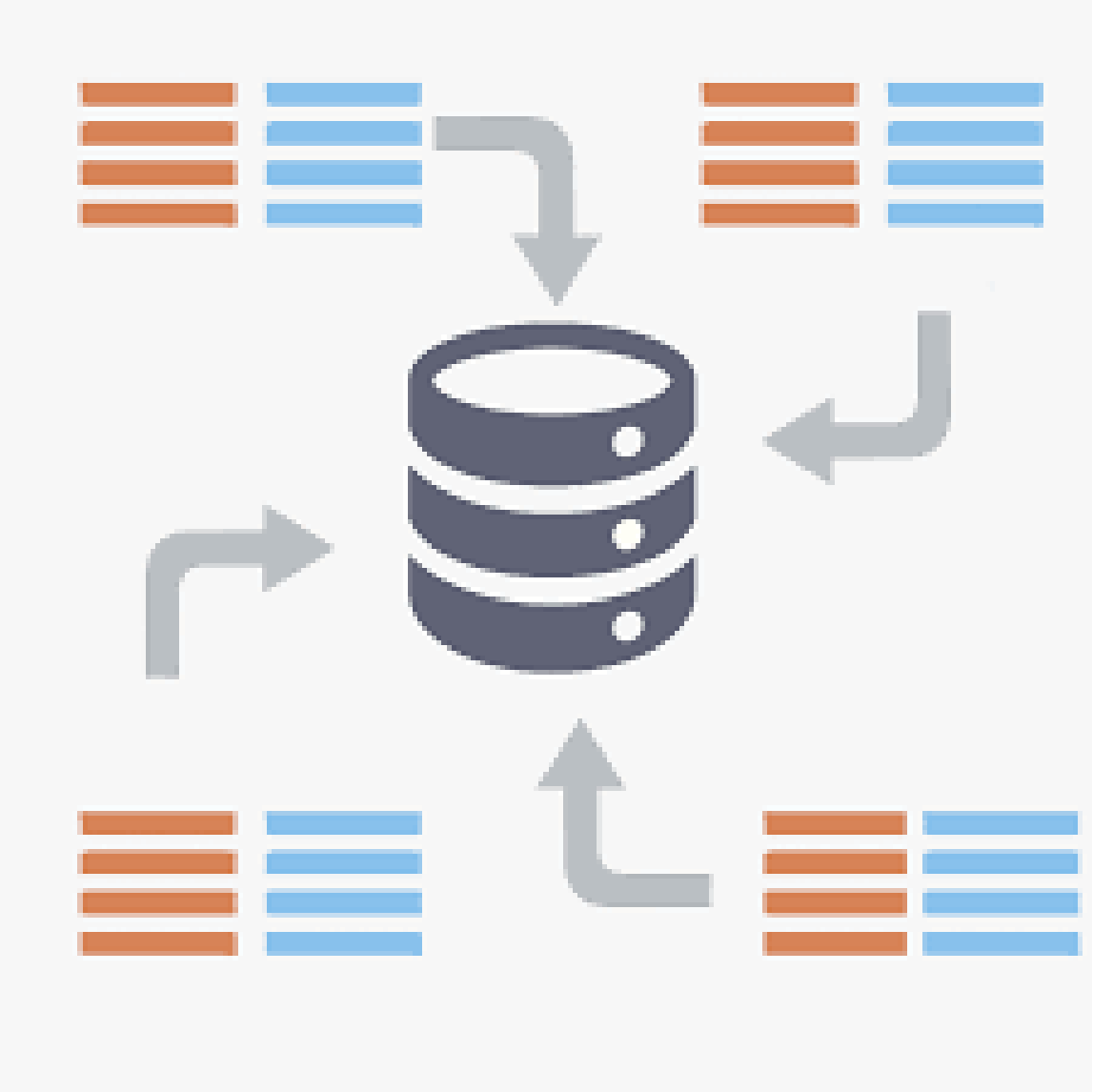
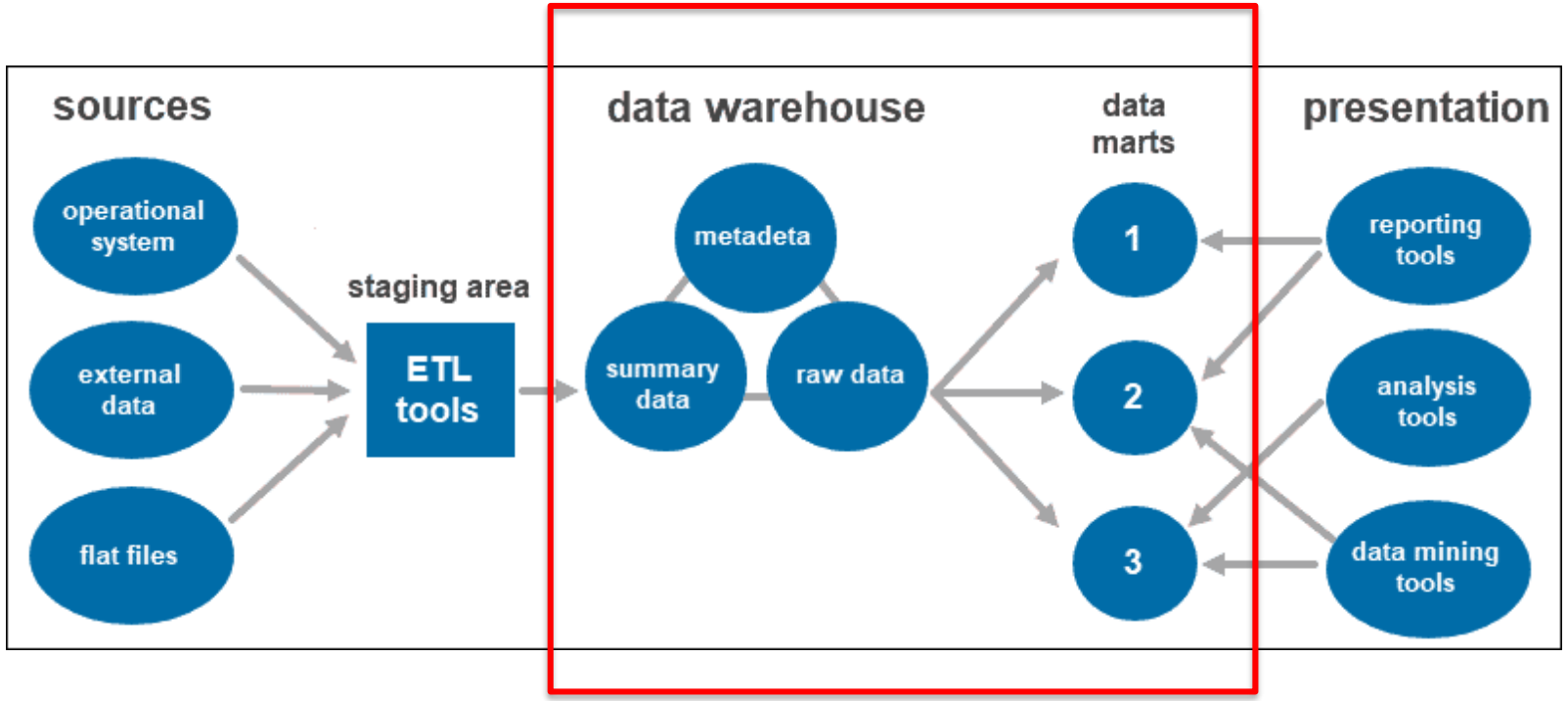
3. Data storage and data structures in Warehouses



Issues with datawarehouses

- Describing the data (metadata)
- Organizing different «views» of the data (data mars)
- Creating high-level data schemas for better analysis
- Operations with DW (how do we get information)

Elements in a datawarehouse



•**Metadata** is the information that **defines** the data. Its primary role is to simplify working with data instances, adding sematics and clarifying the meaning of collected data and descriptors.

•**Summary data** is generated by applying specific OPERATIONS to the data, with the purpose of aggregating information along specific dimensions (such as time, location, product type..)

•**Raw data** is the actual data loading into the repository, which has not been processed. Having the data in its raw form makes it accessible for further processing and analysis.

•**Data marts** are specific, department or application dependent views of the data.

Issues with datawarehouses

- **Describing the data (metadata)**
- Organizing different «views» of the data (data mars)
- Creating high-level data schemas for better analysis
- Operations with DW (how do we get information)

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 <u>and business users</u>
- 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 <u>definitions</u> 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 is the «who, what, where, why, when and how» of data

Who	What	Where	Why	When	How
Who created this data?	What is the business definition of this data element?	Where is this data stored?	Why are we storing this data?	When was this data created?	How is this data formatted? (character, numeric, etc.)
Who is the Steward of this data?	What are the business rules for this data?	Where did this data come from?	What is its usage & purpose?	When was this data last updated?	How many databases or data sources store this data?
Who is using this data?	What is the security level or privacy level of this data?	Where is this data used & shared?	What are the business drivers for using this data?	How long should it be stored?	
Who "owns" this data?	What is the abbreviation or acronym for this data element?	Where is the backup for this data?		When does it need to be purged/deleted?	
Who is regulating or auditing this data?	What are the technical naming standards for database implementation?	Are there regional privacy or security policies that regulate this data?			

In a sense, attribute names are the simplest type of metadata

- For a computer software, attribute names could be numeric: instead, we choose **intelligible** labels to help humans understanding the meaning of data.

Customer

First Name	Last Name	Company	City	Year Purchased
Joe	Smith	Komputers R Us	New York	1970
Mary	Jones	The Lord's Store	London	1999
Proful	Bishwal	The Lady's Store	Mumbai	1998
Ming	Lee	My Favorite Store	Beijing	2001

Metadata

Data

Meta-data add context and definitions, and reduce ambiguity

Customer

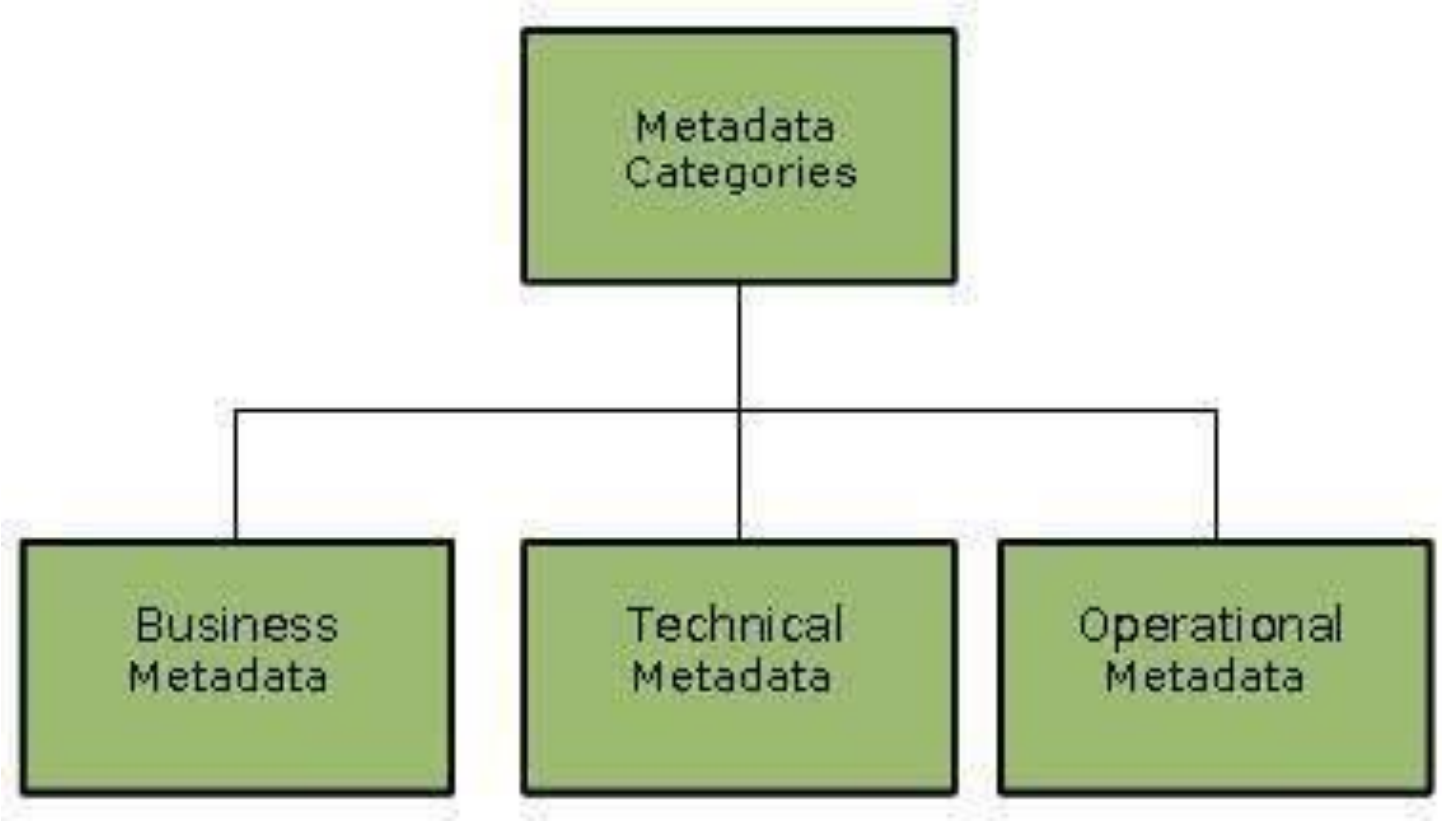
First Name	Last Name	Company	City	Year Purchased
Joe	Smith	Kompu ers R Us	New York	1970
Mary	Jones	The ord's Store	London	1999
Proful	Bishwal	The Lady's Store	Mumbai	1998
Ming	Lee	My Favorite Store	Beijing	2001

Is this the city where the customer lives or where the store is located?

Definition	Last Name represents the surname or family name of an individual.
Business Rules	In the Chinese market, family name is listed first in salutations.
Format	VARCHAR(30)
Abbreviation	LNAME
Required	YES
Etc.	Numerous technical & business metadata including security, privacy, nullability, primary key, etc.

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..



Technical and Business Metadata

Technical Metadata

```
CREATE TABLE EMPLOYEE (
    employee_id        INTEGER NOT NULL,
    department_id      INTEGER NOT NULL,
    employee_fname     VARCHAR(50) NULL,
    employee_lname     VARCHAR(50) NULL,
    employee_ssn       CHAR(9) NULL);

CREATE TABLE CUSTOMER (
    customer_id        INTEGER NOT NULL,
    customer_name      VARCHAR(50) NULL,
    customer_address   VARCHAR(150) NULL,
    customer_city      VARCHAR(50) NULL,
    customer_state     CHAR(2) NULL,
    customer_zip       CHAR(9) NULL);
```

Business Metadata

Term	Definition
Employee	An employee is an individual who currently works for the organization or who has been recently employed within the past 6 months.
Customer	A customer is a person or organization who has purchased from the organization within the past 2 years and has an active loyalty card or maintenance contract.

Data

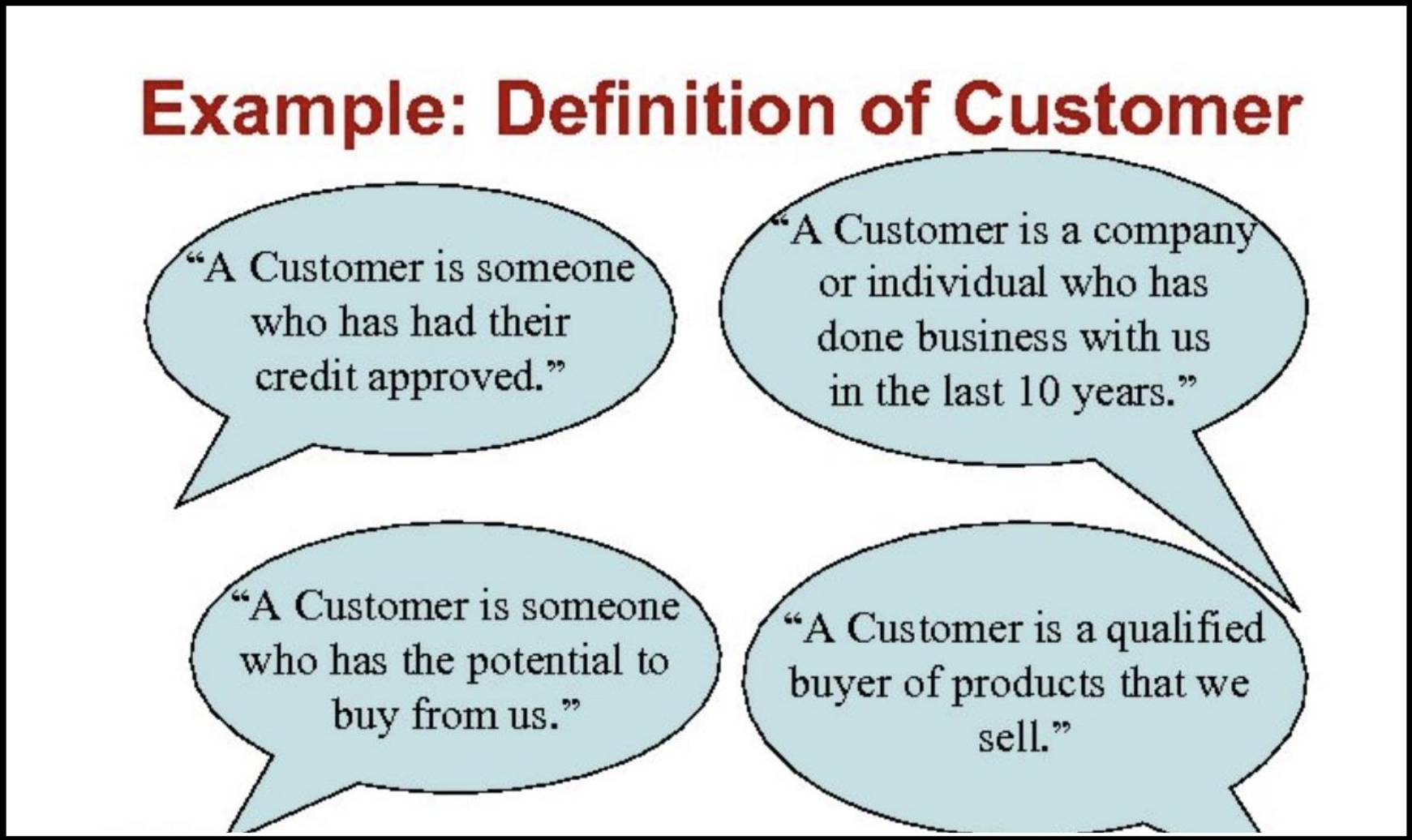
John Smith

Examples of Business and Technical metadata

"A customer is a person or organization who purchases a product or service from..."

"CUST_LNM is VARCHAR(30) on the Oracle database CustDB1.

Business Metadata	Technical Metadata
• Definitions & Glossary • Data Steward • Organization • Privacy Level • Security Level • Acronyms & Abbreviations • Business Rules • Etc.	• Column structure of a database table • Data Type & Length (e.g. VARCHAR(20)) • Domains • Standard abbreviations (e.g. CUSTOMER -> CUST) • Nullability • Keys (primary, foreign, alternate, etc.) • Validation Rules • Data Movement Rules • Permissions • Etc.

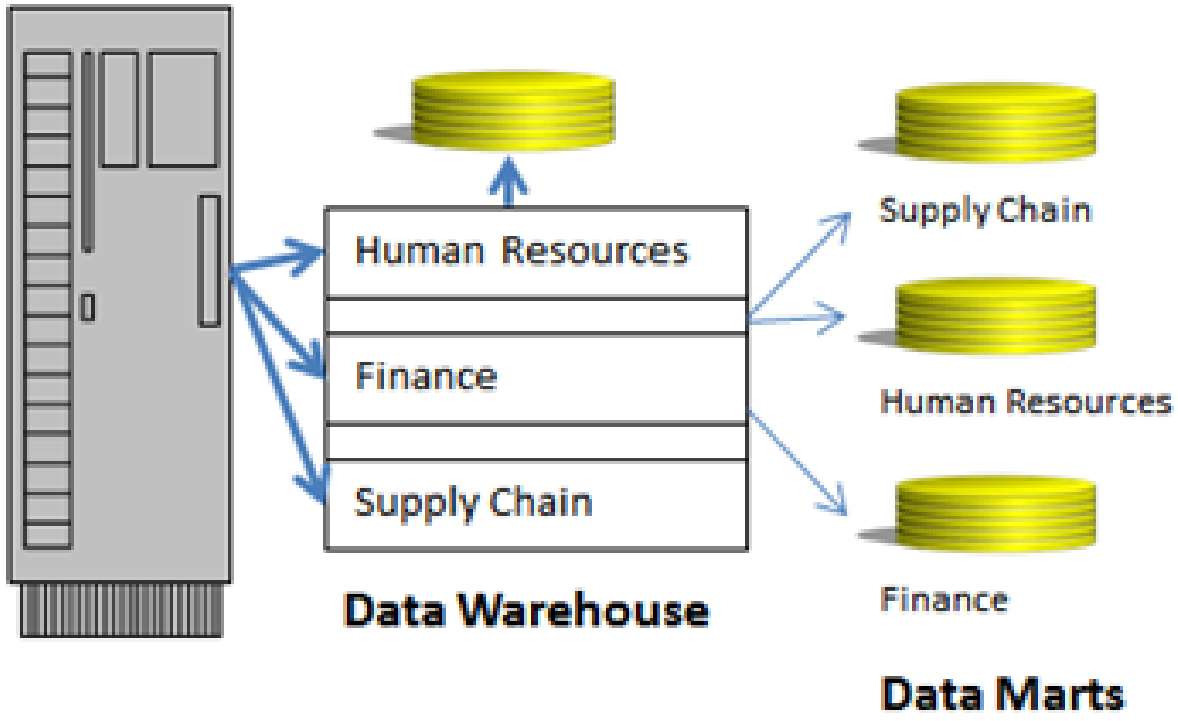


Same entity can have different definitions in different contexts

Issues with datawarehouses

- Describing the data (metadata)
- **Organizing different «views» of the data**
- Creating high-level data schemas for better analysis
- Operations with DW (how do we get information)

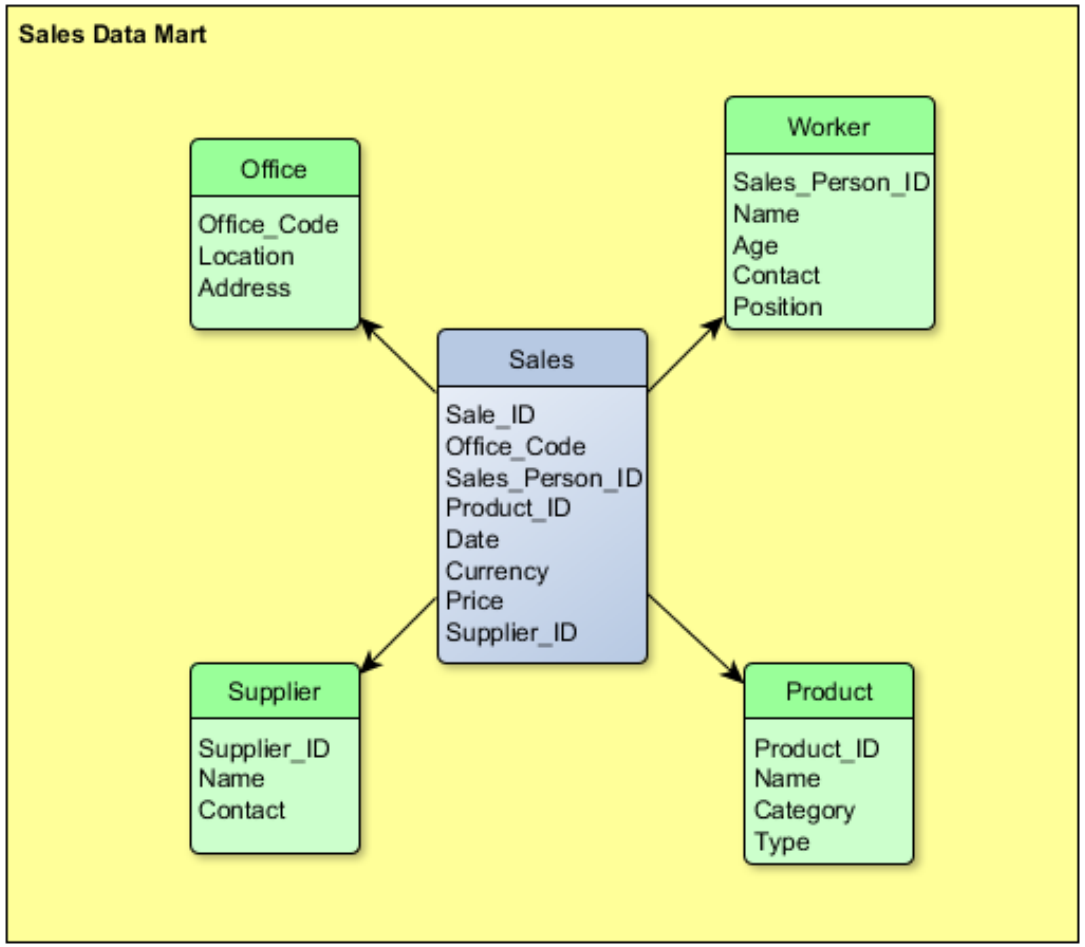
Data storage: Data Marts and data structures



Data Mart

- A **data mart is one piece of a data warehouse** where all the information is related to a **specific business area** (e.g., sales, repairs, shipment, customer care..).
- 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

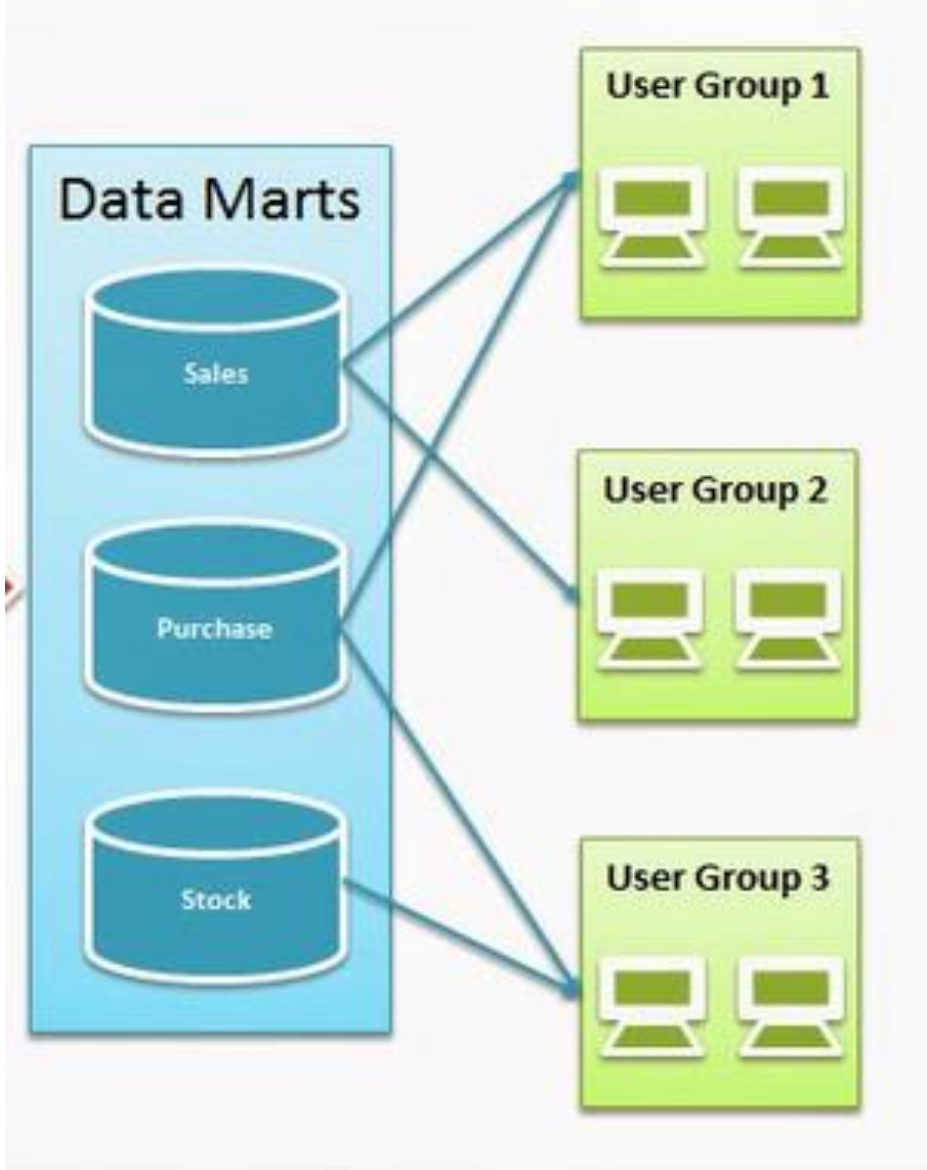
- Since this is a data mart, all the information contained in this data structure is only relative to **sales** and its dependencies.

Why data Marts?

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 (e.g. for customer relationships integrating only social and purchase data of customers)

Why data Marts?

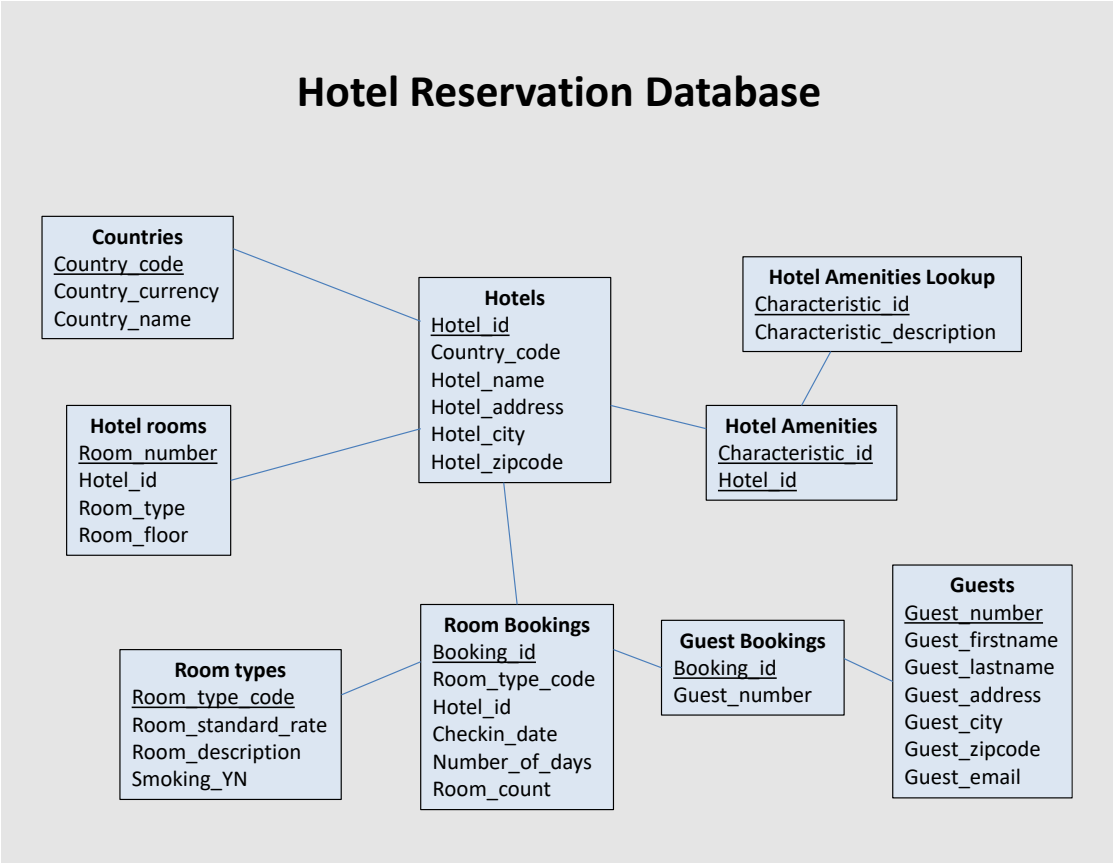
3. Creation

- 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 may create problems with **inconsistency**. Since cleary there is overlapping information among datamarts, changing one piece of info on a mart without updating others, might create inconsitency.
- This problem has been widely recognized, so data marts exist in two styles:
  - **Independent** data marts are those which are fed directly from the original source data (OLTP). 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

Example: TU hotel chains

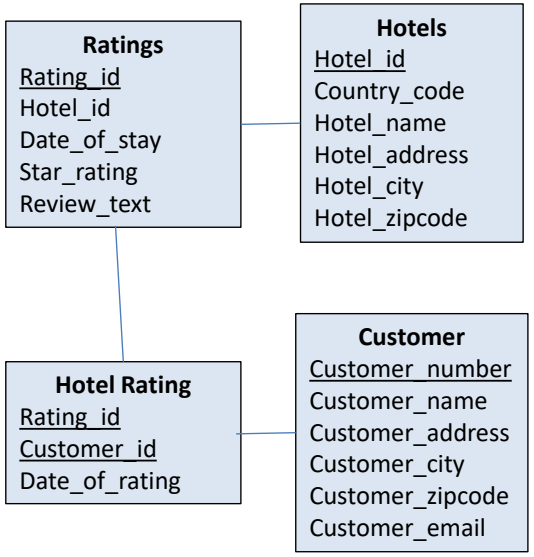


- This is the "view" of the reservation department

Ratings and Café are possible additional Marts for the same domain

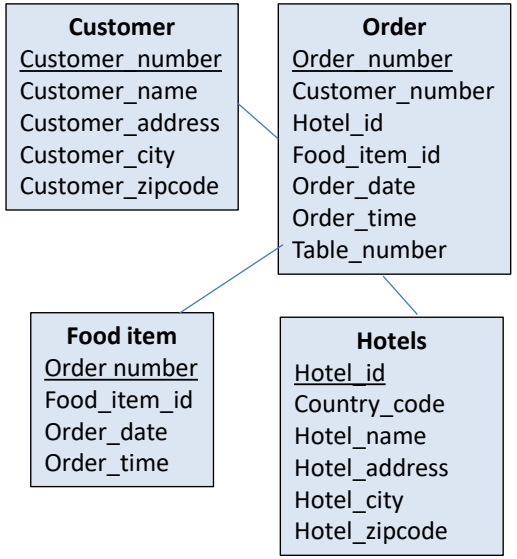
HotelComplainer Ratings Database

(totally external company)



Café in the Hotel Database

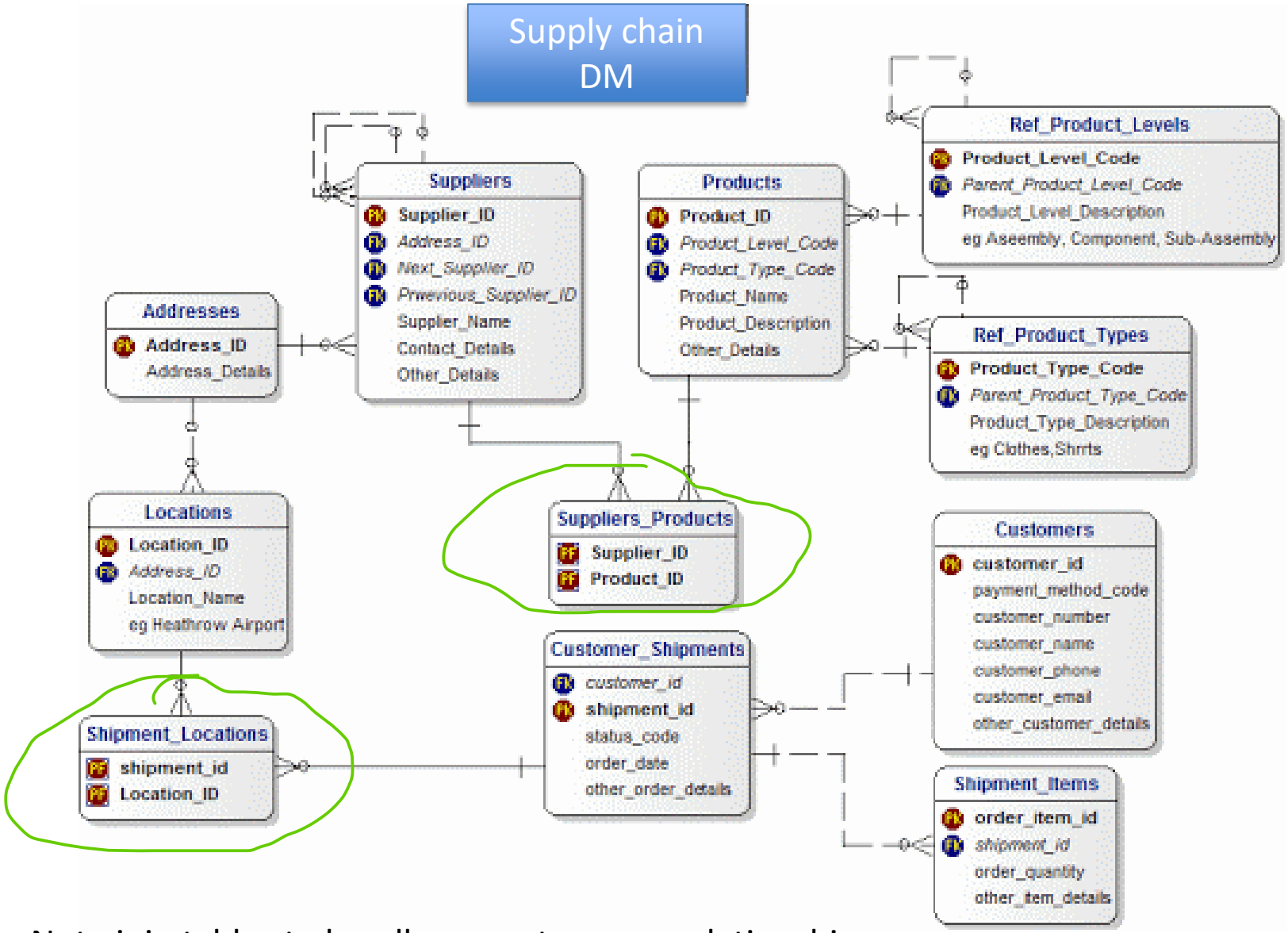
(same company but database is not connected to the hotel)



In class exercise

- Suppose TUCCI is a fashion company. Their datawharehouse store information about sales, customer care & complaints, supply chain, employees in the various point of sales.
- How many data marts would you organize?
- Can you show two of these datamarts (relevant entities and connections)
- Can you add business metadata to (at least some of the) entities and attributes of one datamart?

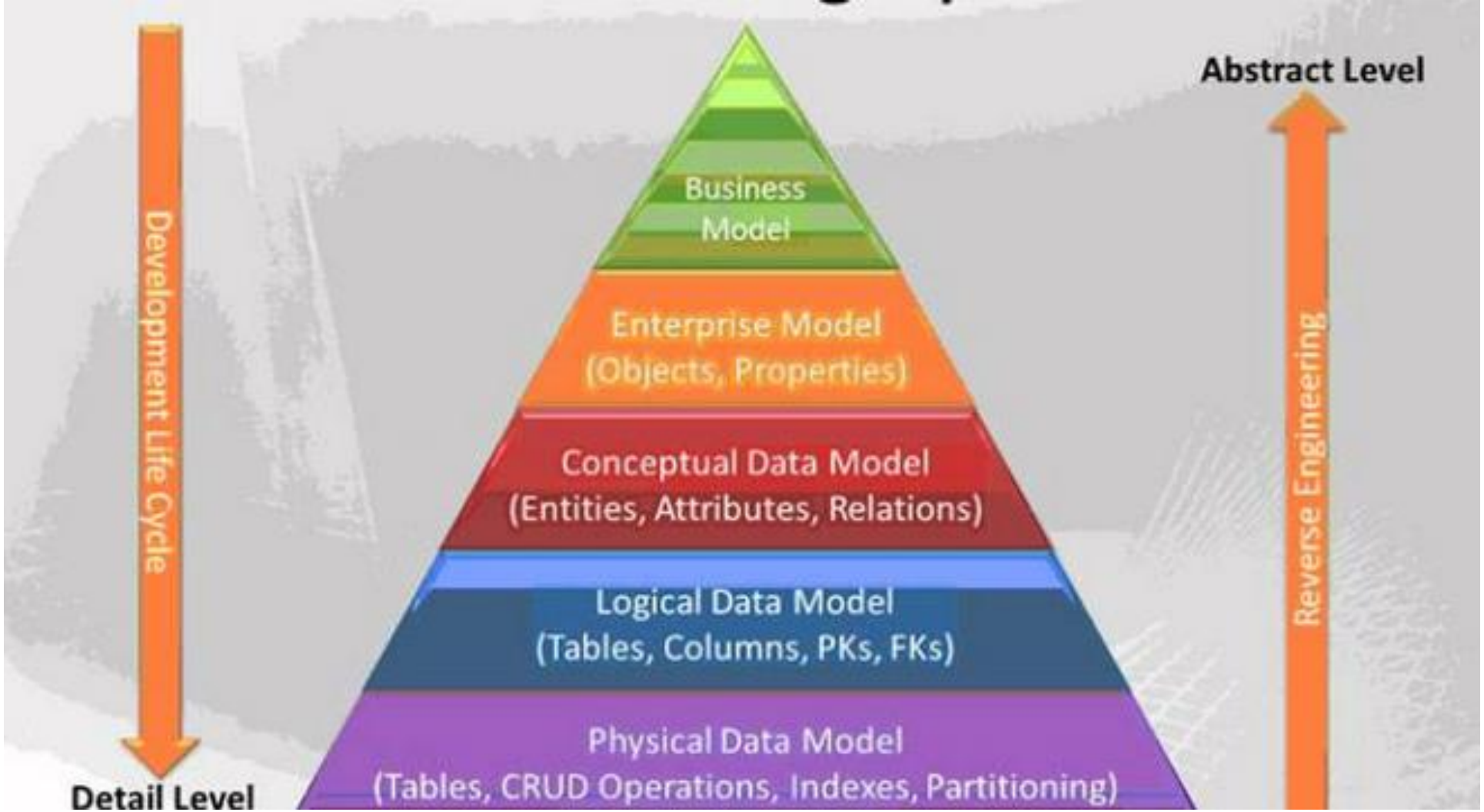
Supply chain DM



Note join tables to handle many-to-many relationships

Issues with datawarehouses

- Describing the data (metadata)
- **Organizing different «views» of the data**
- **Creating high-level data schemas for better analysis**
- Operations with DW (how do we get information)



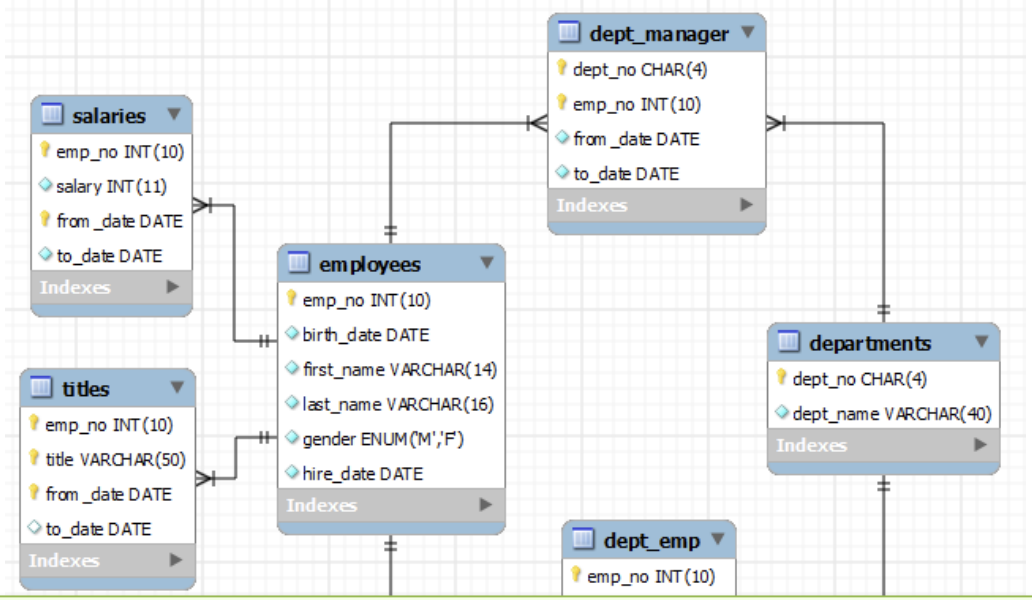
Data models: describing your data at different levels of abstraction

Process the warehouse: models and operations

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

Data schemas

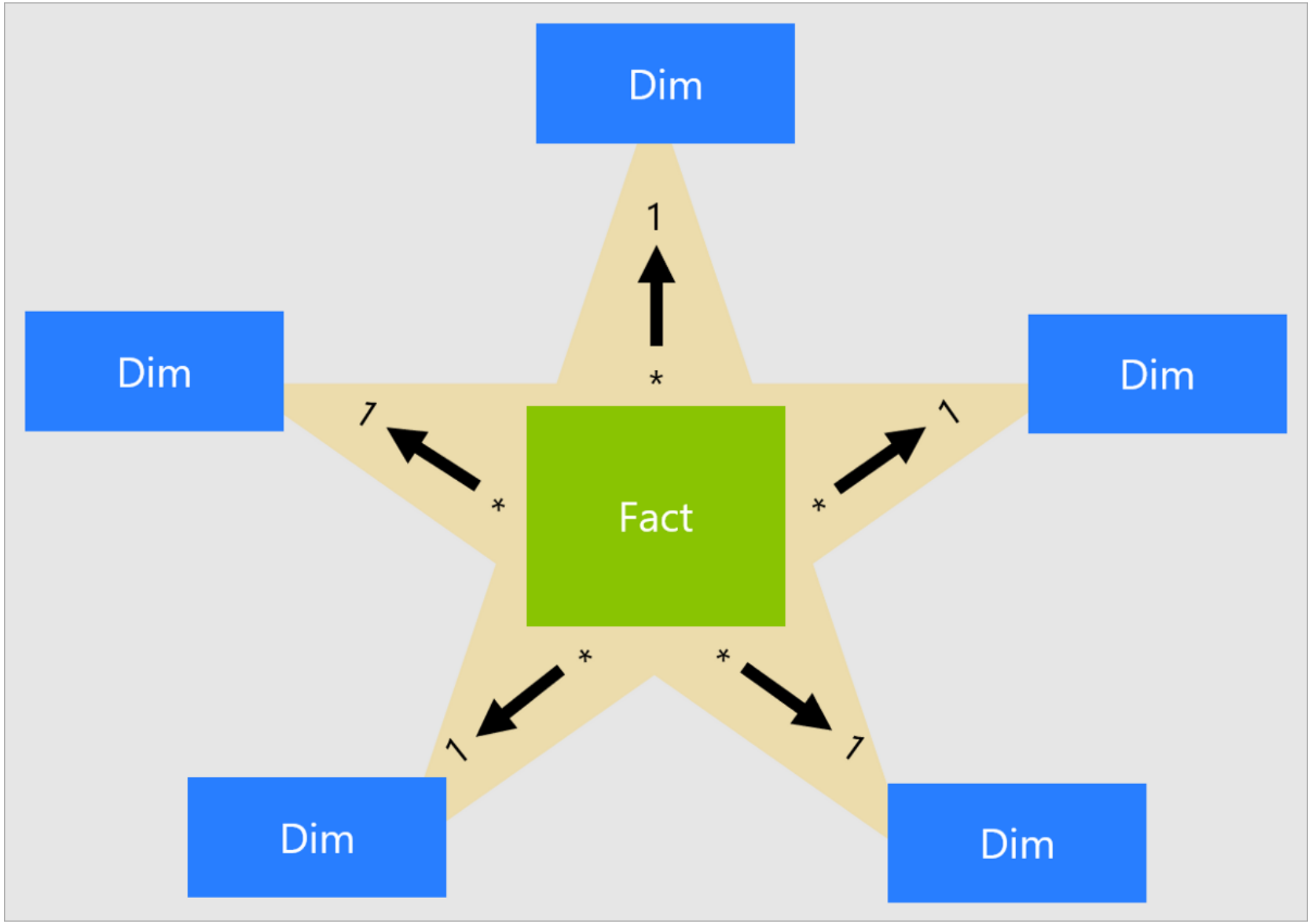
- You already are familiar with schemas: in simple words, they are set of connected tables describing relationships among your data. They allow to answer “descriptive” questions about your data.

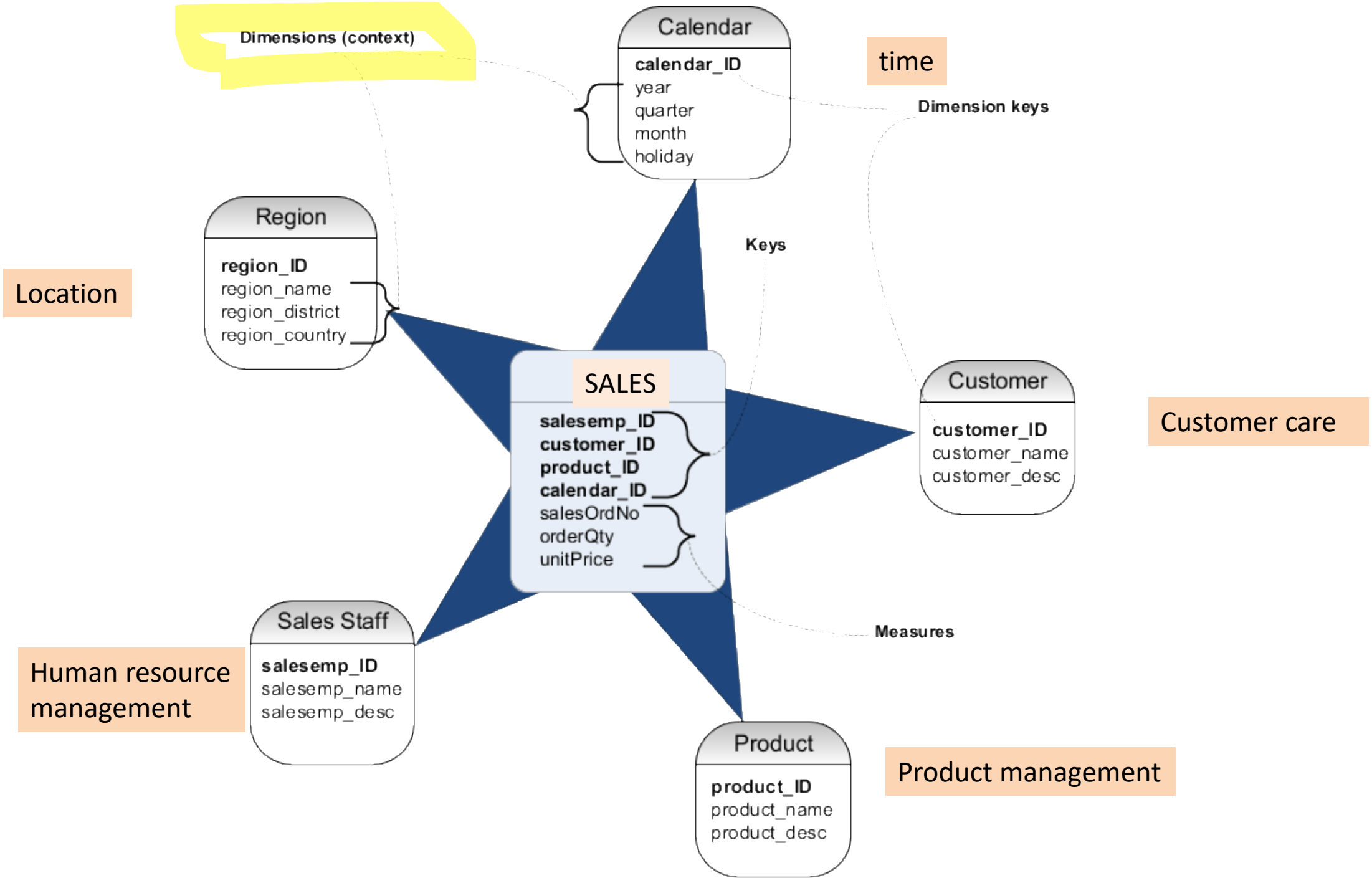


Schemas in data wharehouses must be arranged in a way that <u>facilitates the aggregation/summarization</u> of (and deeper analytics of) data. We need **specific types of schemas**.

DW 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 main idea is that the **star center** represents the entity type (named **FACT**) which is the **main “focus”** (e.g., sales) and the points of the star are entity types called **dimension (= relevant ways of aggregating data)**.
- The attributes (fields) of the central table include these dimensions, plus other dependent attributes. Dimensions are «expanded» in tables associated with star ponts.


Dim
1
*
Dim
Dim
1
*
Fact
*
1
*
*
1
1
Dim
Dim


Dimensions (context)
Calendar
calendar_ID
year
quarter
month
holiday
time
Dimension keys
Region
region_ID
region_name
region_district
region_country
Location
Keys
SALES
salesemp_ID
customer_ID
product_ID
calendar_ID
salesOrdNo
orderQty
unitPrice
Customer
customer_ID
customer_name
customer_desc
Customer care
Measures
Sales Staff
salesemp_ID
salesemp_name
salesemp_desc
Human resource management
Product
product_ID
product_name
product_desc
Product management

DW Star schema: dimensions

- **Dimensions** are similar to the entity type concept in standard schema, but their use is mostly intended for.
  - filtering,
  - aggregation and
  - labelling of our data
- Common dimensions are *people, products, places and time*.
- Each dimension is described by its own dimension table in a star schema
- Dimension tables have corresponding dimension *attributes*
- *NOTE:* ***identifying dimensions is important*** *– it affects the type of operations you can later perform to* ***aggregate*** *your data in an informative way.*

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
• • •			

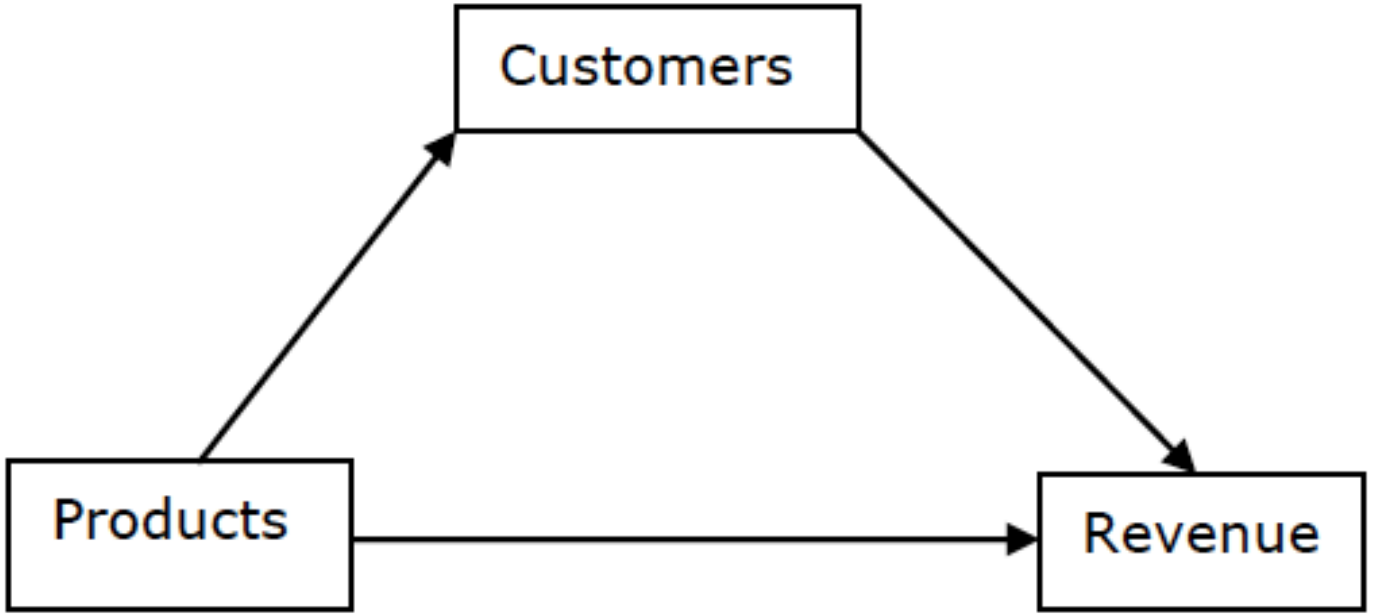
Sales

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

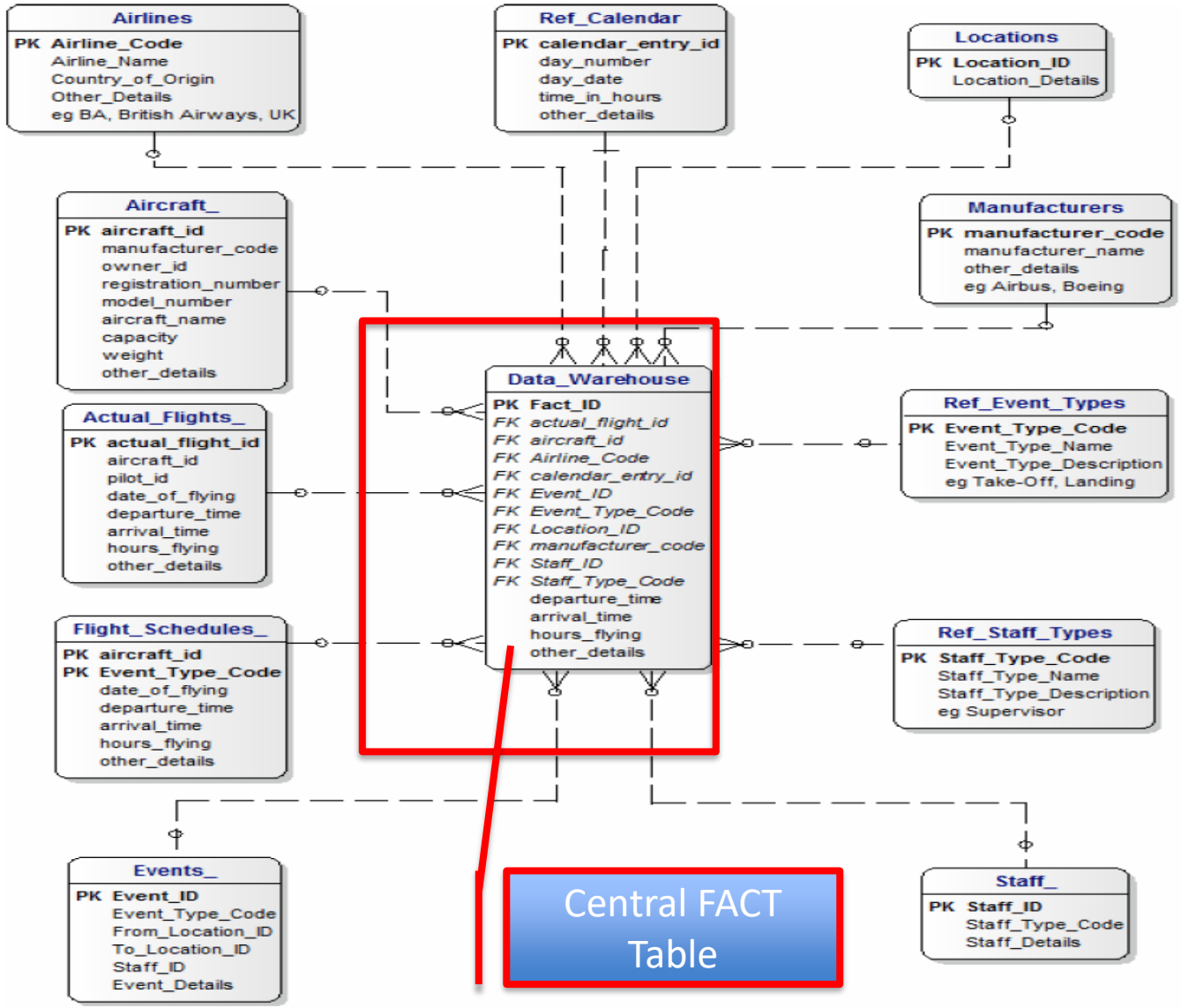
Store_Code	Store_Name	City	Telephone	Manager
S1	Jan's	San Antonio	683-192-1400	Burgess
S2	Bill's	Portland	943-681-2135	Thomas
S3	Ed's	Boulder	417-196-8037	Perry
• • •				

RECORDS: each line shows the VALUES of each attribute

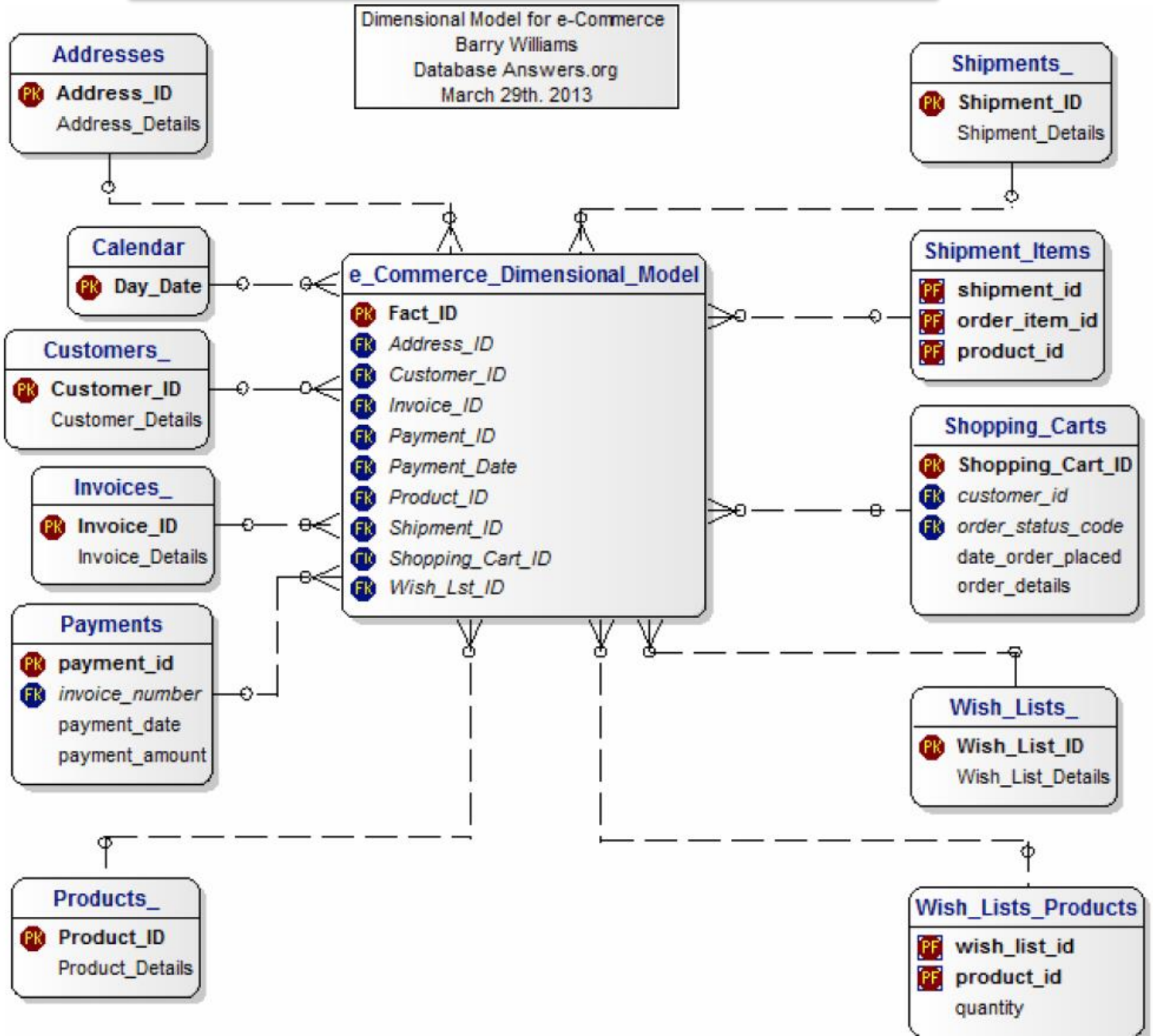


All business cases reflect at least 3 main dimensions (but many other are possible)

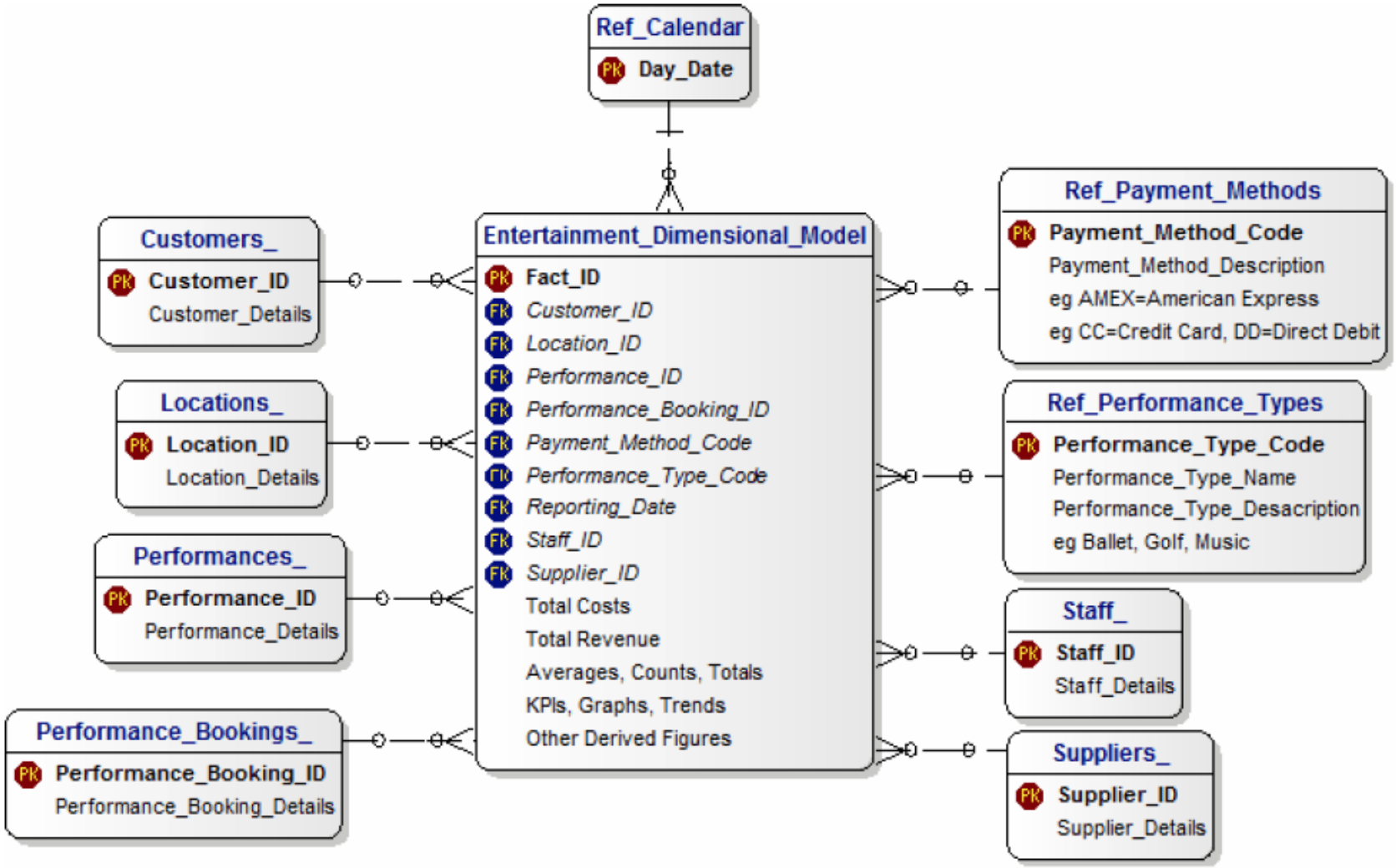
Example 2: A real-case Airline star-schema

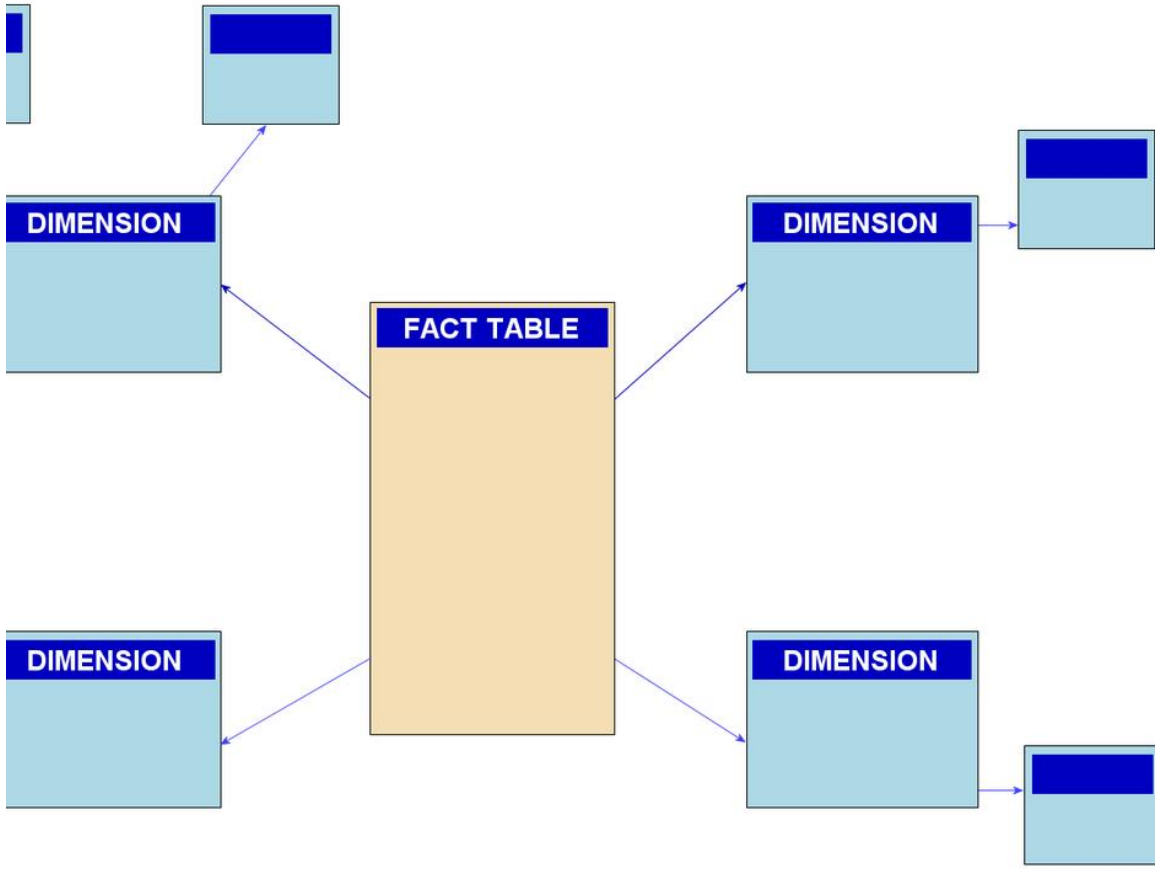


Example 3: e-commerce star schema



Example 4: Entertainment star schema





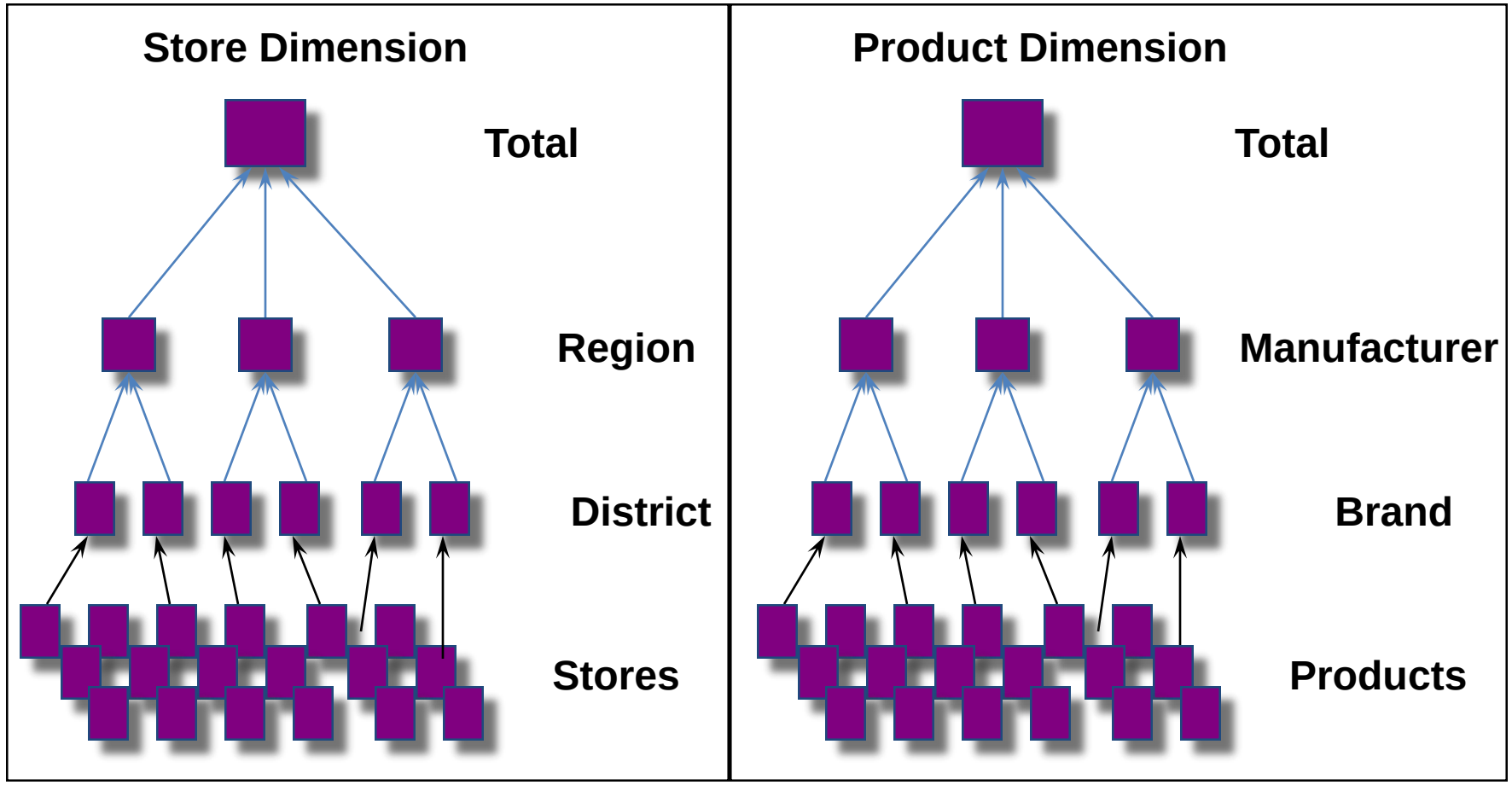
DW Schema for hierarchical dimensions: Snowflake

- It is an extension of star schema, in which normalized **dimension tables** have dimensions themselves
- More suitable to represent **hierarchical** dimensions

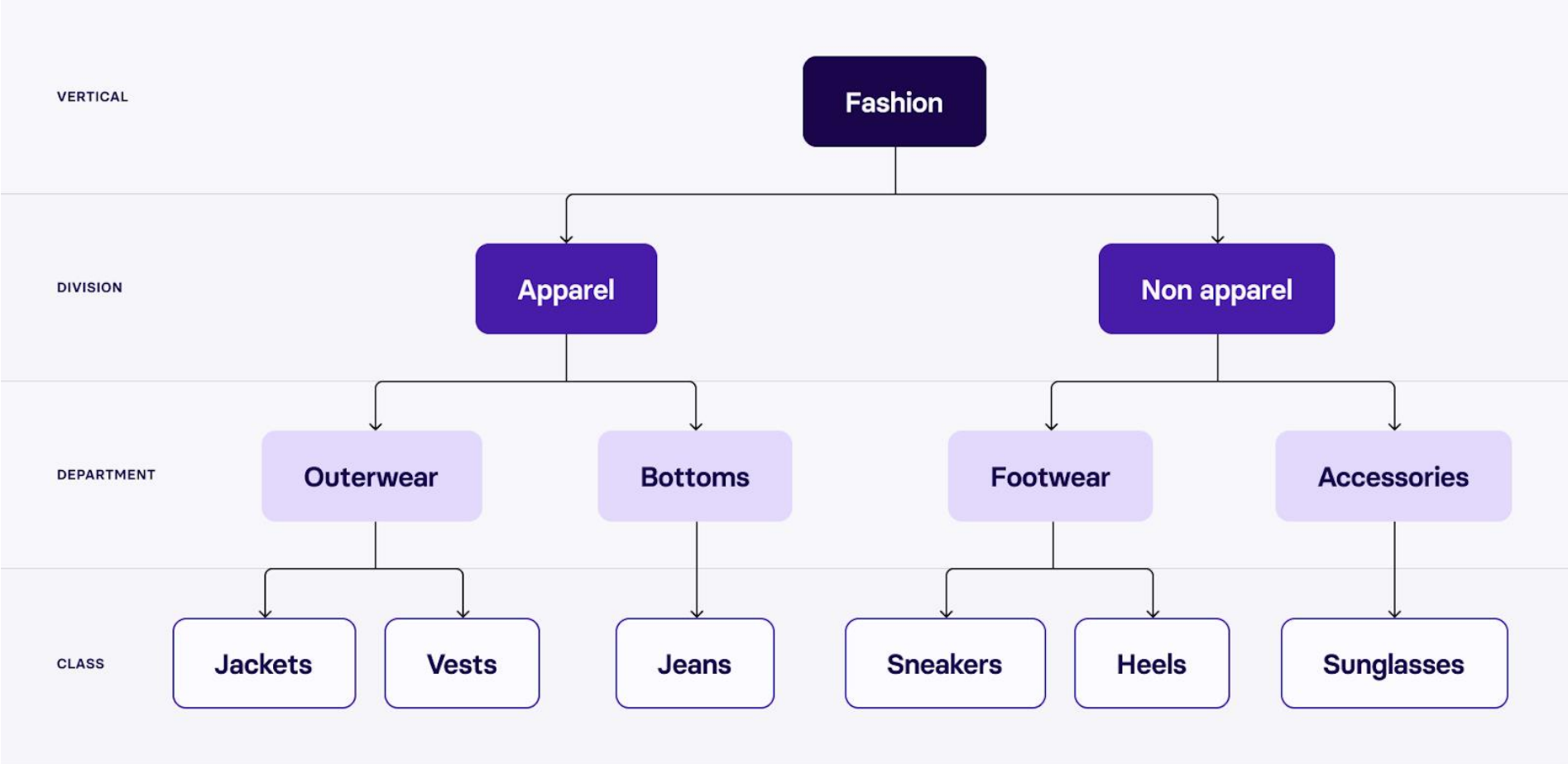
More on Dimensional Modeling: hierachies

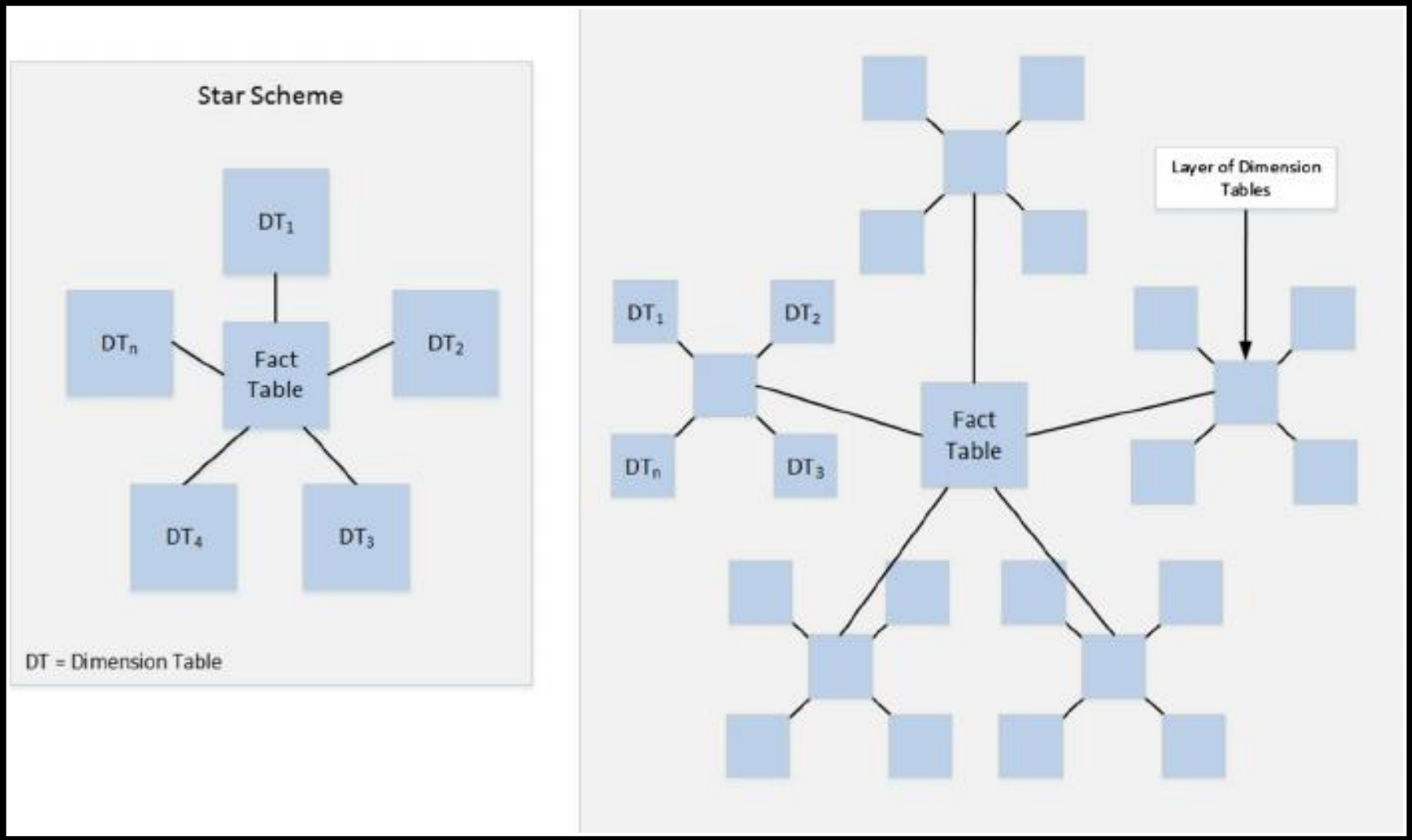
- Dimensions can be further organized into **hierarchies (see Watson labs)**
  - E.g., Time dimension: days ==> weeks ==>quarters
  - E.g., Product dimension: product ==>product line ==>brand
  - E.g. Location dimension: Continent==>nation ==>city==>store

Dimension Hierarchies Examples



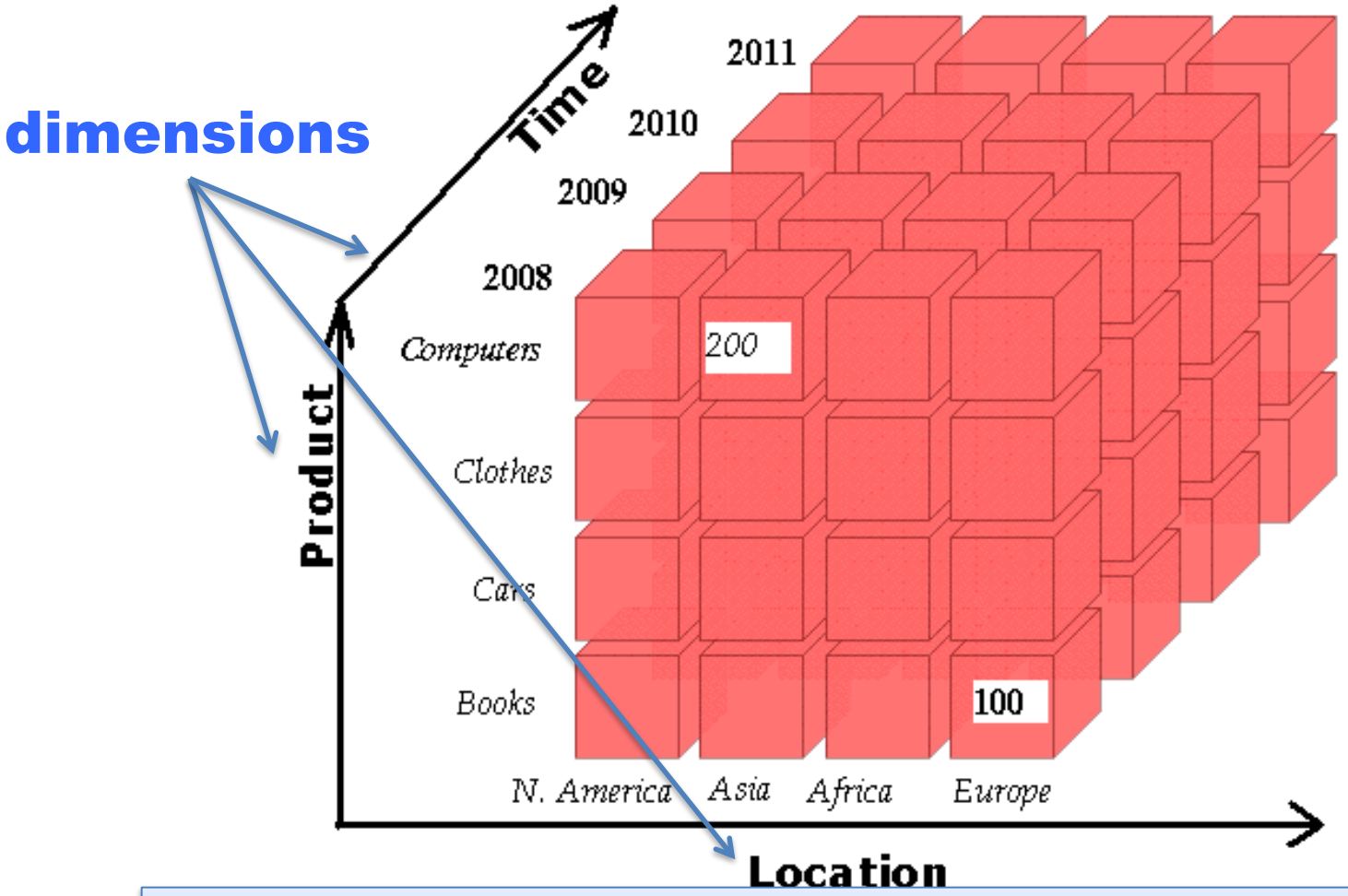
Fashion products taxonomy





Snowflake general scheme: star + hierarchy

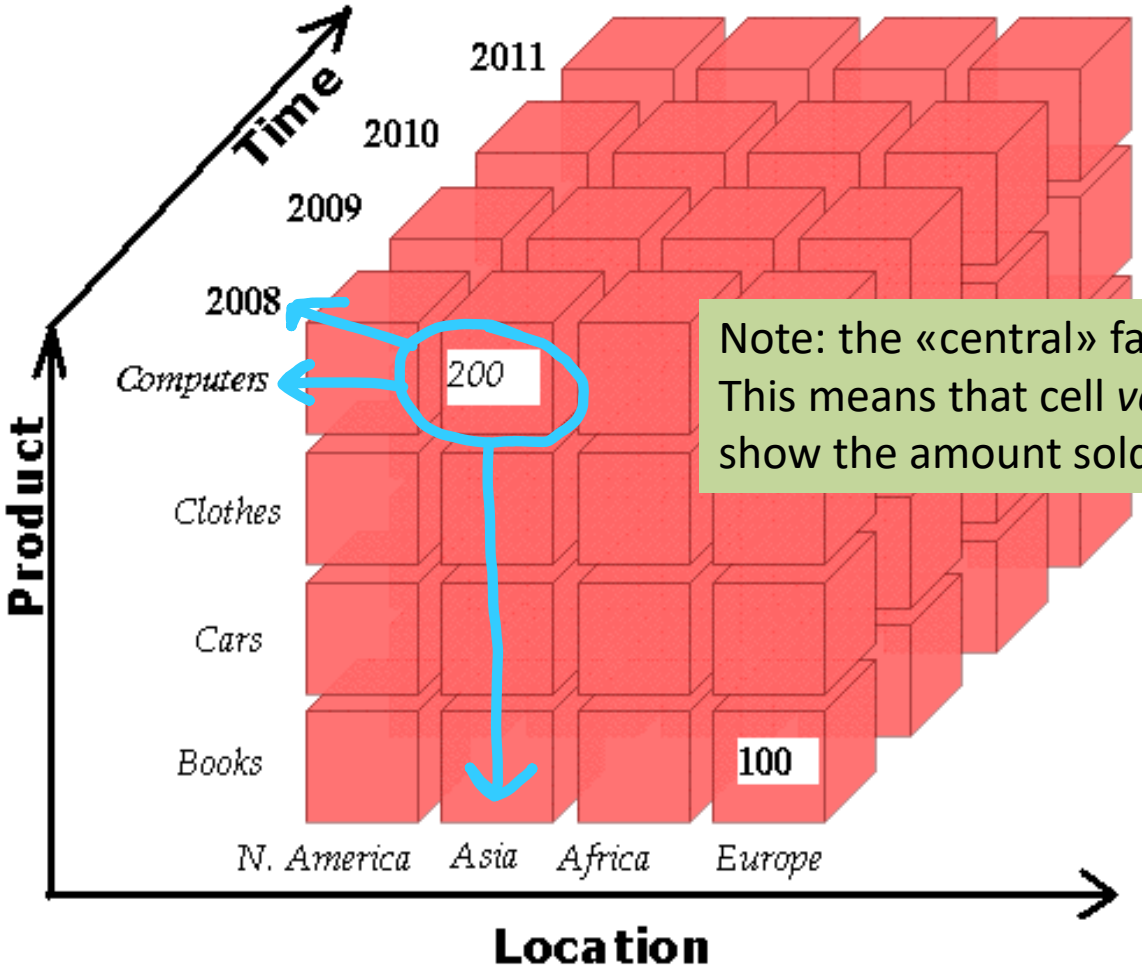
A third very powerful DW schema: Cubes (shows dimensions and measures)



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

"Reading" a cube: the «central» fact is a value in a cell

Dimensions are represented by the axes of the cartesian space

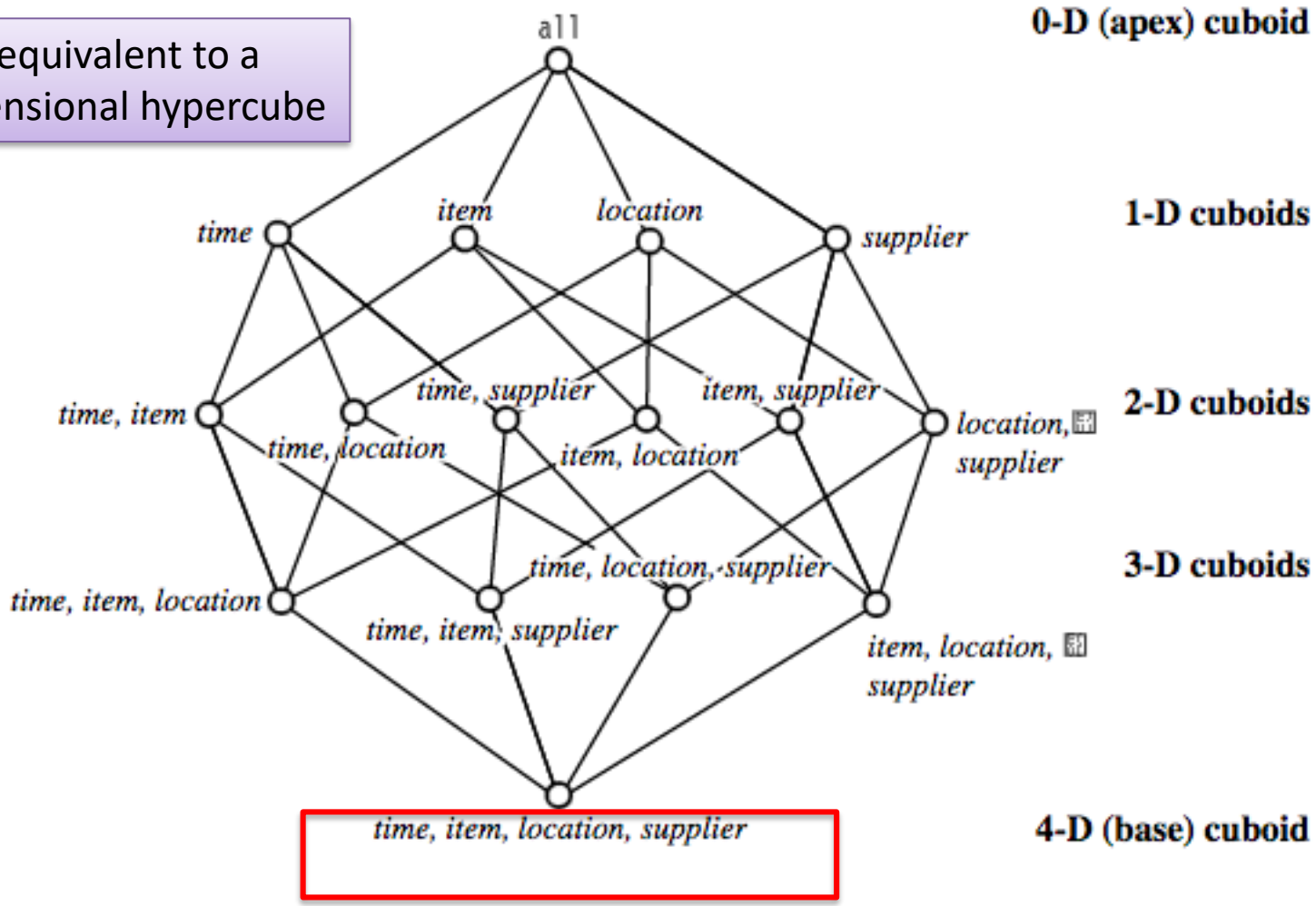


Note: the «central» fact here is **sales**.
This means that cell *values* in the cube show the amount sold (for a given product or total).

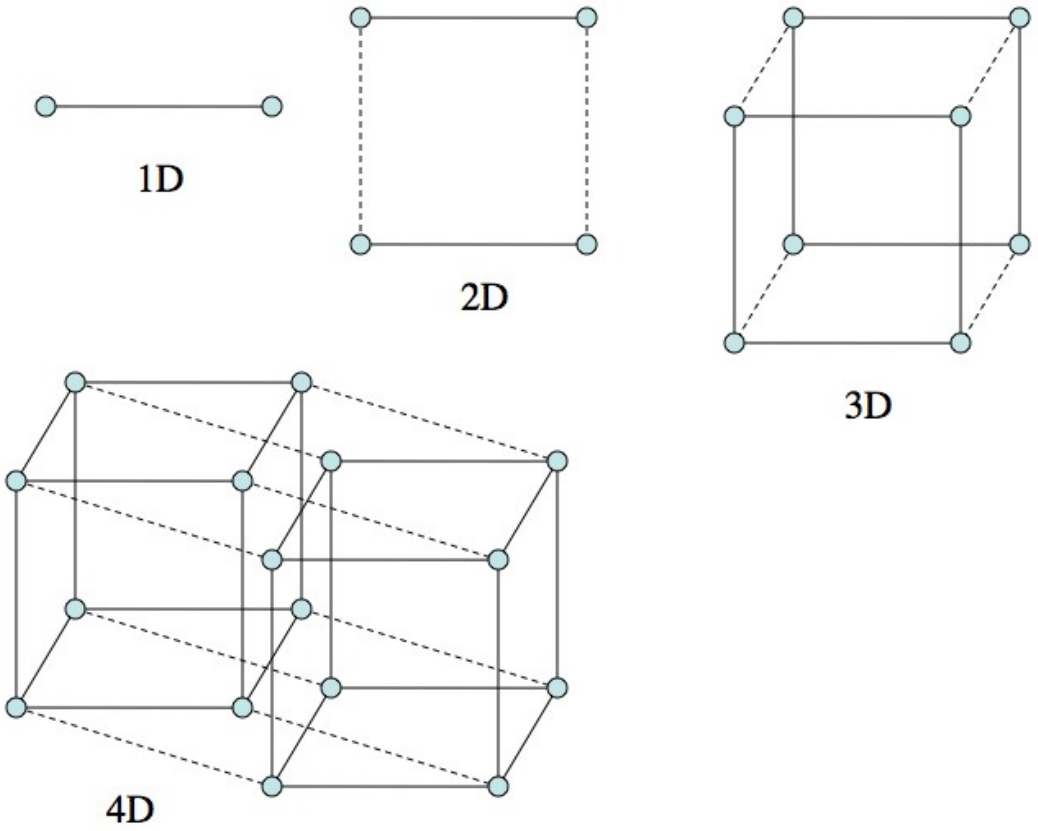
200 computers have been sold in Asia in 2008

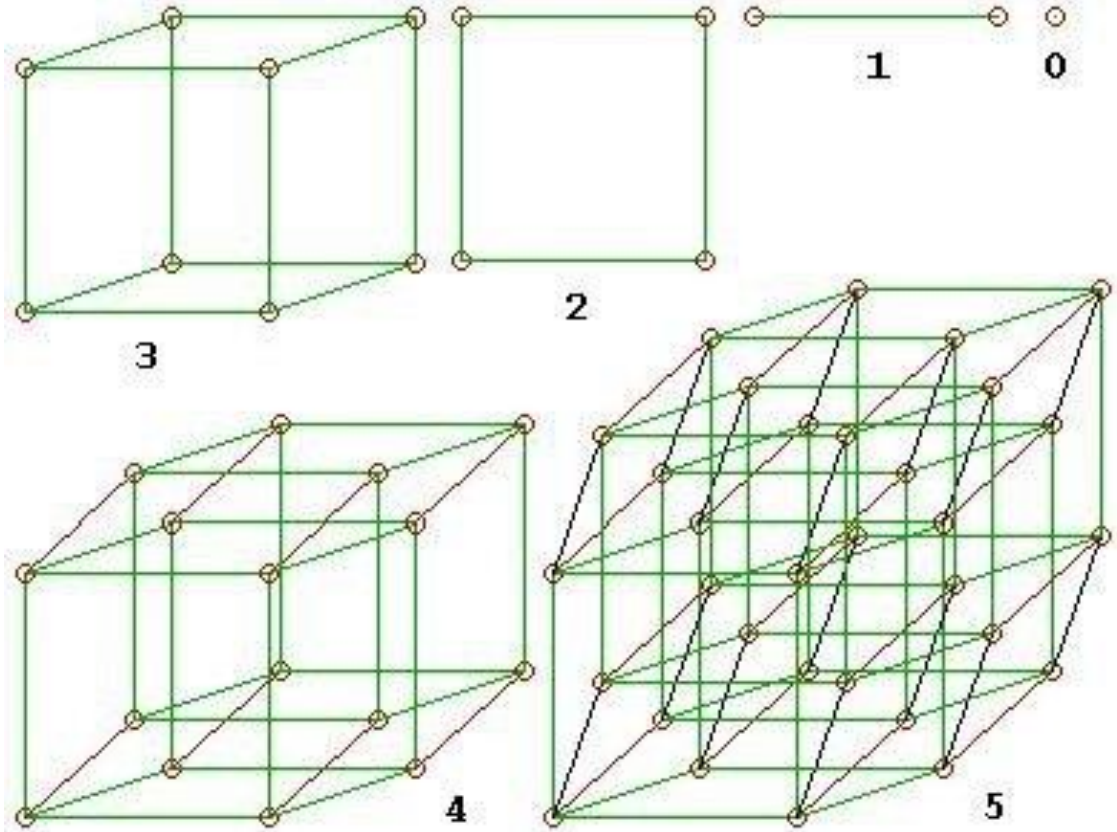
>3 dimensions: cuboids

This is equivalent to a 4-dimensional hypercube



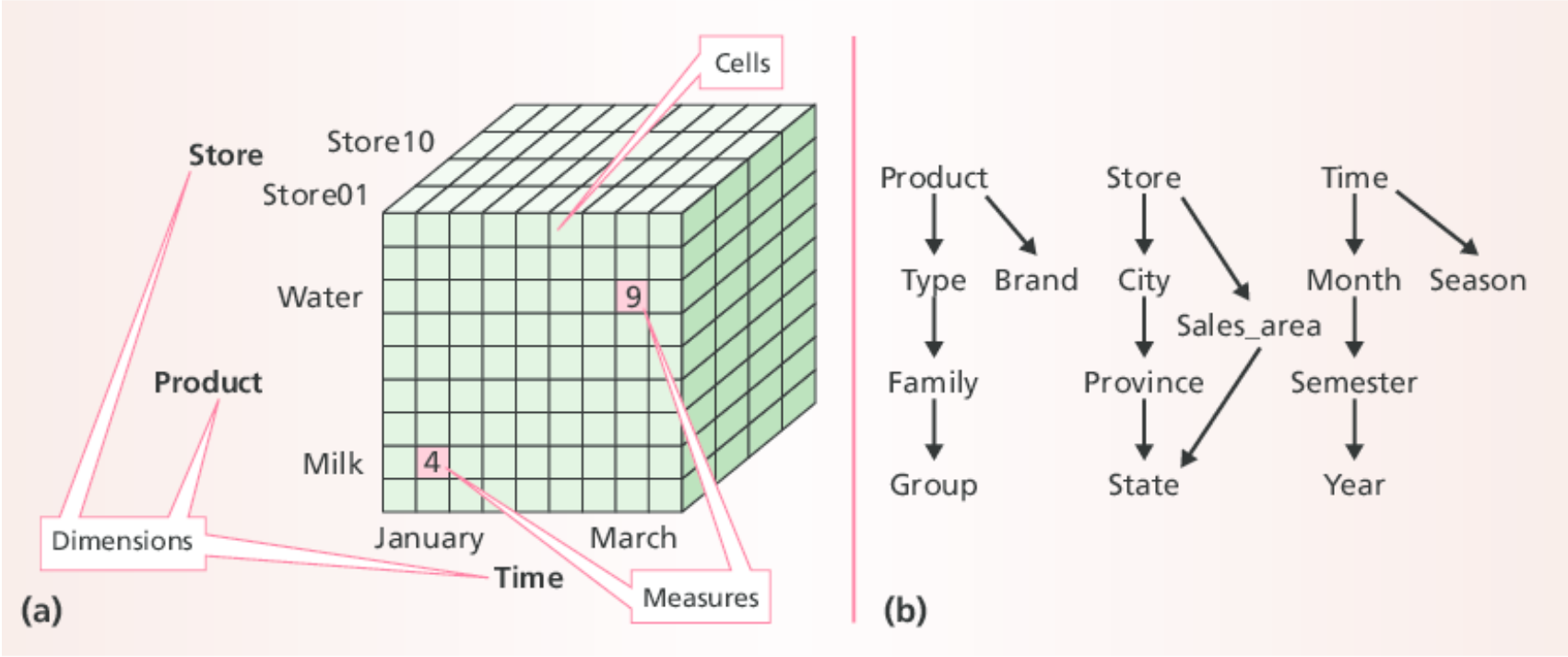
More on cuboids





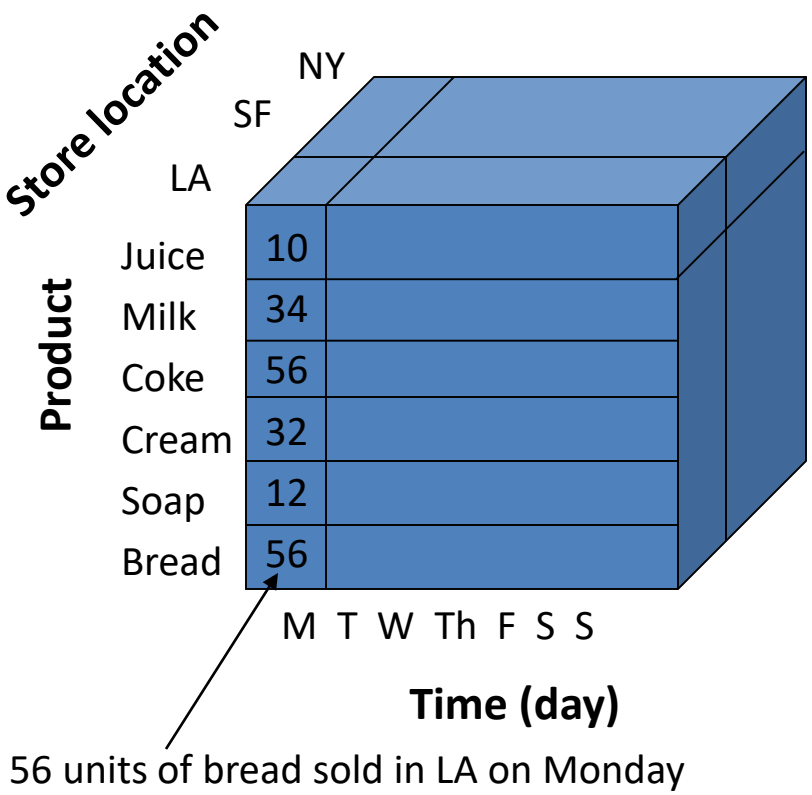
Hierarchies can be «added» to aggregate disaggregate data on cubes

(a) The cube itself is composed of cells that define fact attributes, while (b) the classification hierarchies display the dimensions that define the cube—product, store, and time.



Note: hierachies are defined in separate fact sheets.
E.g.: January→1° Semester→Year1

Cubes with hierarchies



Dimensions:

Time, Product, Store

Attributes:

Product (orders, price, …)

Store …

…

We can add hierarchies:

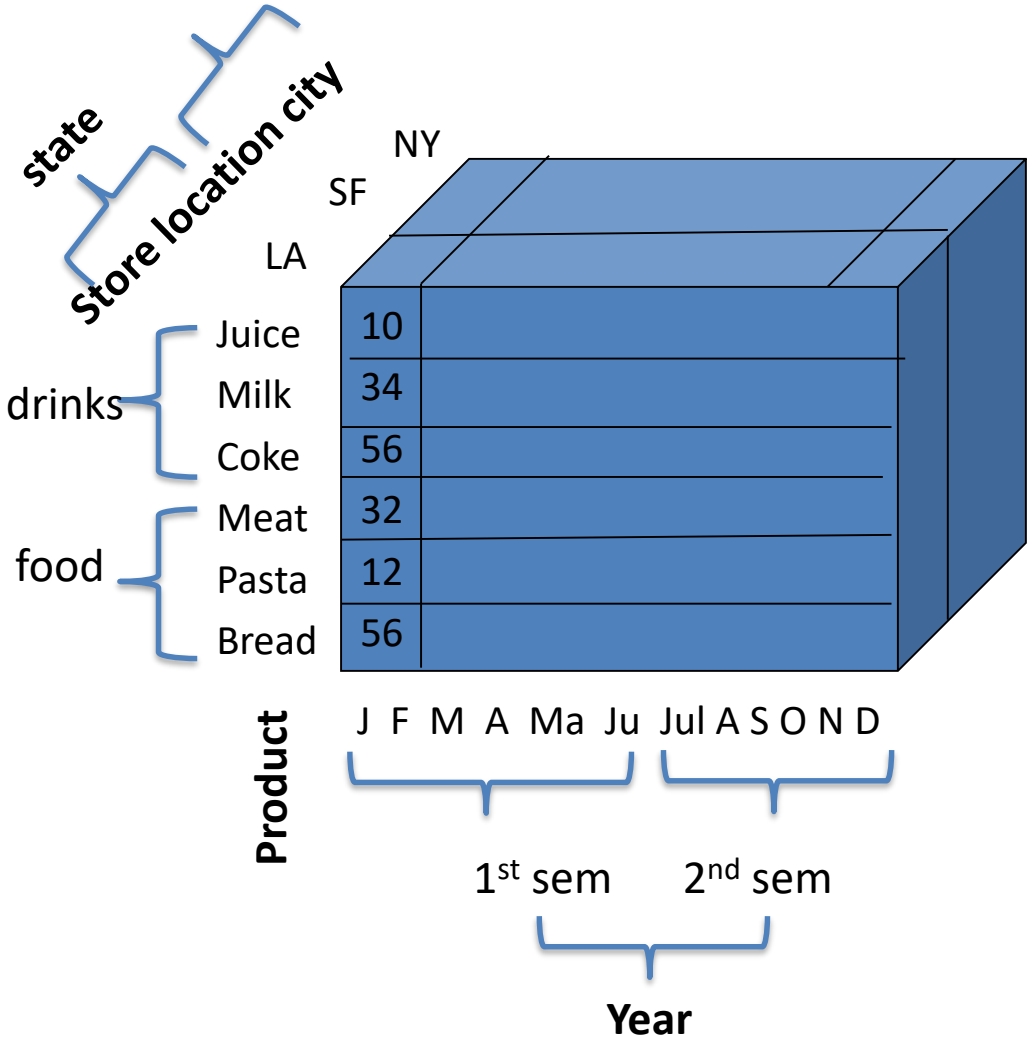
Product → Brand → …

Day → Week → Quarter

Store → Region → Country

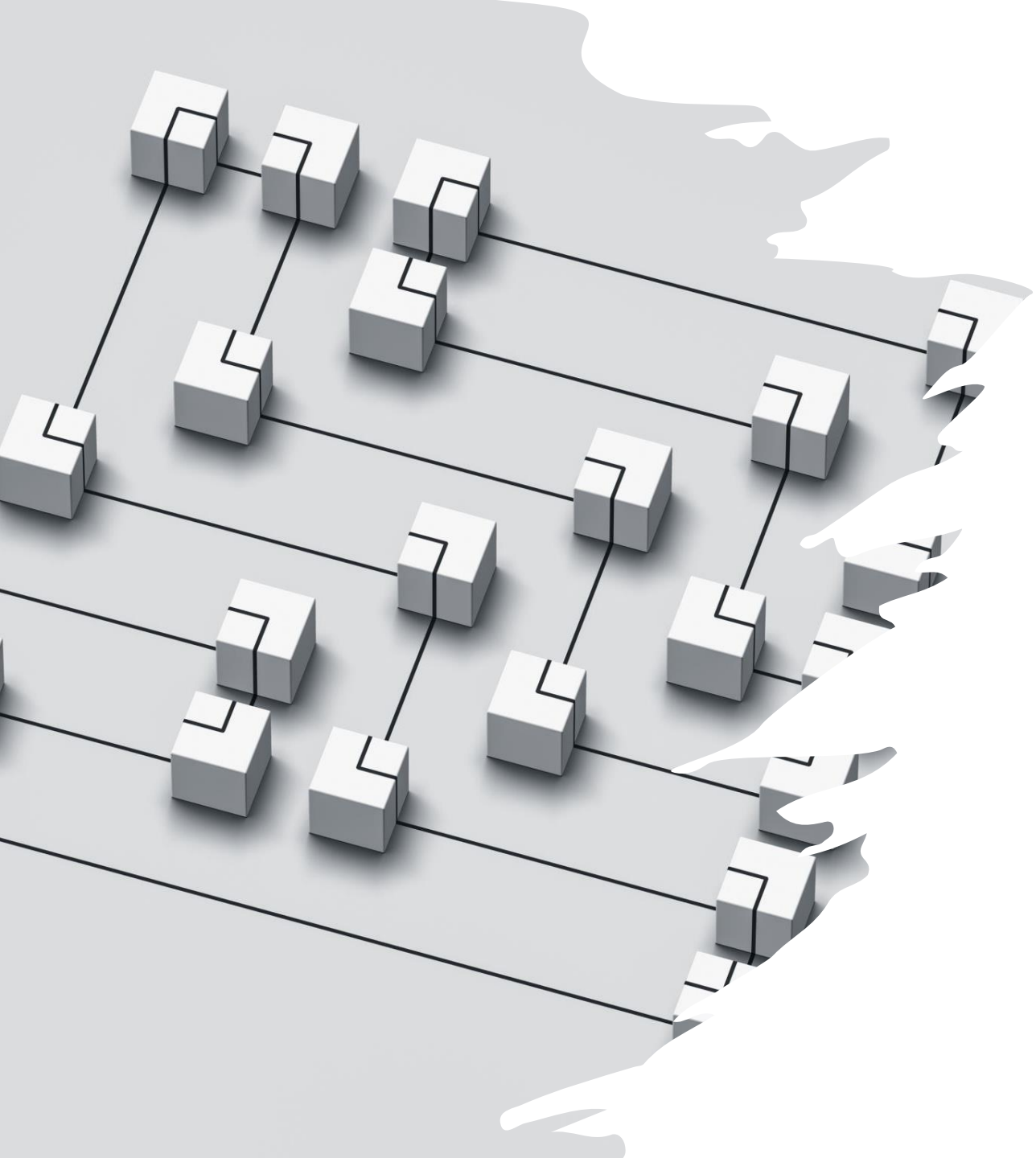
Organizing dimensions in categories

Note: hierarchies can ne associated to dimensions, but, in order to USE them for data summarization, we need to introduce OPERATIONS



Issues with datawarehouses

- Describing the data (metadata)
- Organizing different «views» of the data
- Creating high-level data schemas for better analysis
- **Operations with DW (how do we get information)**



Process the warehouse: models and operations

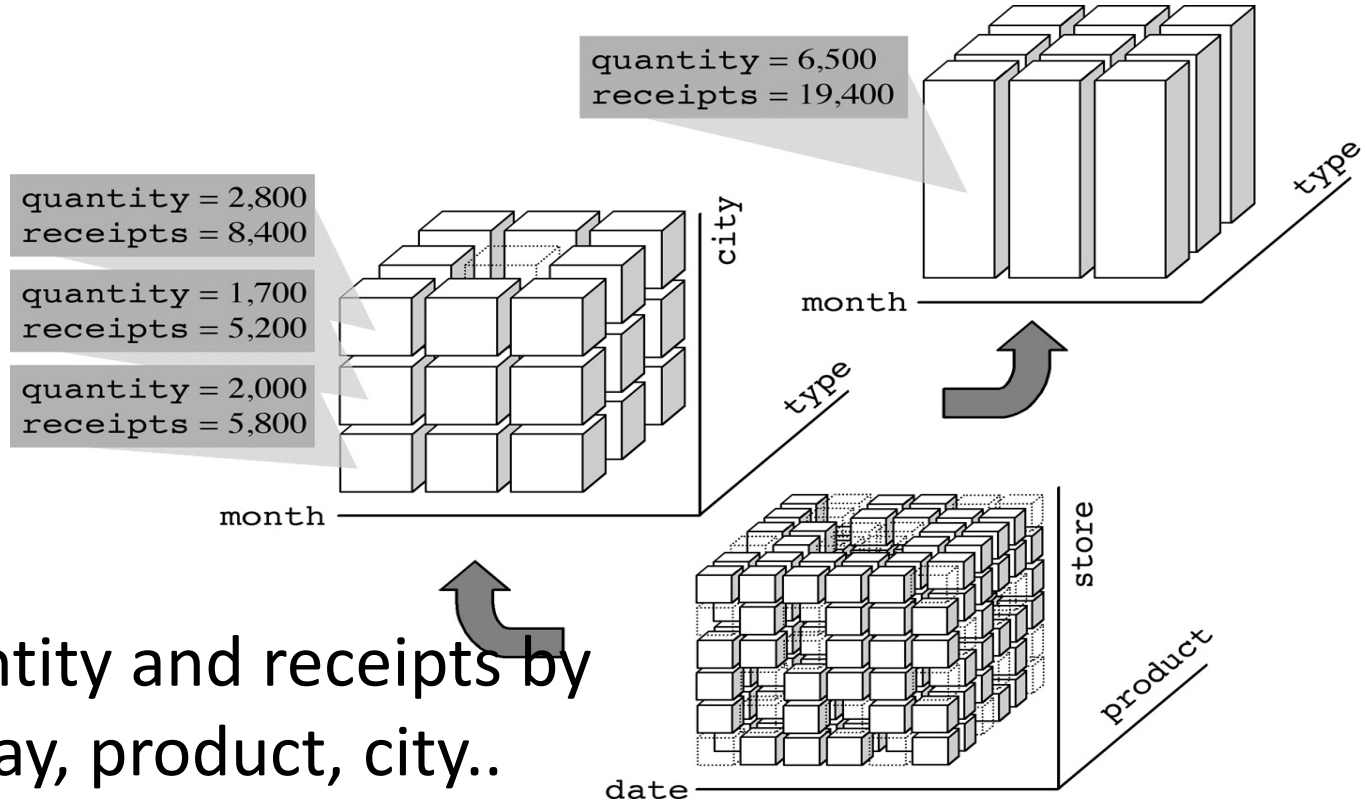
- **Data Schema:** In which data structures we organize the loaded data?
  - Star
  - Snowflake
  - Multidimensional data (cubes)
- **Operators: Which operations can we perform on the data?**
  - **slice & dice**
  - **roll-up, drill down**
  - **pivoting**
  - **other**

What kind of queries in a DW?

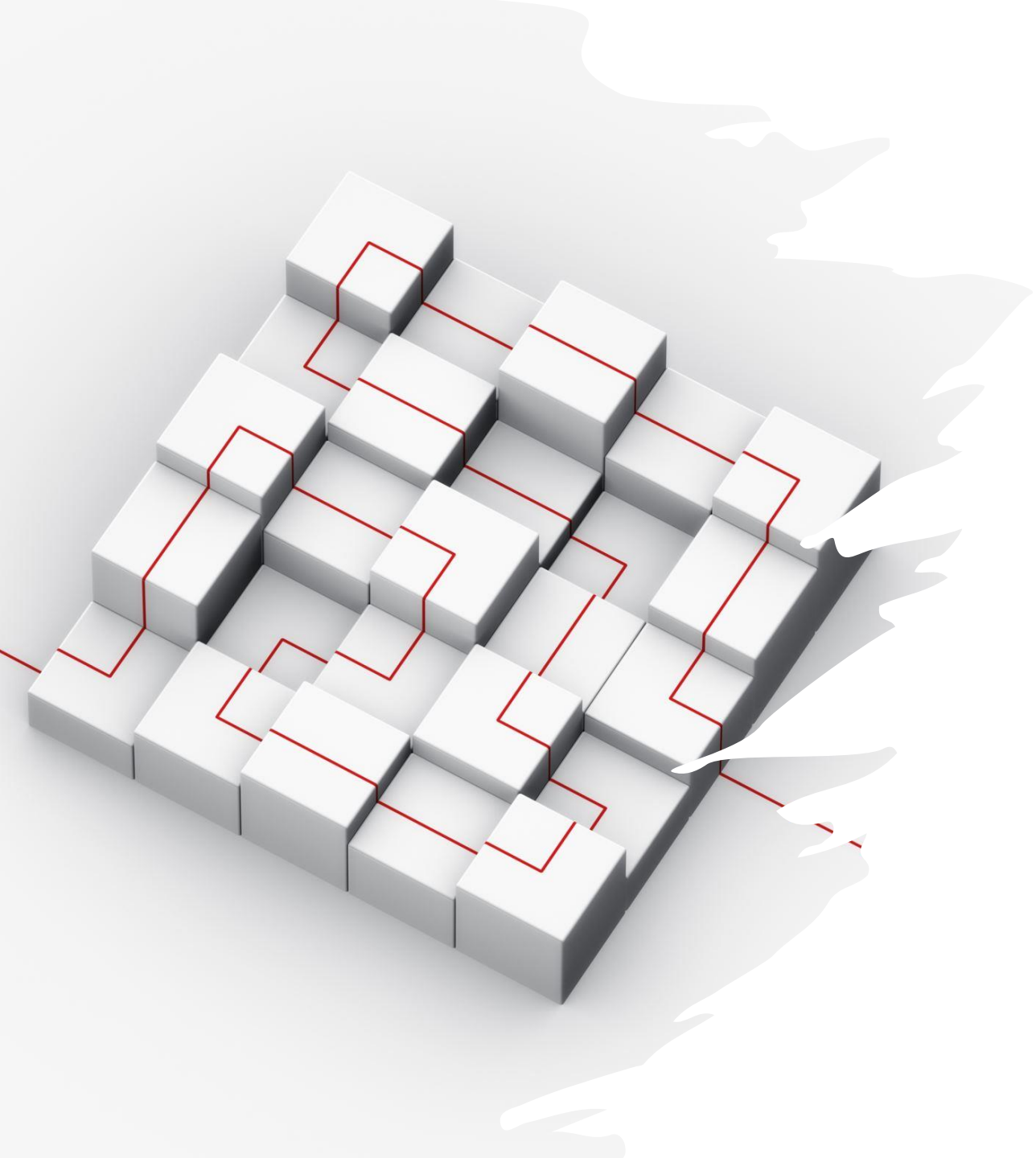
- Users of the data warehouse perform data analyses that require to "slice and dice" their data: summarize, obtain insight, and then disaggregate again
- DW users will sometimes need **highly aggregated data**, and other times they will need to **drill down to details**.
- Often temporal analyses are required. More sophisticated analyses include trend analyses and data mining, which **use existing data for predictive and prescriprive analytics**.
- The data warehouse acts as the **underlying engine used by business intelligence environments** that serve reports, dashboards and other interfaces to end users.
- For these reasons, **DW need mechanisms to aggregate/disaggregate data**

Operations on data cubes

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



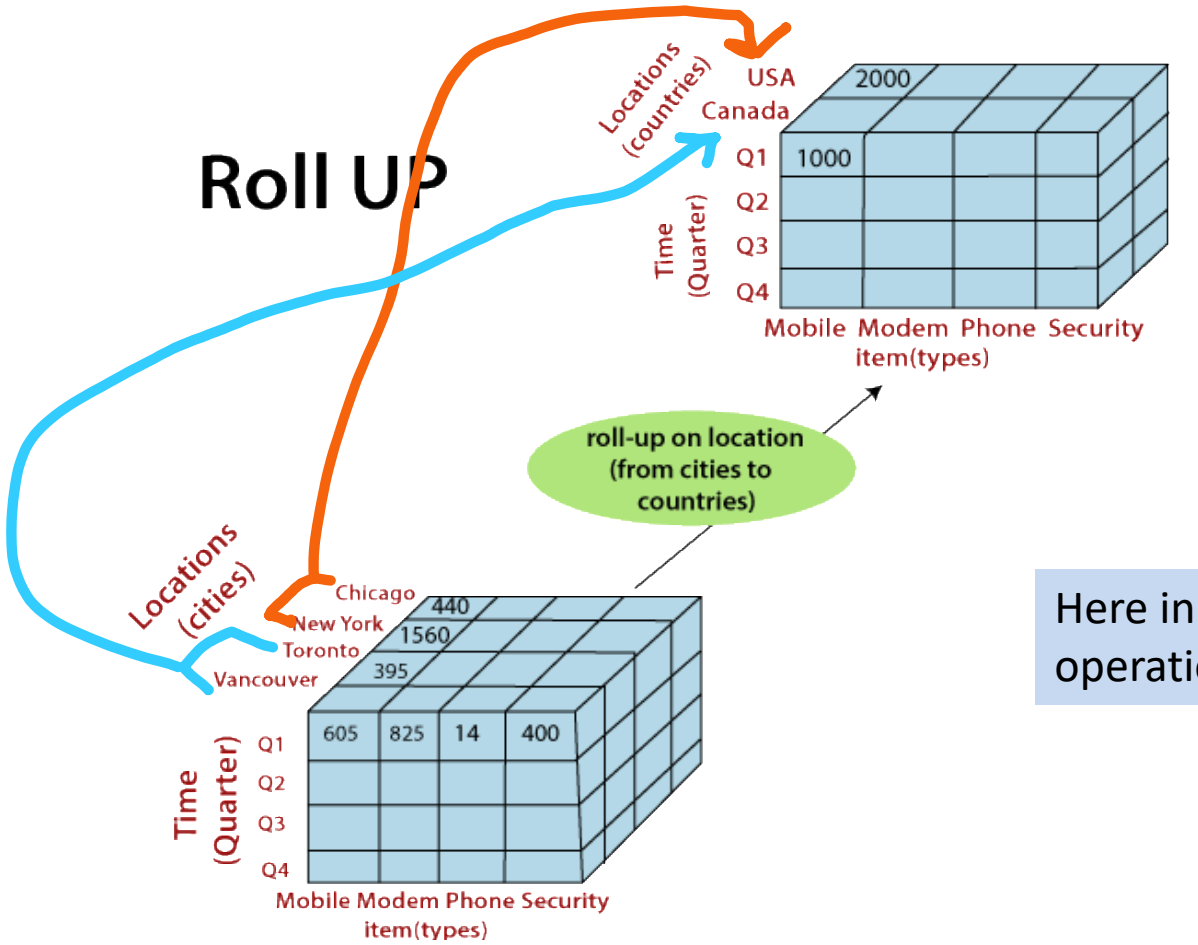
e.g., quantity and receipts by month, day, product, city..



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**.

Roll-up



Here in this example roll up/down operations climb a geographic hierarchy

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

day 2

	s1	s2	s3
p1	44	4	

day 1

	s1	s2	s3
p1	12		50
p2	11	8	

. . .

	s1	s2	s3
p1	56	4	50
p2	11	8	

sale(s1,p1,*)=44+12

	s1	s2	s3
sum	67	12	50

sale(s1,*,*)=44+12+11

	sum
p1	110
p2	19

sale(*,p1,*)=56+4+50

129

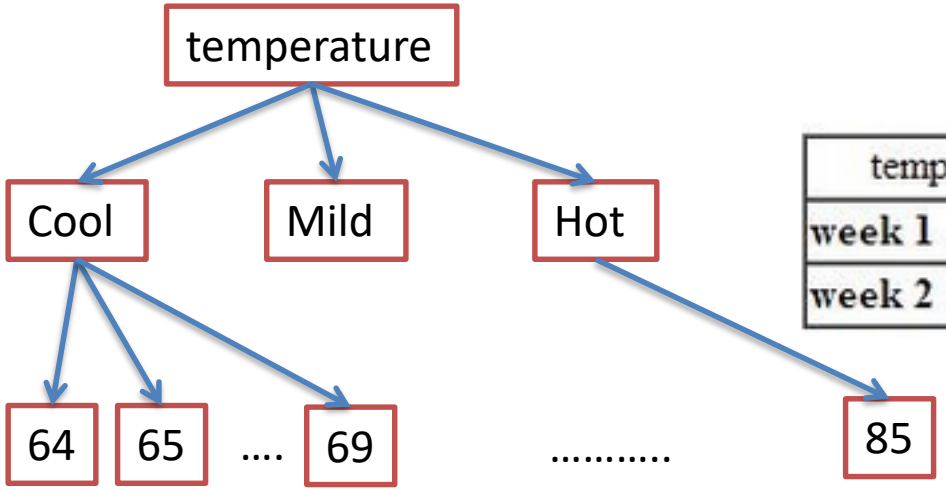
sale(*,*,*)=110+19

Roll-up using hierarchies

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 that Watson studio allows you to add hierarchies easily –in alternative, specific code must be written

Roll-up Using Hierarchies (2)

day 2

	s1	s2	s3
p1	44	4	

day 1

	s1	s2	s3
p1	12		50
p2	11	8	

	region A	region B
p1	56	54
p2	11	8

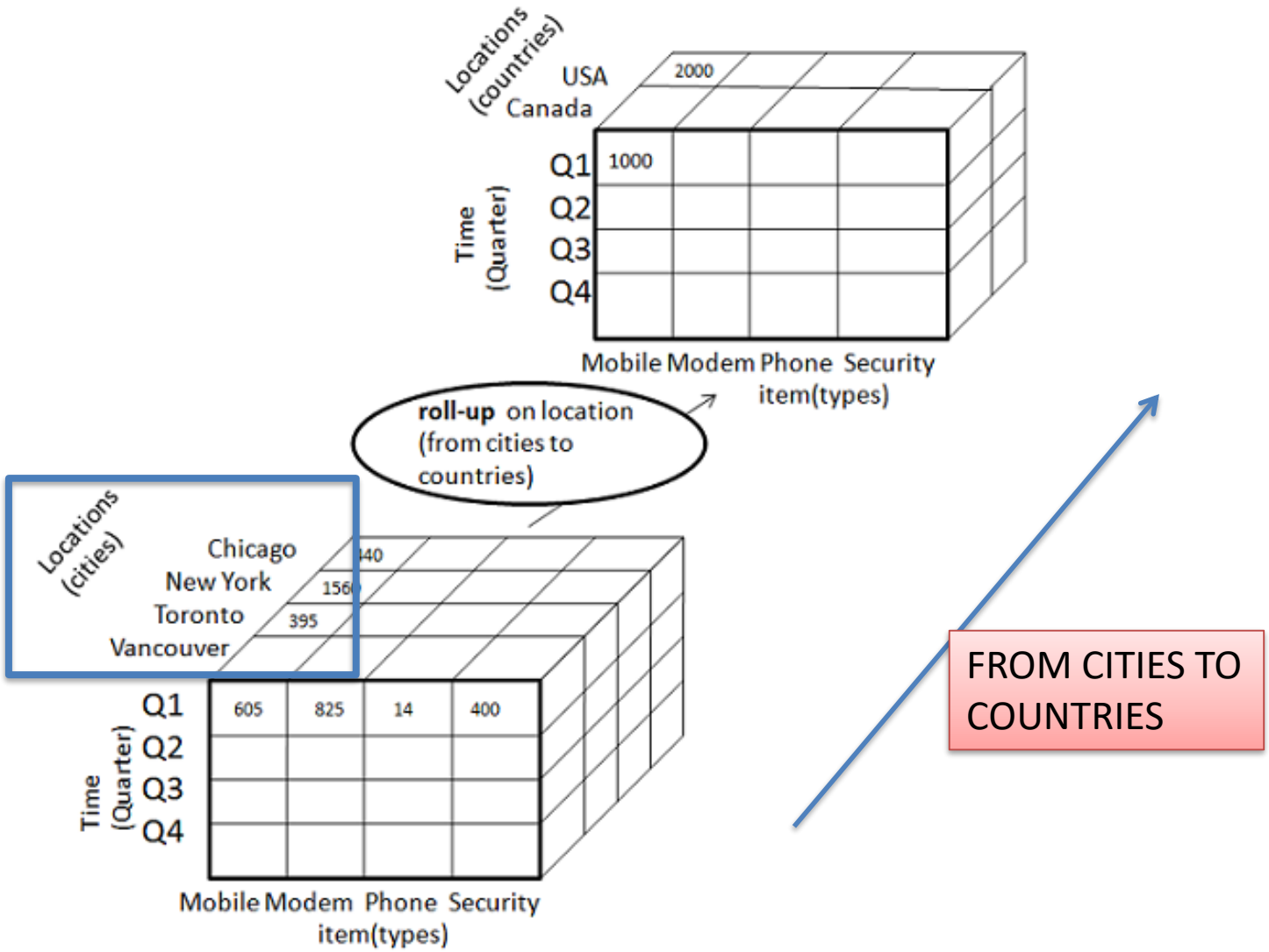
store

region

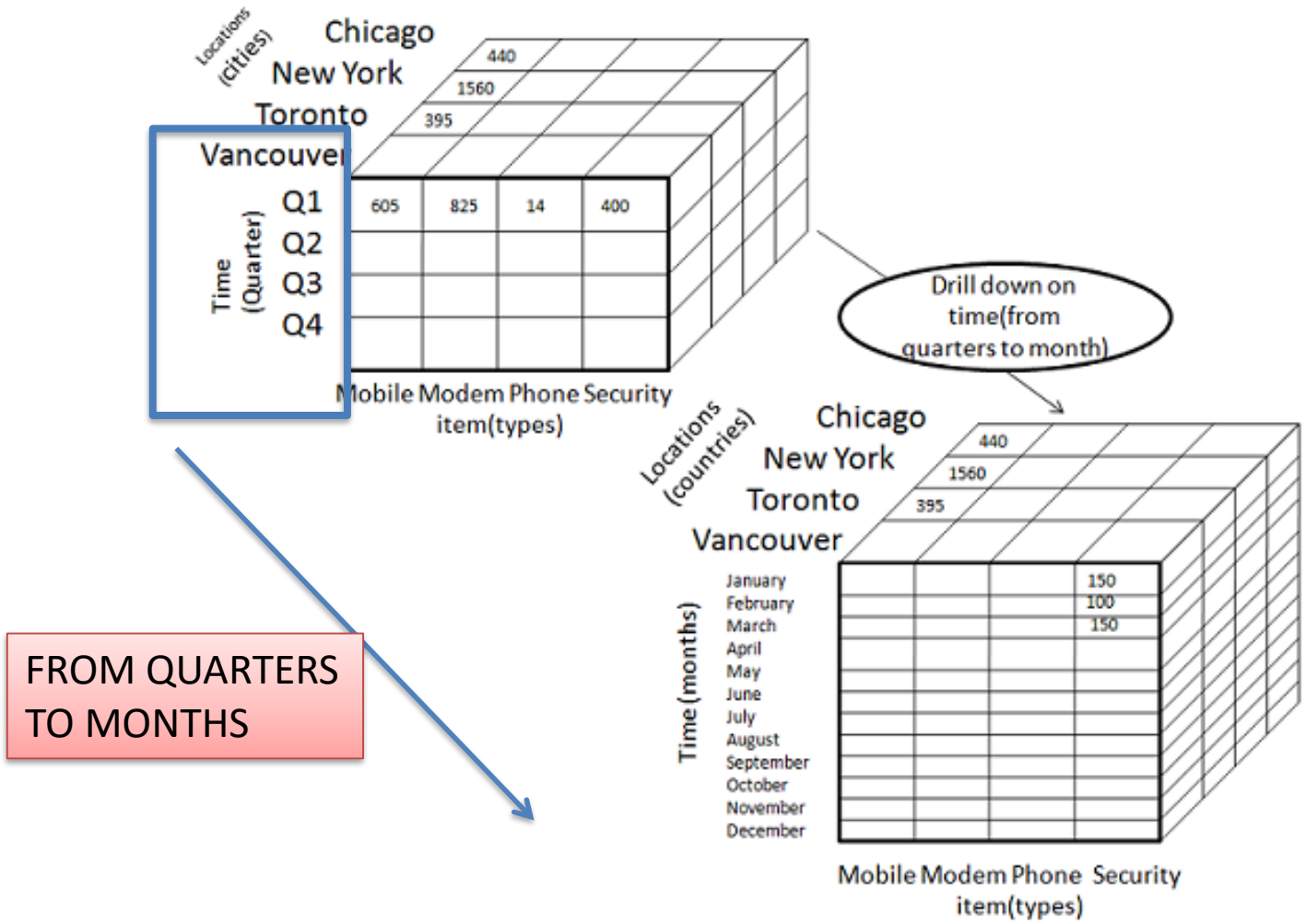
country

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

Another example of roll up with hierarchies

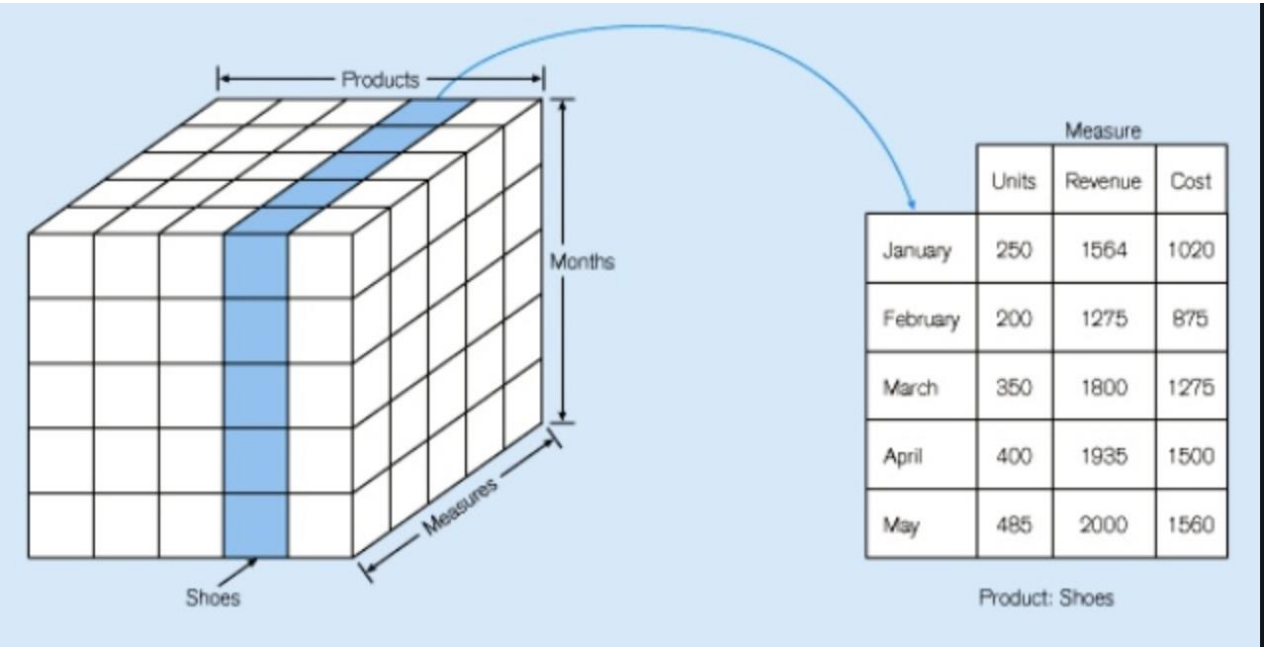


Roll-down example



Operations on cube: 2) Slicing

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



Another example of slicing

day 2

	s1	s2	s3
p1	44	4	

day 1

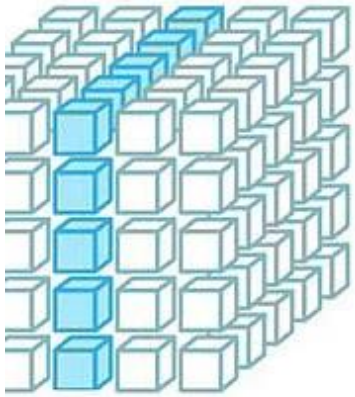
	s1	s2	s3
p1	12		50
p2	11	8	

TIME = day 1

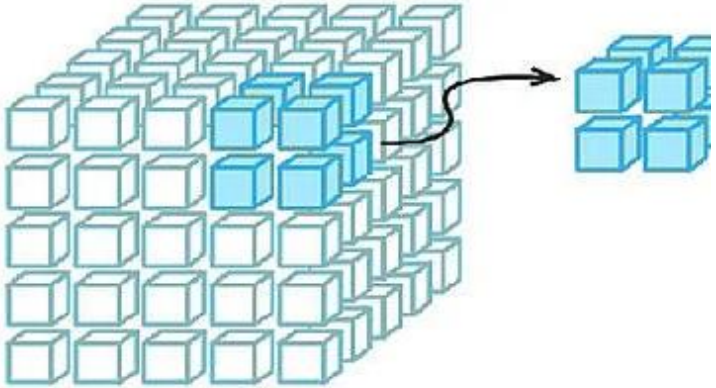
	s1	s2	s3
p1	12		50
p2	11	8	

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.



SLICING



DICING

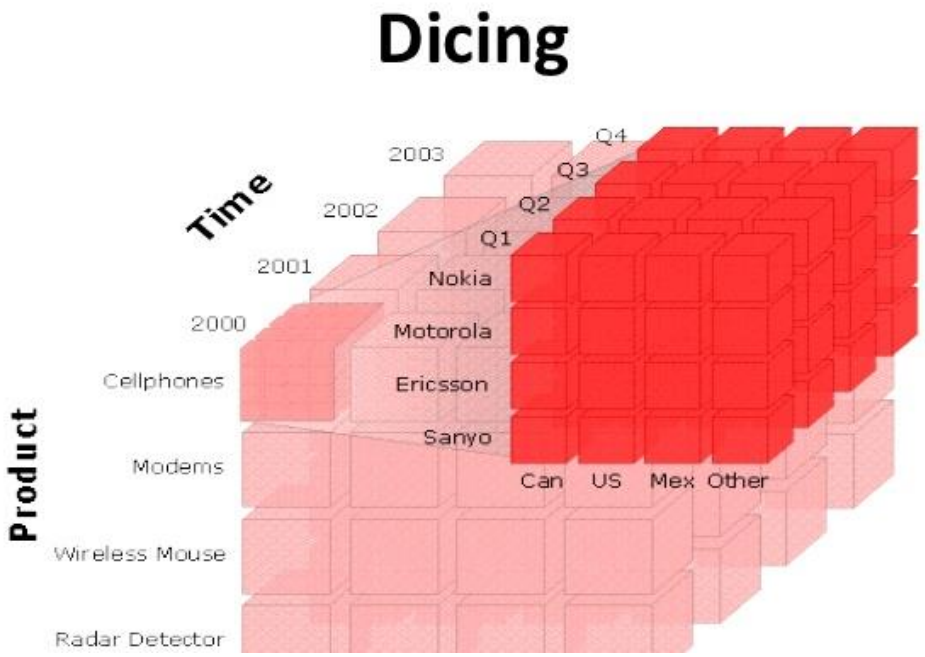
Example of Dicing

	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

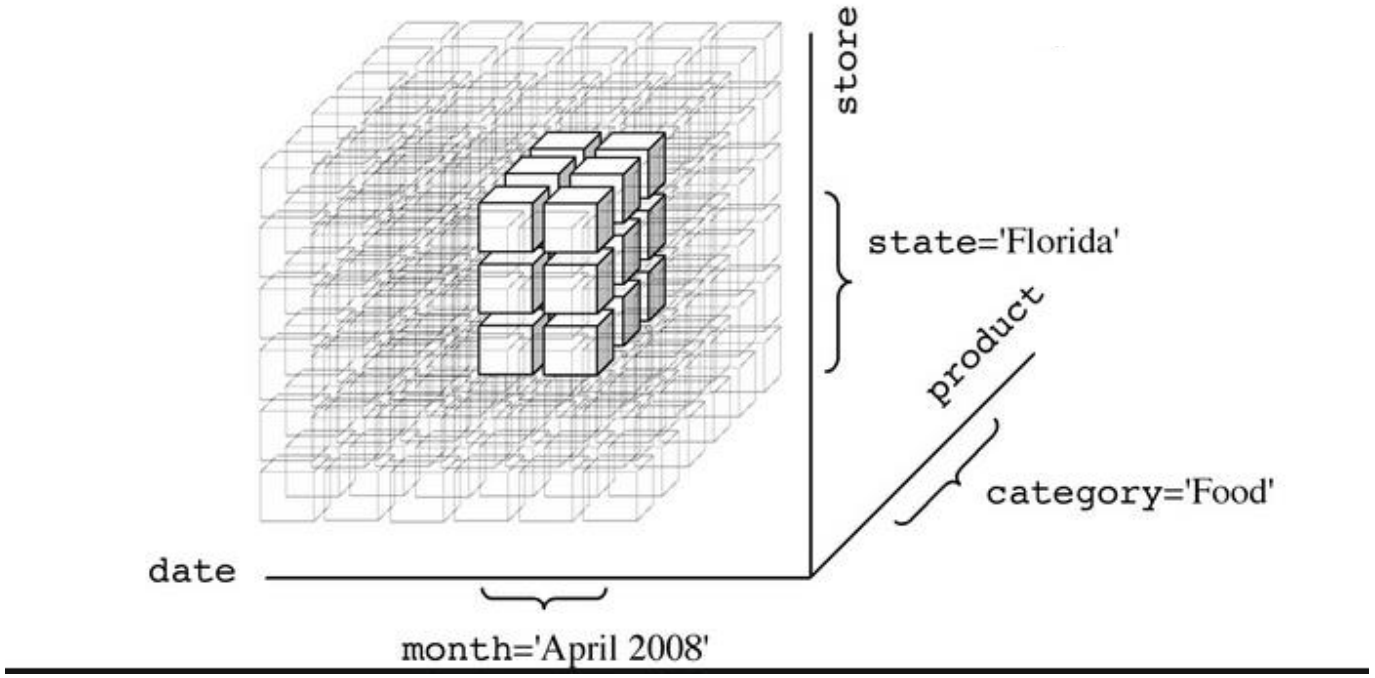
Time=(day 3)OR(day 4)

	cool	hot
day 3	0	1
day 4	0	0

More dicing examples

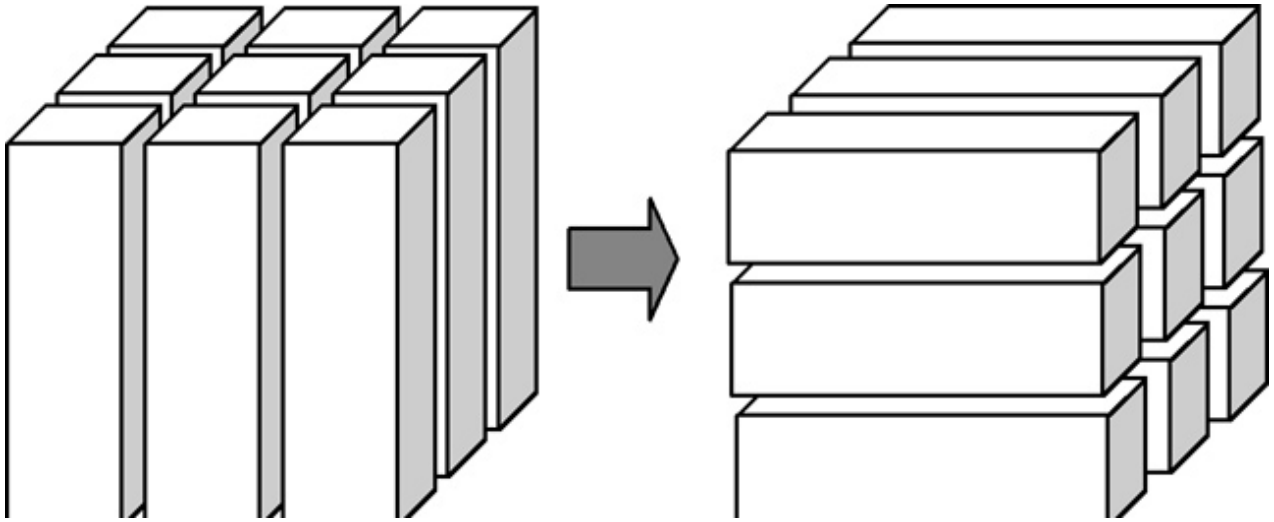


Dicing by exploiting hierarchies

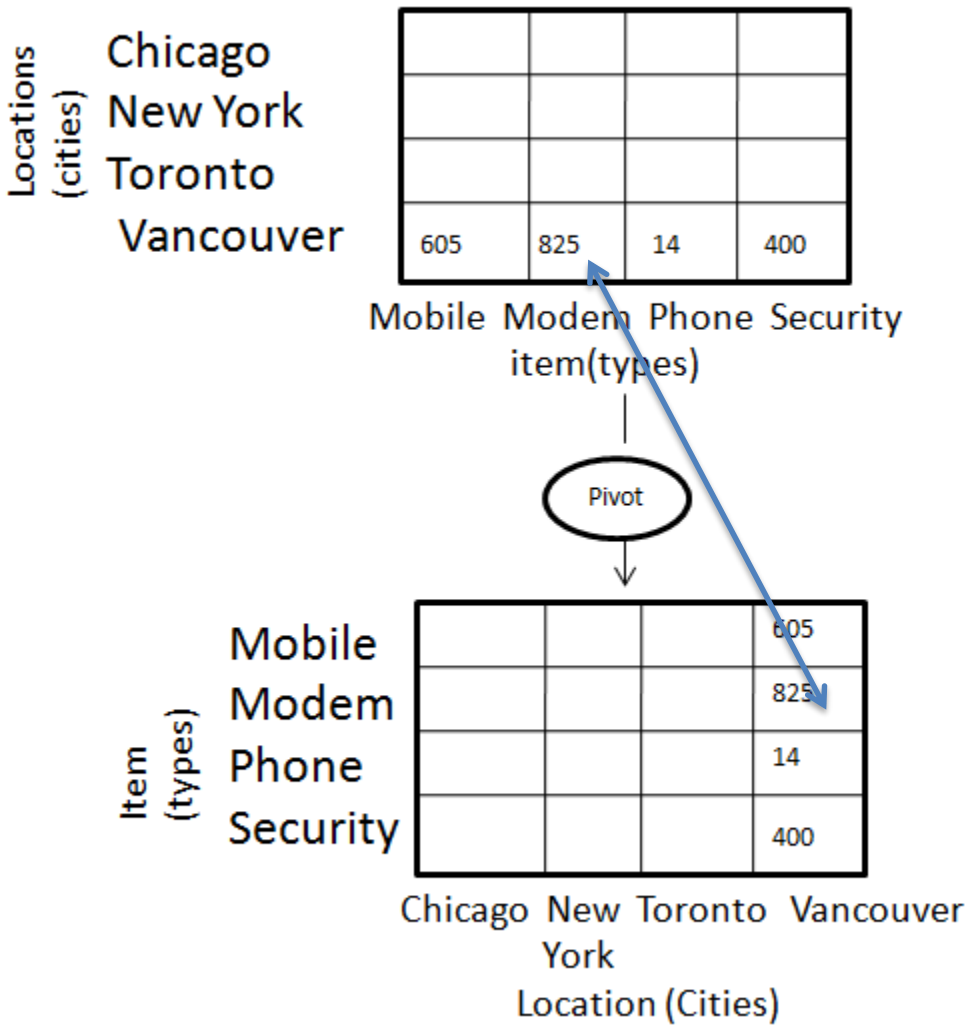


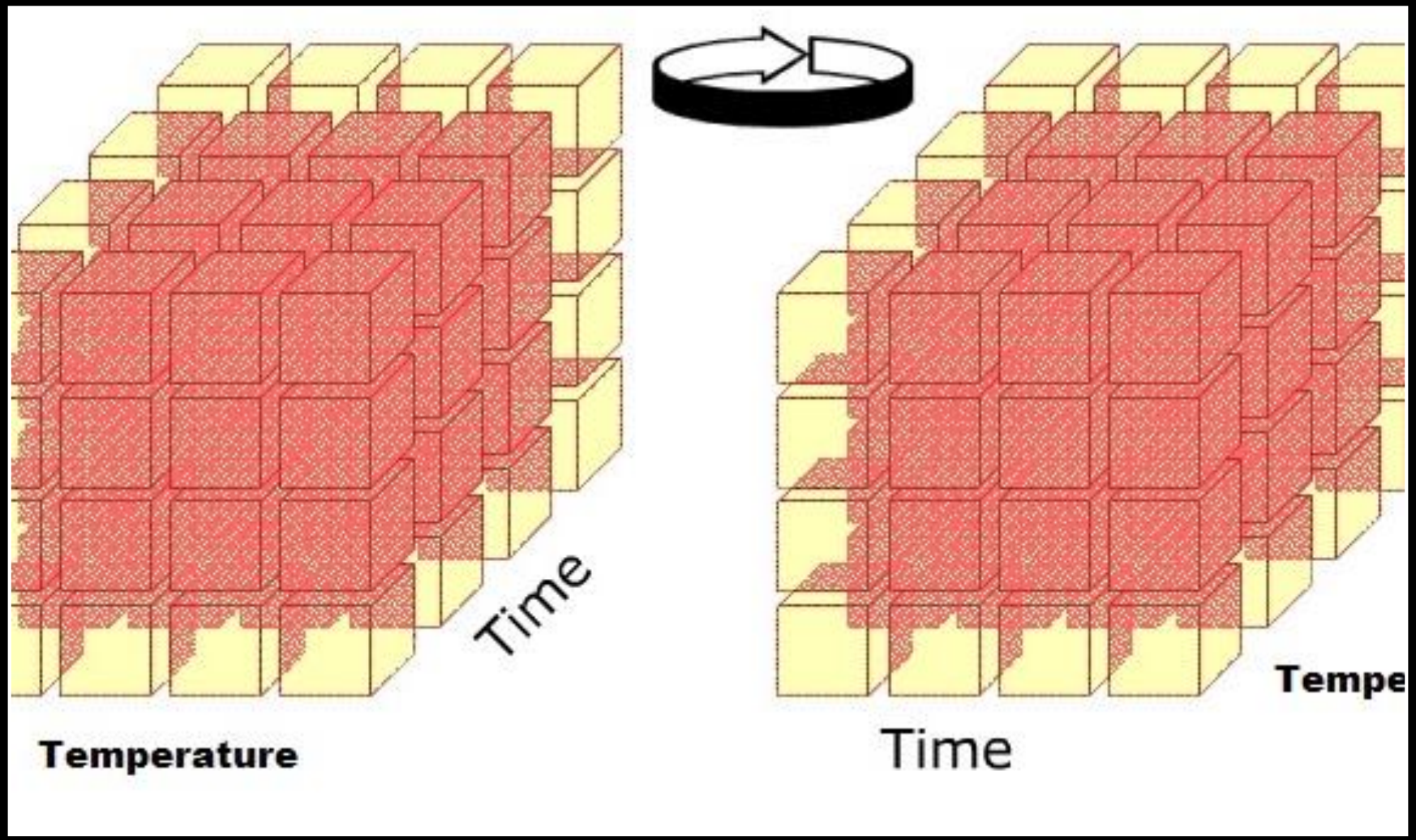
Operations on cubes: 4)Pivoting

- 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.



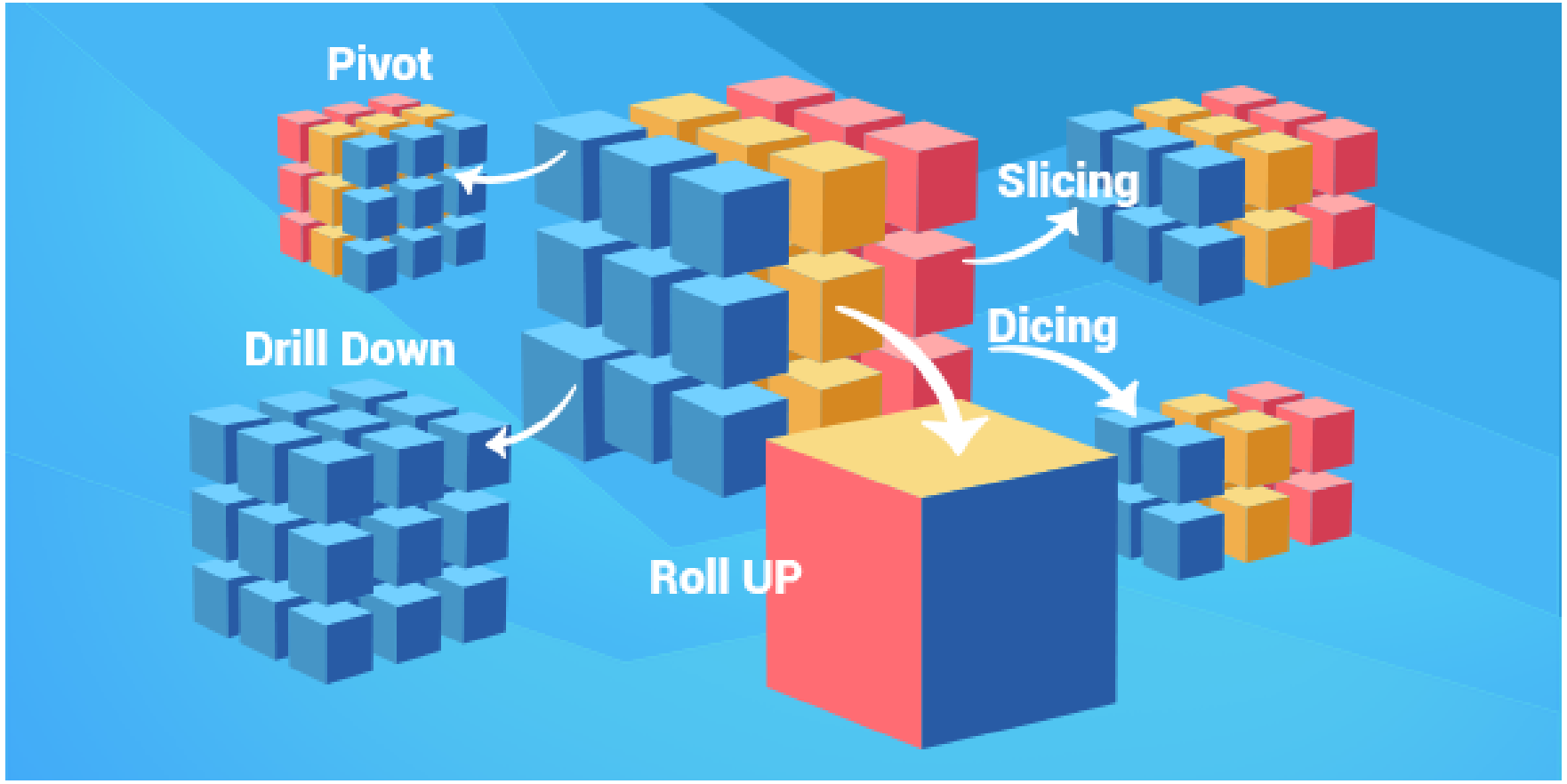
Example pivoting





Example of pivoting on cubes

All together



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

In Class Exercise

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

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
.....	.....	.....	.....
.....	.....	.....	.....

Next Topics

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

- Visualization interfaces