COMP810 Data Warehousing and Big Data

Semester 2 2024

Dr Victor Miranda

COMP810

Week 2 Data Warehousing

- Data Warehousing
- The Logical Model

Week 1 (Week 7) Summary

- Database Concepts
- Data Warehouse Concepts
- Introduction to SQL

Week 2 (Week 8) Outline

- Data Warehouse – OLTP vs OLAP
- The Multidimensional & Logical Models
- Operations in SQL

/* Lecture: 65 min

/* Lab 45-50 min

MOTIVATION - What's a Data Warehouse?

Defined in many different ways, but not rigorously

- *"A data warehouse is a system that **extracts, cleans, conforms, and delivers** source data into a dimensional data store and then supports and implements **querying and analysis** for the purpose of decision making."*

Source: Ralph Kimball, Joe Caserta: The Data Warehouse ETL Toolkit; Wiley 2004

.. Repository, server

The most complex and time-consuming part is

"extracts, cleans, conforms, and deliver"

How complex? 70-80% is basically ETL

Motivation

1. ETL := Extraction, Transformation and Load

- Extract
  - Get the data from different sources as efficiently as possible
- Transform
  - Perform calculations on the data
- Load
  - Load the data into the 'target storage'

Motivation

1. ETL := Extraction, Transformation and Load

- Extract
  - Get the data from different sources as efficiently as possible
- **CLEAN**
  - Perform data cleansing and dimension conforming
- Transform
  - Perform calculations on the data
- Load
  - Load the data into the 'target storage'

Motivation

ETL := Extraction, Transformation and Load

- A piece of software designed
  - To streamline the three (four) E – (C) – T – L steps
  - to perform data transformations
  - Not specifically tight to DW

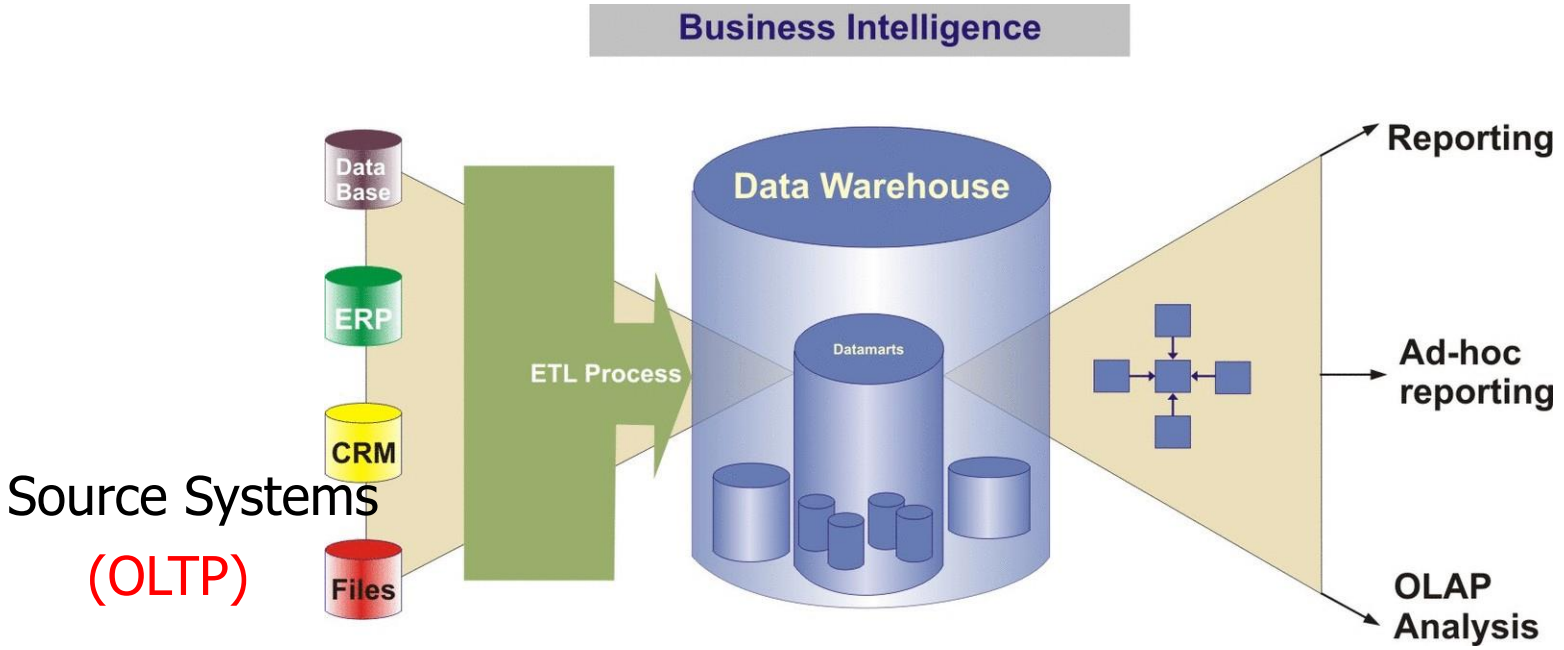
- The **most-underestimated** process un DW development
- The **most time – consuming** process in DW development

80% of development time is spent on ETL!

Loading the Data Warehouse - complex process

In 'practice' (business – wise), it's a central 'store' of all metadata, concepts, and historical information. Serves as a reference to all the entities in the organization.

- Data validation, complex mining, analysis, prediction, etc.



https://www.linkedin.com/pulse/importance-data-warehouse-business-intelligence-ecosystem-patil

Data Integration is Hard

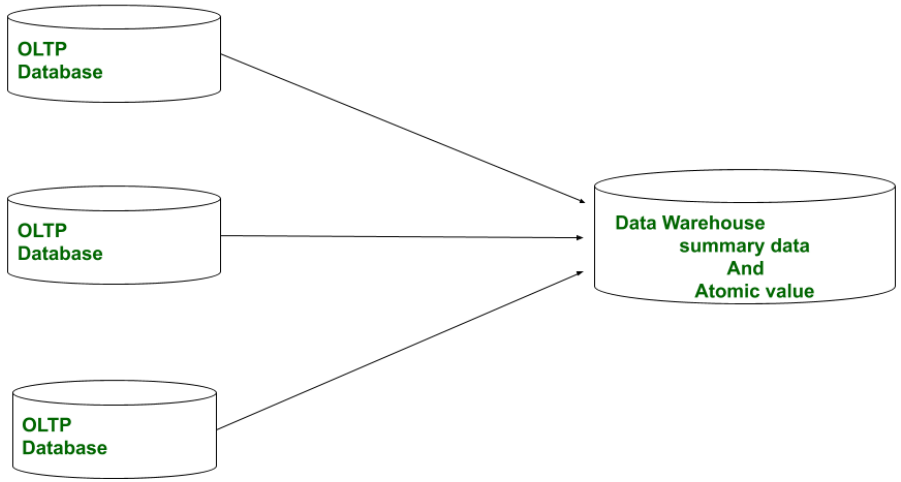
- Data warehouses combine data from multiple sources
- Data must be translated into a consistent format
- Data integration represents ~80% of effort for a typical data warehouse project!
- Some reasons why it's hard:
  - Metadata is poor or non-existent
  - Data quality is often bad
    - Missing or default values
    - Multiple spellings of the same thing
      (Cal vs. UC Berkeley vs. University of California)
  - Inconsistent semantics
    - What is an airline passenger?

OLTP & OLAP Fundamentals

What's OLTP?

> OLTP stands for Online Transaction Processing

OLTP is a type of data processing system used in transaction-oriented applications many operational systems. Supports day-to-day operations.

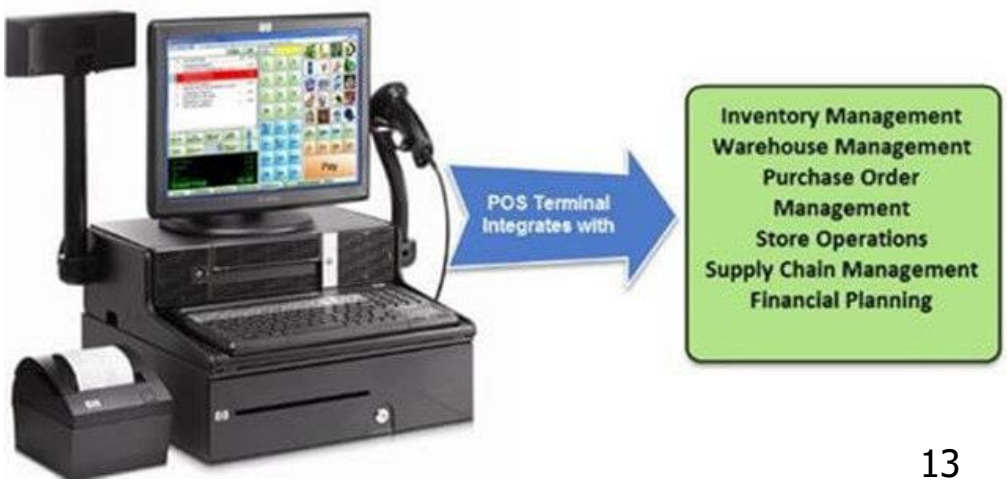


*A data processing system is **a combination of machines, people, and processes that for a set of inputs produces a defined set of outputs**

What's OLTP? Example

> OLTP stands for Online Transaction Processing

Consider a point of sale (POS) system in a supermarket.
You pick a chocolate bar and stand in the line for the self-checkout. For payment, you scan the item's bar code… at the back-end some 'transactions' take place:

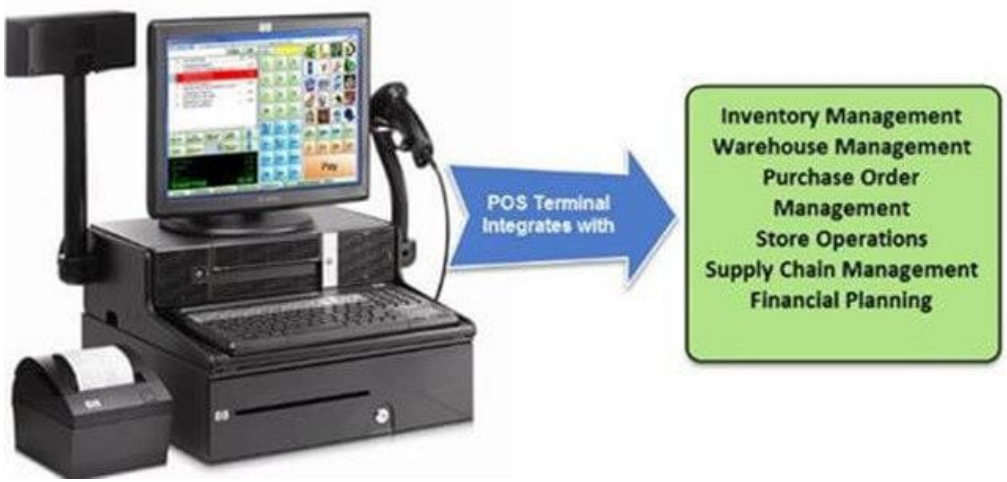


What's OLTP? Example

> OLTP stands for Online Transaction Processing

Consider a point of sale (POS) system in a supermarket.
You pick a chocolate bar and stand in the line for the self-checkout. For payment, you scan the item's bar code… at the back-end some 'transactions' take place:

- ✓ The supermarket database is accessed;
- ✓ The price and product information is retrieved and displayed on screen
- ✓ The machine/cashier feeds in the quantity;
- ✓ The application finally computes the total, generates and prints the purchase receipt. You pay and leave.



What's OLTP? Example

> OLTP stands for Online Transaction Processing

Consider a point of sale (POS) system in a supermarket.
You pick a chocolate bar and stand in the line for the self-checkout. For payment, you scan the item's bar code… at the back-end some 'transactions' take place:

- ✓ The supermarket database is accessed;
- ✓ The price and product information is retrieved and displayed on screen
- ✓ The machine/cashier feeds in the quantity;
- ✓ The application finally computes the total, generates and prints the purchase receipt. You pay and leave.

The 'system' has just added a record of your purchase in the database. This is an example of an on-line transaction processing (OLTP) system (online transaction + query processing)

** The POS of this supermarket is supported by an OLTP system

What's OLTP? (Example 2)

> OLTP stands for Online Transaction Processing

What's OLTP?

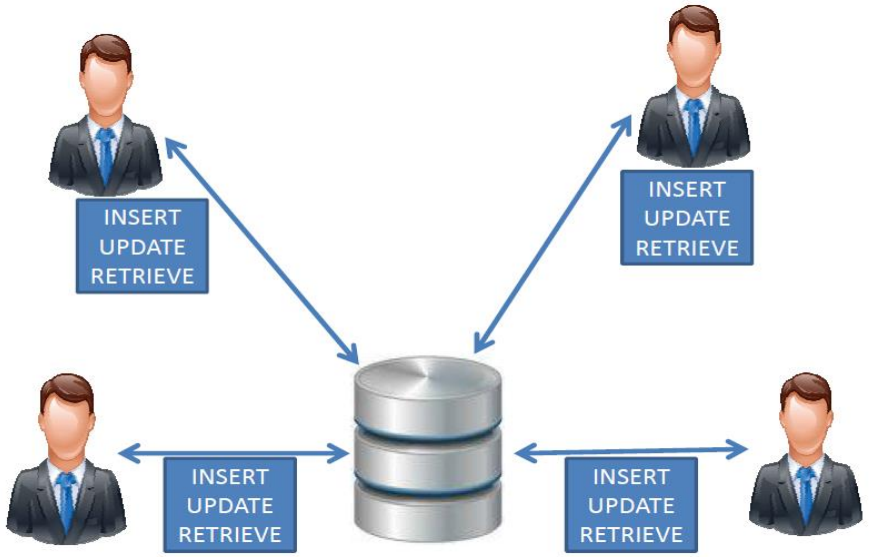
It's about TRANSACTIONS

- Multiple users can fetch the information
- Very fast response rate
- Transactions processed immediately
- Everything is processed in real time
- Frequent updates

OLTP Segmentation

– Real-time Transaction Processing

– Batch Processing



>> Queries an OLTP system can process

- Search for a particular customer's record.
- Retrieve the product description and unit price of a particular product.
- Filter all products with a unit price equal to or above Rs. 25.
- Filter all products supplied by a particular supplier.
- Search and display the record of a particular supplier.

Queries an OLTP system CANNOT process

> **The supermarket plans on introducing a new product.**

(A) "Which product should they introduce?"
(B) "Should it be specific to a few customer segments?"

Queries an OLTP system CANNOT process

> **The supermarket plans on introducing a new product.**

(A) "Which product should they introduce?"
(B) "Should it be specific to a few customer segments?"

> **The supermarket will reward (offer discounts to) loyal clients.**

(A) "How much discount should they offer?"
(B) "Difference rates for different customer segments?"

Queries an OLTP system CANNOT process

> **The supermarket plans on introducing a new product.**

(A) "Which product should they introduce?"
(B) "Should it be specific to a few customer segments?"

> **The supermarket will reward (offer discounts to) loyal clients.**

(A) "How much discount should they offer?"
(B) "Difference rates for different customer segments?"

> **The supermarket plans on opening a branch.**

(A) "Location?"

These queries are not meant to be solved by an OLTP system.

What's OLAP?

Online Analytical Processing (OLAP) is essentially technology (software) designed to organize large business databases and support / guide strategic decisions.

- Provides multidimensional **view** of data
- Data can be viewed from different perspectives
- Determine why data appears the way it does
- **Drill down approach** is used to further dig down deep into the data

What's OLAP?

Online Analytical Processing (OLAP) is essentially technology (software) designed to organize large business databases and support / guide strategic decisions.

- Provides multidimensional **view** of data
- Data can be viewed from different perspectives
- Determine why data appears the way it does
- **Drill down approach** is used to further dig down deep into the data

However:

- Complex queries
- Infrequent updates
- Transactions access a large fraction of the database
- Data need not be up-to-date

OLAP - Example

Think of a supermarket, and its entire database for the year 2021

The data is captured by the OLTP system, columns:

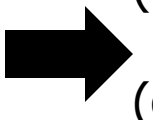
Section, Product-CategoryName, YearQuarter, and SalesAmount.
We have a total of 32 records/rows.

OLAP - Example

Think of a supermarket, and its entire database for the year 2021

The data is captured by the OLTP system, columns:

Section, Product-CategoryName, YearQuarter, and SalesAmount.
We have a total of 32 records/rows.



(a) Section -> 'Female', 'Male', 'Kid', 'Infant'

(b) ProductCategory – 'Accessories' 'Clothing'

(c) YearQuarter -> 'Q1', 'Q2', 'Q3', 'Q4'

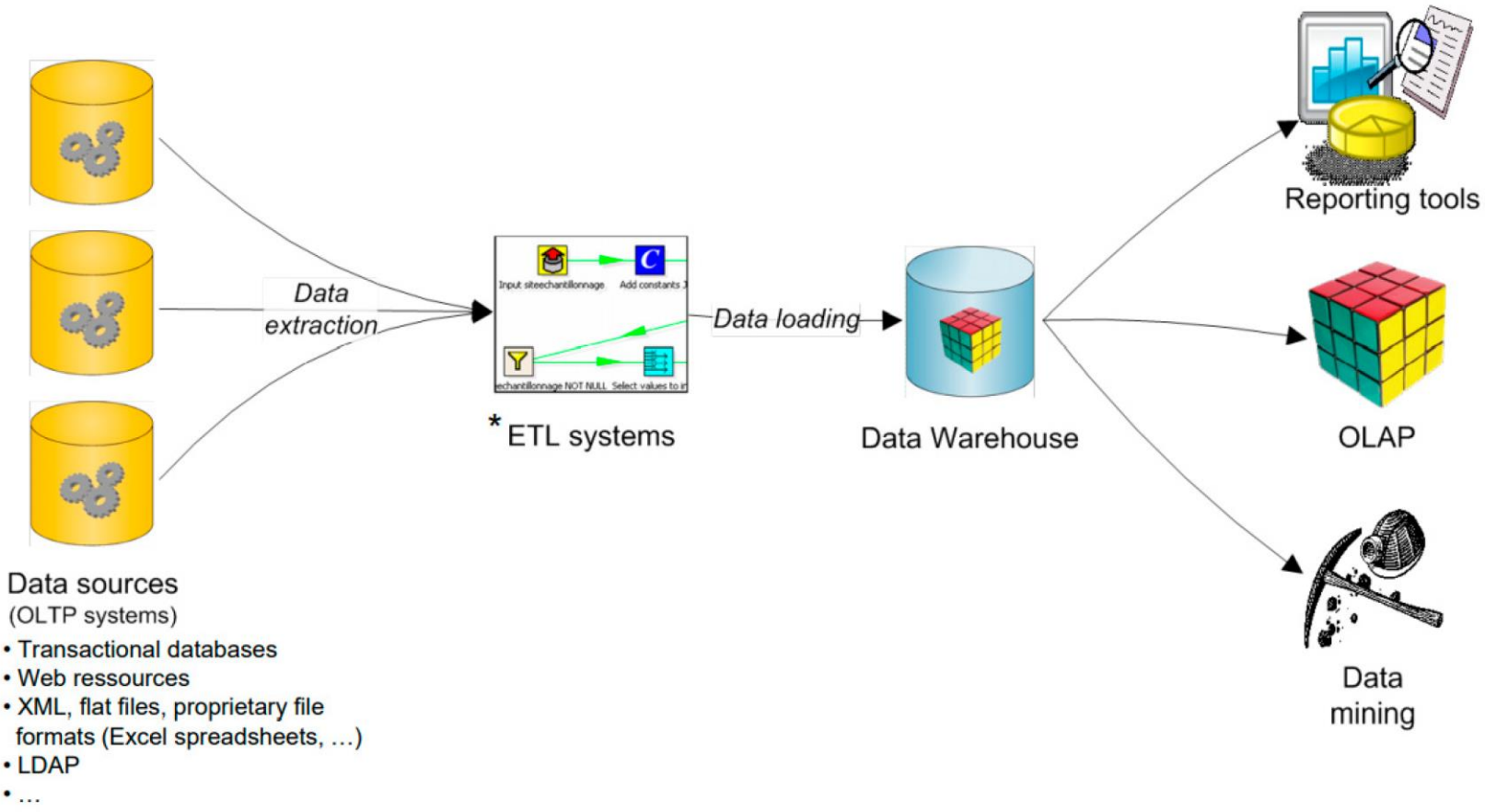
(d) SalesAmount column record the sales figures for each Section, ProductCategory Name, and Year Quarter.

OLTP

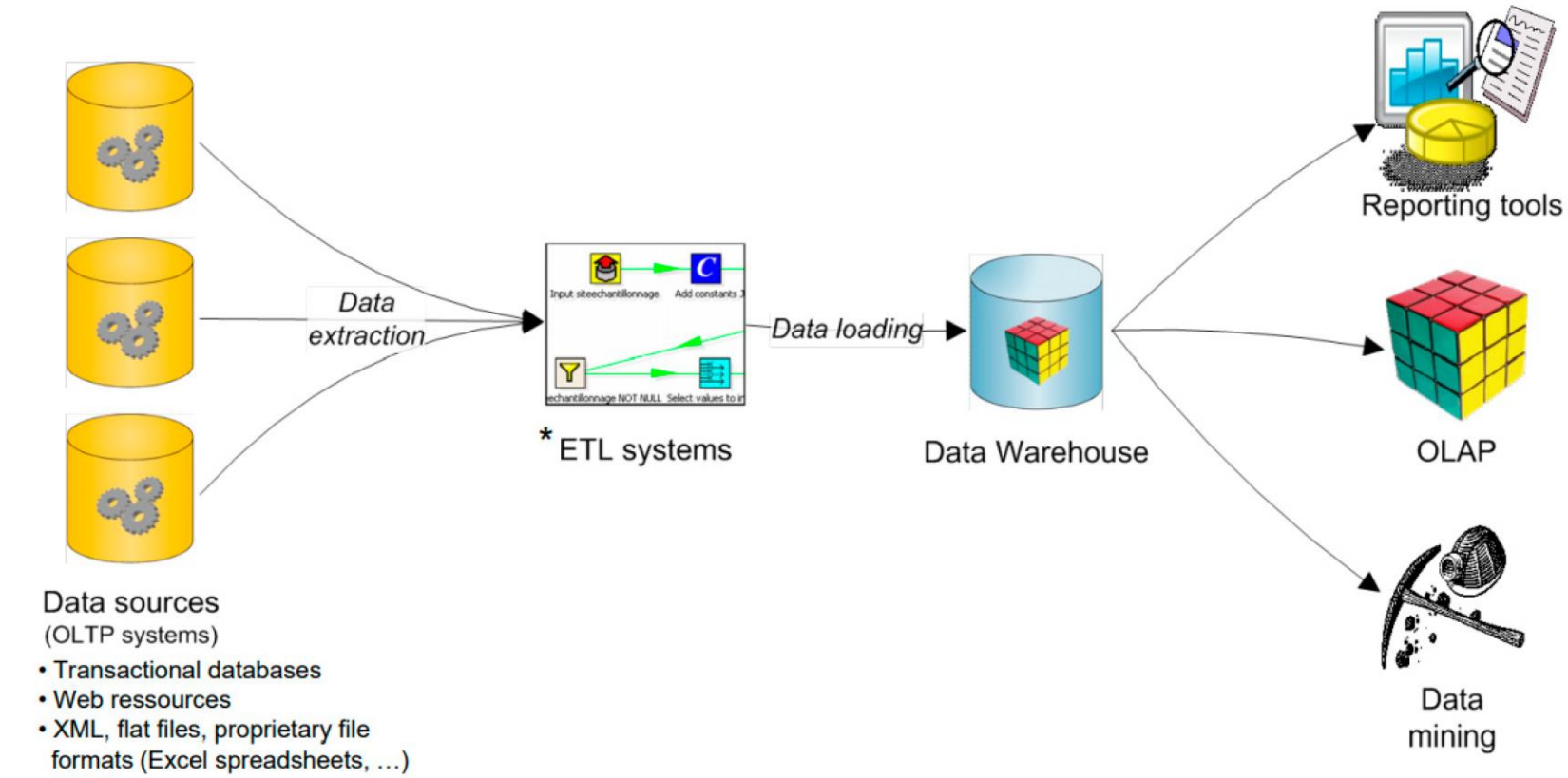
OLAP

* Here, entries for the Section column are 'Female', 'Male', 'Kid', 'Infant'. Etc…

OLTP + ETL + DW + OLAP



OLTP + ETL + DW + OLAP



OLTP vs OLAP

- On Line Transaction Processing – *OLTP*
  - Maintains a database embedded as an accurate model of some real-world enterprise. Supports day-to-day operations.
    Characteristics:
    - Short simple transactions
    - Relatively frequent updates
    - Transactions access only a small fraction of the database

OLTP vs OLAP

- On Line Transaction Processing – *OLTP*
  - Maintains a database embedded as an accurate model of some real-world enterprise. Supports day-to-day operations.
    Characteristics:
    - Short simple transactions
    - Relatively frequent updates
    - Transactions access only a small fraction of the database

- On Line Analytic Processing – *OLAP*
  - Uses information in database to guide strategic decisions.
    Characteristics:
    - Complex queries
    - Infrequent updates
    - Transactions access a large fraction of the database
    - Data need not be up-to-date

DW Project – like queries

- OLTP-style transaction:
  - John Smith, from Schenectady, N.Y., just bought a box of tomatoes; charge his account; deliver the tomatoes from our Schenectady warehouse; decrease our inventory of tomatoes from that warehouse

DW Project - like

- OLTP-style transaction:
  - John Smith, from Schenectady, N.Y., just bought a box of tomatoes; charge his account; deliver the tomatoes from our Schenectady warehouse; decrease our inventory of tomatoes from that warehouse
- OLAP-style transaction - I:
  - How many cases of tomatoes were sold in all northeast warehouses in the years 2000 and 2001?

DW Project - like

- OLTP-style transaction:
  - John Smith, from Schenectady, N.Y., just bought a box of tomatoes; charge his account; deliver the tomatoes from our Schenectady warehouse; decrease our inventory of tomatoes from that warehouse
- OLAP-style transaction – II (specific):
  - Prepare a profile of the grocery purchases of John Smith for the years 2000 and 2001 (so that we can customize our marketing to him and get more of his business)

Data Mining

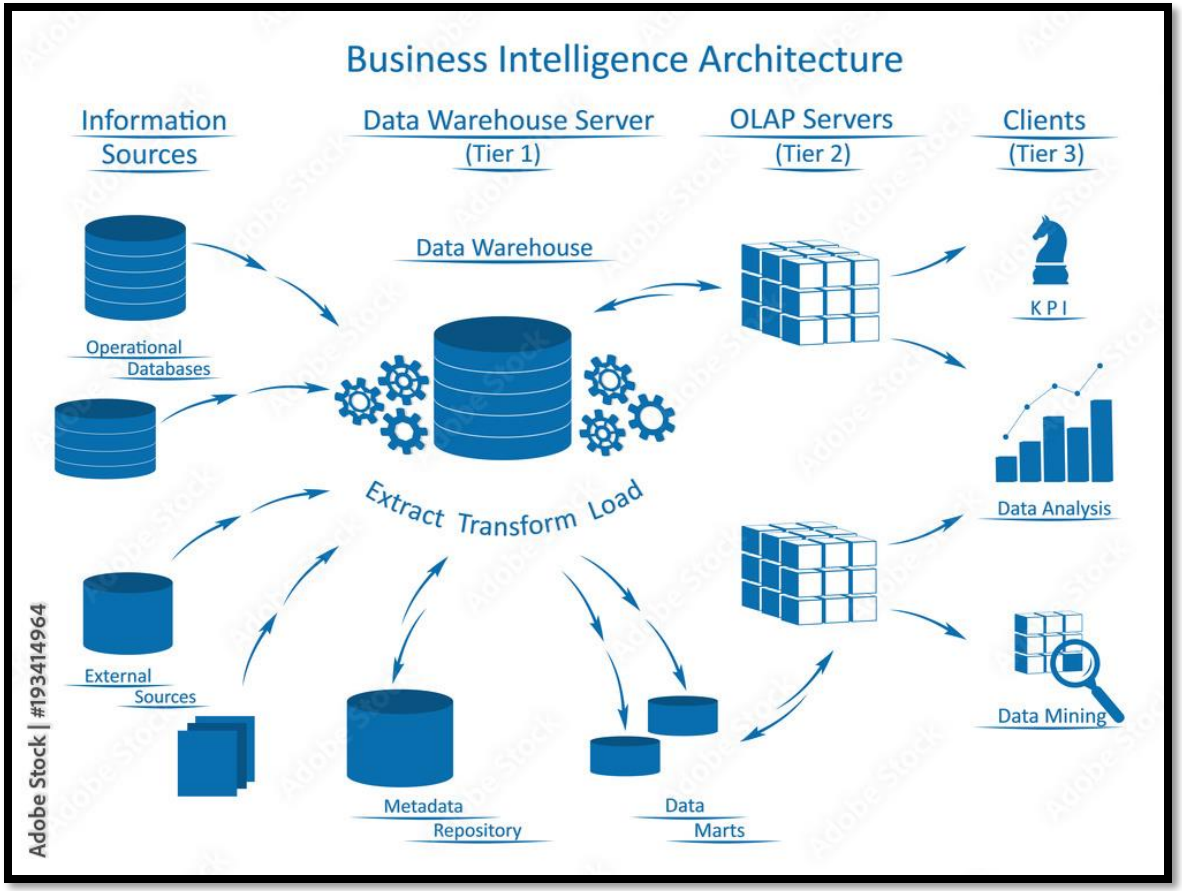
- *Data Mining* - **Identify patterns and relationships that can help solve business problems**
  - *OLAP* :
    - What percentage of people who make over $50,000 defaulted on their mortgage in the year 2000?

Data Mining

- *Data Mining* - **Identify patterns and relationships that can help solve business problems**
  - *OLAP* :
    - What percentage of people who make over $50,000 defaulted on their mortgage in the year 2000?
  - *Data Mining* :
    - How can information about salary, net worth, and other historical data be used to *predict* who will default on their mortgage?

DW + DW Server + Data Mining

- OLAP and data mining databases are stored on special servers called **data warehouses** to accommodate the huge amount of data generated by OLTP systems



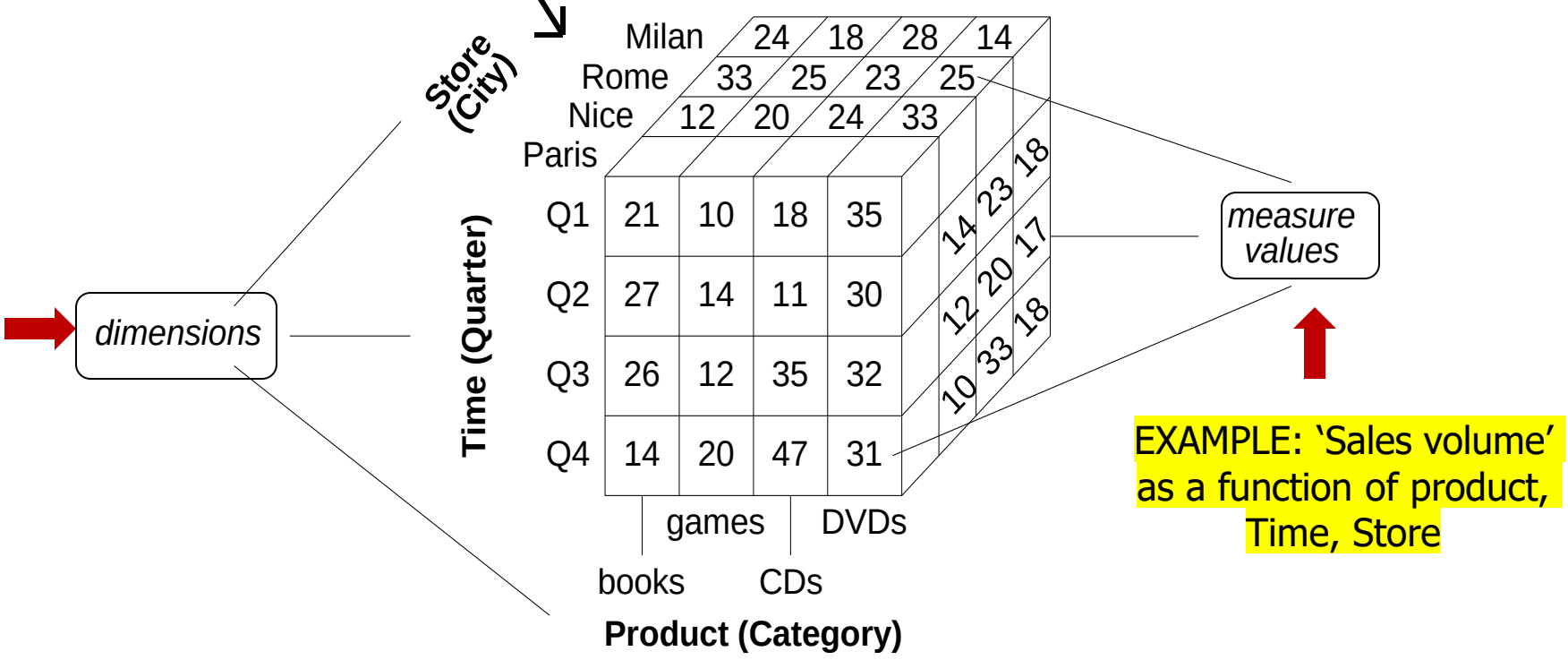
Source: adobeStock

Data Warehouse Multidimensional Model

Multidimensional view of data

Every data warehouse can be seen as a multidimensional data model represented as a **data cube** or a **hypercube**

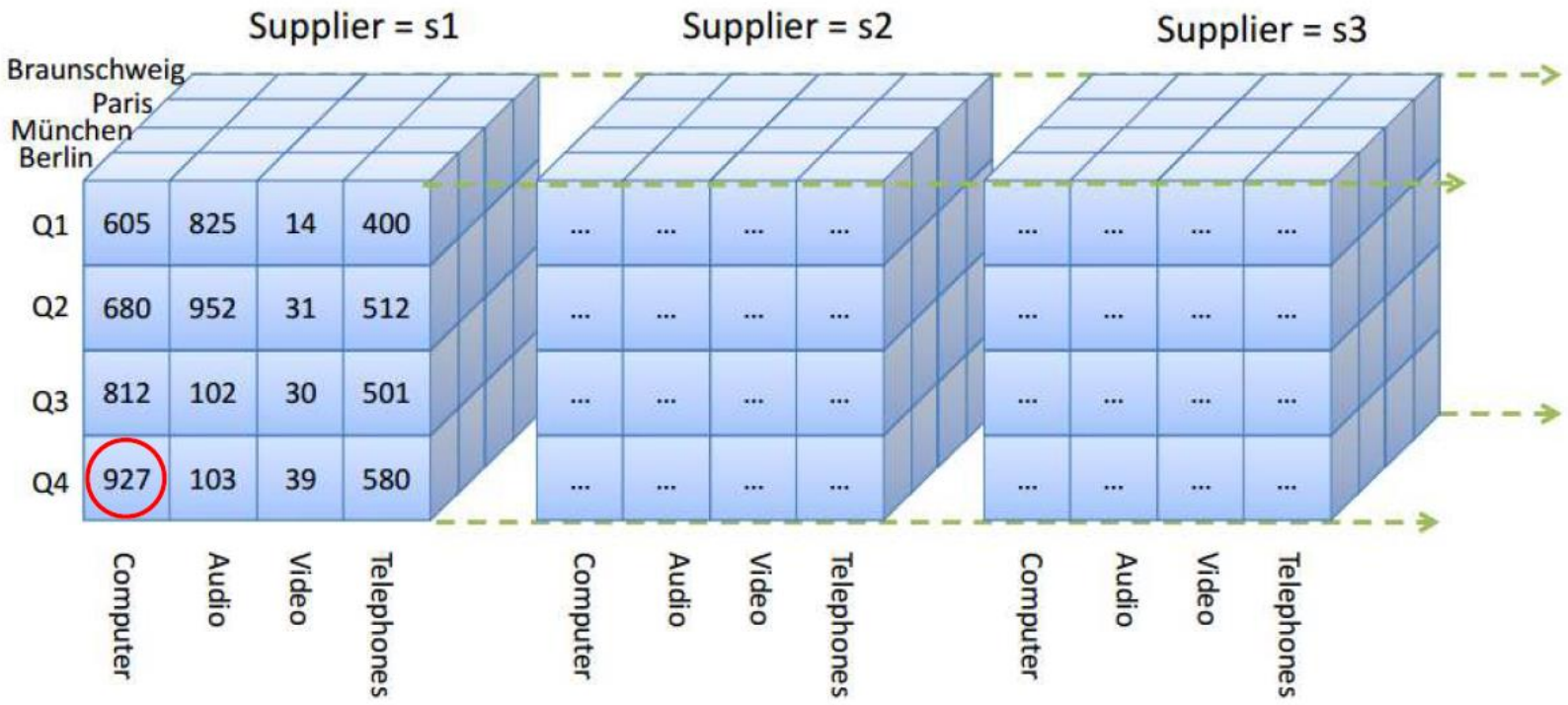
- Dimensions: Perspectives for analyzing data
- Cells (facts): Contain measures, values that are to be analyzed



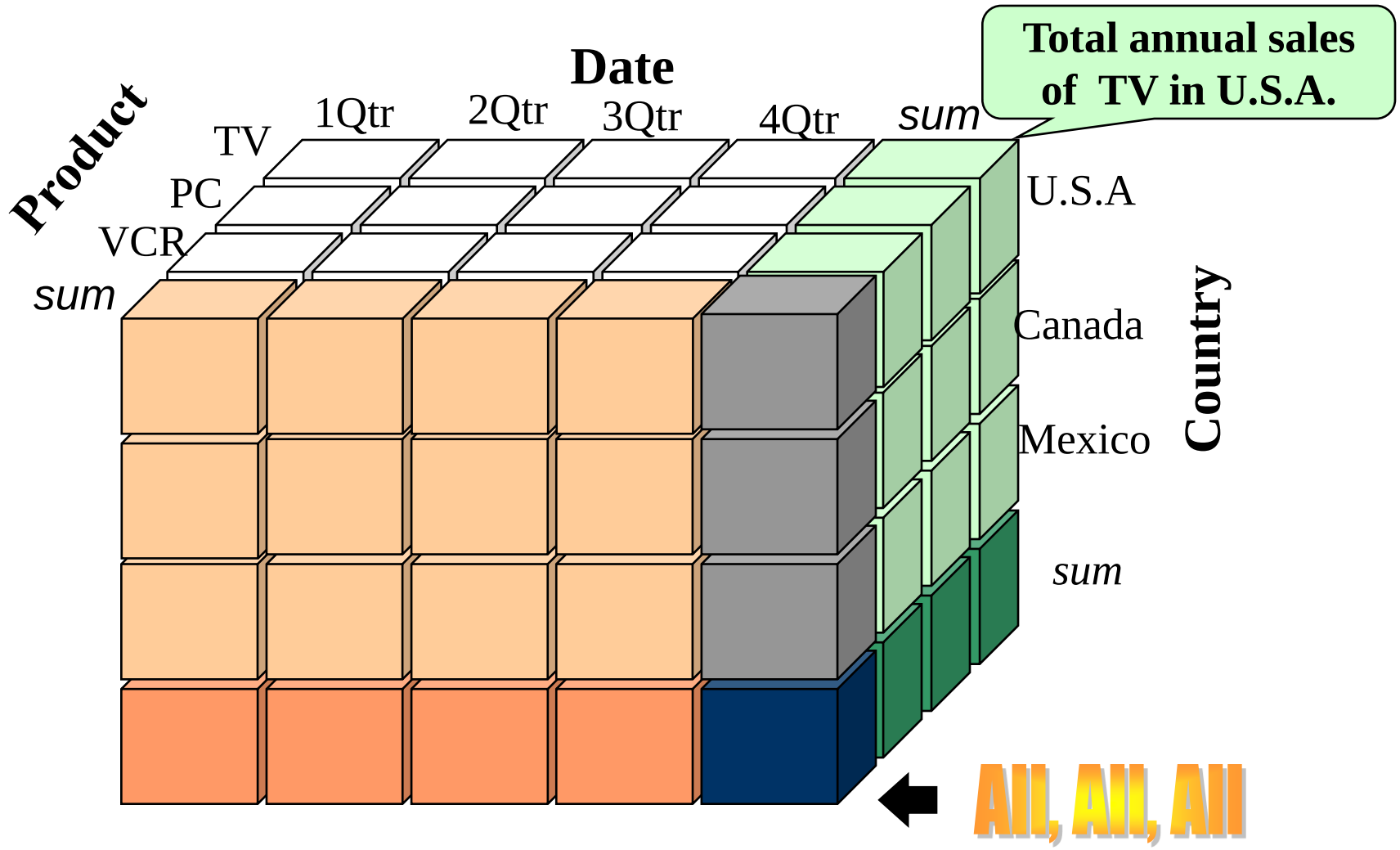
Visualizing data in cubes

Four dimensions (example 2)

REMEMBER: Cubes consist of Fact data with one or more observations

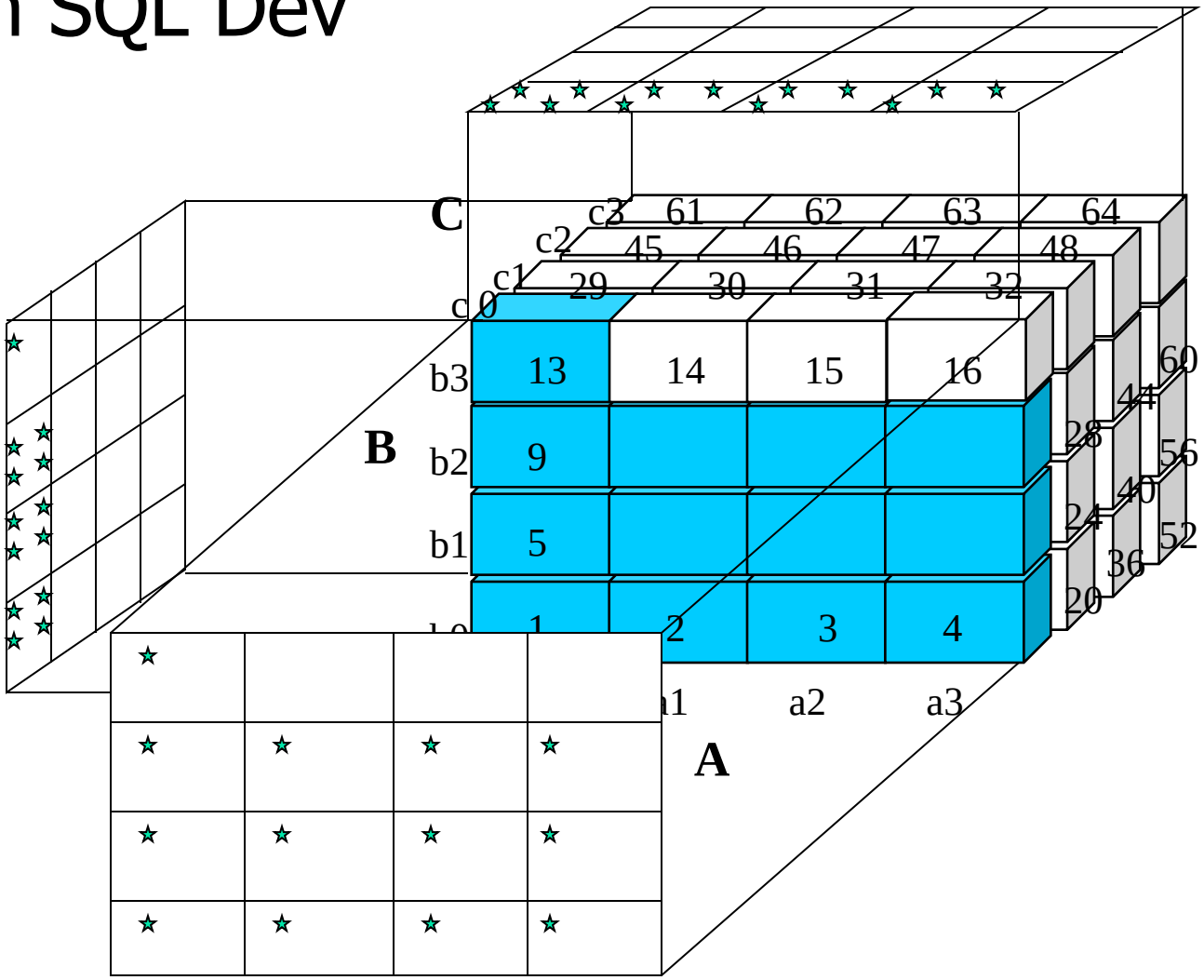


A Sample Data Cube



In practice, the task could be hard to complete

In SQL Dev



The Data Warehouse Logical Data Model

DW Logical Data Model

DW Logical (Conceptual) data models are used to visualize data entities, attributes, keys and relationships. It establishes the structure of data elements and the relationships among them. It is independent of the physical database

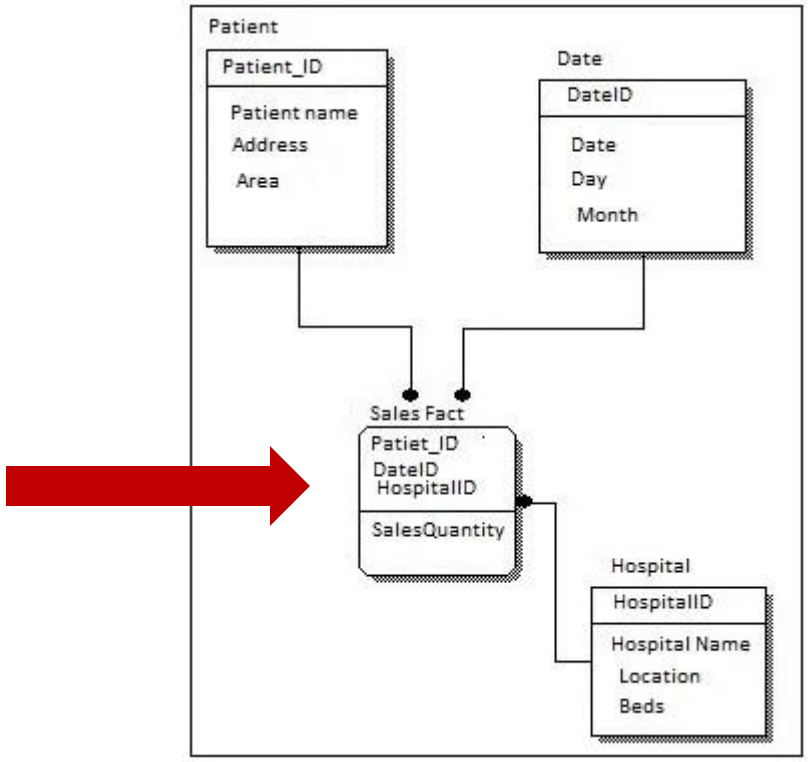


Figure. Example of a logical data model

Conceptual Modelling of Data Warehouses

- Modeling data warehouses: dimensions & measures
  - Star schema: A fact table in the middle connected to a set of dimension tables
  - Snowflake schema: A refinement of star schema where some dimensional hierarchy is normalized into a set of smaller dimension tables, forming a shape similar to snowflake
  - Fact constellations: Multiple fact tables share dimension tables, viewed as a collection of stars, therefore called galaxy schema or fact constellation

Star Schema

* Every data warehouse can be seen as a multidimensional data model *

>> In this course we explore - and adopt –

(i) a widely-used approach for 'modelling' (graphics) data warehouses known as **star schema,** and

(ii) its two primary components: **Fact** and **Dimension Tables**

It is composed of a single **fact** table that references any number of **dimension tables.**

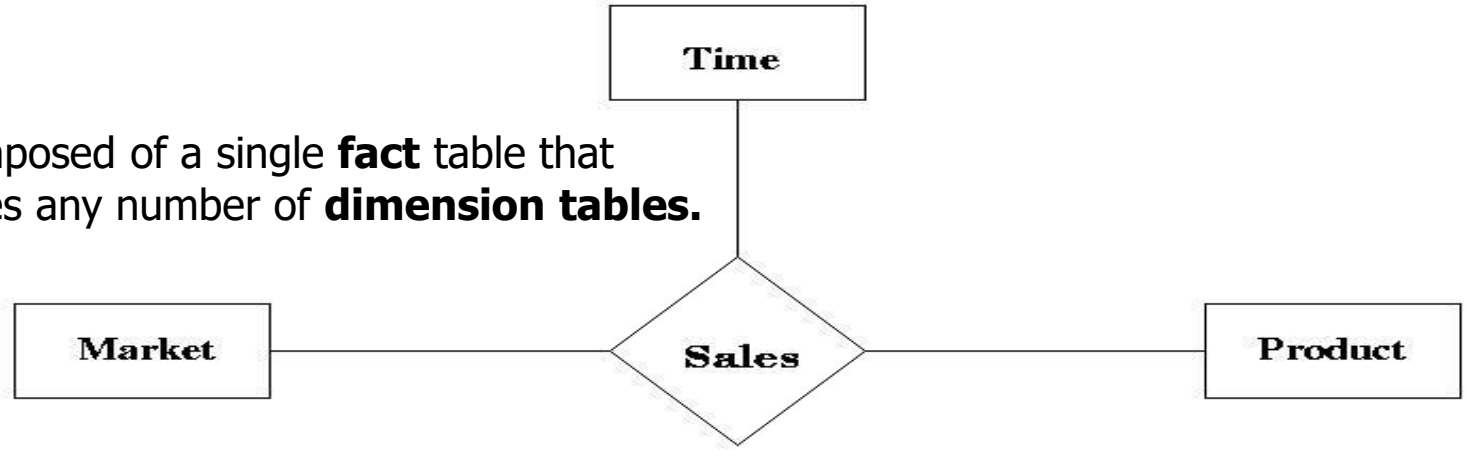


Figure. Example of Star Schema

‘Schema’ of an SQL Table & Key attributes

- The *schema* of a table is the table name and its attributes – Notation:

  Product(PName, Price, Category, Manfacturer)

- A *key* is an attribute whose values are unique; we underline a key – Notation:

Product(PName, Price, Category, Manfacturer)

Star Schema design – Theory

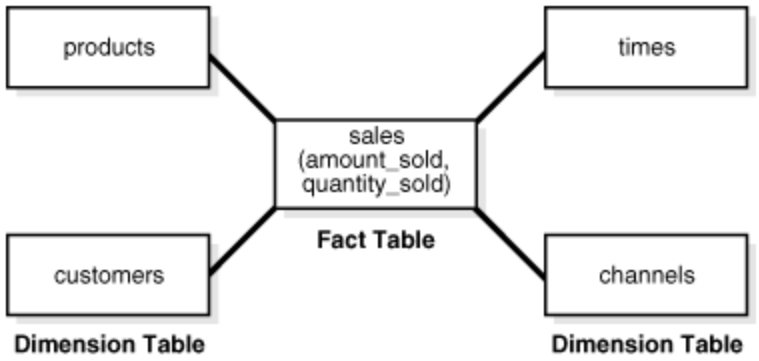
A data cube, such as sales, allows data to be modeled and viewed in multiple dimensions:

- **Fact tables -** Hold the **primary keys** of the referenced **dimension tables** along with some **quantitative metrics** over which some sort of calculation can be performed.

Examples: product (p_name, brand, type), or time(day, week, month, quarter, year)

- **Dimension tables –** These hold, on the other hand, the **descriptive information** for all related fields that are included in the fact table's record.

Examples: sales, orders, time series financial data.



Star Schema design – In practice

Most data warehouses use a star schema to represent multi-dimensional model.

- Each dimension is represented by a **dimension table** that describes it (attributes).
- A **fact table** connects to all dimension tables with a multiple join. Each tuple in the fact table consists of a pointer to each of the dimension tables that provide its multi-dimensional coordinates and stores measures for those coordinates.
- The links between the fact table in the center and the dimension tables in the extremities form a shape like a star.

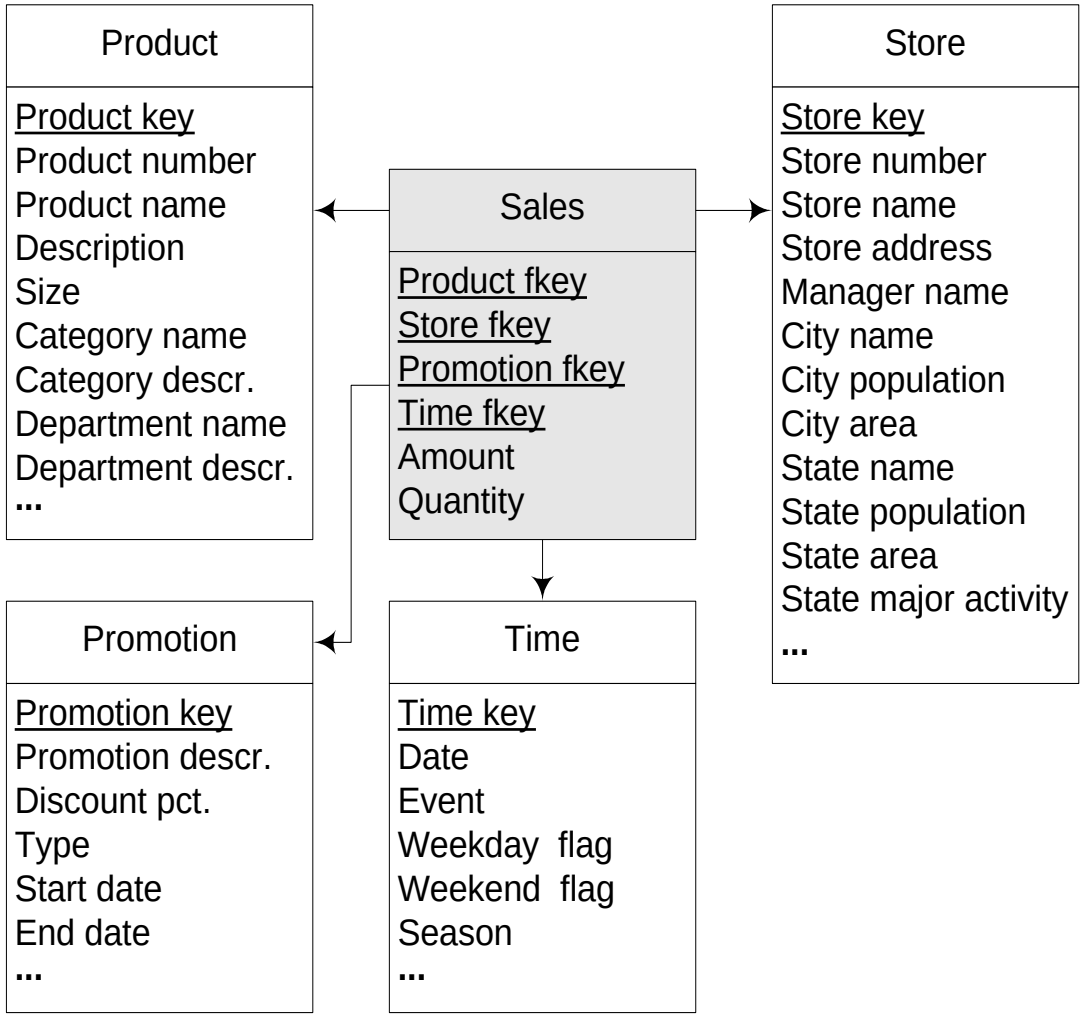
Logical structure of the model - Star Schema

E1

Fact: Sales

Measures:
Amount,
Quantity

Dimension tables:
Product,
Promotion,
Time,
Location



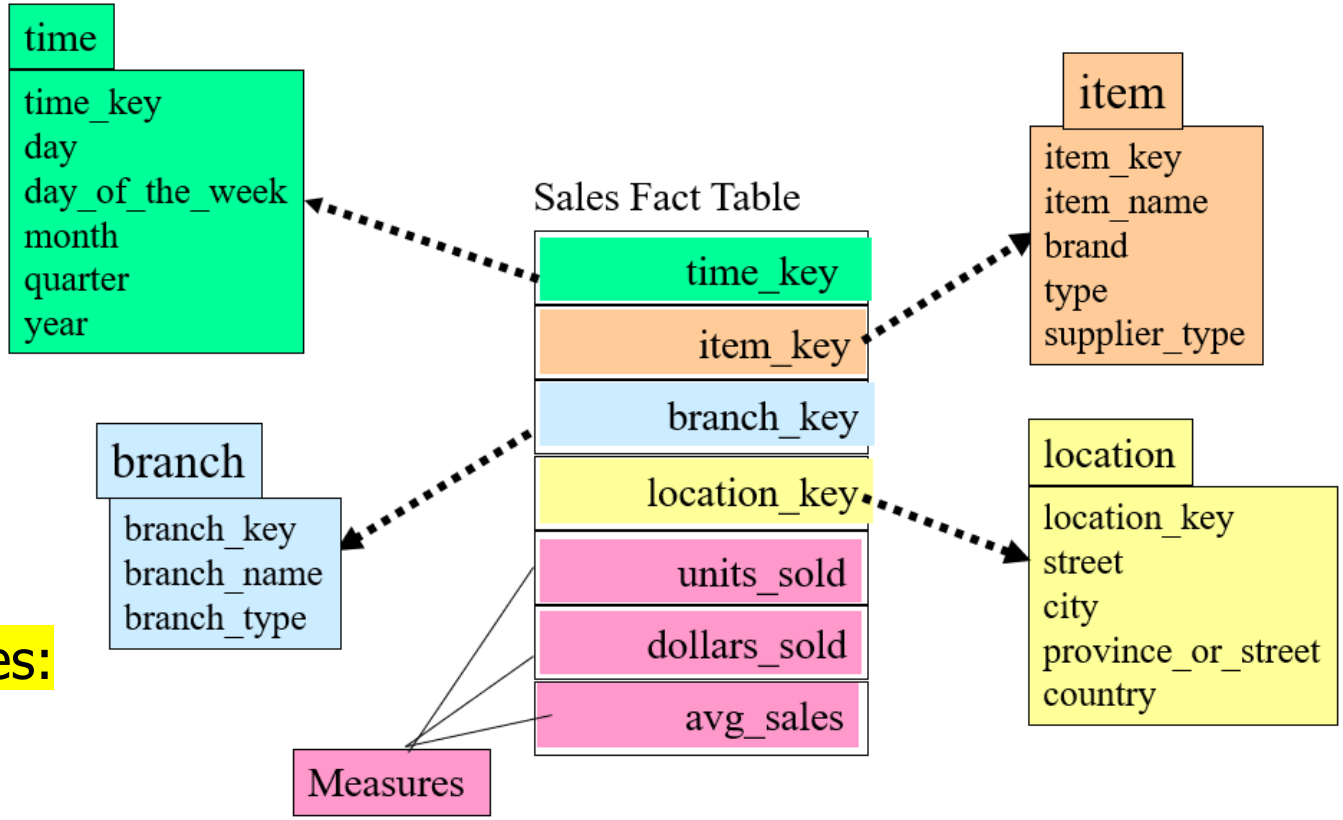
Logical structure of the model - Star Schema

E2

Fact: Sales

Measures:
- Units_sold
- dollars_sold
- avg_sales

Dimension tables:
- Time,
- Branch,
- item,
- Location



Warehouse Logical Modelling (WLM)

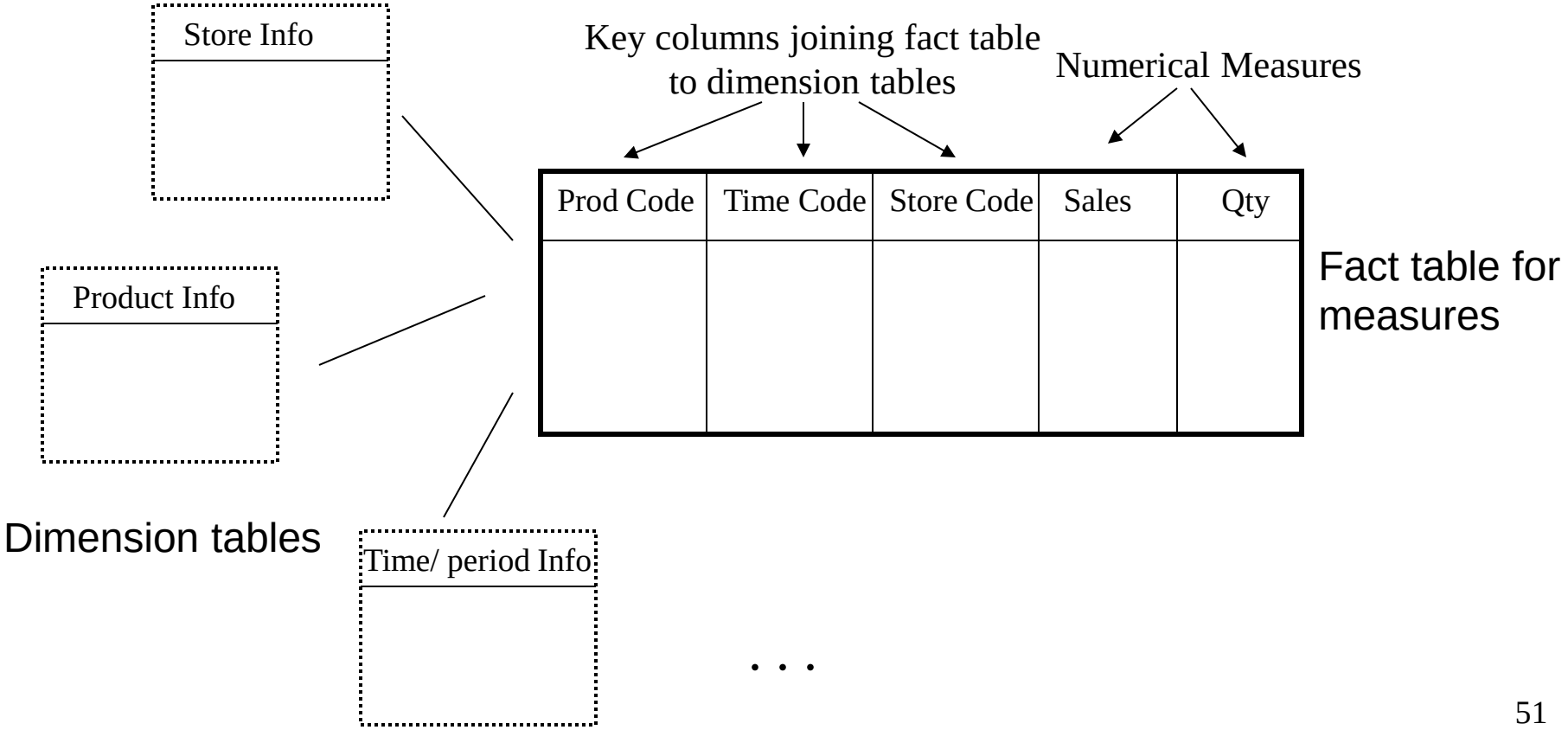
SQL WLM design = Multidimensional model + Logical Schema + implementation

In this course our main interest is in the database design and implementation in SQL Dev

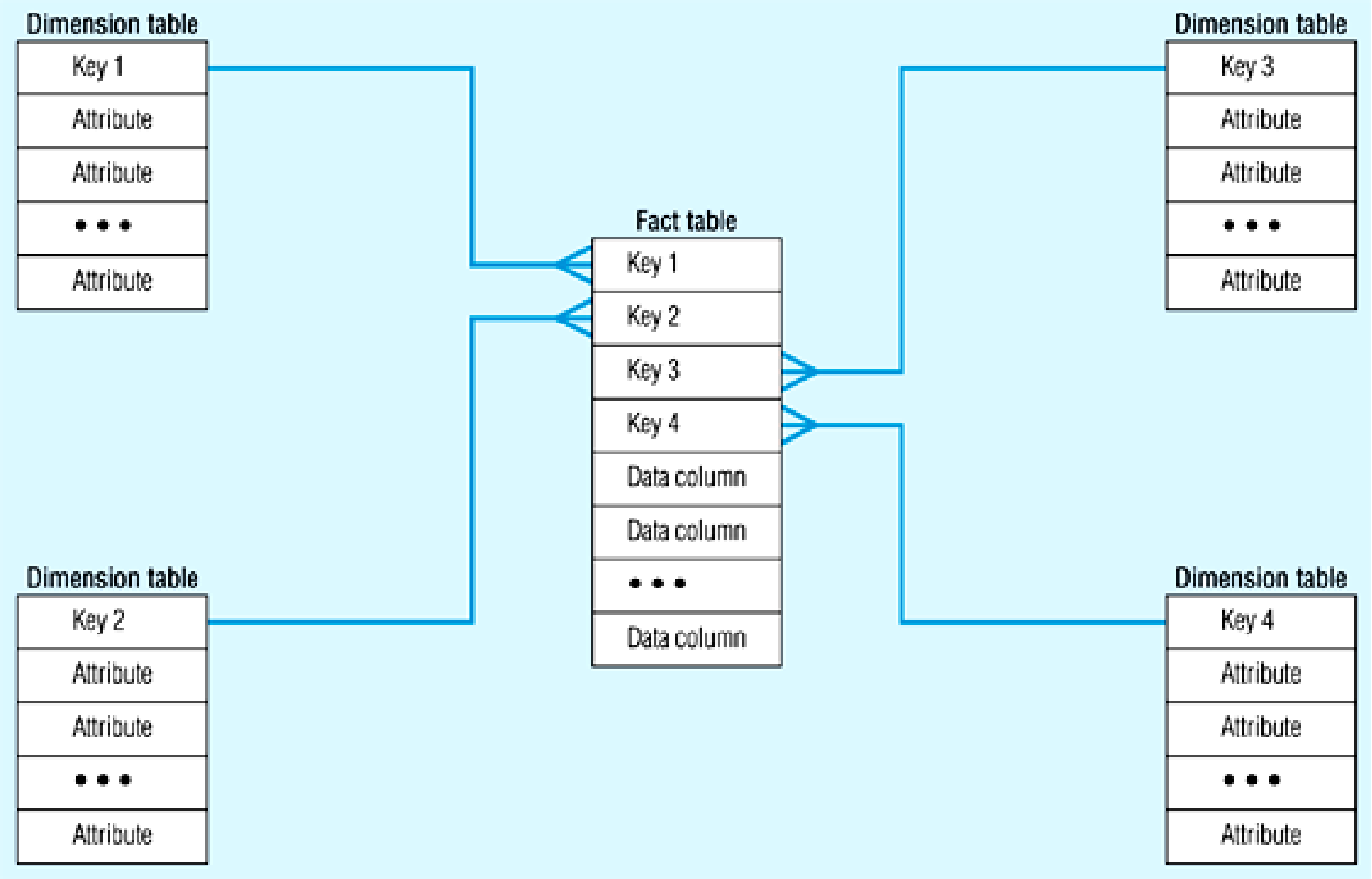
The Multi-Dimensional Model

"Sales by product line over the past six months"

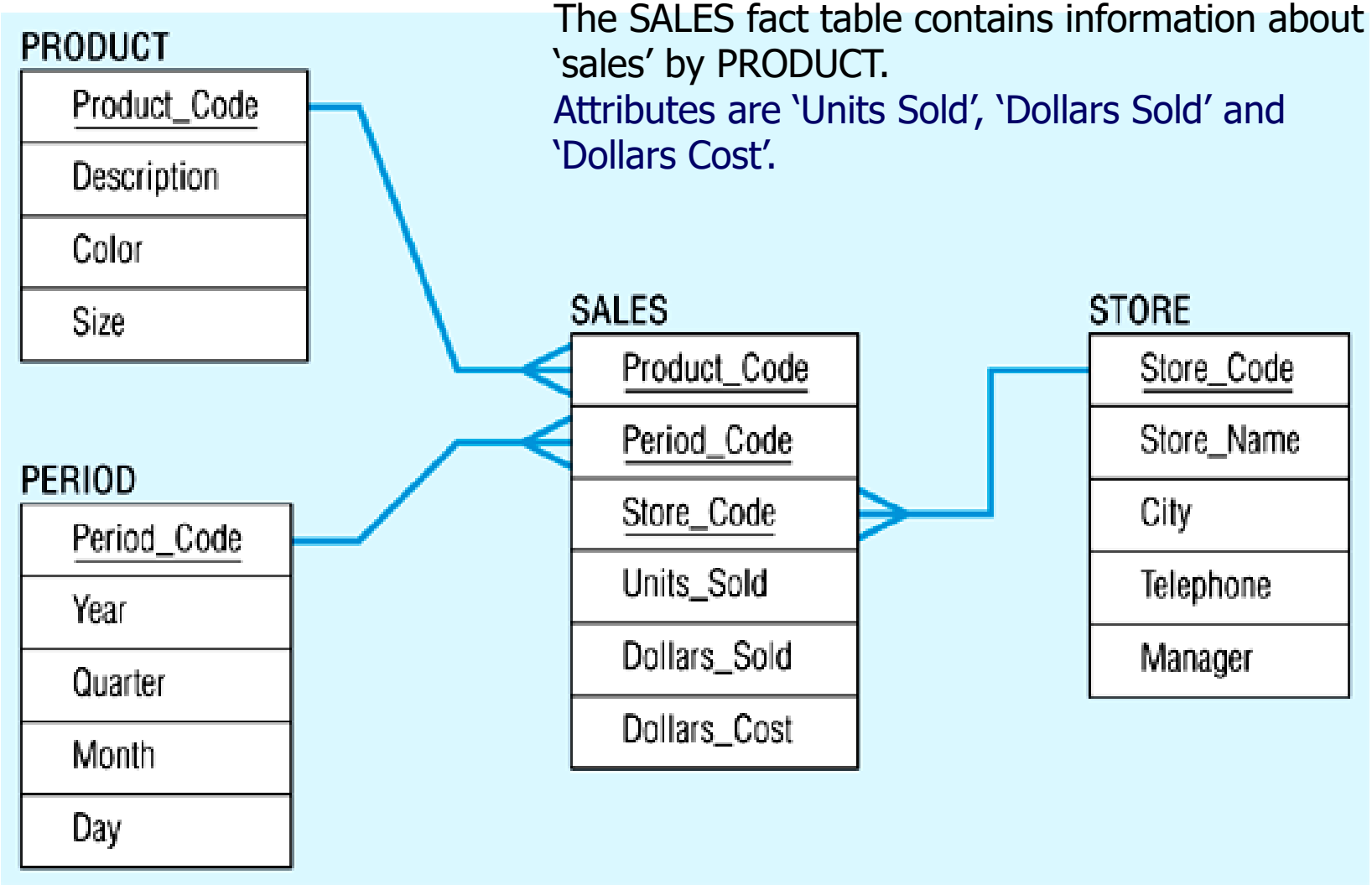
"Sales by store between 1990 and 1995"



Star Schema (in RDBMS)



Star Schema Example



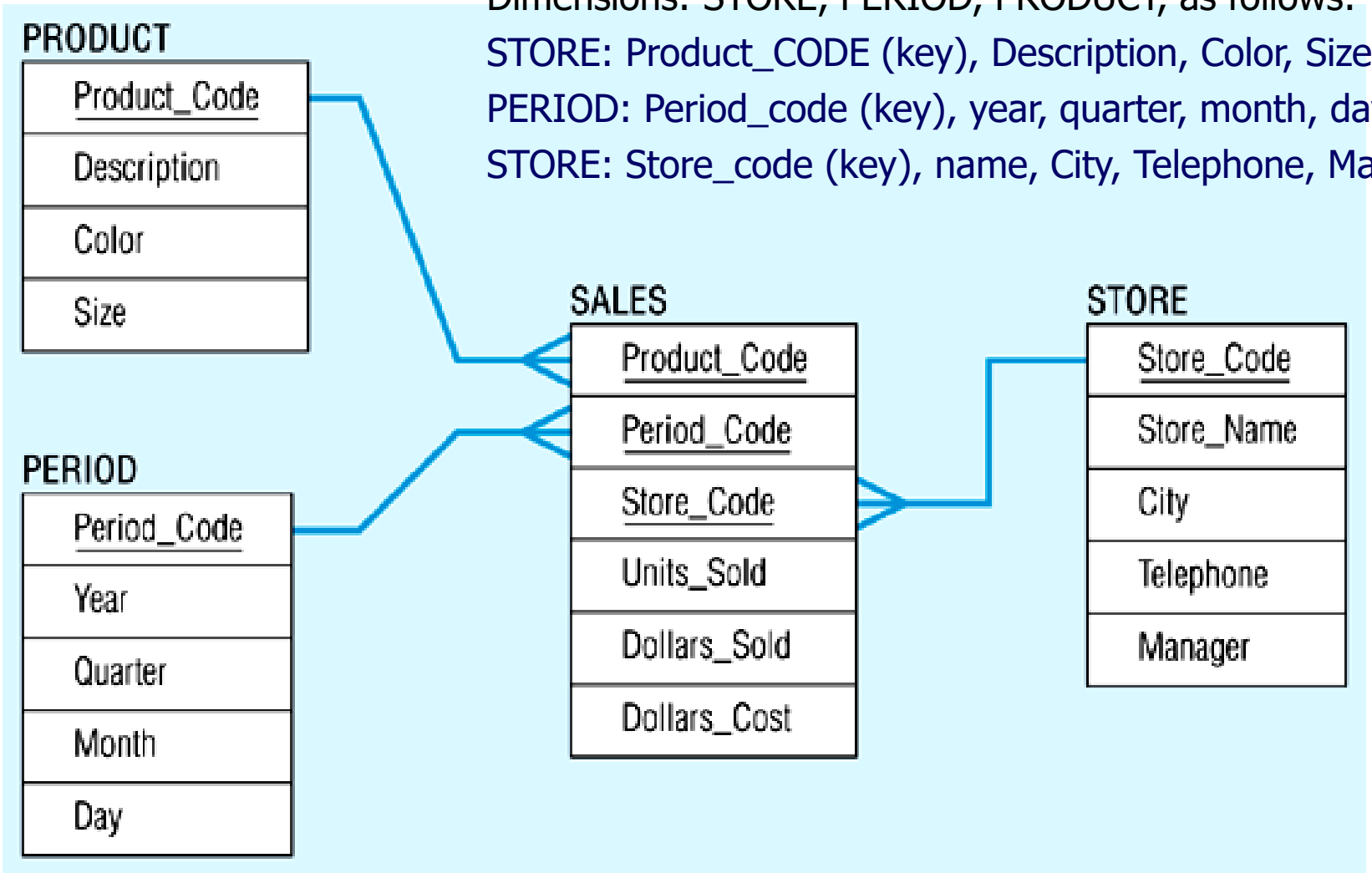
Star Schema Example

Dimensions: STORE, PERIOD, PRODUCT, as follows:

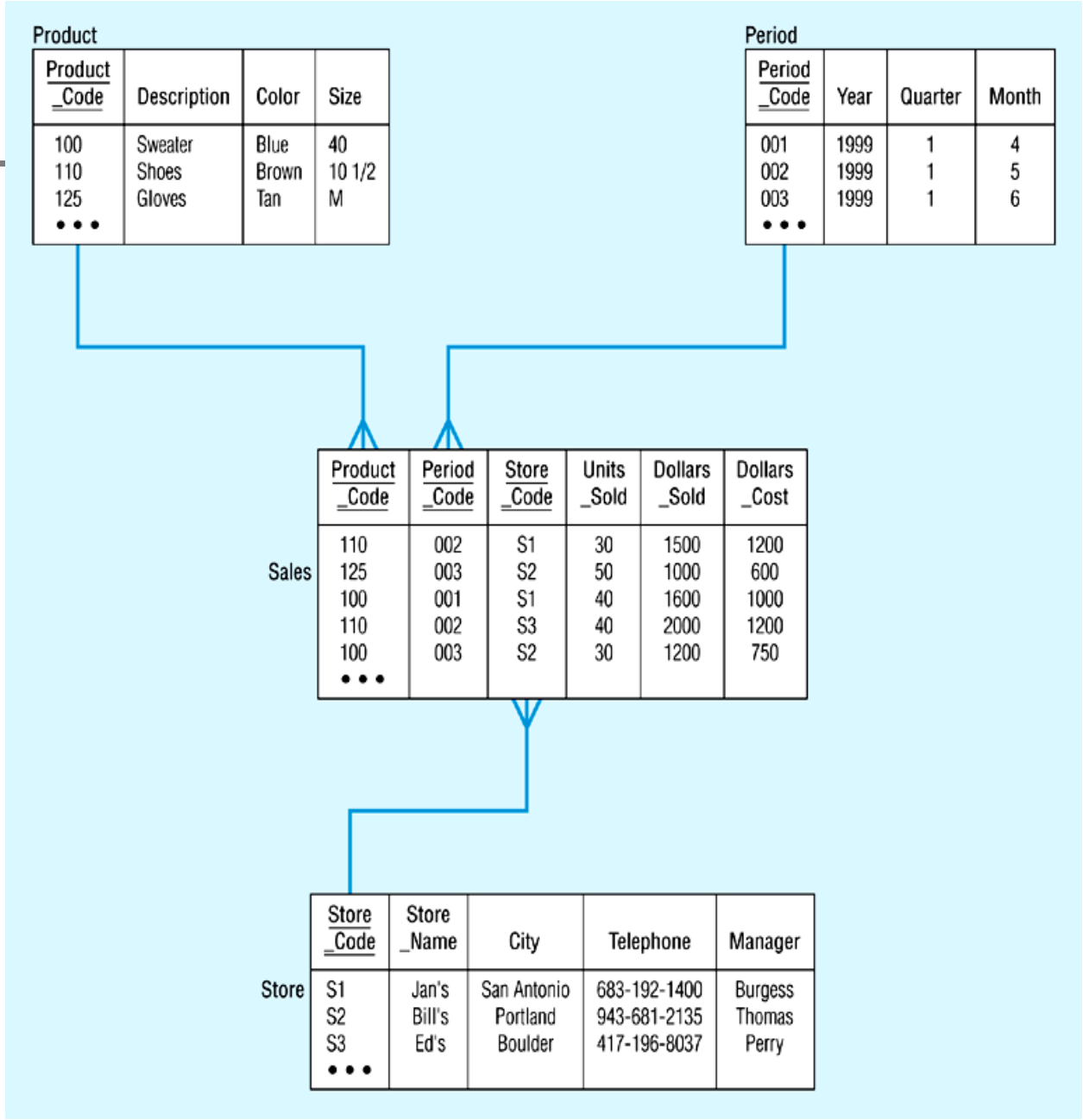
STORE: Product_CODE (key), Description, Color, Size

PERIOD: Period_code (key), year, quarter, month, day

STORE: Store_code (key), name, City, Telephone, Manager



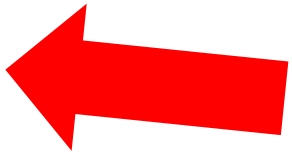
Star Schema with Sample Data



Operations of the multidimensional model on the logical (conceptual) level with SQL

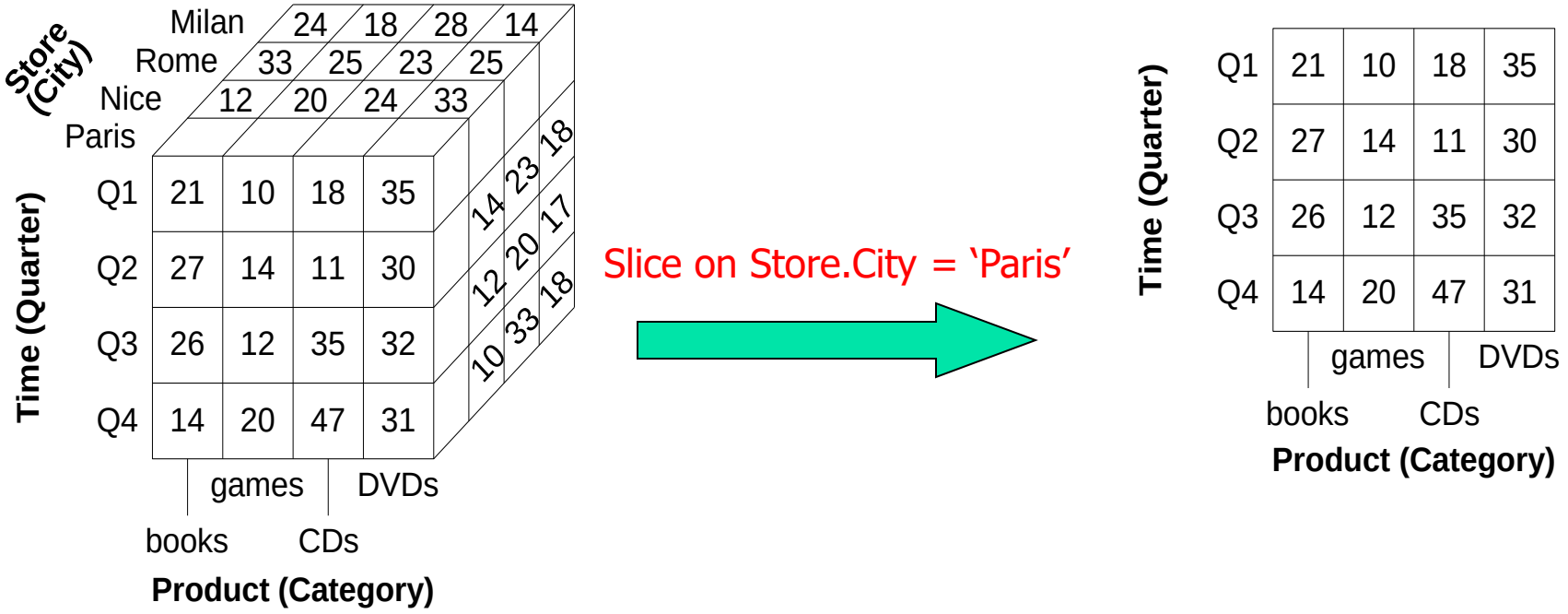
Operations in Multidimensional Data Model

- Selection (Slice)
- Projection
- Aggregation (roll- up)
- Navigation (drill down)



Selection (slices)

- Performs a selection on one dimension of a cube, resulting in a subcube (in practice, select tuples or rows)



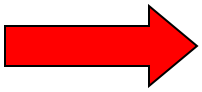
In practice, select tuples

- Select students with gpa higher than 3.3 from S1:

$\sigma_{gpa>3.3}(S1)$

S1

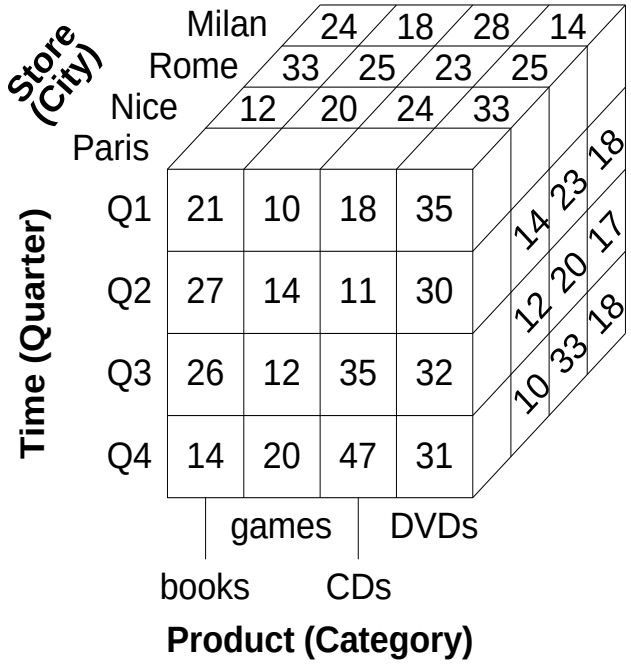
sid	name	gpa
50000	Dave	3.3
53666	Jones	3.4
53688	Smith	3.2
53650	Smith	3.8
53831	Madayan	1.8
53832	Guldu	2.0



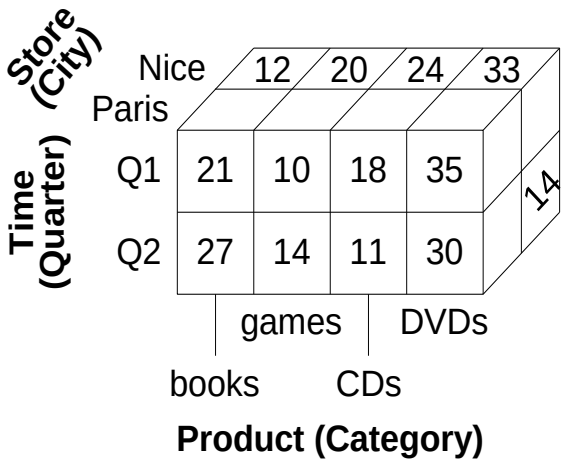
sid	name	gpa
53666	Jones	3.4
53650	Smith	3.8

Projection

- Defines a selection on two or more dimensions, thus again defining a subcube (in practice, select columns)



Dice on Store.Country = 'France' and Time.Quarter= 'Q1' or 'Q2'



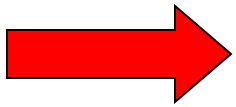
In practice, select columns

Project name and gpa of all students in S1:

$\Pi_{name, gpa}(S1)$

S1

Sid	name	gpa
50000	Dave	3.3
53666	Jones	3.4
53688	Smith	3.2
53650	Smith	3.8
53831	Madayar	1.8
53832	Guldu	2.0



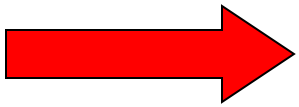
name	gpa
Dave	3.3
Jones	3.4
Smith	3.2
Smith	3.8
Madayan	1.8
Guldu	2.0

Combine Selection and Projection

- Project name and gpa of students in S1 with gpa higher than 3.3:

$$\Pi_{name,gpa}(\sigma_{gpa>3.3}(S1))$$

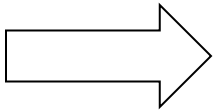
Sid	name	gpa
50000	Dave	3.3
53666	Jones	3.4
53688	Smith	3.2
53650	Smith	3.8
53831	Madayan	1.8
53832	Guldu	2.0



name	gpa
Jones	3.4
Smith	3.8

Eliminating Duplicates

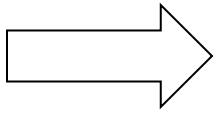
```
SELECT   DISTINCT category
FROM     Product;
```



Category
Gadgets
Photography
Household

Compare to:

```
SELECT   category
FROM     Product;
```



Category
Gadgets
Gadgets
Photography
Household

Ordering the Results

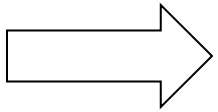
```
SELECT   pname, price, manufacturer
FROM     Product
WHERE    category='gizmo' AND price > 50
ORDER BY  price, pname;
```

Ties are broken by the second attribute on the ORDER BY list, etc.

Ordering is ascending, unless you specify the DESC keyword.

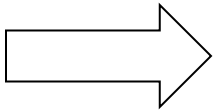
PName	Price	Category	Manufacturer
Gizmo	$19.99	Gadgets	GizmoWorks
Powergizmo	$29.99	Gadgets	GizmoWorks
SingleTouch	$149.99	Photography	Canon
MultiTouch	$203.99	Household	Hitachi

```
SELECT   DISTINCT category
FROM     Product
ORDER BY category;
```



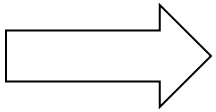
?

```
SELECT   Category
FROM     Product
ORDER BY  PName;
```



?

```
SELECT   DISTINCT category
FROM     Product
ORDER BY PName;
```



?

Keys and Foreign Keys

Company

CName	StockPrice	Country
GizmoWorks	25	USA
Canon	65	Japan
Hitachi	15	Japan

Key

Product

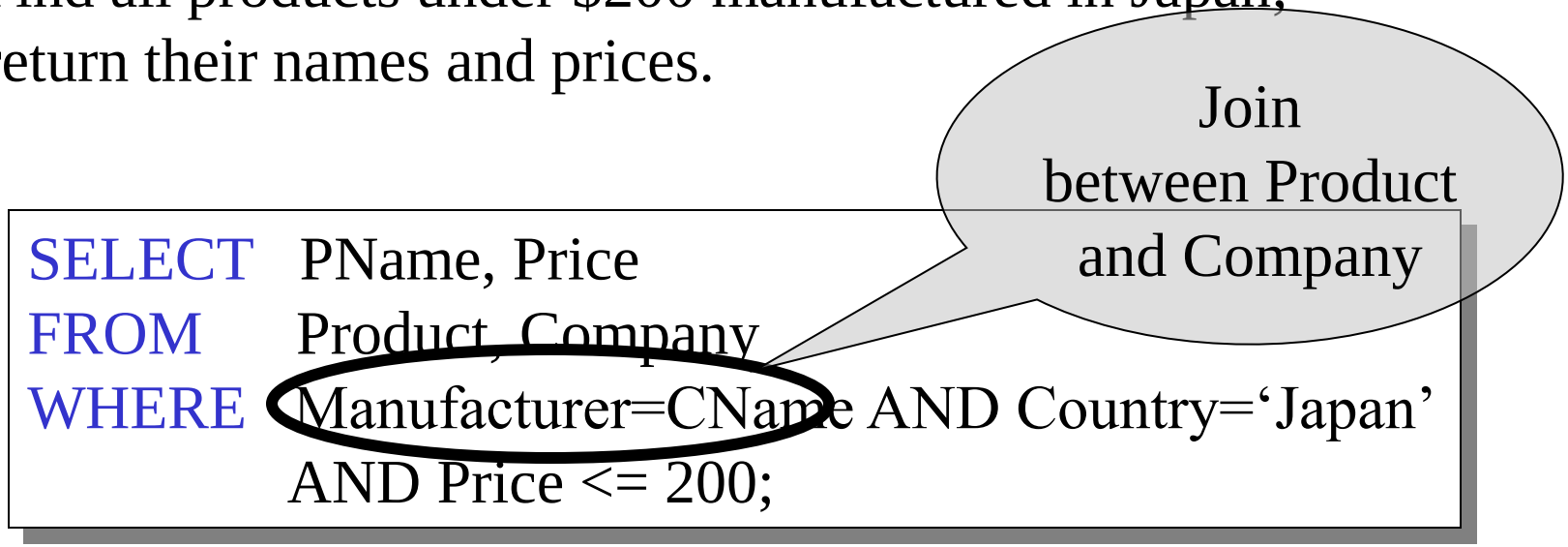
PName	Price	Category	Manufacturer
Gizmo	$19.99	Gadgets	GizmoWorks
Powergizmo	$29.99	Gadgets	GizmoWorks
SingleTouch	$149.99	Photography	Canon
MultiTouch	$203.99	Household	Hitachi

Foreign key

Joins

Product (pname, price, category, manufacturer)
Company (cname, stockPrice, country)

Find all products under $200 manufactured in Japan; return their names and prices.



Joins

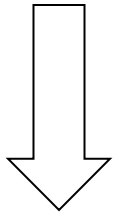
Product

PName	Price	Category	Manufacturer
Gizmo	$19.99	Gadgets	GizmoWorks
Powergizmo	$29.99	Gadgets	GizmoWorks
SingleTouch	$149.99	Photography	Canon
MultiTouch	$203.99	Household	Hitachi

Company

Cname	StockPrice	Country
GizmoWorks	25	USA
Canon	65	Japan
Hitachi	15	Japan

```
SELECT   PName, Price
FROM     Product, Company
WHERE    Manufacturer=CName AND Country='Japan'
         AND Price <= 200;
```



PName	Price
SingleTouch	$149.99

A Subtlety about Joins

Product (pname, price, category, manufacturer)
Company (cname, stockPrice, country)

Find all countries that manufacture some product in the 'Gadgets' category.

```
SELECT   Country
FROM     Product, Company
WHERE    Manufacturer=CName AND Category='Gadgets';
```

Unexpected duplicates

A Subtlety about Joins

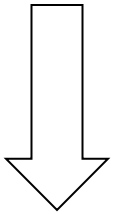
Product

Name	Price	Category	Manufacturer
Gizmo	$19.99	Gadgets	GizmoWorks
Powergizmo	$29.99	Gadgets	GizmoWorks
SingleTouch	$149.99	Photography	Canon
MultiTouch	$203.99	Household	Hitachi

Company

Cname	StockPrice	Country
GizmoWorks	25	USA
Canon	65	Japan
Hitachi	15	Japan

```
SELECT   Country
FROM     Product, Company
WHERE    Manufacturer=CName AND Category='Gadgets';
```



Country
??
??

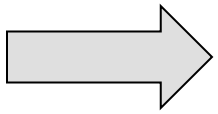
What is the problem ? What's the solution ?

Tuple Variables

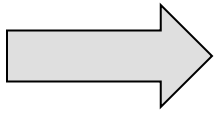
Person(pname, address, worksfor)
Company(cname, address)

```
SELECT   DISTINCT pname, address
FROM     Person, Company
WHERE    worksfor = cname;
```

Which address ?



```
SELECT   DISTINCT Person.pname, Company.address
FROM     Person, Company
WHERE    Person.worksfor = Company.cname;
```



```
SELECT   DISTINCT x.pname, y.address
FROM     Person AS x, Company AS y
WHERE    x.worksfor = y.cname
```

Next week:

a) DW Architecture

b) Logical Model II (hierarchies)

c) SQL Table creation

Questions ?

References:

(a) A Conceptual Poverty Mapping Data Model
Link: https://www.researchgate.net/figure/Key-thematic-layers-for-poverty-spatial-data-modeling_fig2_229724703

(b) Relational Database relationships
https://www.youtube.com/watch?v=C3icLzBtg8I

(c) https://courses.ischool.berkeley.edu/i202/f97/Lecture13/DatabaseDesign/sld002.htm

(d) https://nexwebsites.com/database/database-management-systems/

(e) Acknowledgement – Thanks to http://courses.cs.washington.edu/courses/cse544/ for providing part of this presentation.

(f) Acknowledgement – Thanks to © Silberchatz, Korth and Surdashan for providing part of this presentation.

(e) Malinowski, Elzbieta, Zimányi, Esteban (2008) *Advanced Data Warehouse Design: From Conventional to Spatial and Temporal Applications .* Springer Berlin Heidelberg. Copyright © 2008 Elzbieta Malinowski & Esteban Zimányi