Introduction to data wharehousing

Data collection

Extraction, trasformation and load

Data storage (wharehouses, data marts)

Data Analytics (prescriptive /descriprive/predictive)

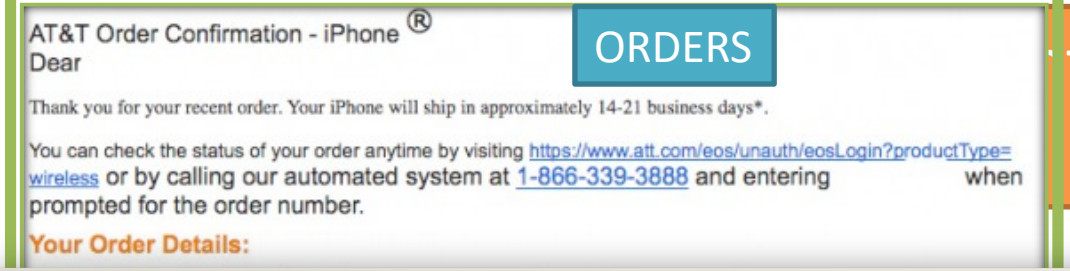
Data Visualization and Reporting

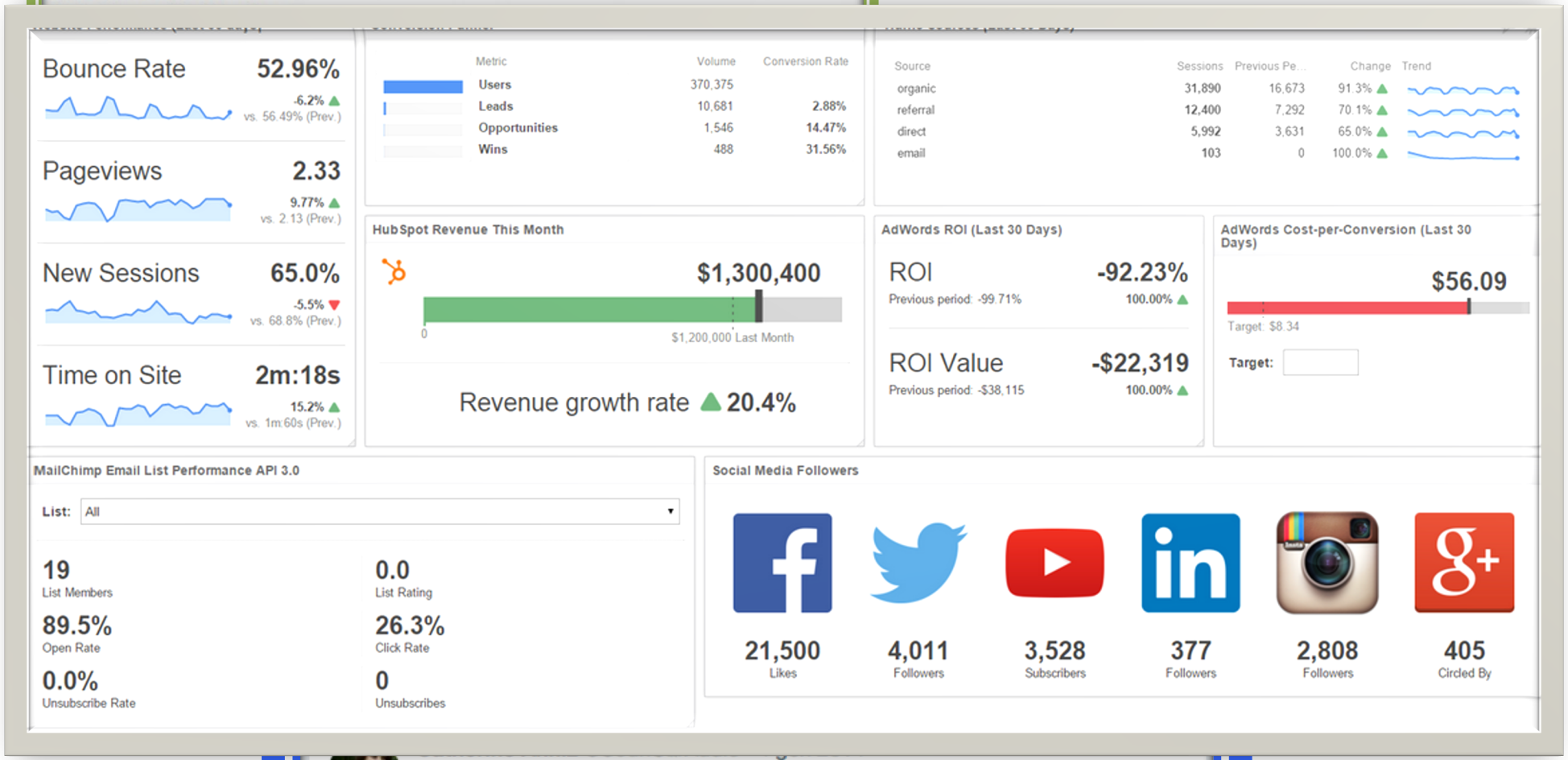
The Business demand for data, information and analytics

- Enterprises today are driven by data: to be more precise, by INFORMATION that can be extracted from data
- Whether BIG DATA or smaller datasets, it requires **a lot of work** before it is actually something useful
- Raw data is incomplete, inconsistent, unformatted, riddled with errors: it is unpalatable to business persons who need to make decisions
- Raw data needs integration, cleaning, design modeling, architecting and other, before it can be transformed in useful information
- In the next lessons will treat the problem of how to **integrate, clean and manage** the data before they can be transformed into INFORMATION

ORDERS

tegration,





And you want to see it all in a nice way

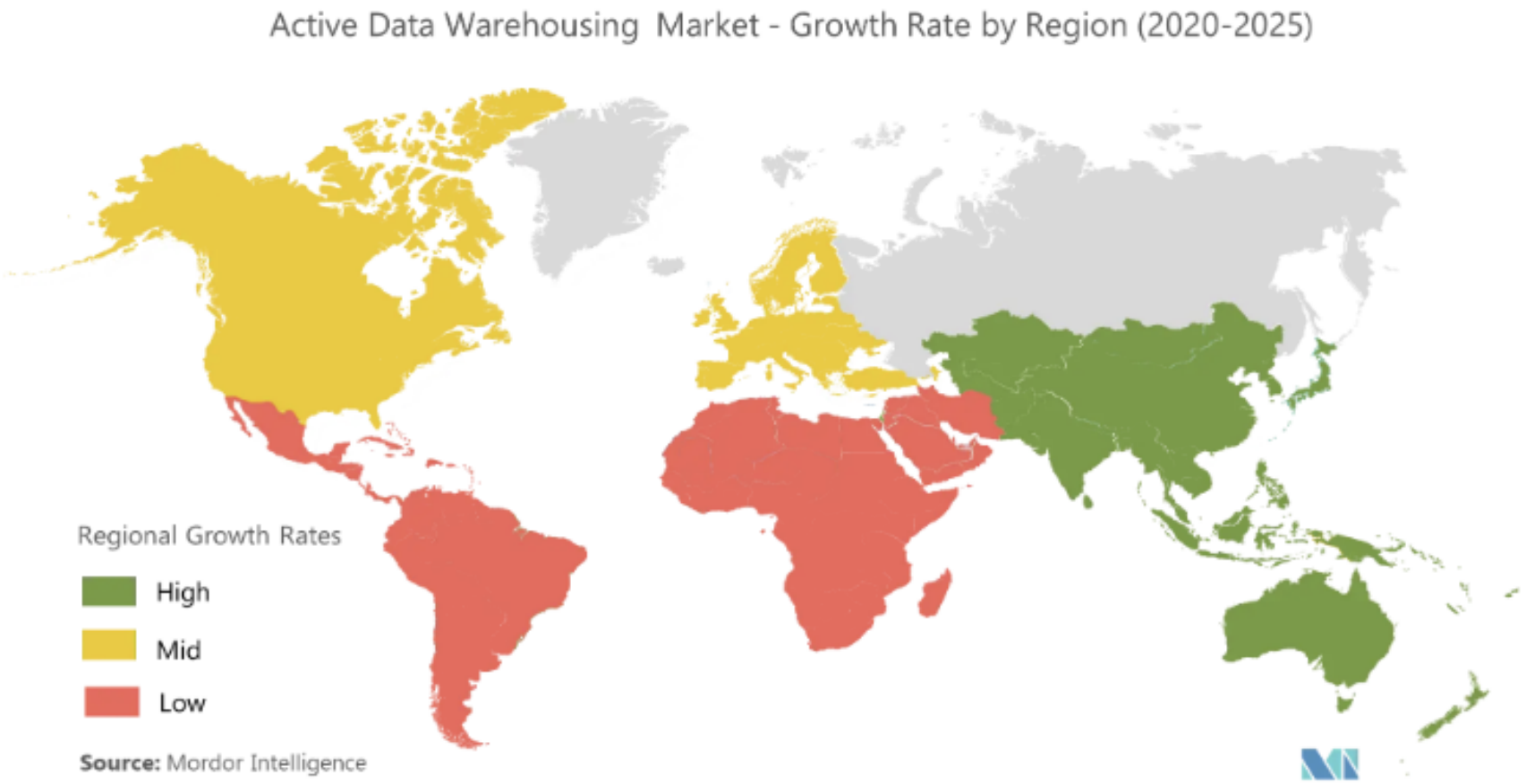
What is a Data Warehouse?

- A Data Warehouse is a collection of data (=**database**) concerning an organisation, used in support of management decisions.
- It is designed for **query and analysis** rather than for **transaction processing** (such as traditional OLTP – on line transaction processing - systems)
- Usually contains historical data derived from transaction data, but can include data from other sources.

Why organizations need DW?

- Organisation may have **many** operational (=for daily operation) databases .
- The different databases are (usually) not synchronised (means that they are not linked and there might be discrepancies). Example: different POS (points of sale), or different types of data (sales, personnel..)
- Management requires an integrated, high-level, company-wide view of all data.
- Data Warehouse separates **informational data**, that can be used for management decisions, from daily operational data.
- Data can be **summarised** as required for management (not relevant details can be omitted).

By geographic area



Use case example: a Regional Health Care (RHC) Group

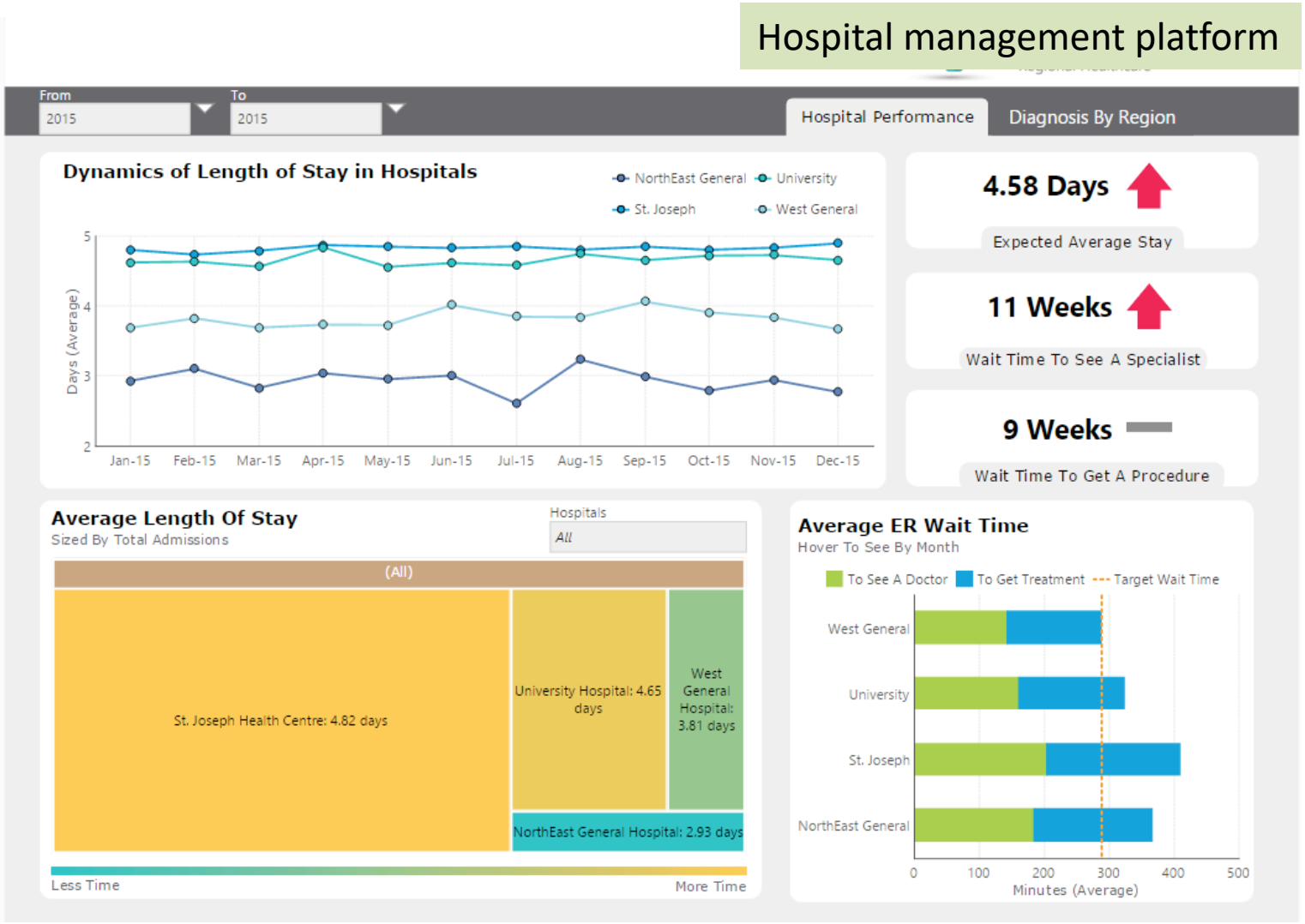
- A RHC organisation may have its data spread across many separate *operational databases*:
- **A health care group** consists of many campuses (formally independent hospitals)
  - Each campus has its own **database for equipment** and minor assets
  - Major assets data is stored on a separate **central database**.
  - Each campus keeps its own **patients database**
  - Each campus employs its own administrative and general (cleaners, gardeners etc.) staff, hence each campus has a separate **payroll database**
  - Doctors and consultants work across the campuses, so there is a **separate database** for them
  - Other data, such as timetables, work rosters, petty cash expenses, etc. are stored in (e.g.) Microsoft Outlook files, spreadsheets and small, local PC databases such as Microsoft Access.

Large organizations need wharehousing the most

- **A large, geographically separate organisation may have hundreds of such 'small' databases**! (e.g., supermarket chains, hotel chains..)
- ✓ A Data Warehouse collects (*copies*) all of this data into a **single (virtual) location**, combines it and puts it into a format for analysing and querying. The information provided from the data warehouse is used to predict trends and help in high-level decision making.
- ✓ The Data Warehouse is separate to the many operational databases in the organisation and should not be used (e.g.,) to "look up who is on duty next Thursday evening" - that information comes from the **operational databases**.
- ✓ Rather, it should provide a high-level, aggregated view of the business: "are our customers happy with our products" "do our POS perform in a comparable way?" "are there products that are best sold in given periods/POS?"

FOR EXAMPLE, THE RHC GROUP WOULD LIKE TO OBTAIN THINGS LIKE THIS…

Hospital management platform



BUT, BEFORE WE CAN EXPLOT DATA IN THIS WAY, WE NEED TO IDENTIFY, COLLECT, CLEAN AND INTEGRATE DATA IN A

DATABASE

DATABASE??????

- …Do you know:
  what a DATABASE is ?
  What is an operational database?

DBs for the non-techies (1)

- A *database* is a digital **collection of data** that is organized so that its contents can easily be accessed, managed, and updated.
- Access to these data is usually provided by a "**database management system**" (DBMS), that is, a computer *software* that allows users to interact with one or more databases and provides access to all of the data contained in the database
- In DBs, data are organized in **Tables**

Table

- TERMINOLOGY: "A *table* is the primary unit of **physical storage** for data in a database."[1]
- It is also a "**logical**" structure: a way of organizing data
- Usually a database contains **more than one table**.

1) Stephens, R.K. and Plew. R.R., 2001. *Database Design*. SAMS, Indianapolis , IN.

Table (example)

Name	Company	Phone Number	E-mail Address
Vedat Diker	CLIS/UMD	(301) 405 9814	vedat@umd.edu
Bugs Bunny	Acme, Inc.	(123) 555 9876	bugs@acme.com
Will E. Coyote	Acme, Inc.	(123) 555 9821	will@acme.com

Tables have NAMES to identify the entities they describe

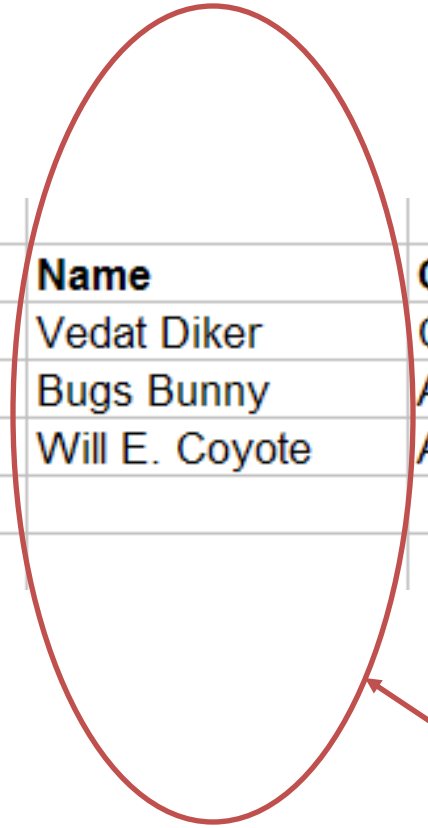
NAME of the Table
(also called ENTITY)

Name	Company	Phone Number	E-mail Address
Vedat Diker	CLIS/UMD	(301) 405 9814	vedat@umd.edu
Bugs Bunny	Acme, Inc.	(123) 555 9876	bugs@acme.com
Will E. Coyote	Acme, Inc.	(123) 555 9821	will@acme.com

They have Fields (Columns) to identify descriptors

Customers

Name	Company	Phone Number	E-mail Address
Vedat Diker	CLIS/UMD	(301) 405 9814	vedat@umd.edu
Bugs Bunny	Acme, Inc.	(123) 555 9876	bugs@acme.com
Will E. Coyote	Acme, Inc.	(123) 555 9821	will@acme.com



a field

Fields are identified by a label or field name (e.g. Name, Company…).
Fields are also called **ATTRIBUTEs** or **DESCRIPTORS** or **DIMENSIONS** or **KEYs or FEATURES** (they can be used interchangeably)

Name	Company	Phone Number	E-mail Address
Vedat Diker	CLIS/UMD	(301) 405 9814	vedat@umd.edu
Bugs Bunny	Acme, Inc.	(123) 555 9876	bugs@acme.com
Will E. Coyote	Acme, Inc.	(123) 555 9821	will@acme.com

In this example, *fields* describe the «entity» Customer.
We say that customers have a *name*, belong to a *company*, have a *phone number* and an *e-mail.*
Deciding what are the relevant entities in a company Database and what are the relevant fields **is a conceptual task** that requires knowledge of a specific Business or business line

Record (Row)

Customers

Name	Company	Phone Number	E-mail Address
Vedat Diker	CLIS/UMD	(301) 405 9814	vedat@umd.edu
Bugs Bunny	Acme, Inc.	(123) 555 9876	bugs@acme.com
Will E. Coyote	Acme, Inc.	(123) 555 9821	will@acme.com

a record

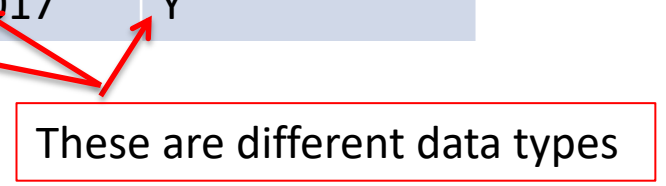
A **record** is a row of the table where fields (attributes, keys) have VALUES
E.g., Name=Bugs Bunny

A record represents a real-world instance of the entity type described by the table (e.g., in this example, the entity type is «customers» and each row is a real-world customer.

Data Types in tables

- Data types describe the type of values a field can take.
- Data types can be:
  - Alphanumeric (Text)
  - Numeric (Number, Currency, etc.)
  - Date/Time
  - Boolean (attributes with only two values, e.g.: Yes/No, true/false, 0/1..)
- Data types impose constraints on the values in a cell. e.g., dates must be expressed in a valid date format. But, **data entry error may occur.** Constraints help detecting these errors

ID	Name-of-product	Order date	availability
37000876	IPhone 7 pink	10/09/2017	Y



Primary Key

Customers

Customer ID	Name	Company	Phone Number	E-mail Address
6273	Vedat Diker	CLIS/UMD	(301) 405 9814	vedat@umd.edu
3245	Bugs Bunny	Acme, Inc.	(123) 555 9876	bugs@acme.com
1324	Will E. Coyote	Acme, Inc.	(123) 555 9821	will@acme.com

primary key field

Primary key is a **unique** identifier of records in a table.
There cannot be records with the same value for the primary key.
Primary key values may be generated manually or automatically.

Primary Key (2)

Roles (Performances)

Actor/Actress	Movie	Character Name
Keanu Reeves	Matrix	Neo
Laurence Fishburne	Matrix	Morpheus
Carrie-Anne Moss	Matrix	Trinity
Keanu Reeves	Sweet November	Nelson Moss
Charlize Theron	Sweet November	Sara Deever
Charlize Theron	Waking Up in Reno	Candy Kirkendall
Laurence Fishburne	Othello	Othello
Ted Lange	Othello	Othello

primary key fields

A primary key can also consist of more than one field (and is not necessarily an ID). What matters is that it is UNIQUE!!
e.g., actors/actresses might have the same name, but the tuple “actor, movie” is (hopefully) unambiguous

In class exercise

- You run a nation-wide bike rental service, with several departments (sales, repairs, customer service, administrative offices, HRM..)
- Can you identify one or two «tables» describing the business entities relevant for this service?
- For each table (entity):
  - Show the name of the described entity type (name of the table)
  - Identify the relevant fields (descriptors, attributes) of the table
  - Specify the type (number, date, string, ..)
  - Identify the primary key
  - From which company department should the data be provided?

Primary keys can be used to connect tables

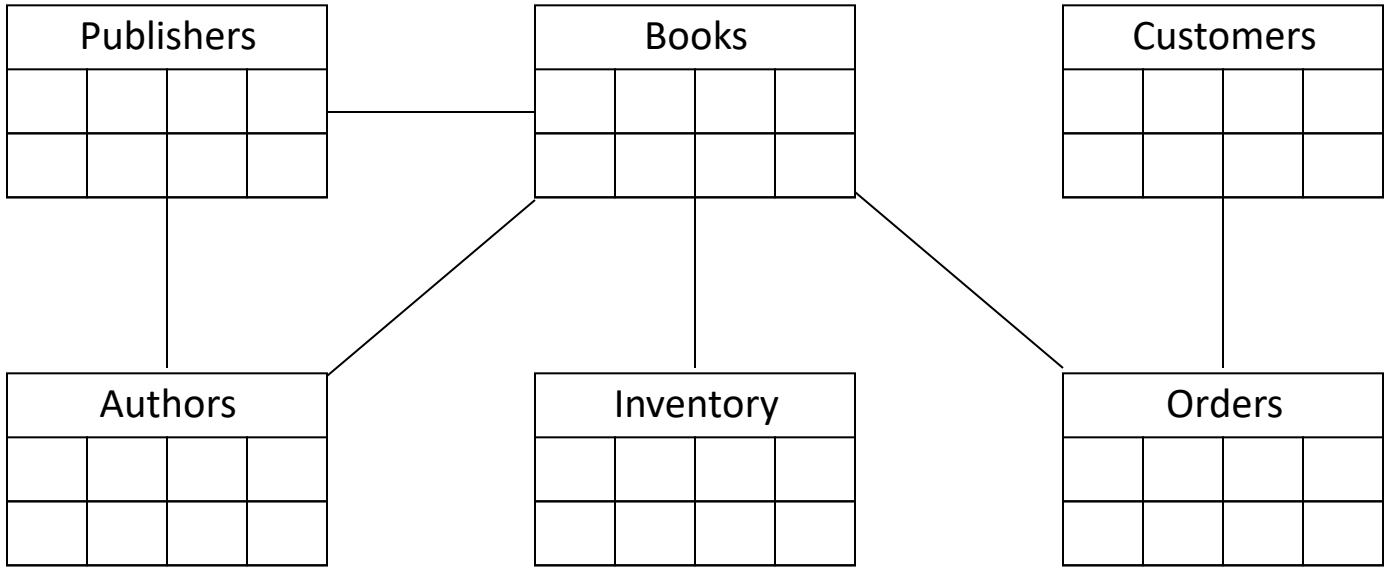
In a database, often we need to describe many entities, each entity has a separate table

E.g., you can have customers, sales, products, point of sales, employees

But in order to answer specific questions, we need to connect data from different tables

So, databases usually are made of multiple connected Tables, each representing a “viewpoint” (entity, concept) on the data.

The example below refers to a chain of libraries



Why we need multiple tables, why not everything in a single table?

- Example: in a **movie database** you would like to know, e.g., all the titles of movies directed by a given director.
- Say that we want to store all the info in an entity table called *Director*
- Since every director has directed a variable number of movies, and since tables must have **a FIXED number of fields**, we can't store a director's movie names (e.g. MOVIE1, MOVIE2…) in the director table!!
- Similarly, if we create a unique table MOVIE, since movies have multiple actors, we can't store the name of all actors (ACTOR1, ACTOR2..) in the Movie table, because the number of actors is variable!
- So we need **multiple tables for multiple entities**, and we need **a method to connect this information**, in order to answer questions like: *How many movies has directed Ridley Scott? What are the actors of Gladiator?*
- Primary and foreign keys are the solution!

To answer the question: «**what are the movies directed by Readly Scott**?» we must (through some dedicated code):

1. Select in table «Directors» the record with Name=Ridley Scott
2. Retrieve the primary key **Director ID** (235)
3. Select in table «Movies» all records with foreign key Director ID=235
4. For these records, retrieve the filed *Title* and create the list.

primary key field

Director ID	Name	Date of Birth	Place of Birth	Biography
785	John Frankenheimer	19-Feb-30	New York, NY	Born in New York and raised in Queens, …
235	Ridley Scott	30-Nov-37	South Shields, UK	Education: Royal College of Art, London…
976	James Foley	28-Dec-53	Brooklyn, NY	Attended the USC Film School…

relationship

child table

Movies

Movie ID	Title	Director ID	Genre	…
4532	Gladiator	235	Action	
8357	Swwet and Lowdown	497	Comedy	
7465	Confidence	976	Drama	

foreign key field

TO CONNECT TABLES: Foreign key is defined in a second table, but it refers to the primary key or a unique key in the first table.

It is a way of connecting information referring to the same item

How does, in practice, this connection process occurs?

- Using query languages (specific programming languages to query databases)
- e.g., Structured Query Language (SQL) is extensively used in data mining, data storage, and OLTP systems.
- Let's say we want to show book titles along with their authors (i.e., the author's first name and last name). The book titles are stored in the *books* table, and the author names are stored in the *authors* table.
- In our SQL query, we'll join these two tables by matching the author id column from the books table b.author_id and the id column from the authors table a.id (*a.id* the primary key of the author table, the *b.author:id* is the foreign key)

Code

```
SELECT b.id, b.title, a.first_name, a.last_name
FROM books b
INNER JOIN authors a
ON b.author_id = a.id
ORDER BY b.id;
```

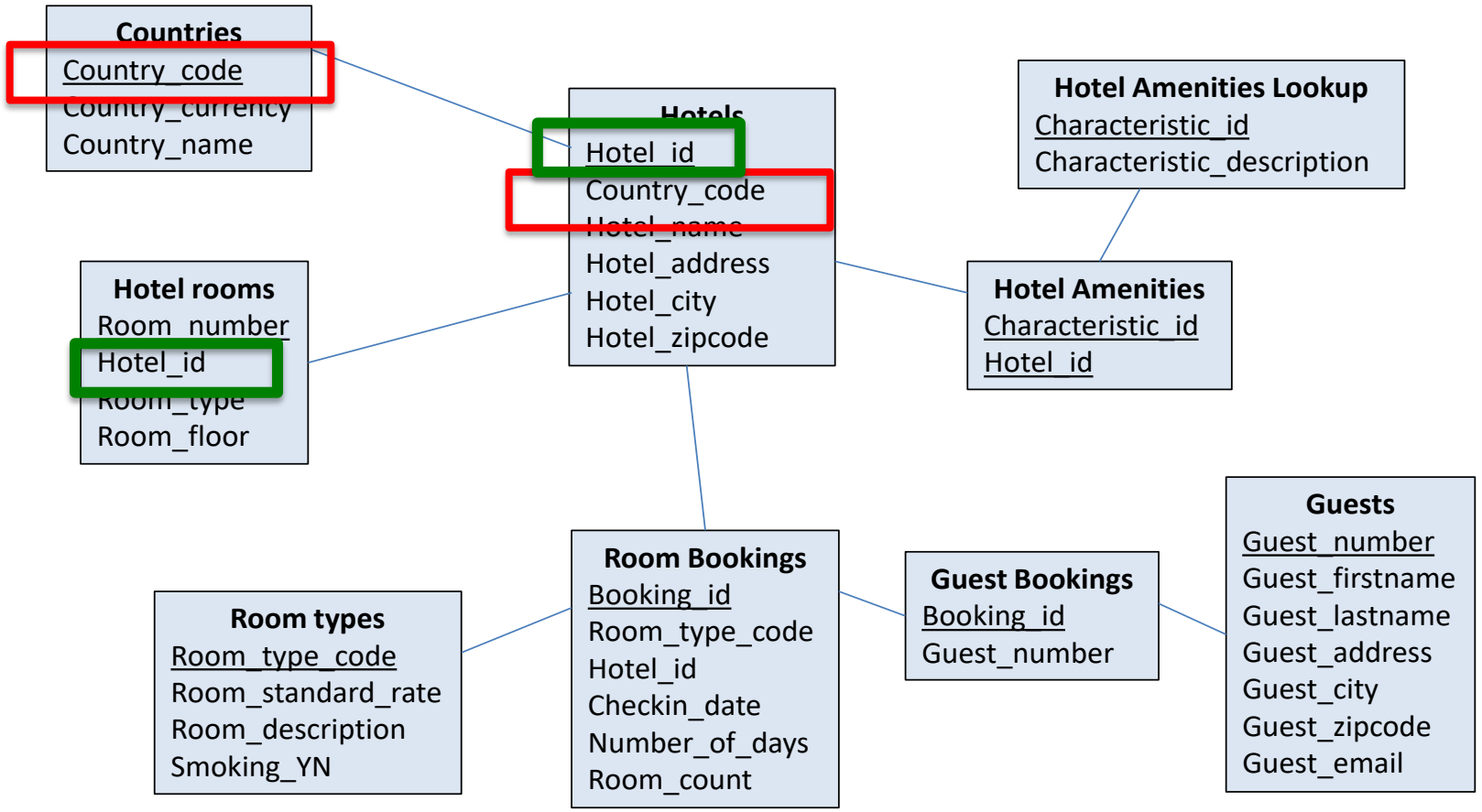
Lucky you, there is GTP!

- GTP (3,4) , given a database, is able to convert natural language queries into SQL code and allows querying databases without writing dedicated software
- Watch this video https://www.youtube.com/watch?v=azFXWEi9Fmk&ab_channel=C-Phrase%3ANaturalLanguagetoSQL%21
- No need to worry about SQL coding (to some extent), but you still need to understand the concept of entity linking for query answering

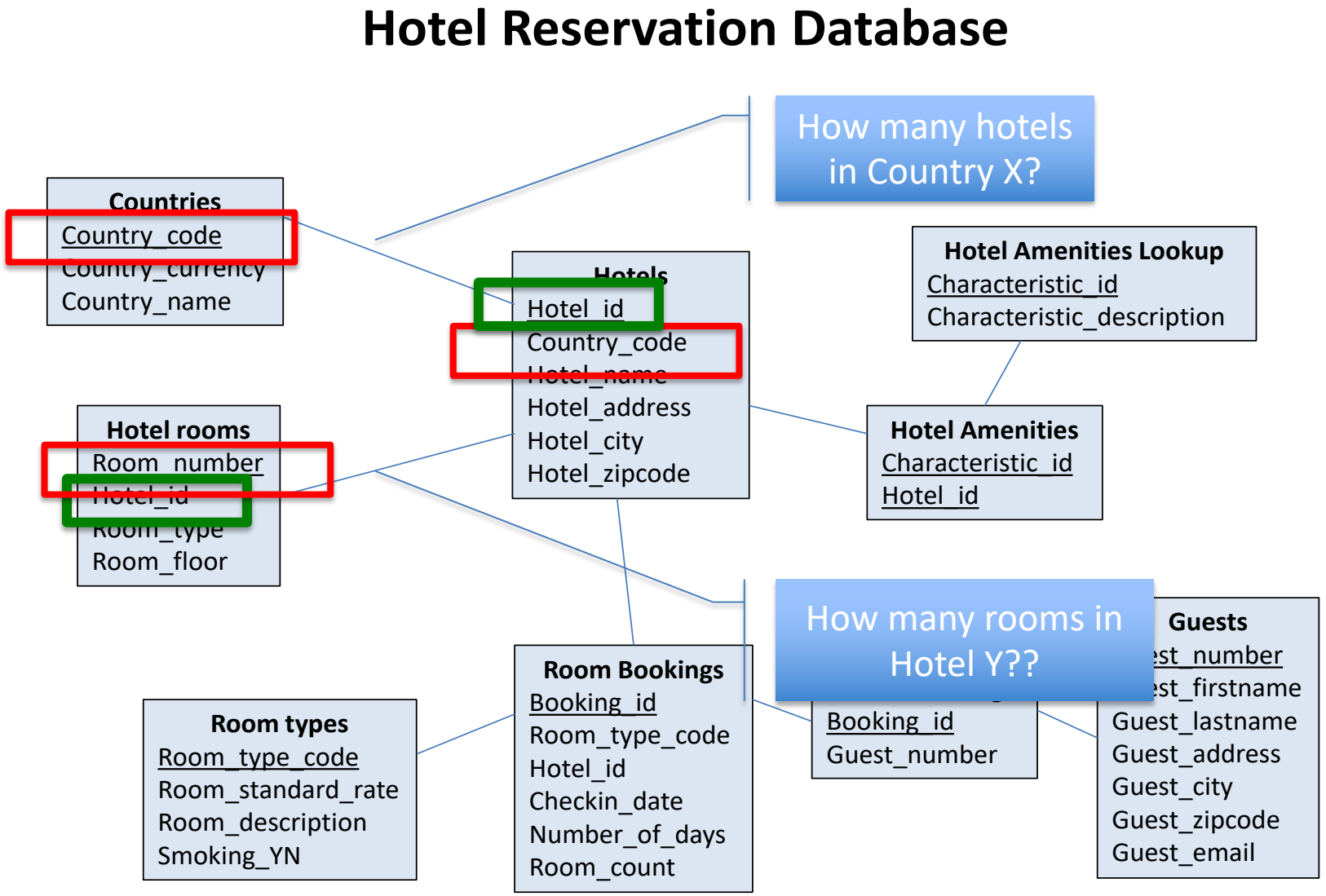
Another example with multiple tables (primary keys are underlined)

Hotel Reservation Database

Relations between records in tables are determined by the primary/foreign keys



“Common” keys are used to answer queries



Summary of terminology so far

Tables describe entities

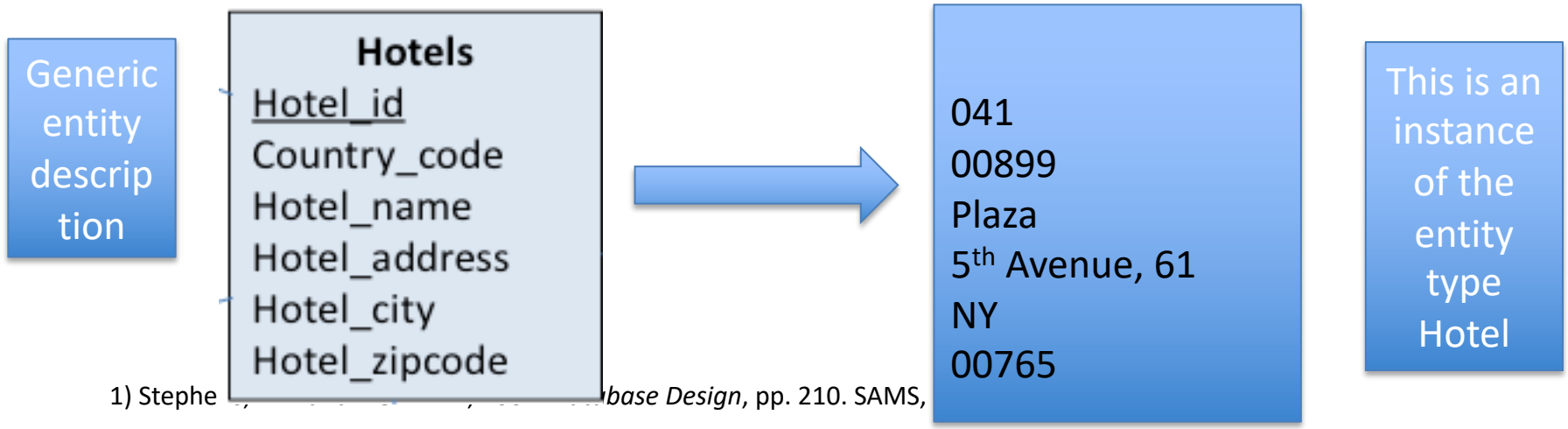
- TERMINOLOGY: An *entity* is a “business object” that represents a group, or category of data.[1]

- Example: hotel, hotel_room, guest..

1) Stephens, R.K. and Plew. R.R., 2001. *Database Design*, pp. 21. SAMS, Indianapolis , IN.

Record (Instance, Tuple)

- TERMINOLOGY A single, specific occurrence of an entity is a *record*. Other terms for an instance are ***instance*** and ***tuple***.[1]
- Hotel: Plaza
- Instances are “valued” entities!



1) Stephe … *base Design*, pp. 210. SAMS, …

Attributes (fields, primary /foreign keys)

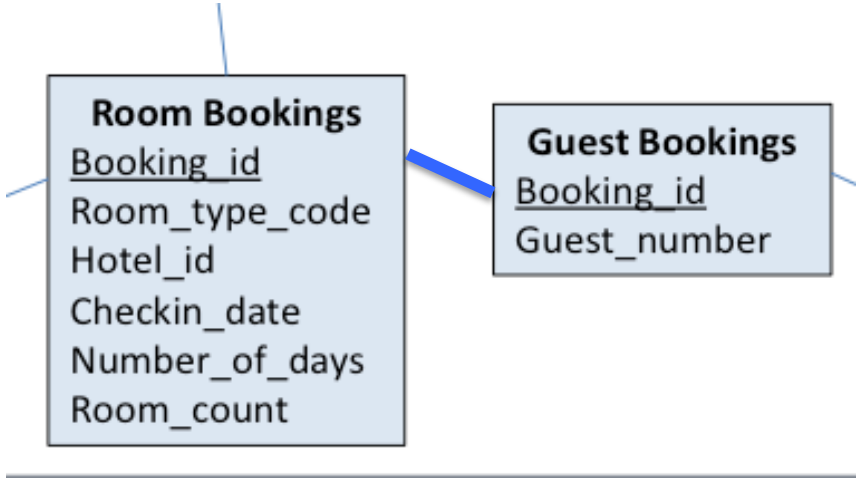
- TERMINOLOGY: An *attribute* (or field or key or descriptor..) is a sub-group of information within an entity.[1]
- Ex: Country_Code is an attribute of the entity type Hotel

- As we said, an attribute can be **a primary key or a foreign key**. In the above example, Hotel_id is primary, country_code is foreign. Primary keys are UNIQUE for each record of a given entity table. For example, Hotel_id is a primary key for the entity table *Hotels*, and is a *foreign key* for the entity table *Hotel Rooms*.

1) Stephens, R.K. and Plew. R.R., 2001. *Database Design*, pp. 21. SAMS, Indianapolis , IN.

Relationship

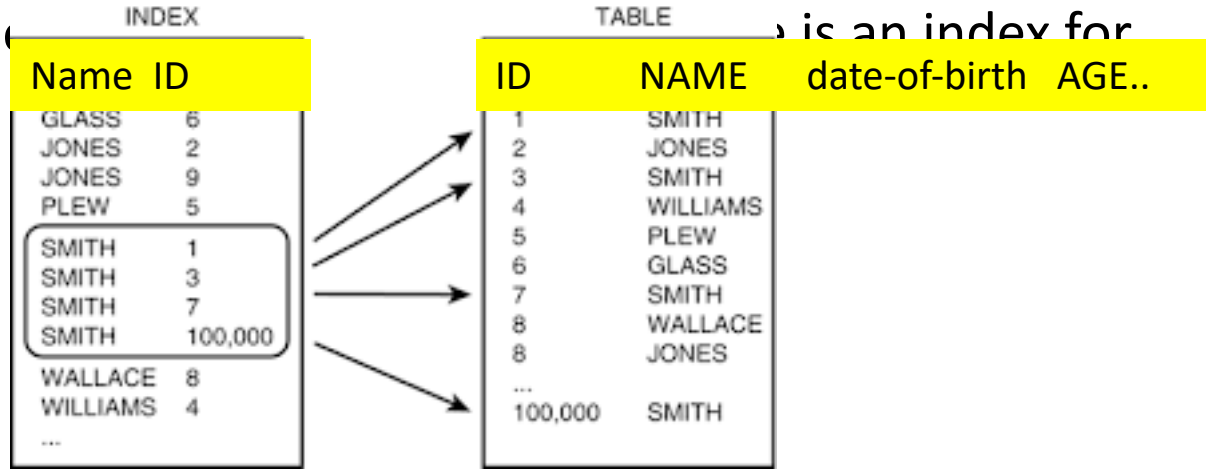
- TERMINOLOGY: A *relationship* is a ***link*** that relates two entities that share one or more attributes (keys, fields).
- Example: **Guest_booking** and **Room_booking** have the same attribute Booking id (since one would like to know which guest reserved a given room, or which room has been reserved for a given guest)



Though often **implicit**, relationships have a *semantics* and a *direction*, e.g.,
Guest – (has booked) →Room
Room – (has been booked by) →Guest

Indexes

- **TERMINOLOGY: Indexes** are data structures (again, tables!) used **for fast look-up in tables** (they are a mechanism to quickly retrieve information in tables)
- E.g. say that you want to know how many Guests have the "Name" attribute = SMITH, **without searching sequentially all the database**
- An index is a **pointer** to the locations (record IDs) of the DB where **the required attribute has the required value**. An index is a bit like an address..
- Clearly, since you have many fields (attributes), you cannot organize your database in alphabetic [illegible] is an index for (nearly) each field.



Summary of terminology

We have

- Entities (*tables*)
- Records (*lines* in tables)
- Attributes (*names of columns* in tables)
- Relationships (two tables are related if there are attributes that are *primary keys* in a table and *foreign keys* in others)
- Indexes (for each attribute name, indexes are list of possible attribute values with pointers to records in tables where that value is found)

So we have these tables (DBs)..

But, what we can actually DO with them??

We want to answer queries!!

Operations

Attribute

Attribute values

Entity name

- What are the main operations in a DB?
- **DELETE, UPDATE, INSERT** (self explanatory operations)
- The **SELECT** operator is used to select those records with given values of one or more attributes (e.g. *SELECT from SALES_DATA where PART_NAME= iPhone 6 and YEAR= 2016*)
- The **JOIN** operator, is used to merge values from different tables:

Employee table

LastName	DepartmentID
Rafferty	31
Jones	33
Heisenberg	33
Robinson	34
Smith	34
Williams	NULL

Department table

DepartmentID	DepartmentName
31	Sales
33	Engineering
34	Clerical
35	Marketing

Joint these 2 tables to learn that Mr. Rafferty works at sales dept.

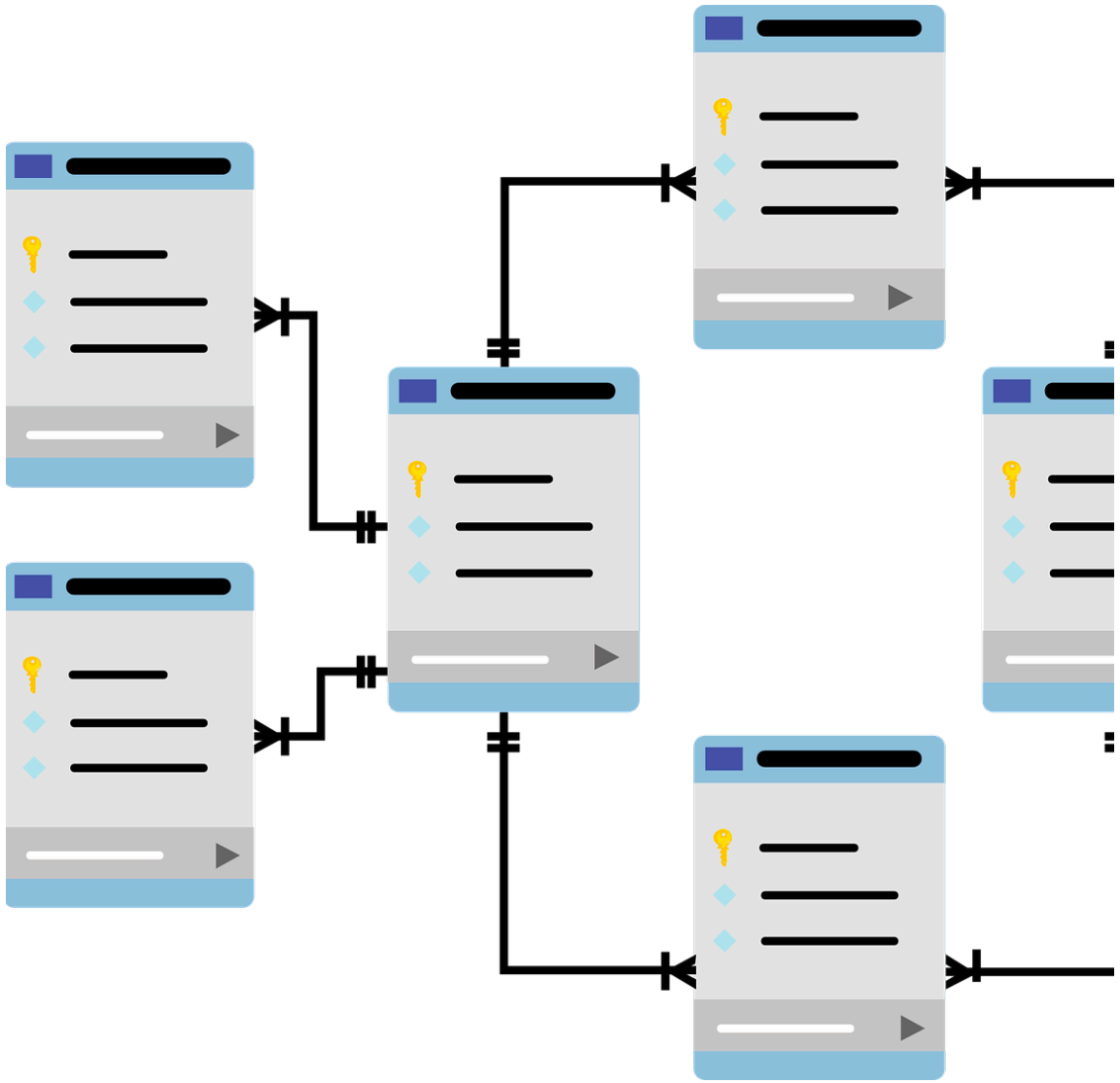
Another Join example

Summary so far

- Data concerning a business are collected in ***tables***.
- The relevant data of a business are organized in many tables, offering different and detailed views of the business (e.g. reservation, restaurant and services, billing, customer care..)
- Tables are linked together via their attributes (*primary* and *foreign* keys). Links are called **relationships** and usually have a (hidden) semantics
- **Operations** (select, join, delete..) and **indexes** are used to QUERY the database and retrieve RELEVANT BUSINESS FACTS (e.g., how many rooms have been reserved on January 2018 ?)
- Usually performing operations on databases need programming languages (e.g. SQL), **but with self-service business analytics you can retrieve facts with very simple interactions** (will see in Labs!!!)
- Or, you can use GTP (but then, you need to share your data ..)

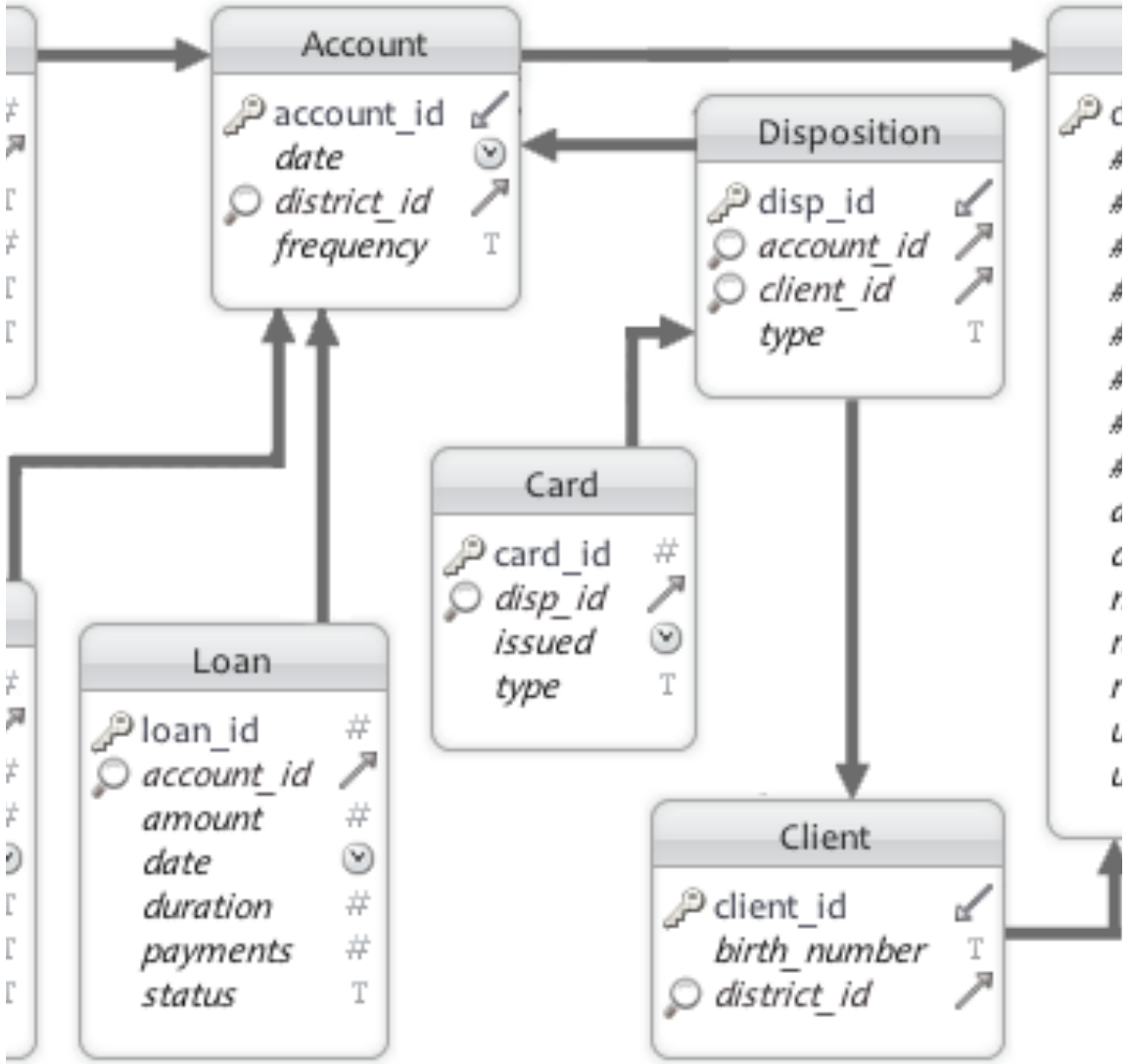
Database schema

- In database terms, a *schema* is the organisation and structure of a database
- A database schema can be represented in a **visual diagram**, which shows the database entities and their relationship with each other.

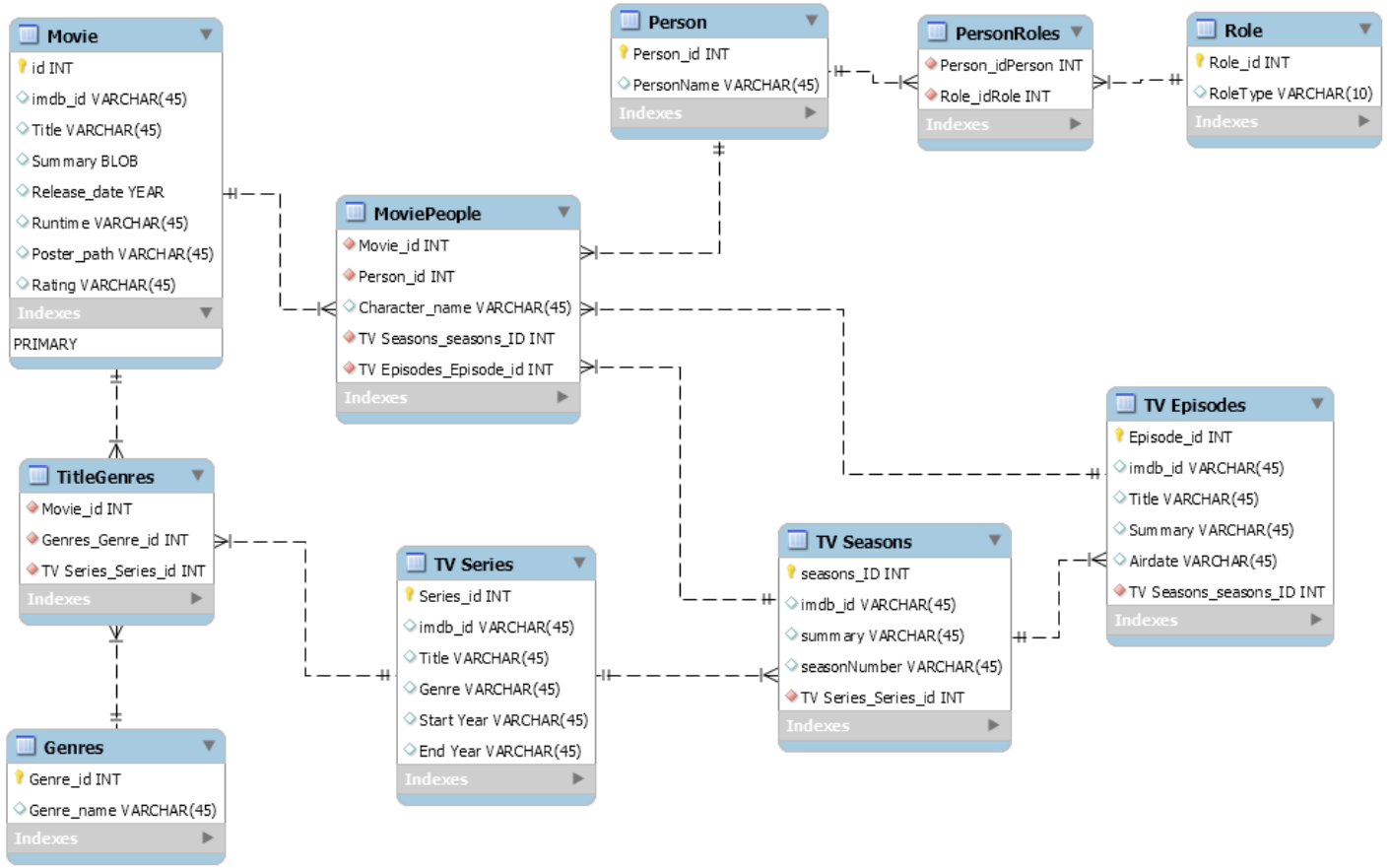


Schema where tables are related by primary-foreign key pairs (Bank database example)

- Entities are those of a Bank: account, loan, disposition, client, credit card..
- Primary keys are those with the «key» symbol
- Foreign keys are those with the magnifying glass



A more complex scheme: movies database



HW 3 (you can start working in class)

- A TV company wishes to develop a database to store data about the **TV series** that the company produces. The database includes information about **actors** who play in the series, and **directors** who direct the **episodes** of the series.
- Actors and directors are **employed** by the company. TV series are divided into **episodes**. Each episode may be **transmitted at several occasions** (timestamps). An actor is hired to participate in a series, but may participate in many series. Each episode of a series is directed by one of the directors, but different episodes may be directed by different directors.
- Develop a **database scheme** of this system (=set of related tables with attributes). 1) Identify *entity types*. 2)Create a table for each entity type 3)Choose *attributes* of the entity sets. 4)Determine which of the attributes can be used as primary keys. 5)Draw connections between tables that are related trough primary/foreign keys

Querying the TV series database

- According to your schema, which tables should be used to answer these types of questions:
  - Which actors play in the series X?
  - In which series does the actor Y participate?
  - Which actors participate in more than one series?
  - How many times has the first episode of the series X been transmitted? At what times?
  - How many directors are employed by the company?
  - Which director has directed the greatest number of episodes?

OLTP and OLAP databases

- We now introduce and compare two types of DB systems:
  - OLTP (on-line transaction processors)
  - OLAP (on-line analytical processors, also called Data Wharehouses)

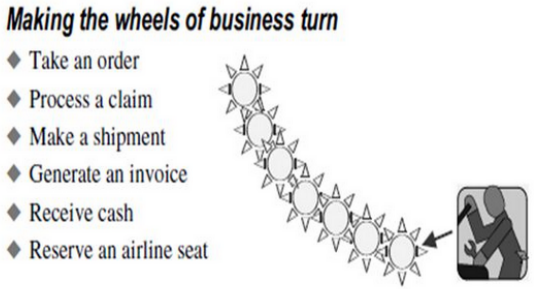
OLTP vrs OLAP (DataWhareh ouses, DW)

- Traditional On Line Transaction Processors (OLTP, introduced in the first lesson!.. Excel-like tables) are *operational* systems tailored for processing **transactional databases**
- A **transactional database** supports business process flows (sales, supply chain, etc.) and is typically an online, real-time system .
- With respect to OLTP, DW (also named OLAP, On-Line Transaction Analytics) are **much more powerful**

OLTP vrs. OLAP (DW)-2

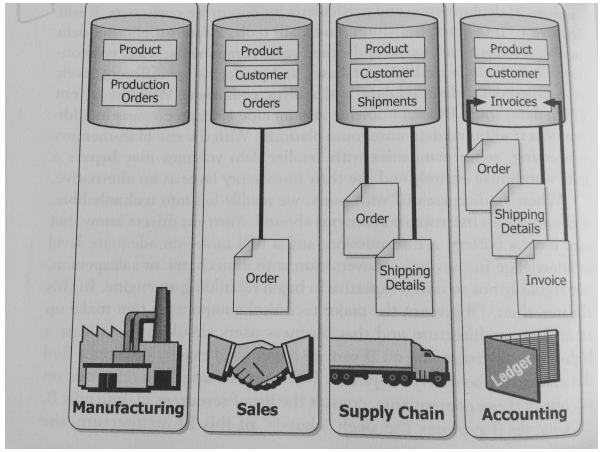
Purpose of data:

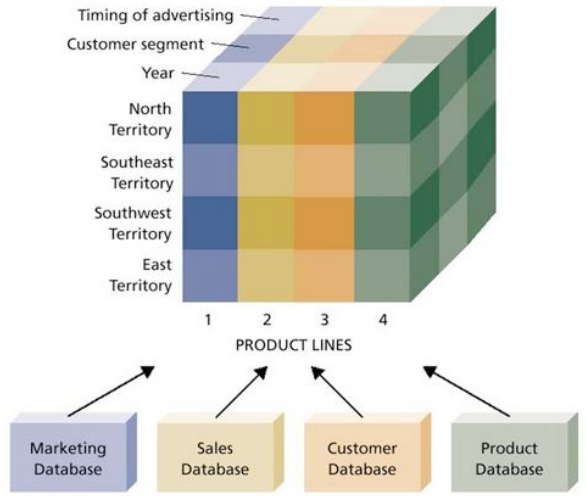
- OLTP: To control and run fundamental **day-to-day business tasks** (e.g., handle guest reservations, room cleaning, payments..)
- OLAP: To help with **planning, problem solving, and decision support**



OLTP vrs. OLAP (DW)-3

- **Source of data**
  OLTP: Operational data; OLTPs are the <u>original</u> source of the data and each system manages a specific transactional database (e.g. shipments, orders..).
- OLAP: OLAP data **comes from the various OLTP Databases + external sources** and are aggregated (also called OLAP cube)



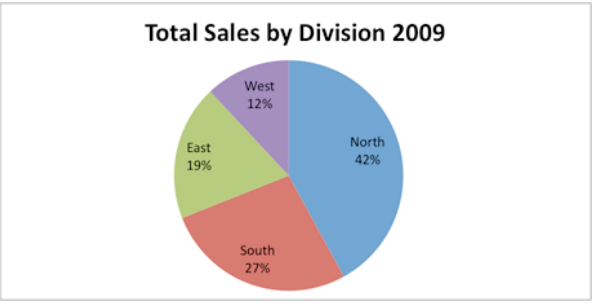


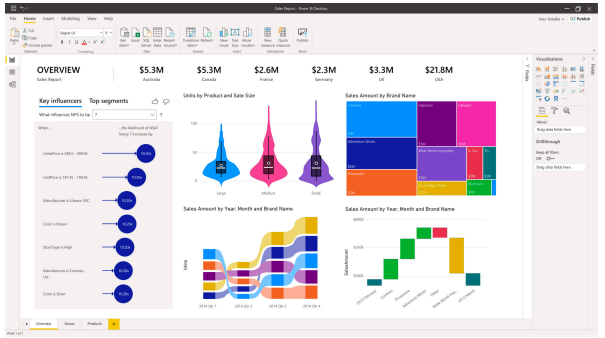
OLTP vrs. OLAP (DW)-4

What the data represent

- OLTP: Reveals a snapshot of ongoing business processes

- OLAP: **Multi-dimensional** views of various kinds of business activities

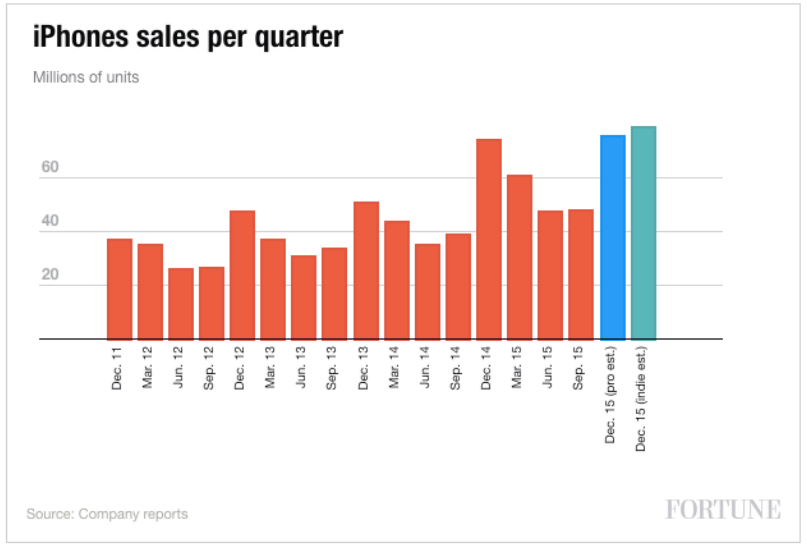




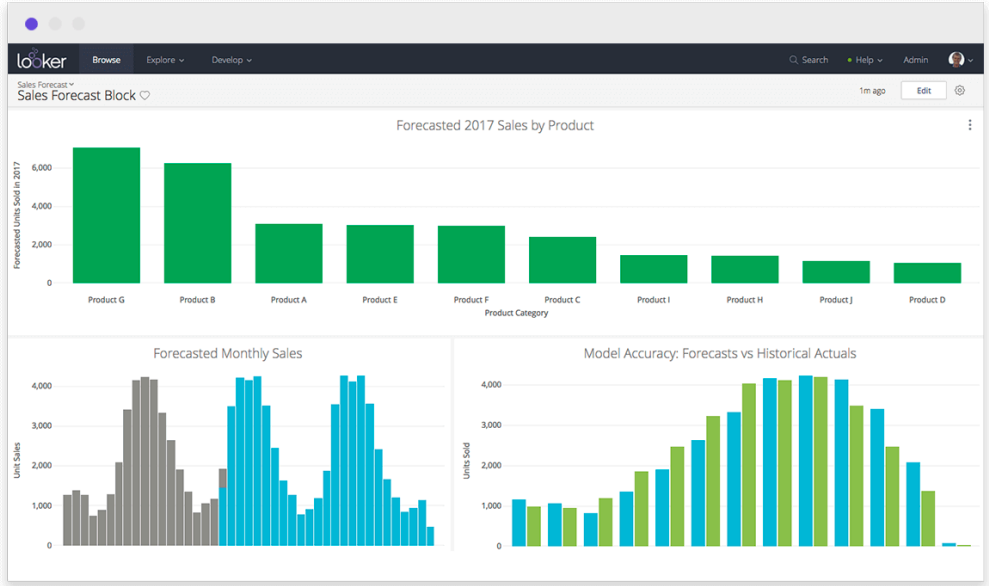
OLTP vrs. OLAP (DW)-5

Queries

- OLTP: Relatively standardized and simple queries; Returning relatively few records (= answers) which are **descriptive** of the current status of a business
- OLAP: Often ***complex queries*** involving aggregation of many data and INFERENCE (**prediction, prescription**)



How many i-Phones sold in this quarter?



How many products of type X can we expect to sell next month in Germany?

OLAP vrs OLTP (DW) –more issues

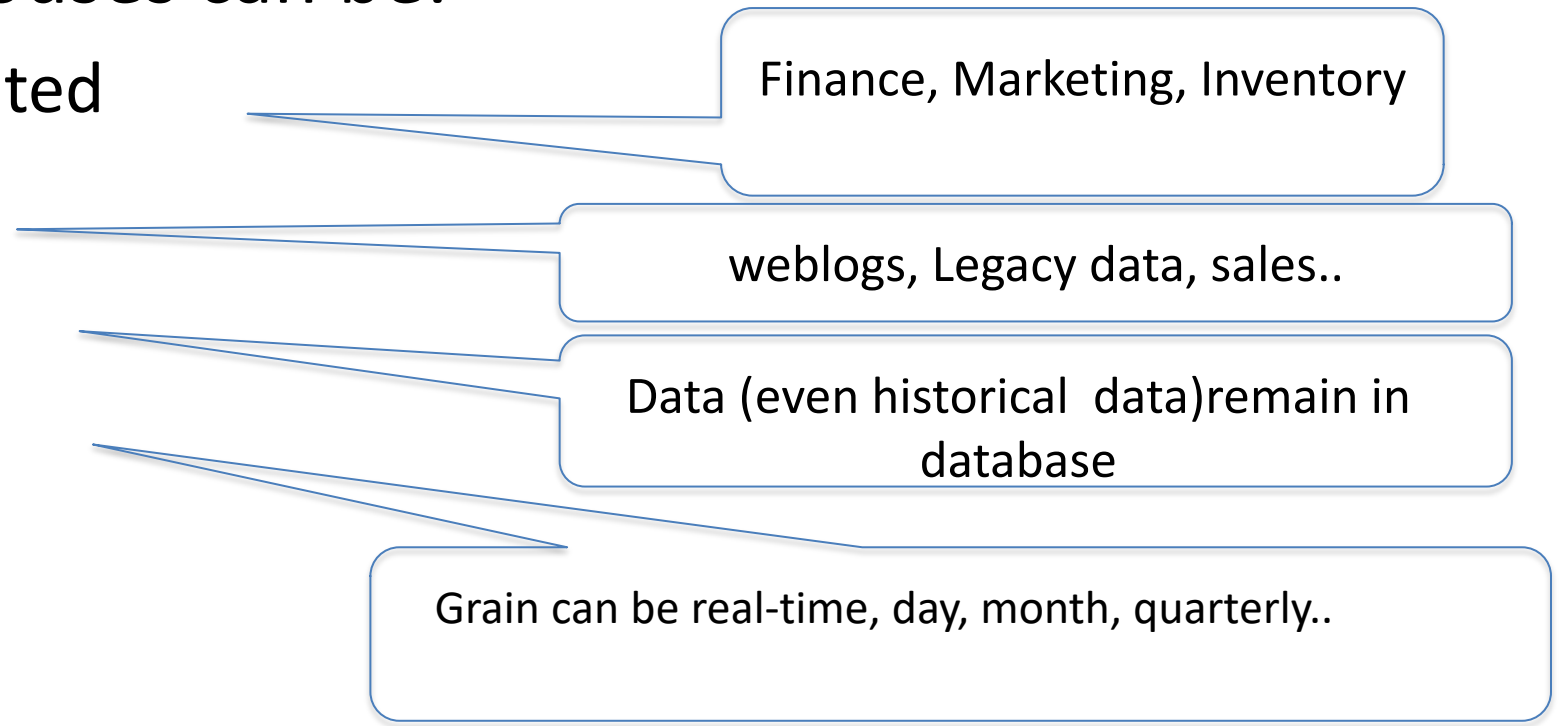
- **Processing Speed**
  OLTP: Typically very fast
  OLAP: Depends on the amount of data involved; Typically needs **Big Data solutions**.

- **Space Requirements**
  OLTP: Can be relatively small if historical data is archived
  OLAP: Larger due to the existence of aggregation structures and history data; requires **more indexes** than OLTP (since more dimensions are available or can be defined)

- **Backup and Recovery**
  OLTP: Backup religiously; operational **data is critical to run the business**, data loss is likely to entail significant **monetary loss** and **legal liability**
  OLAP: Instead of regular backups, some environments may consider simply reloading the OLTP data as a recovery method

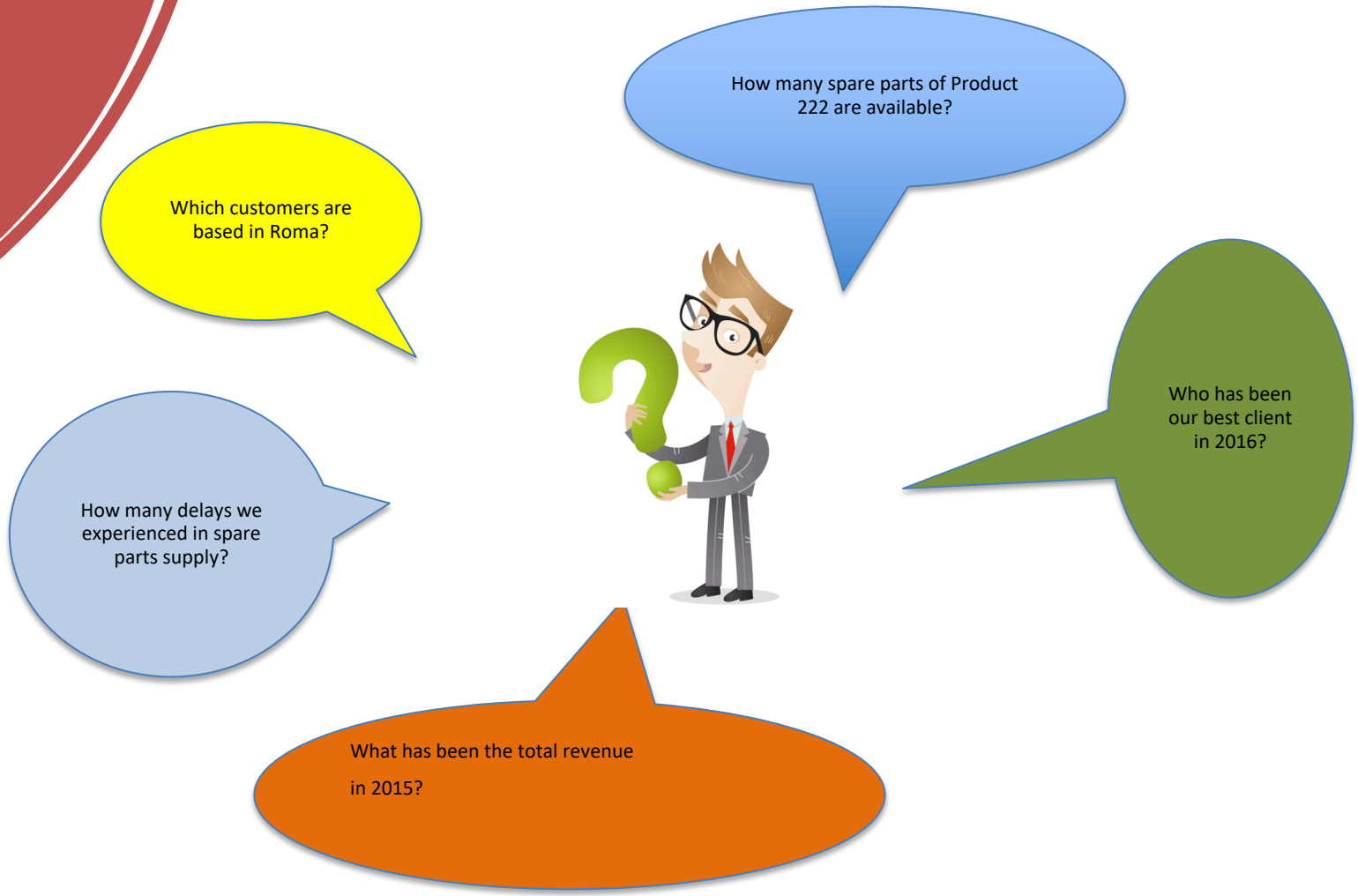
Characteristics of DWs

- Data Wharehouses can be:
  - Subject oriented
  - Integrated
  - Non Volatile
  - Time variant



Querying OLTP and DWs

What kind of queries in an OLTP?



What kind of queries in a OLAP/DW?

Summary OLAP vrs OLTP (DW)

	OLTP	OLAP
users	clerk, IT professional	knowledge worker
function	day to day operations	decision support
DB design	application-oriented	subject-oriented
data	current, up-to-date detailed, flat relational isolated	historical, summarized, multidimensional integrated, consolidated
usage	repetitive	ad-hoc
access	read/write index/hash on prim. key	lots of scans
unit of work	short, simple transaction	complex query
# records accessed	tens	millions
#users	thousands	hundreds
DB size	100MB-GB	100GB-TB
metric	transaction throughput	query throughput, response

In a nutshell..

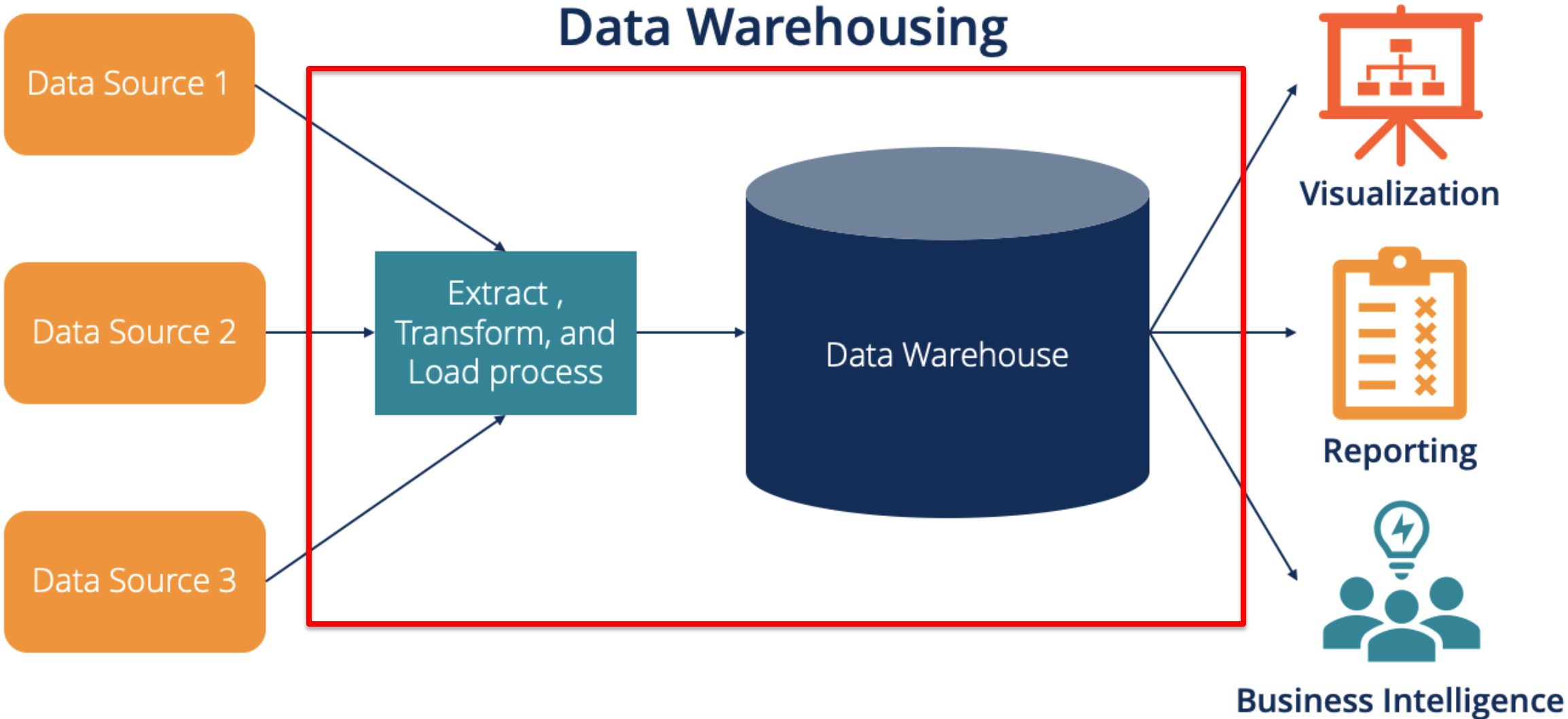
- OLTP Systems are used to *"run"* a business

- The Data Warehouse helps to *"optimize"* the business

Summary so far

- Data Warehouse is a collection of data concerning the organisation used in support of management decisions.
- It is a kind-of database: a data structure organized in tables
- A Data Warehouse allows analytical processing of data (OLAP) for decision support, contrary to operational databases, which support real-time transaction processing (OLTP)

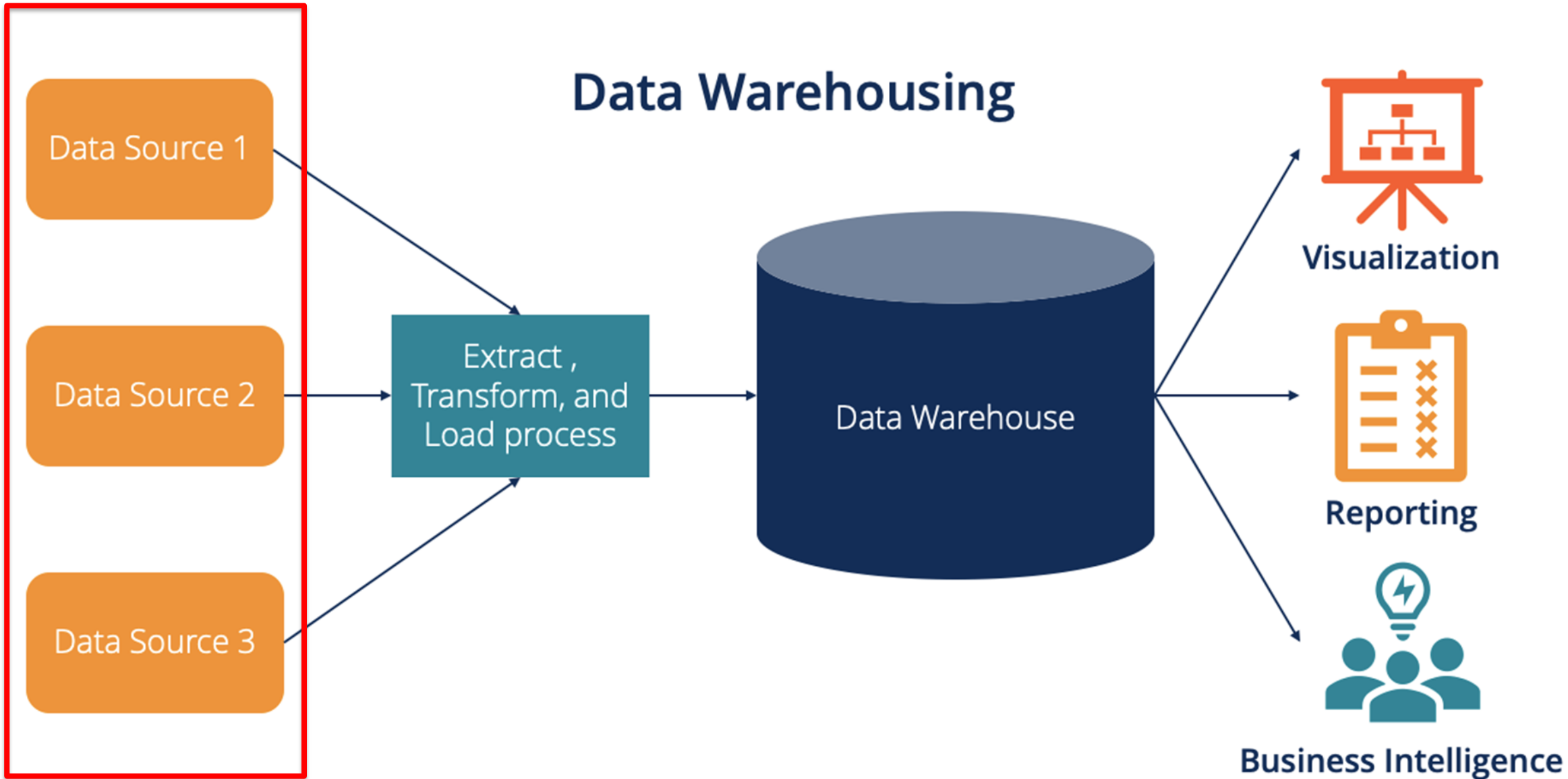
Architecture of a DW



Design aspects of a DW: ETL

1. **Extract:** selecting which data and what for
2. **Transform:** so-called **ETL**: extraction, cleaning, transform and load data
3. **Load:** Store and process data, cretate data Mars, add metadata, aggregate, integrate data

Step 1: Which data and what for?



Which data? Data Sources and Types

- Primarily sources come from legacy data, operational systems
  - Mostly structured and numerical data at the present time. Sales, vendors, transactions..
- **External data** are often included, either purchased from third-party sources, or open source data
- Some types of external data are **unstructured**!
  - Technology exists for storing unstructured data (images, text, sensors) and is becoming more important over time
  - External data (social networks data, user profiles) are also becoming more and more important

Structured vrs. Unstructured data

Structured Data

0.103	0.176	0.387	0.300	0.379
0.333	0.384	0.564	0.587	0.857
0.421	0.309	0.654	0.729	0.228
0.266	0.750	1.056	0.936	0.911
0.225	0.326	0.643	0.337	0.721
0.187	0.586	0.529	0.340	0.829
0.153	0.485	0.560	0.428	0.628

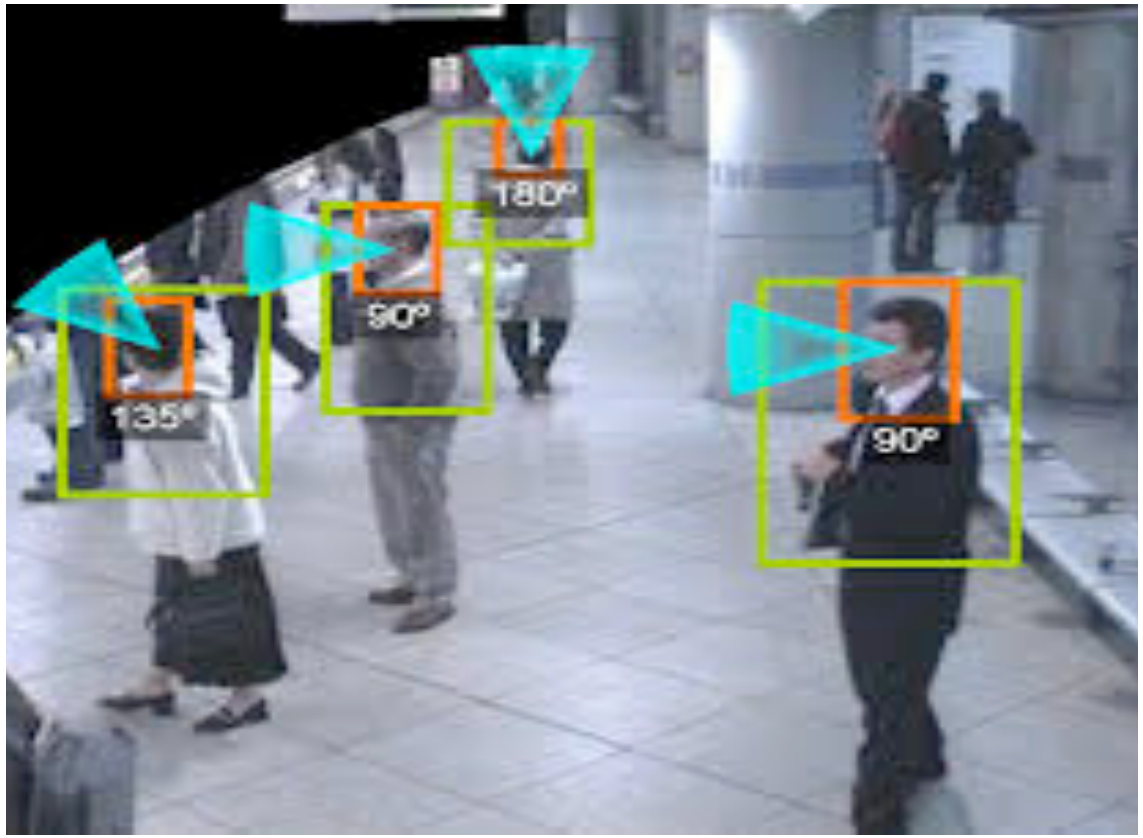
Unstructured Data

What type of unstructured external data data and what for? (2)

- **Social data** (social networks, blogs): to mine user opinions, trending topics, market forecasts
- **Sensors data** (signals from devices e.g. vending machines, packages, wearable devices, sensor networks..): to detect anomalies (remember Magpie vaccines), learn trends..
- **Clickstream data** (cliklogs of web sites): for traffic and e-commerce analysis
- **Environmental data** (geolocations, metereological data): to produce recommendations, supply chain, market forecasts..
- **Images, videos, signals** (medical imaging, landscapes, portraits): to detect anomalies, security, fraud detection..
- **Audio** (speech, sound): to mine opinions, fraud detection, environmental analysis

Example 1: applications of image understanding (people recognition)

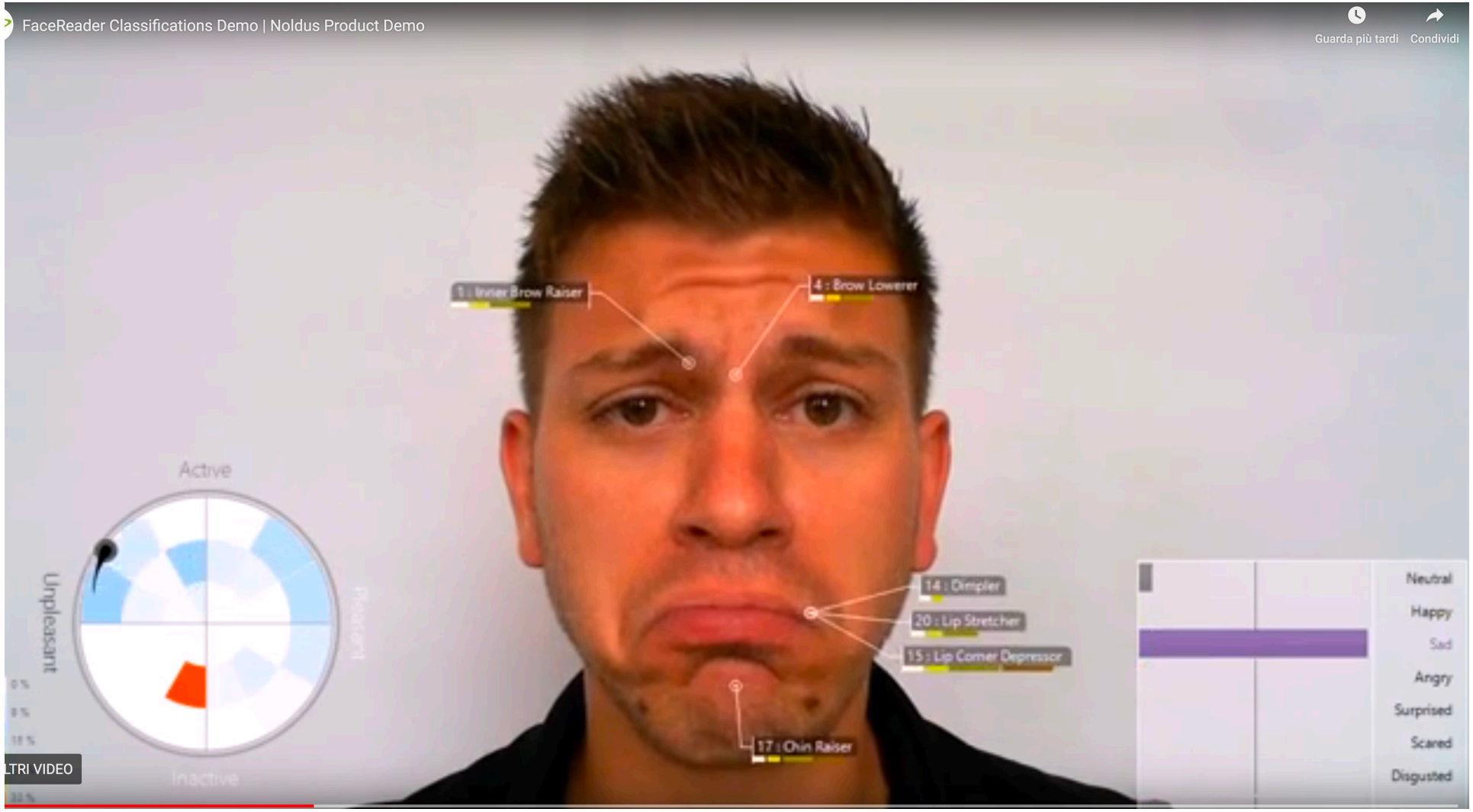
People recognition



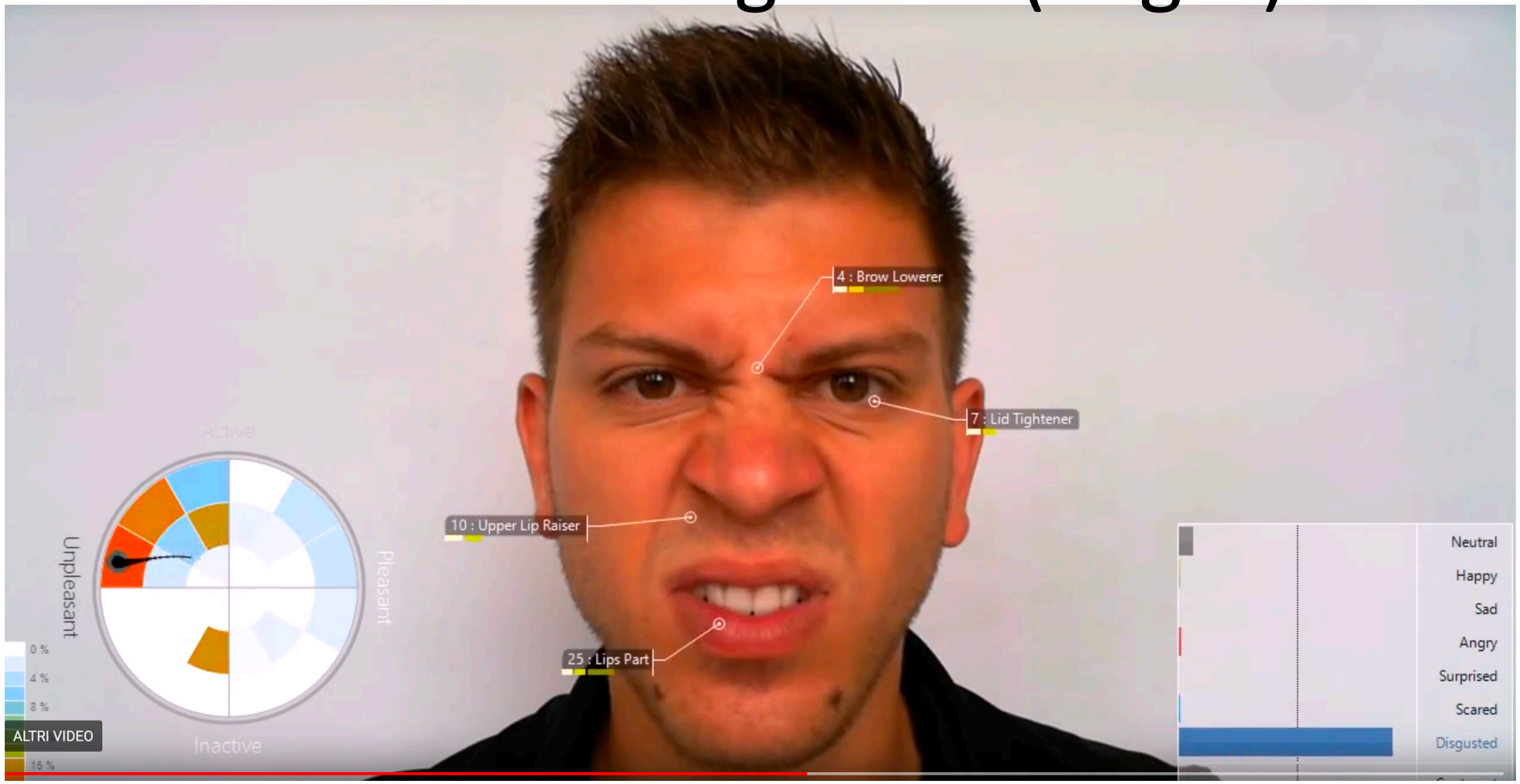
Business applicatons:

- Visitor traffic per hour, day, season, store occupancy vrs opening hours
- Schedule staffing
- Shoplifting, sweetharting
- Customer demographics /satisfaction
- Security

Emotion recognition (unhappy)



Emotion recognition (anger)



Sweethearting

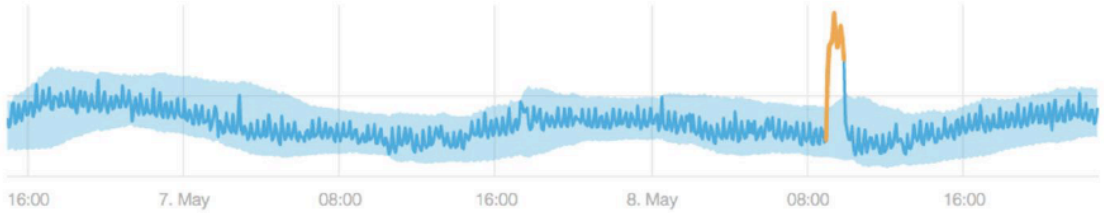
Shoplifting

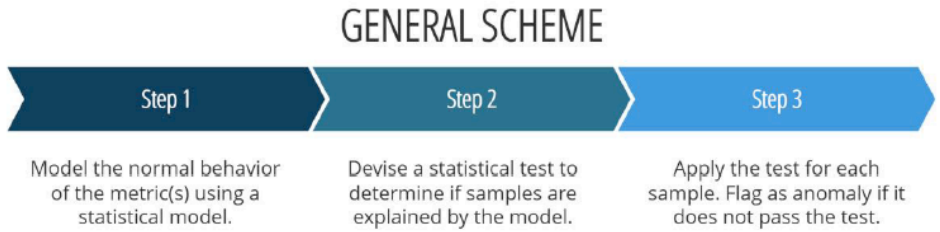
- is a term used in the retail loss prevention industry to mean intentional margin loss through employee theft at the cash register. Sweethearting is the most common type of employee theft.

- (also known in slang as **boosting** and **five-finger discount**) is a popular term used for the unnoticed theft of goods from an open retail establishment.

Example 2: anomaly detection

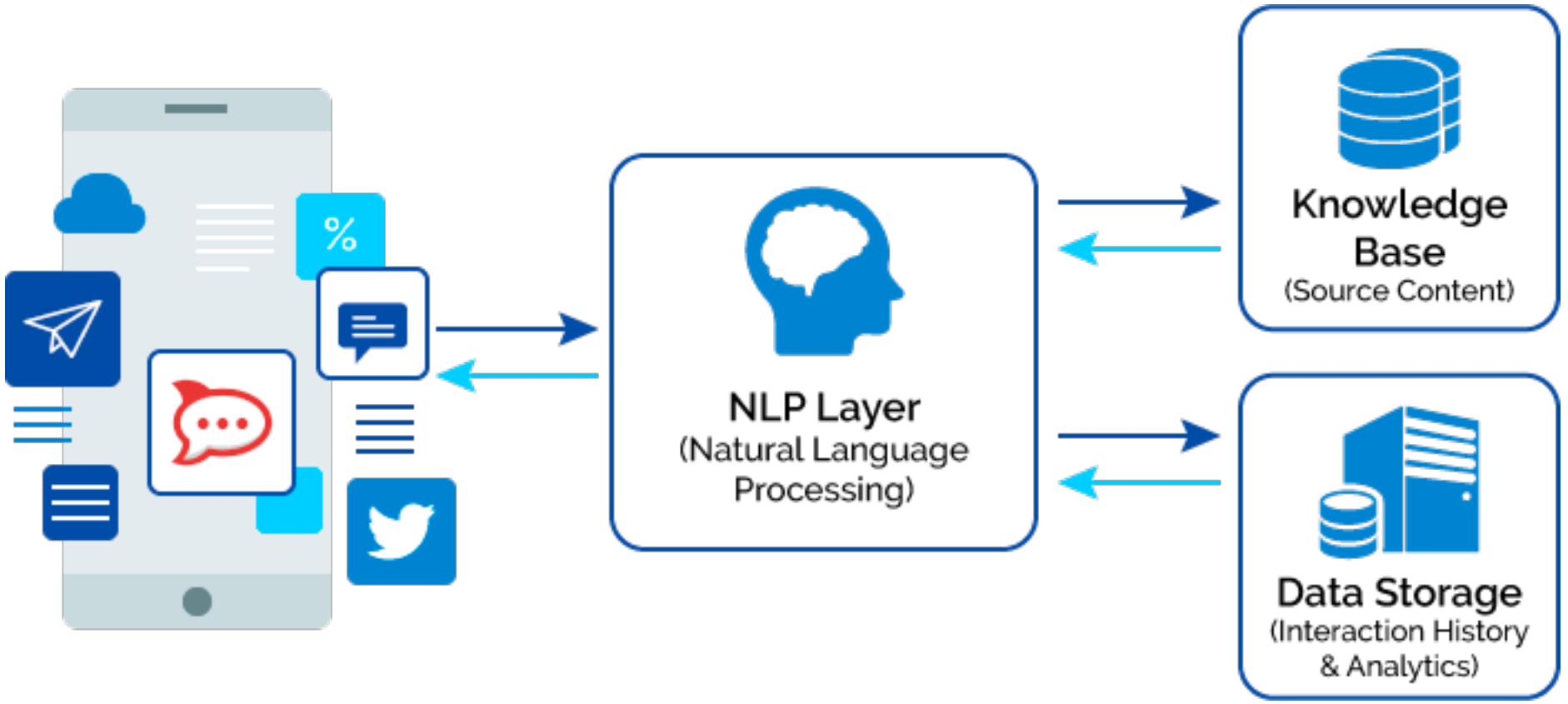
- Can be applied to any signal (output of sensors/medical data etc.) to learn “normal behaviour” and detect/predict anomalies
- Data is collected in real time. Remember Magpie example of cold chain.





In anomaly detection input are continuous signals.

Example 3: Text



Text is pervasive: social media messages, reports, CVs, web data…

Challenges with unstructured data (images, signals, text)

- Need **complex processing** to be useful
  - Text processing, natural language understanding (NLP)
  - Image processing, image understanding
  - Signal processing
- A number of techniques/methods are available (Artificial Intelligence, Machine Learning)
- E.g. see Cognitive Apps in Watson (later in this course)
- Again: GTP offers powerful NLP tools
- Will see something (on text processing) also when talking about Social Analytics

In class exercise

- Smart city is **a city that uses technology to provide services and solve city problems**. A smart city does things like improve transportation and accessibility, improve social services including public health, promote sustainability, and give its citizens a voice.
- Smart City **is business intelligence**: use data and data analytic features to improve the management of resources, and ultimately our lifes
- Consider one of these targets: transportation and accessibility, social services, sustainability, empowerment of citizens. You can be more specific (e.g., reduce food waste)
- For the selected target, which (digital) data? From which sources? How they can be used to meet specific objectives?
- See also https://en.wikipedia.org/wiki/List_of_smart_cities