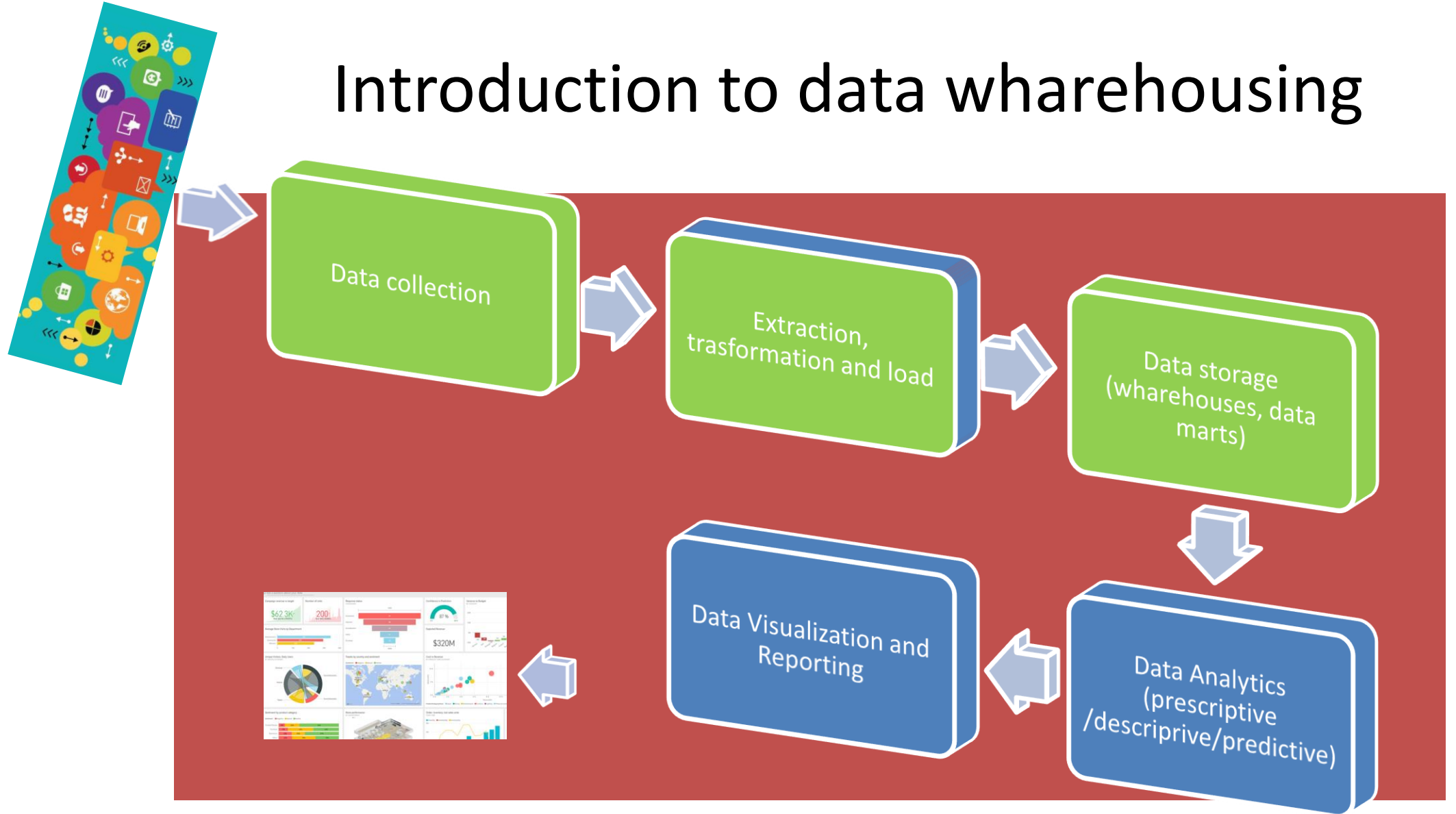


Introduction to data warehousing



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 plain old data, it requires a **lot of work** before its 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
- Next lessons will treat the problem of how to **integrate, clean and manage** the data before they can be transformed into INFORMATION

AT&T Order Confirmation - iPhone®
Dear

ORDERS

Thank you for your recent order. Your iPhone will ship in approximately 14-21 business days*.
You can check the status of your order anytime by visiting <https://www.att.com/eos/unauth/eosLogin?productType=wireless> or by calling our automated system at 1-866-339-3888 and entering when prompted for the order number.

Your Order Details:

Integration,

Bounce Rate 52.96%
-6.2% ▲ vs. 56.49% (Prev.)

Pageviews 2.33
9.77% ▲ vs. 2.13 (Prev.)

New Sessions 65.0%
-5.5% ▼ vs. 68.8% (Prev.)

Time on Site 2m:18s
15.2% ▲ vs. 1m:60s (Prev.)

Metric	Volume	Conversion Rate
Users	370,375	
Leads	10,681	2.88%
Opportunities	1,546	14.47%
Wins	488	31.56%

Source	Sessions	Previous Pe	Change	Trend
organic	31,890	16,673	91.3% ▲	
referral	12,400	7,292	70.1% ▲	
direct	5,992	3,631	65.0% ▲	
email	103	0	100.0% ▲	

HubSpot Revenue This Month
\$1,300,400
Revenue growth rate ▲ 20.4%
\$1,200,000 Last Month

AdWords ROI (Last 30 Days)
ROI -92.23%
Previous period: -99.71% 100.00% ▲
ROI Value -\$22,319
Previous period: -\$38,115 100.00% ▲

AdWords Cost-per-Conversion (Last 30 Days)
\$56.09
Target: \$8.34

MailChimp Email List Performance API 3.0
List: All

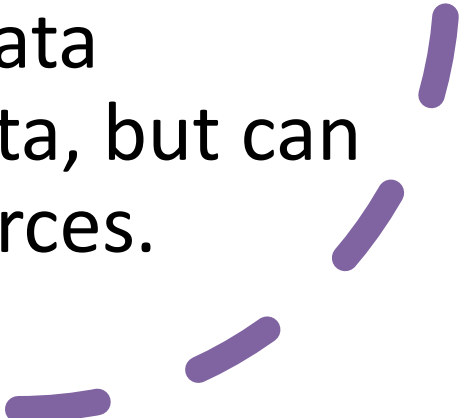
19 List Members	0.0 List Rating
89.5% Open Rate	26.3% Click Rate
0.0% Unsubscribe Rate	0 Unsubscribes

Social Media Followers

21,500 Likes	4,011 Followers	3,528 Subscribers	377 Followers	2,808 Followers	405 Circled By
--------------	-----------------	-------------------	---------------	-----------------	----------------

Opinions: eForCity Black Car Air Vent Phone Holder Cradle compatible with the
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, or different types of data (sales, personnel..)
- Management requires an integrated, 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 omitted).


By geographic area

Active Data Warehousing Market - Growth Rate by Region (2020-2025)



Use case: 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.

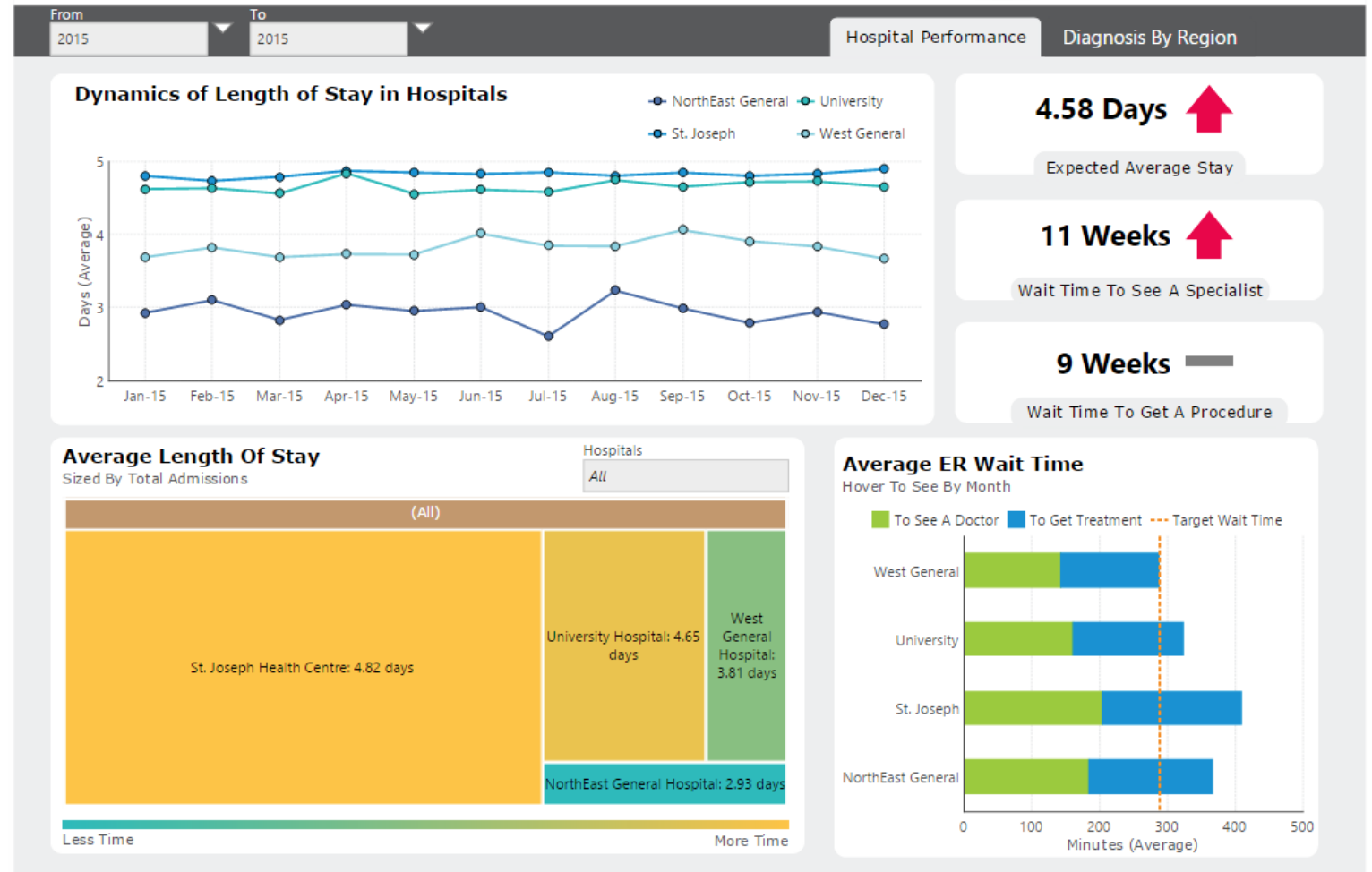


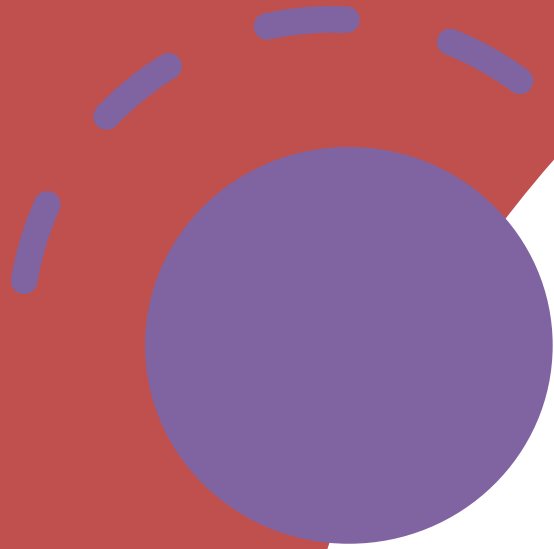
Large organizations need warehousing the most

- **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.**

SO 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

Customers

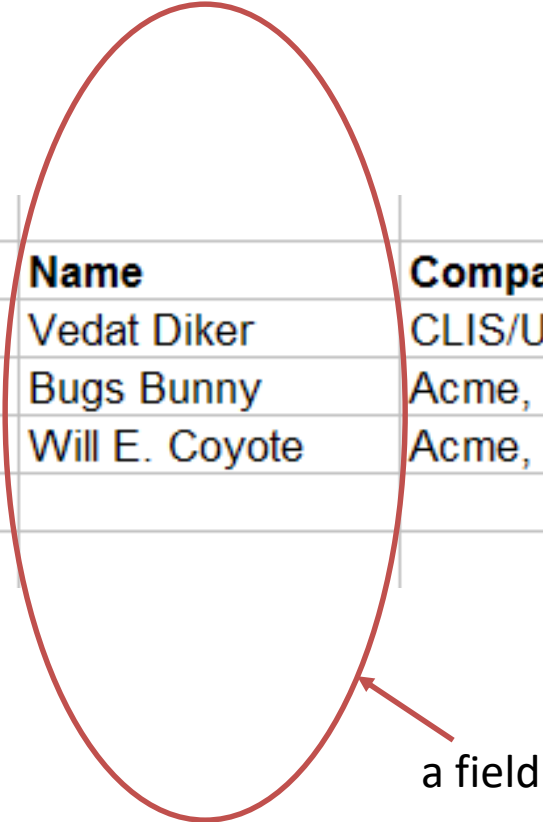
NAME of the Table
(also called ENTITY)

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-worlds 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

These are different data types

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 consist of more than one field (and is not necessarily an ID). What matters is that it is UNIQUE!!
e.g., actors 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 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?

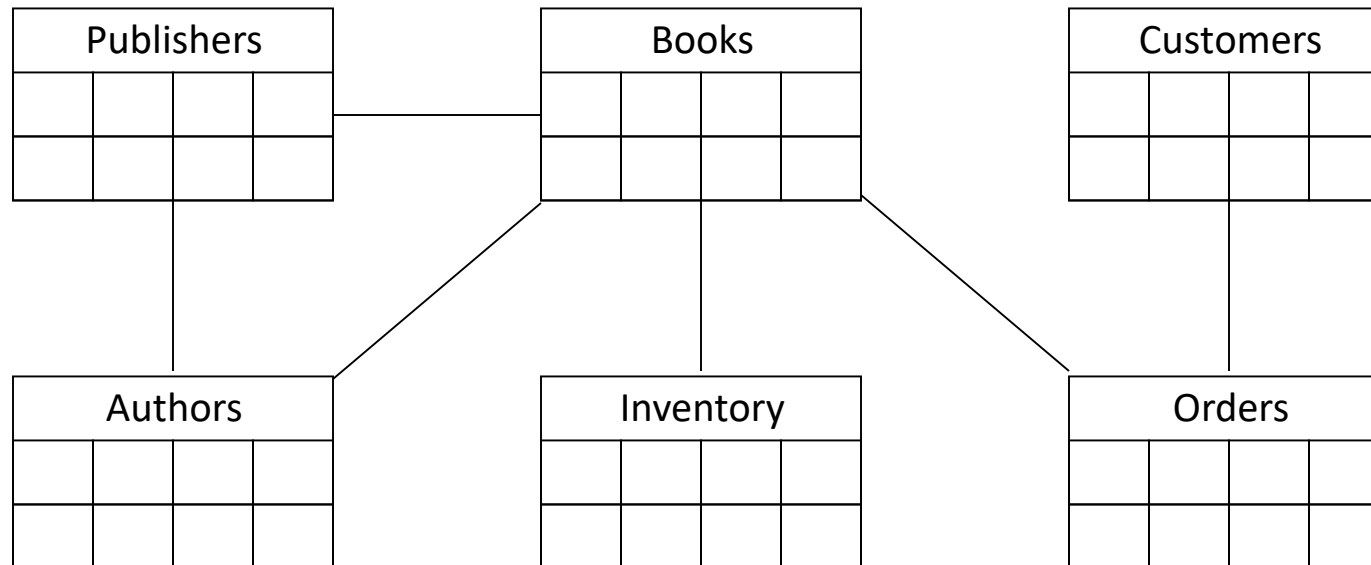
Primary keys can be used to connect tables

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

Databases usually are made of multiple connected Tables, each representing a “viewpoint” on the data (e.g., the example below for a chain of libraries)



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?**»

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

Directors

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

Movies

child table

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

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.

foreign key field

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

How does, in practice, this connection process occur?

- Using query languages (specific programming languages to query databases)
- 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)

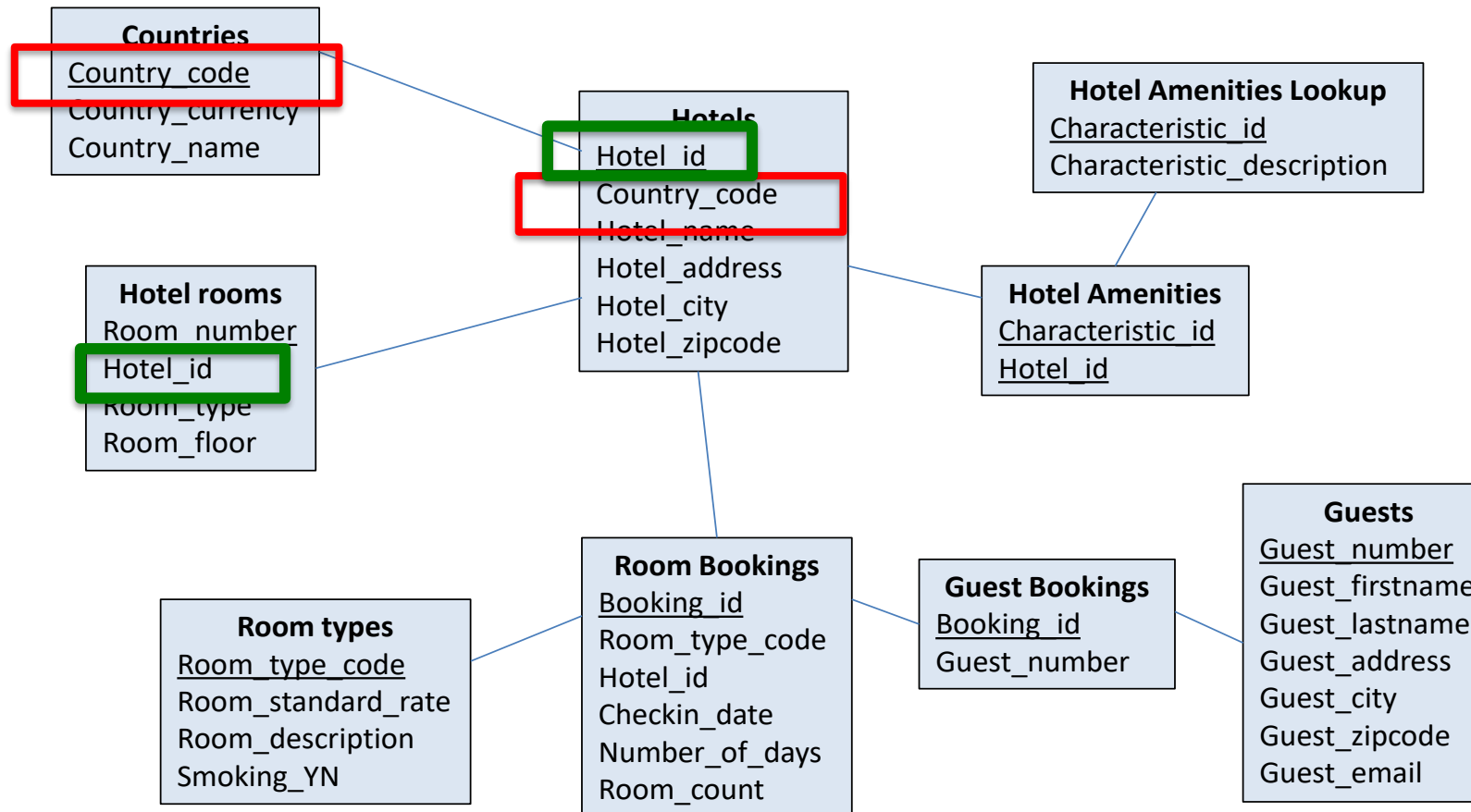
Code

```
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;
```

Another example with multiple tables (primary keys are underlined)

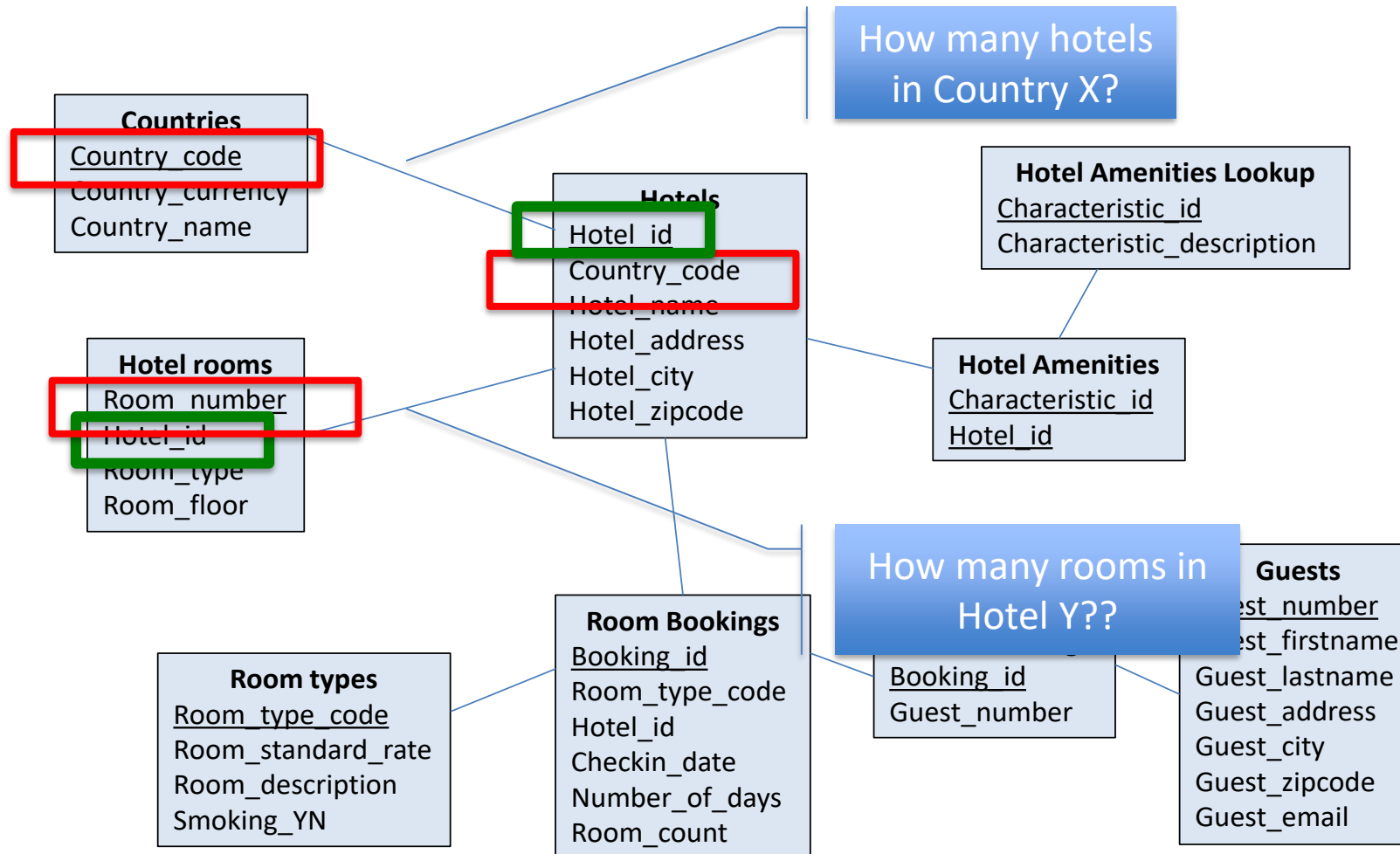
Hotel Reservation Database

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



“Common” keys are used to answer queries

Hotel Reservation Database





Summary of terminology so far

Tables describe entities

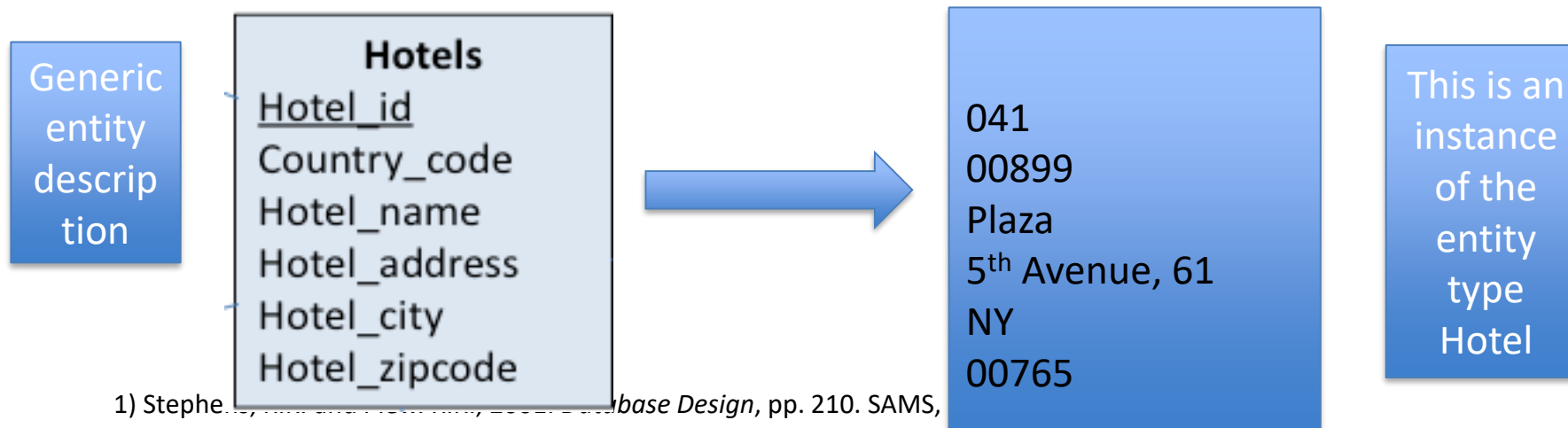
- **TERMINOLOGY:** An *entity* is a “business object” that represents a group, or category of data.¹
- Example: hotel, hotel_room, guest..



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!



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

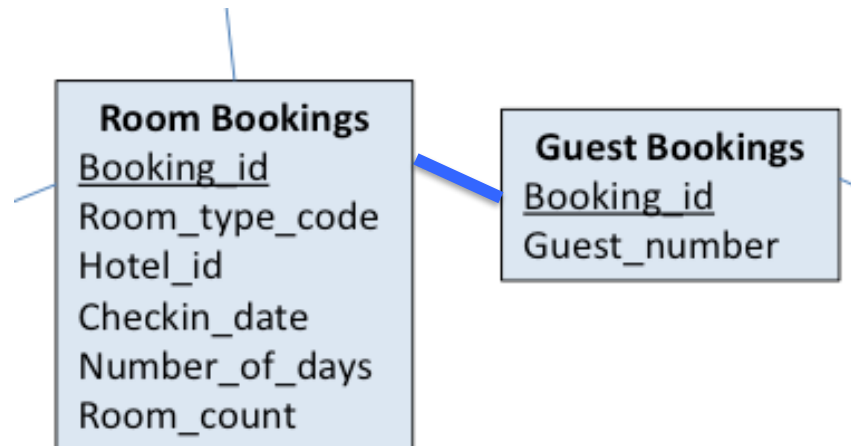


- As we said, an attribute can be a **primary key** or a **foreign key**. In the above example, Hotel_id is primary, country_code is foreign. Primary keys are UNIQUE for each record of a given entity table. For example, Hotel_id is a primary key for the entity table *Hotels*, and is a *foreign key* for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

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

Relationship

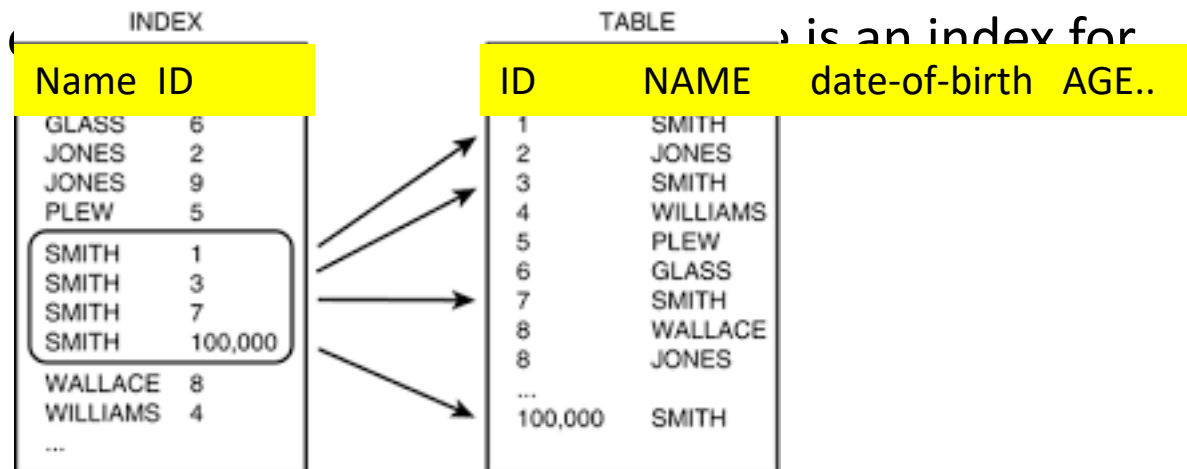
- TERMINOLOGY: A *relationship* is a **link** that relates two entities that share one or more attributes (keys, fields).
- Example: **Guest_booking** and **Room_booking** have the same attribute Booking id (since one would like to know which guest reserved a given room, or which room has been reserved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

One-to-One Merging

geography.dta

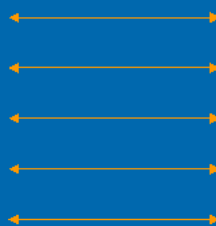
country Land Area (sq km)

ARG	2,736,690
FRA	640,053
GER	349,223
ITA	294,020
USA	9,161,923

economy.dta

country GDP per Capita

ARG	12,468
FRA	27,913
GER	28,889
ITA	28,172
USA	39,498



country Land Area (sq km) GDP per Capita

ARG	2,736,690	12,468
FRA	640,053	27,913
GER	349,223	28,889
ITA	294,020	28,172
USA	9,161,923	39,498

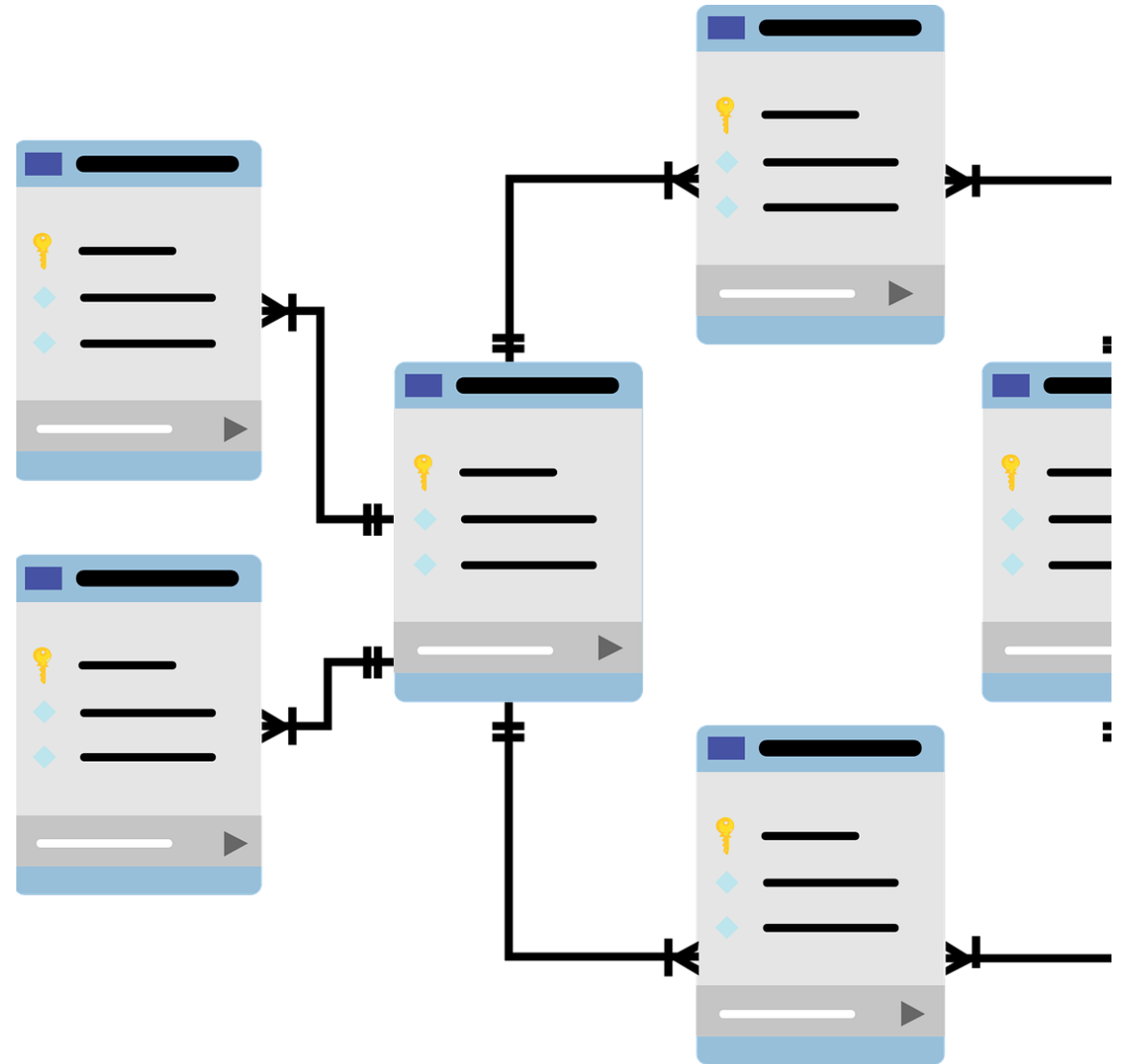


Summary so far

- 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!!!)
- Remember the first lesson: some recent OpenAI tools directly translate descriptive NL queries into SQL code.

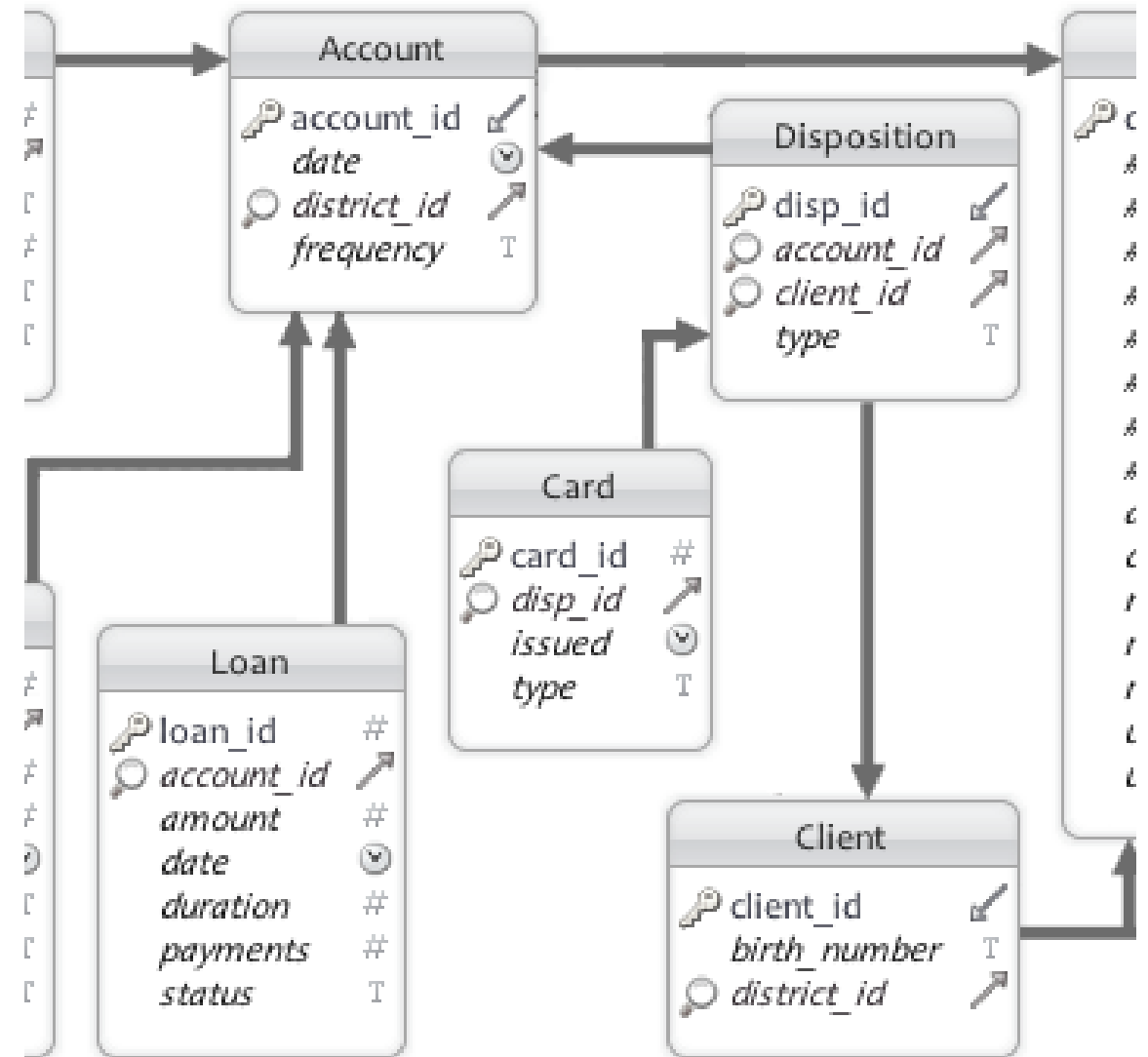
Database schema

- 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.

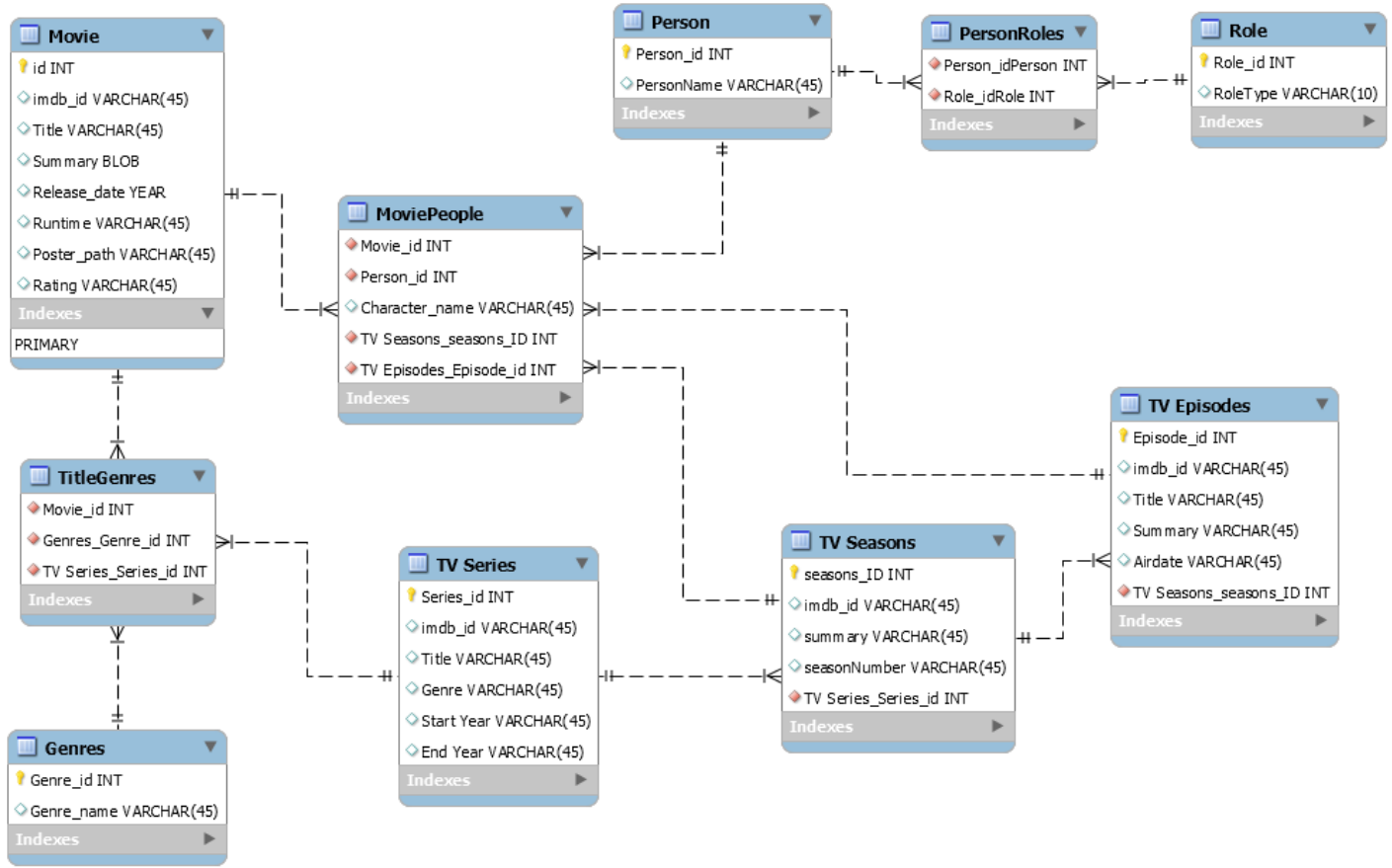


Schema where tables are related by primary-foreign key pairs

- Primary keys are those with the «key» symbol
- Foreign keys are those with the magnifying glass



A more complex scheme



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 through 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 (DataWharehouses, 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**

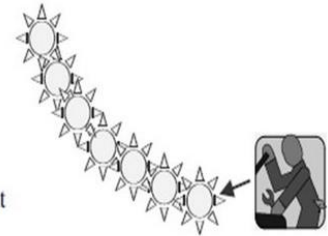
OLTP vrs. OLAP (DW)- 2

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**

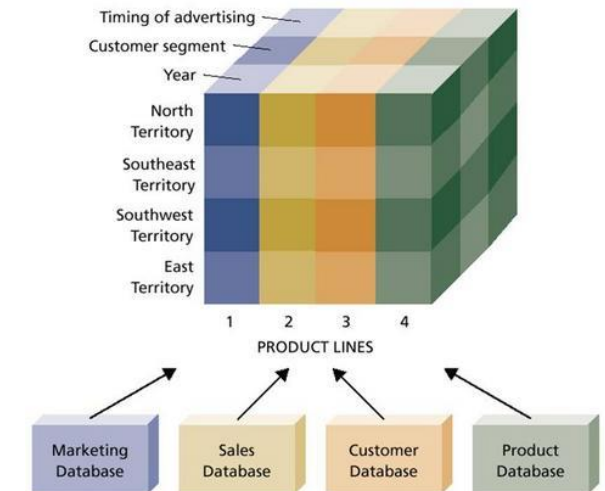
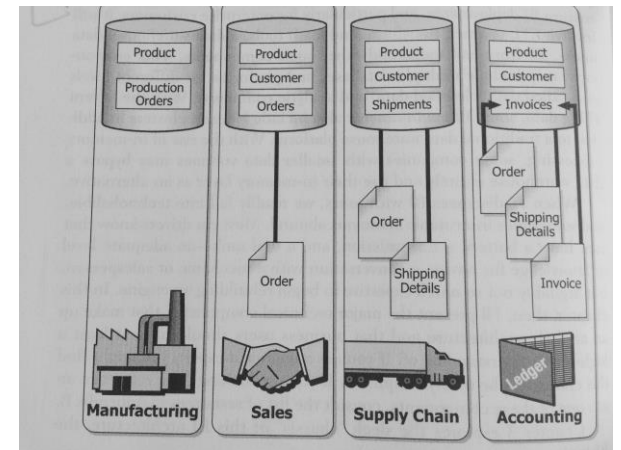
Making the wheels of business turn

- ◆ Take an order
- ◆ Process a claim
- ◆ Make a shipment
- ◆ Generate an invoice
- ◆ Receive cash
- ◆ Reserve an airline seat



OLTP vrs. OLAP (DW)- 3

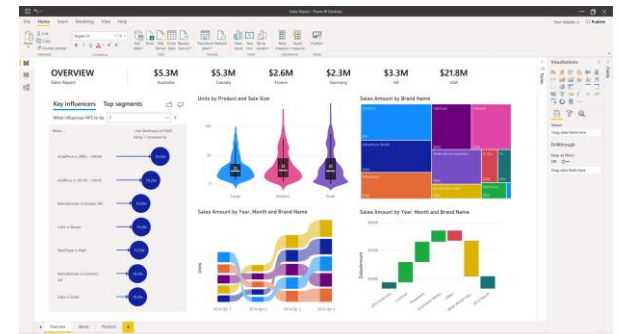
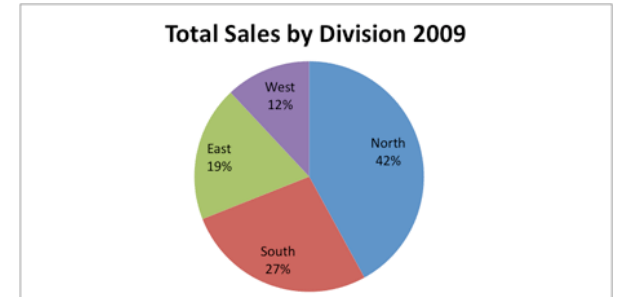
- **Source of data**
OLTP: Operational data; OLTPs are the original source of the data and each system manages a specific transactional database (e.g. shipments, orders..).
- OLAP: OLAP data **comes from the various OLTP Databases + external sources** and are aggregated (also called OLAP cube)



OLTP vrs. OLAP (DW)- 4

What the data represent

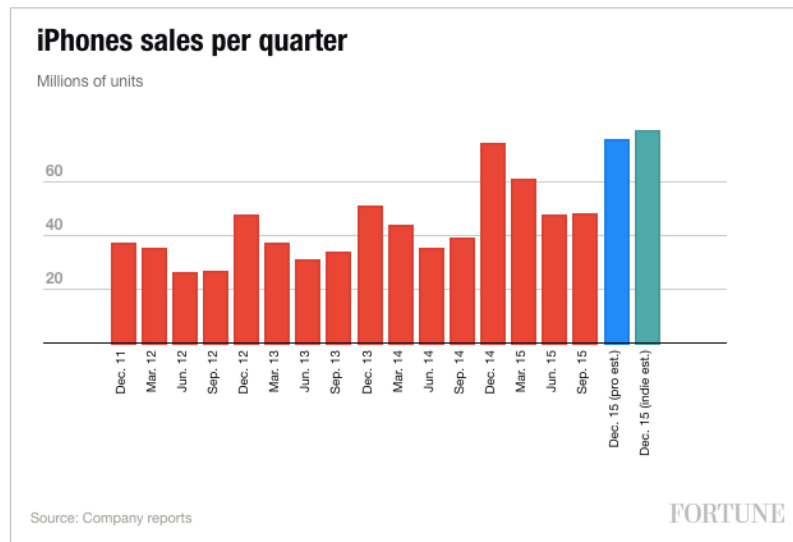
- OLTP: Reveals a snapshot of ongoing business processes
- OLAP: **Multi-dimensional** views of various kinds of business activities



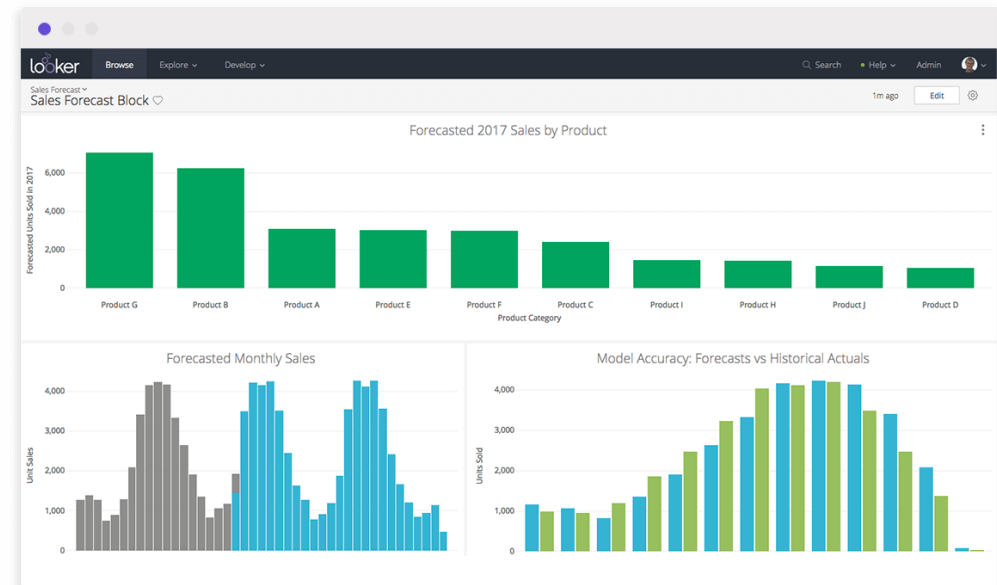
OLTP vrs. OLAP (DW)-5

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)



OLTP: How many i-Phones sold in this quarter?



OLAP: 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 Warehouses can be:

- Subject oriented

Finance, Marketing, Inventory

- Integrated

weblogs, Legacy data, sales..

- Non Volatile

Data (even historical data) remain in database

- Time variant

Grain can be real-time, day, month, quarterly..

Querying OLTP and DWs



What kind of queries in an OLTP?

Which customers are based in Roma?

How many delays we experienced in spare parts supply?

How many spare parts of Product 222 are available?

Who has been our best client in 2016?

What has been the total revenue in 2015?



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

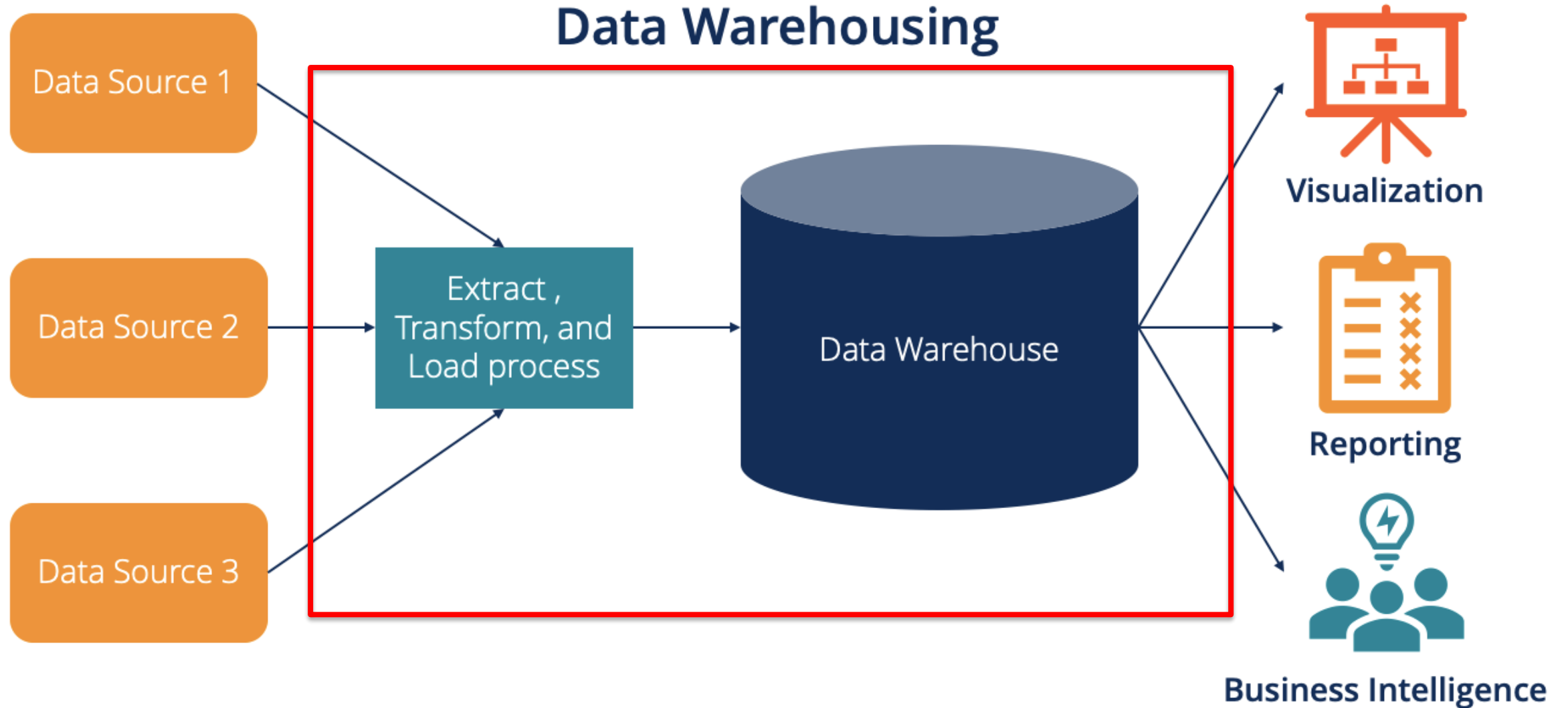


❖ The Data Warehouse
helps to *“optimize”* the
business

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

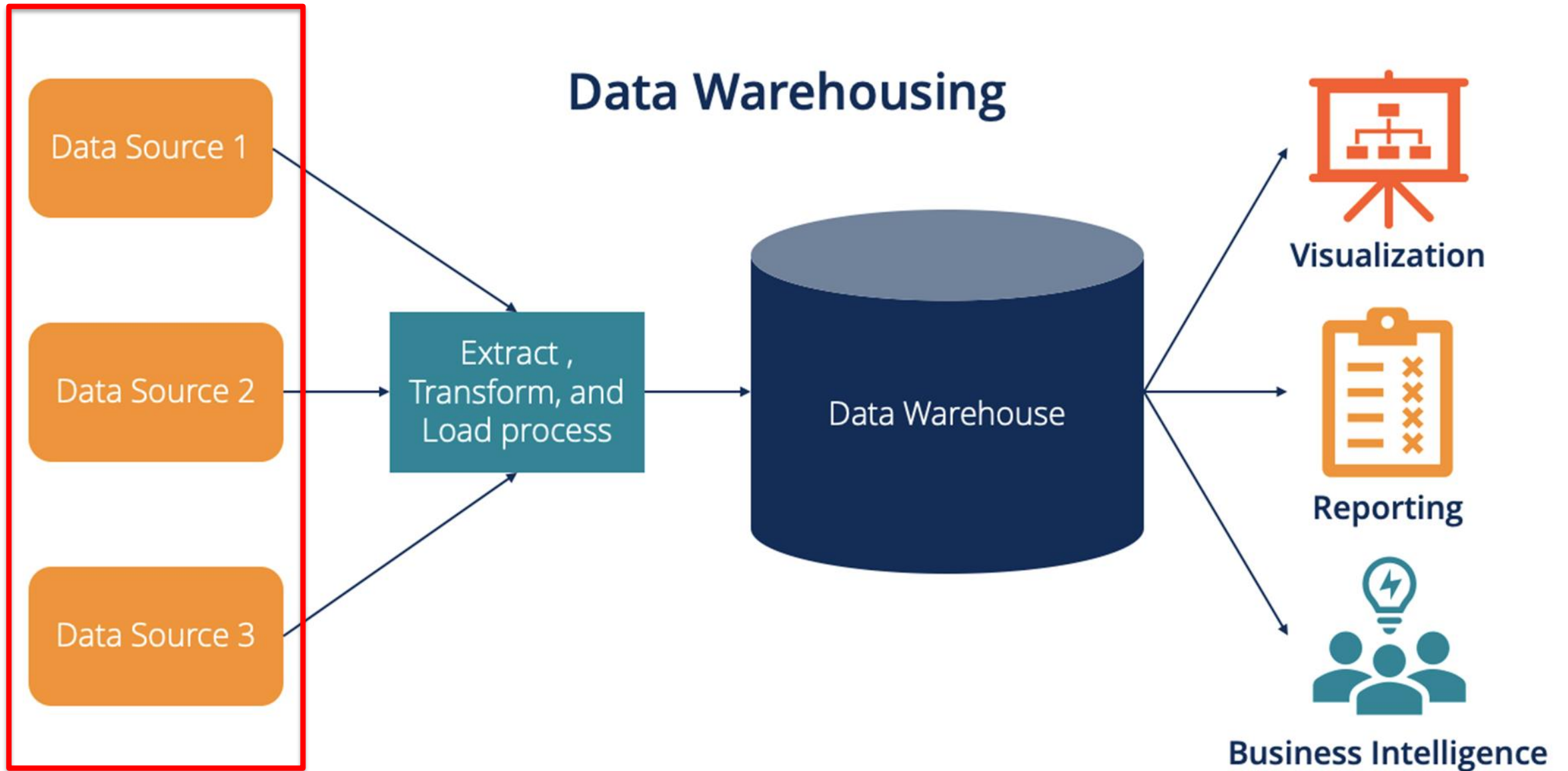


A large red circle on the left side of the slide, partially cut off by the edge.

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, create data Marts, add metadata, aggregate, integrate data
- 
- A decorative purple dashed line in the bottom right corner of the slide.

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 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** (clicklogs of web sites): for traffic and e-commerce analysis
- **Environmental data** (geolocations, meteorological 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)

FaceReader Classifications Demo | Noldus Product Demo

Guarda più tardi Condividi

1: Inner Brow Raiser

4: Brow Lowerer

14: Dimpler

20: Lip Stretcher

15: Lip Corner Depressor

17: Chin Raiser

Active

Unpleasant

Pleasant

Inactive

Neutral

Happy

Sad

Angry

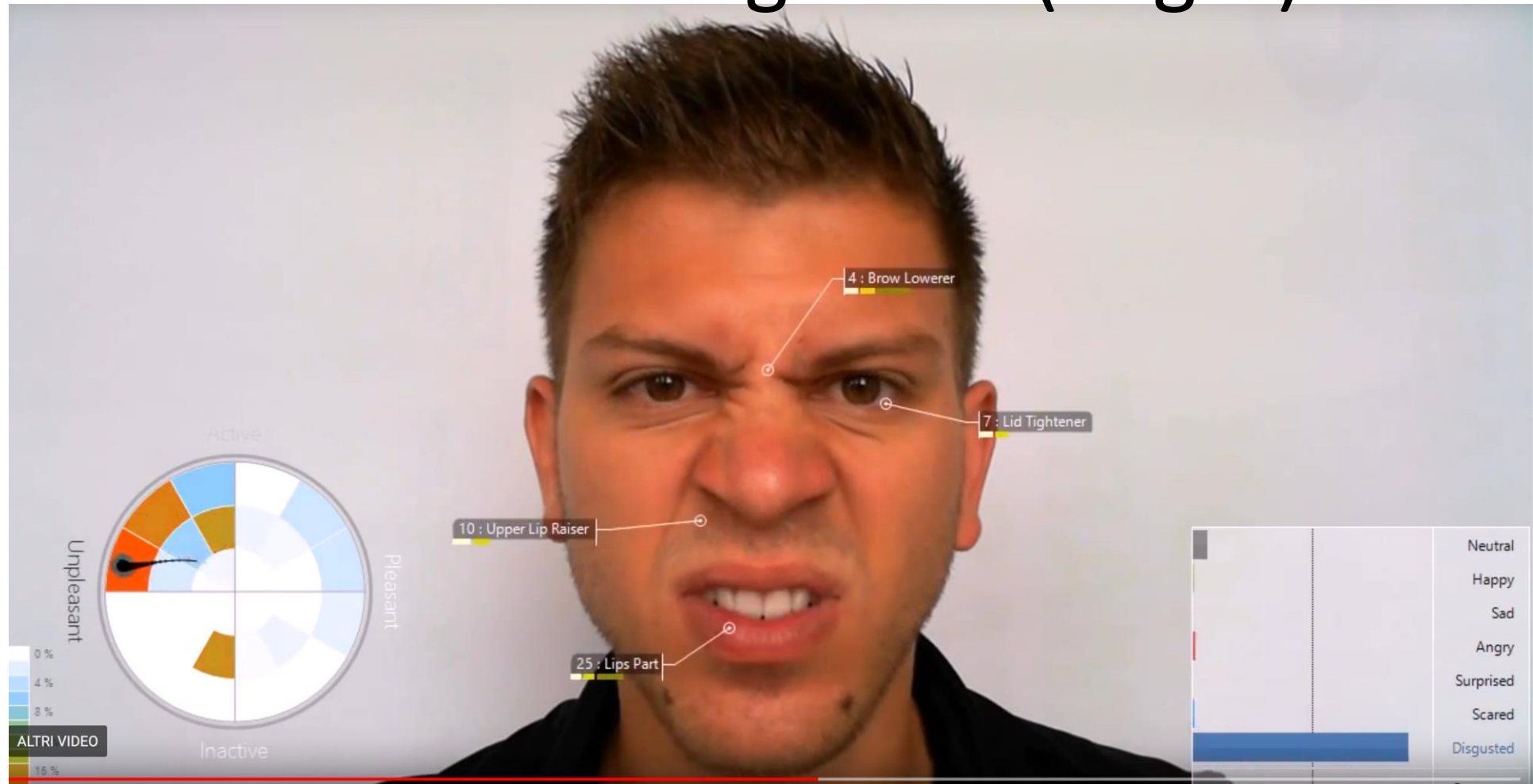
Surprised

Scared

Disgusted

ULTRI VIDEO

Emotion recognition (anger)





Sweethearting

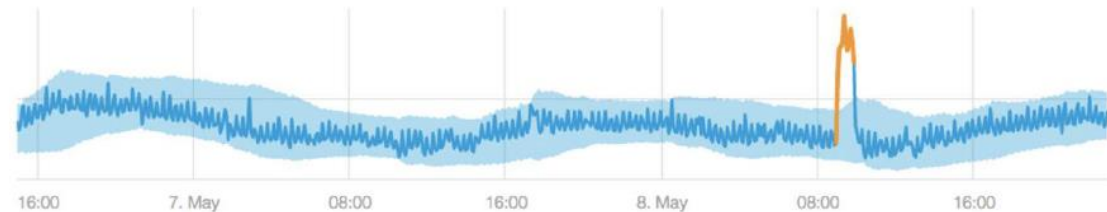
- is a term used in the [retail loss prevention](#) industry to mean intentional margin loss through employee theft at the [cash register](#). Sweethearting is the most common type of employee theft.

Shoplifting

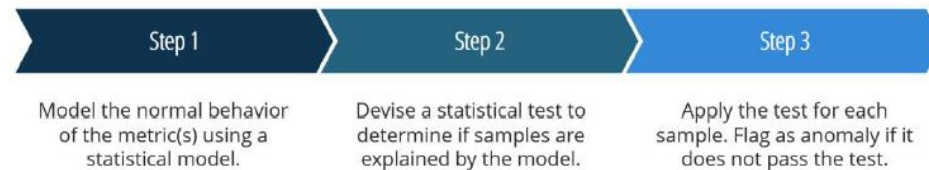
- (also known in slang as **boosting** and [five-finger discount](#)) is a popular term used for the unnoticed [theft](#) of goods from an open [retail](#) establishment.

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.

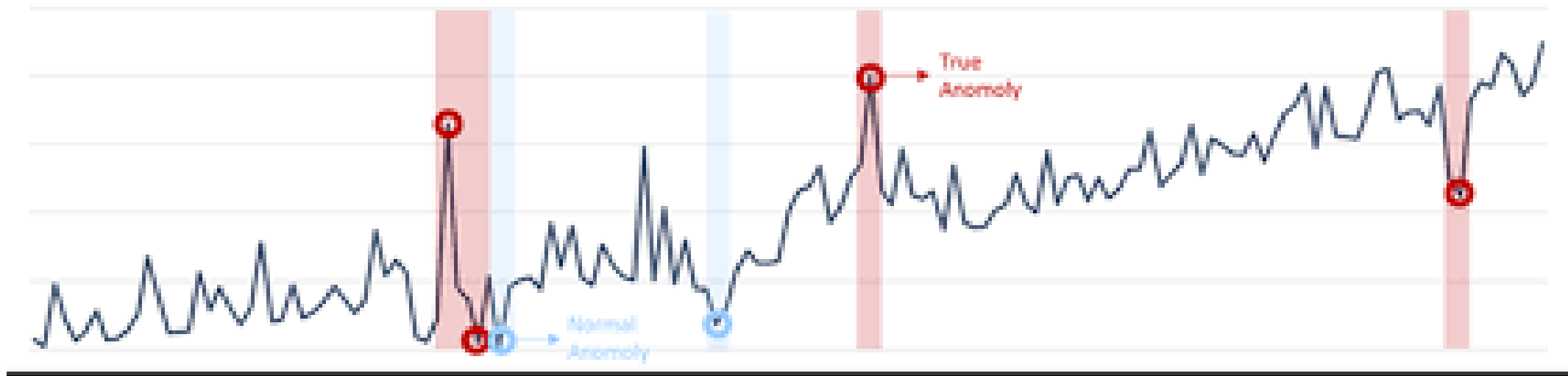


GENERAL SCHEME

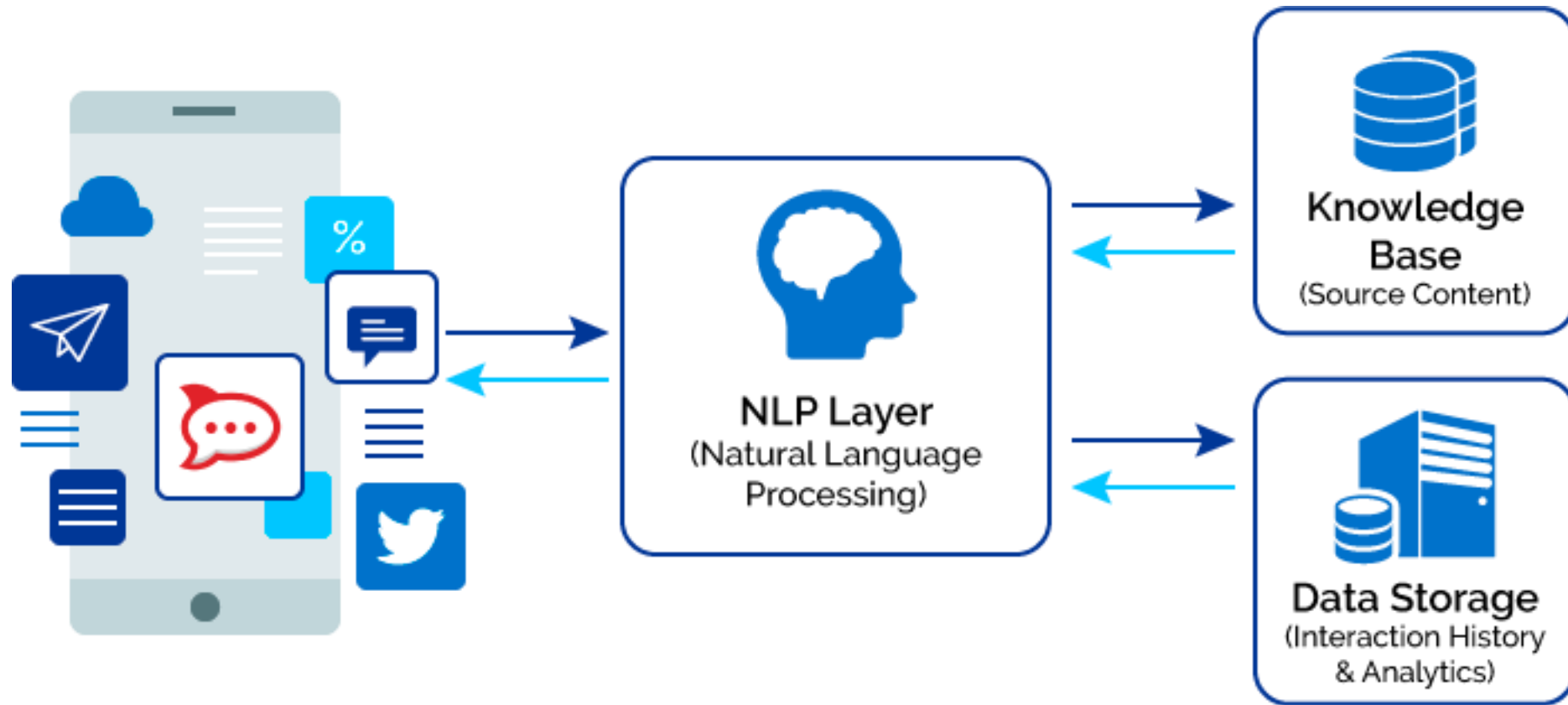


In anomaly detection input are continuous signals.

FRAUD DETECTION



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
 - 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)
- Will see something (on text processing) also when talking about Social Analytics

In class exercise

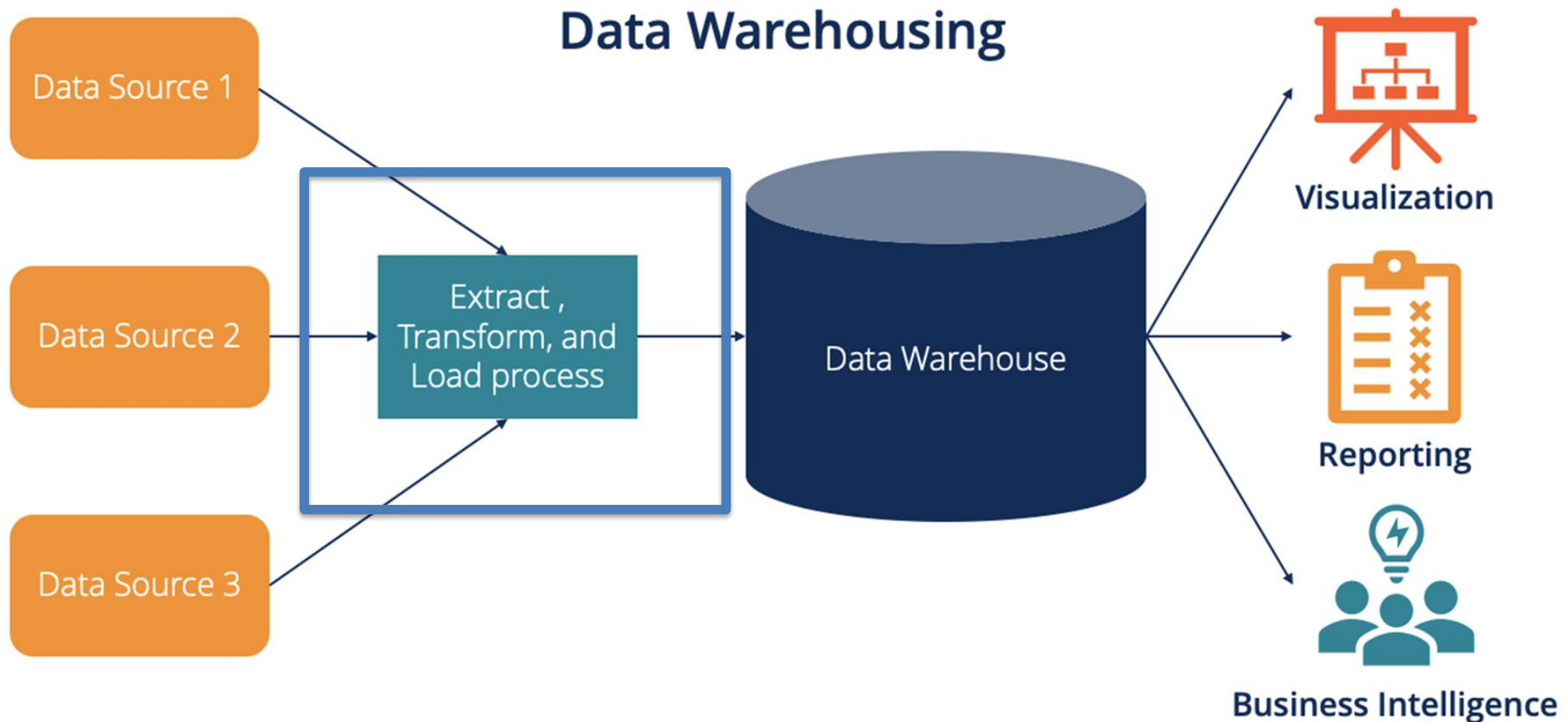
- Smart city is a **city that uses technology to provide services and solve city problems**. A smart city does things like improve transportation and accessibility, improve social services including public health, promote sustainability, and give its citizens a voice.
- Smart City is business intelligence: use data and data analytic features to improve the management of resources, and ultimately our lives
- 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?

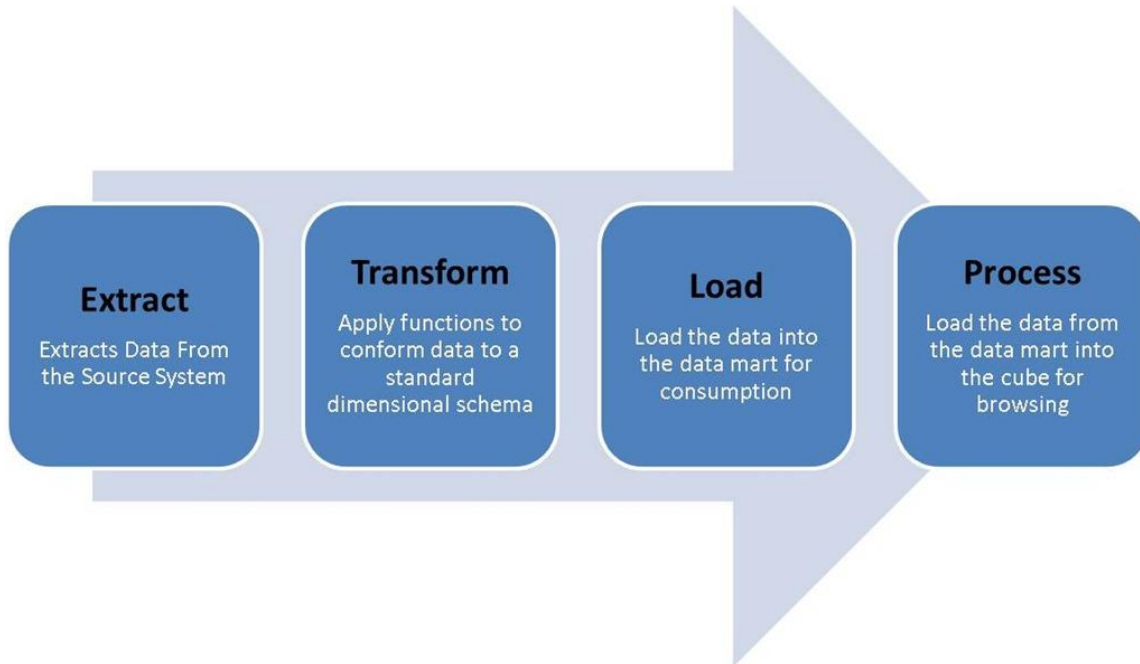
A large red circle on the left side of the slide, partially cut off by the edge.

Design aspects of a DW

- 1. Select:** Which data and what for
 - 2. Transform:** so-called **ETL**: Extraction, cleaning, Transform and Load data
 - 3. Store and process data:** data Marts, metadata, aggregations
- 
- A decorative graphic in the bottom right corner consisting of several short, thick, purple dashed lines arranged in a curved, upward-sloping pattern.

Step 2: ETL: extraction, cleaning, transform and load data





- It is important to understand that a data warehouse has the purpose of integrating **different sources** of data, not just COLLECTING new data.
- So, new data are added, deleted, and updated in the ORIGINAL sources (e.g. an OLTP, or in the original source).
- The data warehouse must **extract** new data as they are generated , detect and handle **changes** in old data, and **integrate** data from the different sources.

What is ETL

- Extraction–transformation–loading (ETL) tools are pieces of software responsible for
 - the extraction of data from several sources,
 - its cleansing, customization, reformatting, integration, and
 - storage into a data warehouse.
- Building the ETL process is potentially **one of the biggest tasks of creating a warehouse**; it is complex, time consuming, and consumes most of data warehouse project's implementation efforts, costs, and resources.

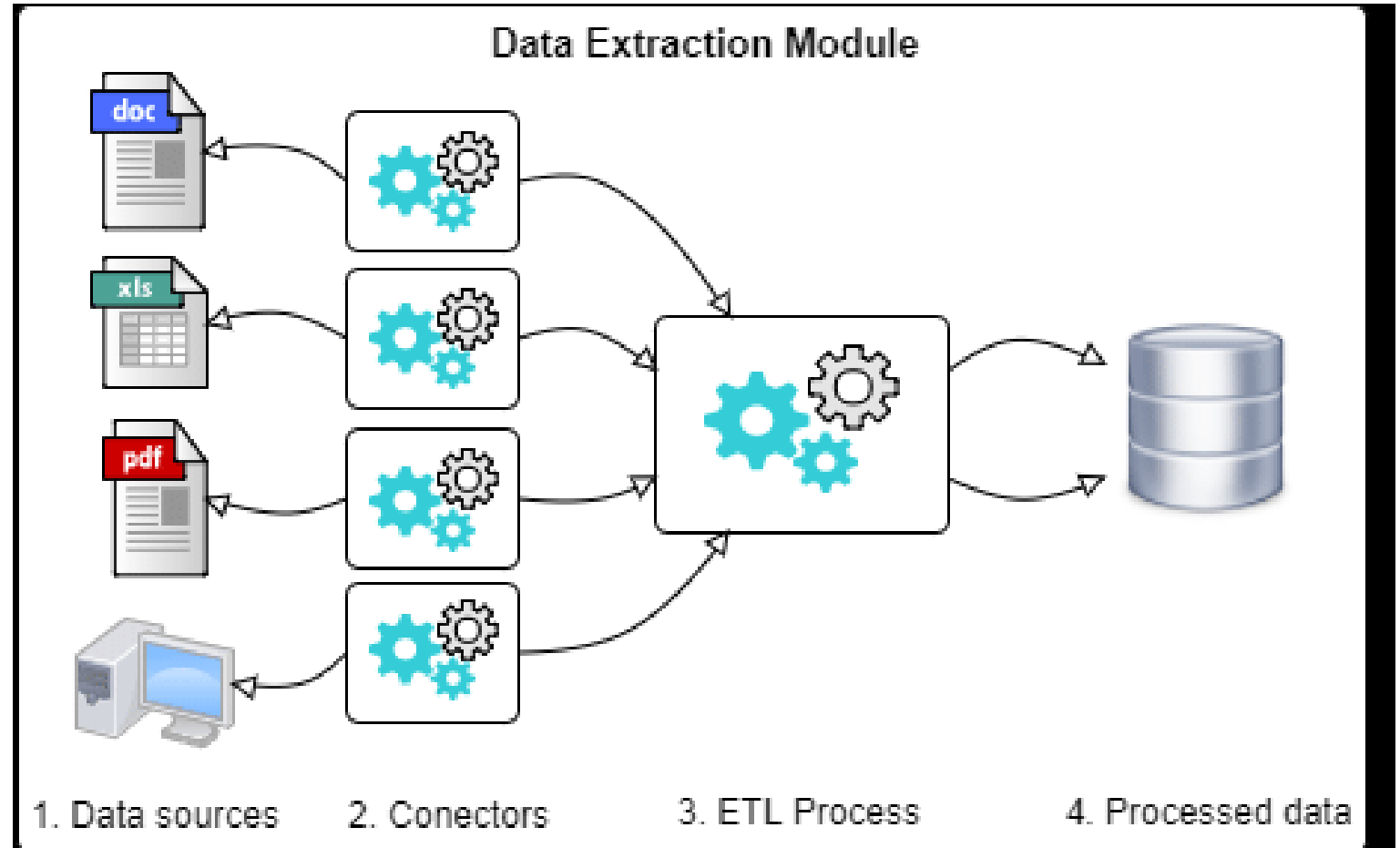
ETL Functional Elements

- ETL systems have a common purpose: **they move data from one database to another.**
- Generally, ETL systems move data from OLTP systems (or from external sources) to a data warehouse.
- An ETL system consists of **four distinct functional elements**:
 - Extraction
 - Transformation (cleaning, alignment of data, ecc. ..will see)
 - Loading (the result of Extraction and Transformation on the DW)
 - Adding Metadata to the DW

ETL 1. Extraction

- The first step in any ETL scenario is data extraction.
- The ETL extraction step is responsible for extracting data from the source systems.
- Each data source has its distinct set of characteristics that need to be managed in order to effectively extract data for the ETL process.
- The process needs to integrate systems that have different platforms, such as different database management systems, different operating systems, and different communications protocols.


Extraction



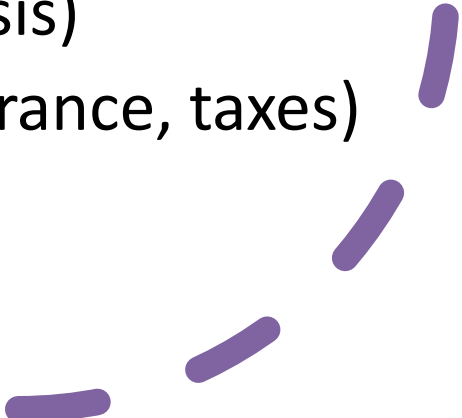
Issues: Extraction frequency

- There are several ways to perform the extract:
 - **Update notification** - if the **source system** is able to provide a notification that a record has been changed in the original data source and describe the change (e.g. a new shipment has been completed, and order has been filed..), this is the easiest way to get the data.
 - **Incremental extract** – No notifications, so in given **time intervals** the extraction process start, source system should be able to identify which records have been modified and provide an extract of such records. During further ETL steps, the system needs to identify changes and propagate it down.
 - **Full extract** - some systems are not able to identify which data has been changed at all, so a full extract is the only way one can get the data out of the system. The full extract requires keeping a copy of the last extract in the same format in order to compare and be identify changes. Full extract handles deletions as well.
 - **Extract from unstructured resources** – If data are not structured (not a database) system extracts *either in real time (in streaming) or incrementally*, but new data are simply added to old data (e.g. new tweets discussing about a given product).

Take away message

- You are not responsible for the extraction process, **IT people will be**
 - Your responsibility is to help deciding – having in mind objectives of the analysis and timing constraints – **which data should be extracted, and (about) what frequency of extraction.**
 - E.g., if the objective is to predict credit card frauds, need *real-time updating*. If objective is to analyze and compare point-of-sales, weekly or monthly extraction can be enough
- 

In class
exercise: what
updating policy
and which
sources would
you use for
these
applications?
(real-time vrs
incremental)

- Telephony (Churn prediction)
 - Transportation (traffic management)
 - Energy and utilities (energy savings)
 - Health (remote healthcare; epidemic warning systems)
 - Natural systems (water management)
 - Law, defense, cybersecurity (surveillance systems, cybersecurity detection)
 - Stock market (market data analysis)
 - Fraud detection (credit card, insurance, taxes)
 - eScience (weather prediction)
- 

2. Transformation


- The second step in any ETL scenario is data transformation.
- Objective: make some cleaning and conforming on the incoming data to gain accurate data which is **correct, complete, consistent, and unambiguous**.
- For all datatypes, this process includes data cleaning, transformation, and integration. It defines the granularity of fact tables, the dimension tables, data structures, etc.
- **Note:** if source data is unstructured (images, text, signals), transformation also imply converting from unstructured to structured (tables)!! We now consider only transformation of structured data, text image and signal processing will be introduced later!
- **All transformation rules and the resulting schemas must be described in the metadata repository.**
- Will see later, but **your responsibility** (as business experts in a BI project) is that a **comprehensible** (by business people) description of what kind of transformations are performed on the data is maintained!

Data transformation and cleaning


Data in the real world is **dirty**

 **incomplete**: lacking attribute values, lacking certain attributes of interest, or containing only aggregate data

- e.g., occupation=""

 **noisy**: containing errors or outliers (spelling, phonetic and typing errors, word transpositions, multiple values in a single free-form field)

- e.g., Salary="-10"

 **inconsistent**: containing discrepancies in codes or names (synonyms and nicknames, prefix and suffix variations, abbreviations, truncation and initials)

- e.g., Age="42" Birthday="03/07/1997"
- e.g., Was rating "1,2,3", now rating "A, B, C"
- e.g., discrepancy between duplicate records

Why data is dirty?

- Incomplete data comes from:
 - Not available data when collected
 - Criteria changed (e.g. collect twitter messages with user ID, then GDR rules e no more user ID collected)
 - Human/hardware/software problems
- Noisy data come from:
 - Faulty instruments, Human errors, transmission errors
- Inconsistent/redundant data comes from:
 - Different data sources with different data description models



What can we do with «dirty» data?

- You work on data cleaning when using Watson Studio (the process is called data REFINERY)
- Incomplete, noisy data: if an attribute in a table has mostly noisy or empty values, better to cancel the entire column!
- If a column has a *missing* or unclear attribute name, you can change it
- If the format of data in the same column are different (e.g., for dates: 11/06/23 or April 11, 2023), we need to replace with a unique format
- There are, however, many machine learning algorithms to cope with incomplete data (automated «imputation»)
- Inconsistent data are a more complex problem

Example of inconsistent data

- As a small example, assume you have data coming from two different source systems which you want to merge in the data warehouse: there might be some differences between the two.
- For example, one source may denote the *gender* as Male and Female while other may denote as F and M.

Customer				
CustomerId	Name	EmailAddress	Gender	EmailVerified
1	Jack Frost	jfrost@winter.com	Male	1
2	Miss Piggy	queen@muppets.com	Female	1
3	Dr. Octopus	doc@octopus.net	Male	0

Student ID	First Name	Last Name	Date of Birth	Gender	Contact Num	Address	Class
ST0001	Minahil	Adeel	2/6/1991	F	(042) 35769018	23 A, H-Block, C	A2
ST0002	Eemaan	Ali	3/7/1992	F	(042) 39293847	45 C, B-Block, G	A2
ST0003	Momina	Ahmed	11/12/1994	F	(042) 38833138	65 P, D-Block, C	A1
ST0004	Nisa	Ahmed	8/3/1991	F	(042) 34811145	14 F, Y-Block, D	A1
ST0005	Sana	Shah	10/10/1991	F	(042) 25222886	124/2, X-Block, A2	A2

Comparing these two Tables there is another **mismatch** in the way the same information is encoded. Which one?

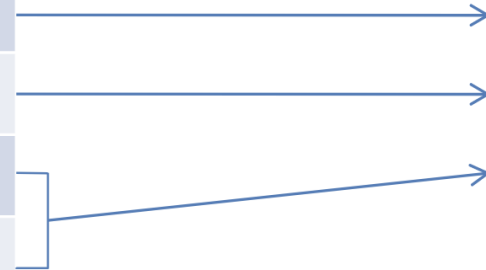
Mismatches in the schemas require CONFORMATION (also called reconciliation)

Schema 1

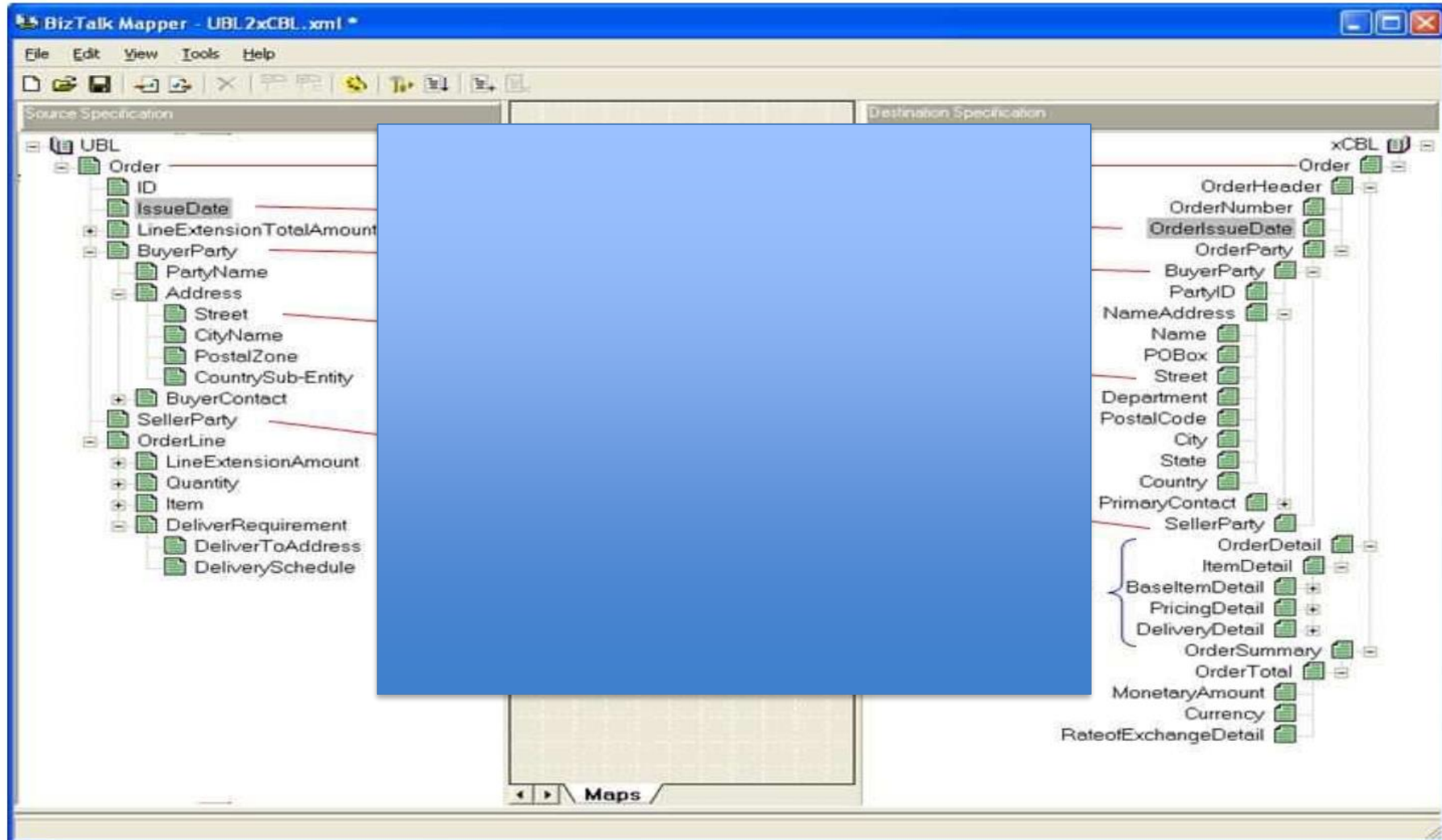
Cust
C#
Cname
FirstName
LastName

Schema 2

Customer
CustID
Company
Contact



Example: aligning attribute names (“reconciling” data)



Summary of types of transformations: MODIFICATION, CONFORMATION, ADDITION

- **MODIFICATION** (changing the name of an attribute or value):
 - Example of **attribute** modification: If you are storing the gender in target as M and F, you may need to "transform" Male and Female to M and F (or viceversa). You may write a simple CASE statement (a RULE), or you may just write code which translates Male --> M and Female --> F.
 - Example of **values** modification: **Discretizing** attribute values: e.g., if you have «Age» as an attribute, you can define rules to change all values according. e.g., to the rule: IF age<=20 then change value to YOUNG; IF 20 < age <55 then change value to ADULT; ELSE change value to ELDER (e.g., this is handled with conditional replace in Watson Studio)
 - You can also **define a hierarchy** of values for subsequent AGGREGATION operations. For example , if you have dates in your dataset (D1: 08/12/2020) you can define a time hierarchy day→week→month→semester→year. Now D1 can be replaced by different values according to the hierarchy:
D1 →week2 →december →semester2 →year2020

Example of
MODIFICATION
(discretization)

Discretization

NUM	Total Spend	CAT	Spend Category
	7.342,99		Top 33%
	304,12		Bottom 33%
	4,56		Bottom 33%
	345,87		Middle 33%
	8.546,32		Top 33%



Summary of types of transformations: MODIFICATION, CONFORMATION, ADDITION (2)

- **CONFORMATION** (making two attribute compatible) : If you want to encode the Name attribute in two attributes: *First Name*, *Family Name*, then you must **split** the values in each record of Table 1 and record the data separately in the Target Table. Again, you do this by writing some code and documenting it with a RULE.

FullName	FirstName	LastName
David Jones	David	Jones
Samual Thomas	Samual	Thomas
Hilary Stiles	Hilary	Stiles
Jennifer Smith	Jennifer	Smith
Owen Lamb	Owen	Lamb

How To Use STRING_SPLIT – Split Delimited Row Into Single Column

Name	Trophies
Federer	Wells,Miami,Halle
Nadal	Madrid,Italian
Djokovic	Paris


Name	Trophies
Federer	Wells
Federer	Miami
Federer	Halle
Nadal	Madrid
Nadal	Italian
Djokovic	Paris

Summary of types of transformations: MODIFICATION, CONFORMATION, ADDITION (3)

- **ADDITION** (adding a new attribute, also called *augmentation*): In the same way, if you have a *Revenue* field in a Table maintained in Italy and another *Revenue* Field from Germany, and you need a Total Revenue in your target warehouse, you will write a function which **calculates** the sum and stores it in another column. All these modifications, additions, conformation are part of the Transform stage. These transformations must be encoded in **RULES** readable by non-ICT users.
- **IMPORTANT: the SYNTAX and SEMANTICS of the data you combine and store is a CRITICAL FACTOR. Syntactic and semantic mismatches are a major source of problems when aggregating data!**
- You will practice on these transformations during Labs

Example of ADDITION: computing a new attribute

NUM	Debt	Income
	10.134	100.000
	85.234	134.000
	8.112	21.500
	0	45.900
	17.534	52.000



$\frac{\text{Debt}}{\text{Income}}$

NUM	Debt to Income Ratio
	0,10
	0,64
	0,38
	0
	0,34

Transforming unstructured data

- Way more complex! First, we need to transform from unstructured to structured
- Example: sentiment analysis in Twitter

F	G	H	I	J
location	sentiment	contents	authname	
Omaha, Neb	-1	Starbucks computer glitch means free drinks http://t.co/cD4Lf6GHaj	KETV NewsWatch 7	
	0	RT @ArianaGrande so starbucks is closing i'm pregnant my nudes leaked & s aœ`ã...%æ		
Stark County	0	RT @sanctuarymg Will you be cheering at the @YMCASTark #NorthCanton4ti	YMCA of Cntrl Stark	
valdivia - chil	0	OHhh muy cansada pero alfin en starbucks ðŸ™ˆ	frafrafran	
Tampa, FL	1	I live off of Starbucks Arizona tea and blue cherry Gatorade ðŸ™ˆ...	Sky	
	1	RT @camillacabello97 today i walked around New York City with a hot Starbu	LERN JERGI	
Hawthorne, N	0	@JosilynnLoren Starbucks on el segundo and hawthorne	senpai	
	1	Time to relax ðŸ™ˆ #Starbucks http://t.co/u2oEkloMdV	Elizabethã ðŸ™ˆ,	
Washington	1	@Alex_is_coded within a year you could be a manager if you work hard enou	Versace Princess	
Cosby, TN	0	RT @EverythingGoats I goat some starbucks ã`*i, ðŸ™ˆ http://t.co/6u2V	Jordan Self	
	1	RT @Luis_v76 I want a vanilla bean frappe from Starbucks	ã` i, ã` i, ã` i,	
	0	RT @AminePosey Vous allez au Starbucks pour la boisson ou pour Instagram	EL MAESTROãœCE	
Buenos Aires	0	"Yo que querã-a ir a starbucks con con vos:ccc" que linda que es como la qu ã`*		
Hawaii Pacifi	0	I'm the asshole that asks Coffee Bean if they have (a variation of) a Starbucks	Michelle C	

Here, the challenge is to analyze text and, first, identify those of interest (e.g. talking about your company or a given product) and then, assign to the text a positive, negative or 0 (neutral) score.

Hermosillo Sonora	Starbucks repara falla tã@cnicã y reanuda el servicio http://t.co/Z5WWgyfzV Rebeca Dessens
italy	@CiccioBa ti fari la musica che sparano qui da Starbucks. che te ne fai del Qui The new londoner

Transforming unstructured data (2)

- What you get from this transformation (let's ignore HOW for now)?

Table: Starbucks Twitter Sentiment

date	positive	negative	neutral
1/04/2016	500	237	1715
2/04/2016	451	277	2015
3/04/2016	816	300	3016

Transformation: aggregating heterogeneous data


- We already mentioned a simple example of aggregation (summing revenues data from different DBs in maintained in different departments)
- Aggregation on heterogeneous data may be far more complex
- E.g. we may want to aggregate *sentiment data* with *sales* to discover what went wrong (or what was the winning move users appreciated best)

Example (Social Engagement Index)

- <http://www.brandamplitude.com/blog/innovation/item>
- /announcing-breakthrough-in-measuring-the-impact-of-social-media-on-sales



Summary of data transformation

- It may be relatively simple if data are homogeneous (come from the same source and are structured)
 - But this is the dream.. Usually data transformation is very complex and time-consuming and needs state-of-the-art software tools and also human supervision
 - By far the most complex step in ETL
- 

- Third step of ETL is Loading
- The ETL loading element is responsible for loading transformed data into the data warehouse database.
- The data warehouse is often taken *offline* during update operations so that data can be loaded faster
- If data are real-time streams (sensor data, social data..), or near-real time approach is used, then out-of-service is not perceived
- Loading basically implies to decide the destination and the updating frequency, that can be different for different sources (plus other security requirements)

DW ETL Tools

- **Some of the Well Known ETL Tools**
- The most well known commercial tools are [Ab Initio](#), [IBM InfoSphere DataStage](#), [Informatica](#), [Oracle Data Integrator](#) and [SAP Data Integrator](#).



Case Study (HW 4)

- Download the paper at <http://bmjopen.bmj.com/content/bmjopen/6/8/e010962.full.pdf> describing the use case of Dutch Red Cross data warehouse (also on course web site)
- Answer the following:
 - What type of data have been integrated, from which sources?
 - Can you draw the schema of all needed tables?
 - What are the objects? What are the attributes? What are the relationships? What is the “semantics” of relationships?
 - Can you list some of the TRANSFORM operations that were needed to harmonize data during the ETL process?
 - Which additional challenges are posed to the warehouse by the specific application domain?
 - Can you list the main categories of data which have been integrated?
 - Can you list and summarize the main data analytic tasks supported by the warehouse?