Data Wrangling and Data Analysis

Data Models

Hakim Qahtan

Department of Information and Computing Sciences
Utrecht University

Utrecht University

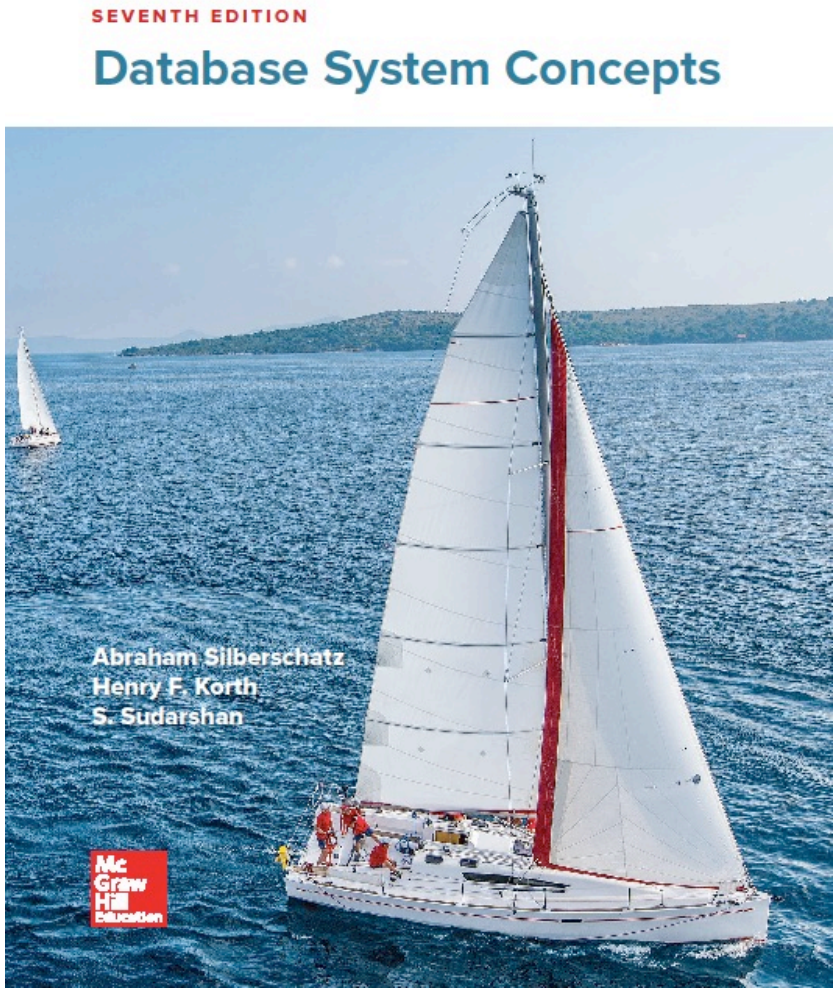
Reading Material for Today

Database System Concepts (7th Edition)

CH 1, 3.2, 3.9

Office Hours: Friday 10:00 – 11:45

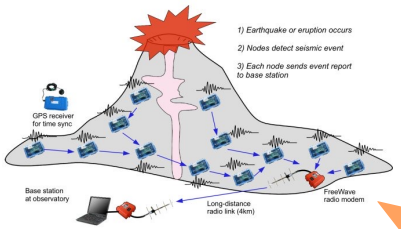
No office hours next week

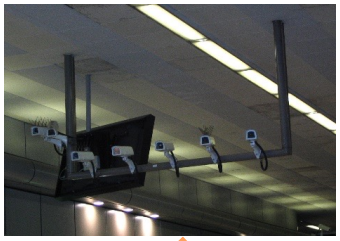


Utrecht University

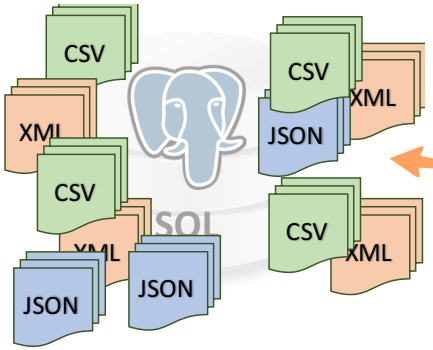
Introduction

We Collect Huge Amount of Data





Data Streams

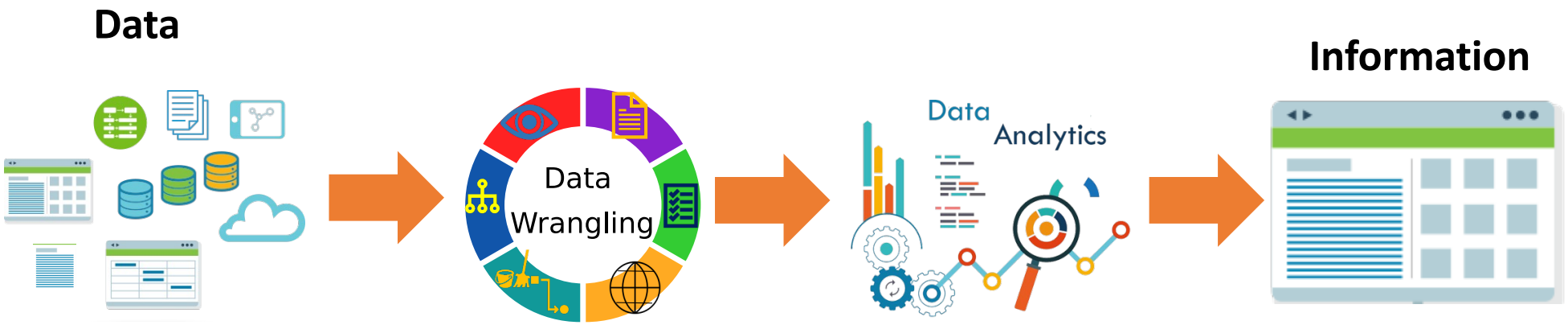


Static Data

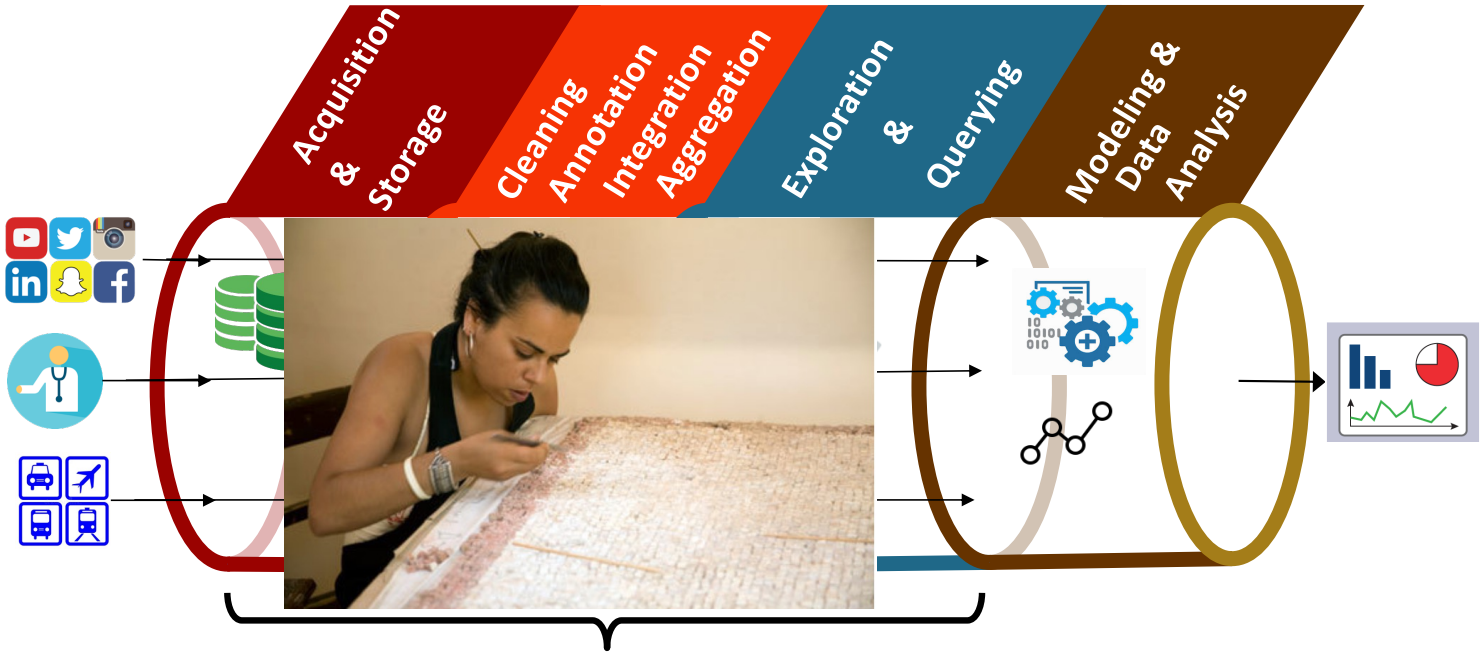
❖Merck (a pharma company) has:

- ~ 4000 Oracle databases
- Plus a data lake
- Plus untold files
- Plus untold spreadsheets
- Plus they are interested in public data from the web

Extracting Value from the Data



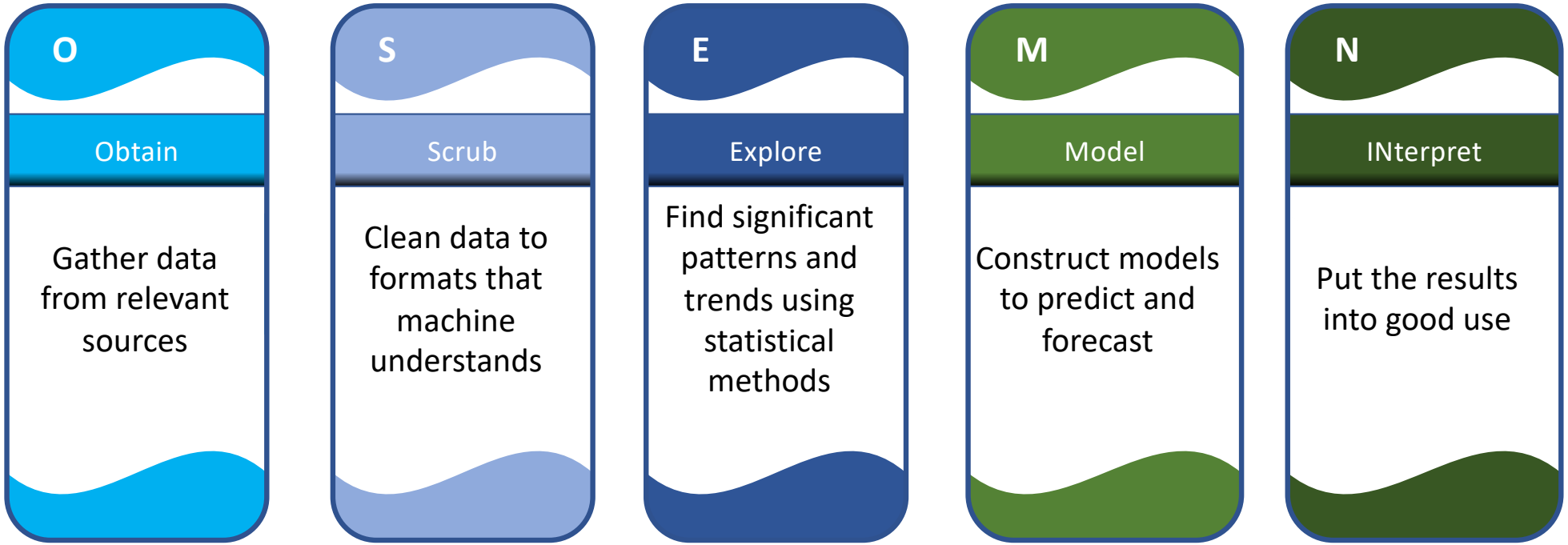
The Data Processing Pipeline



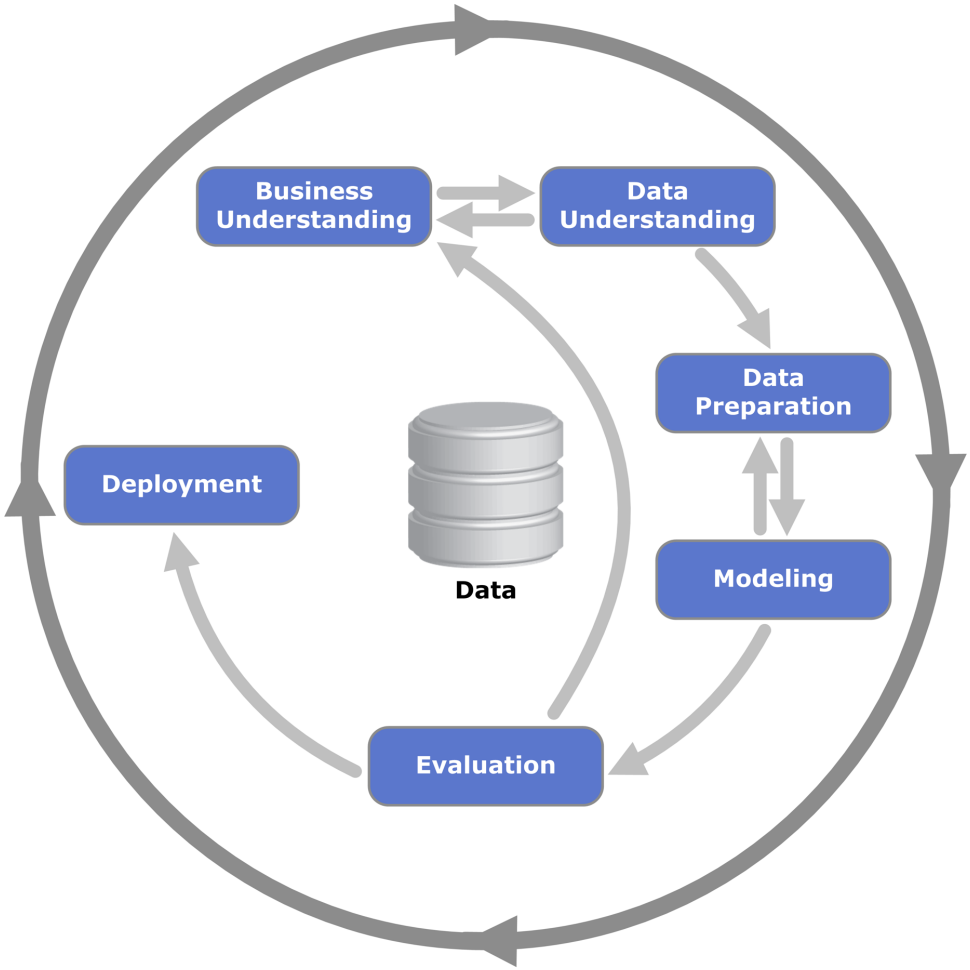
Data scientists, according to interviews and expert estimates, spend **50% to 80%** of their time mired in the mundane labor of collecting and preparing unruly digital data, before it can be explored for useful nuggets

(New York Times)

OSEMN Framework

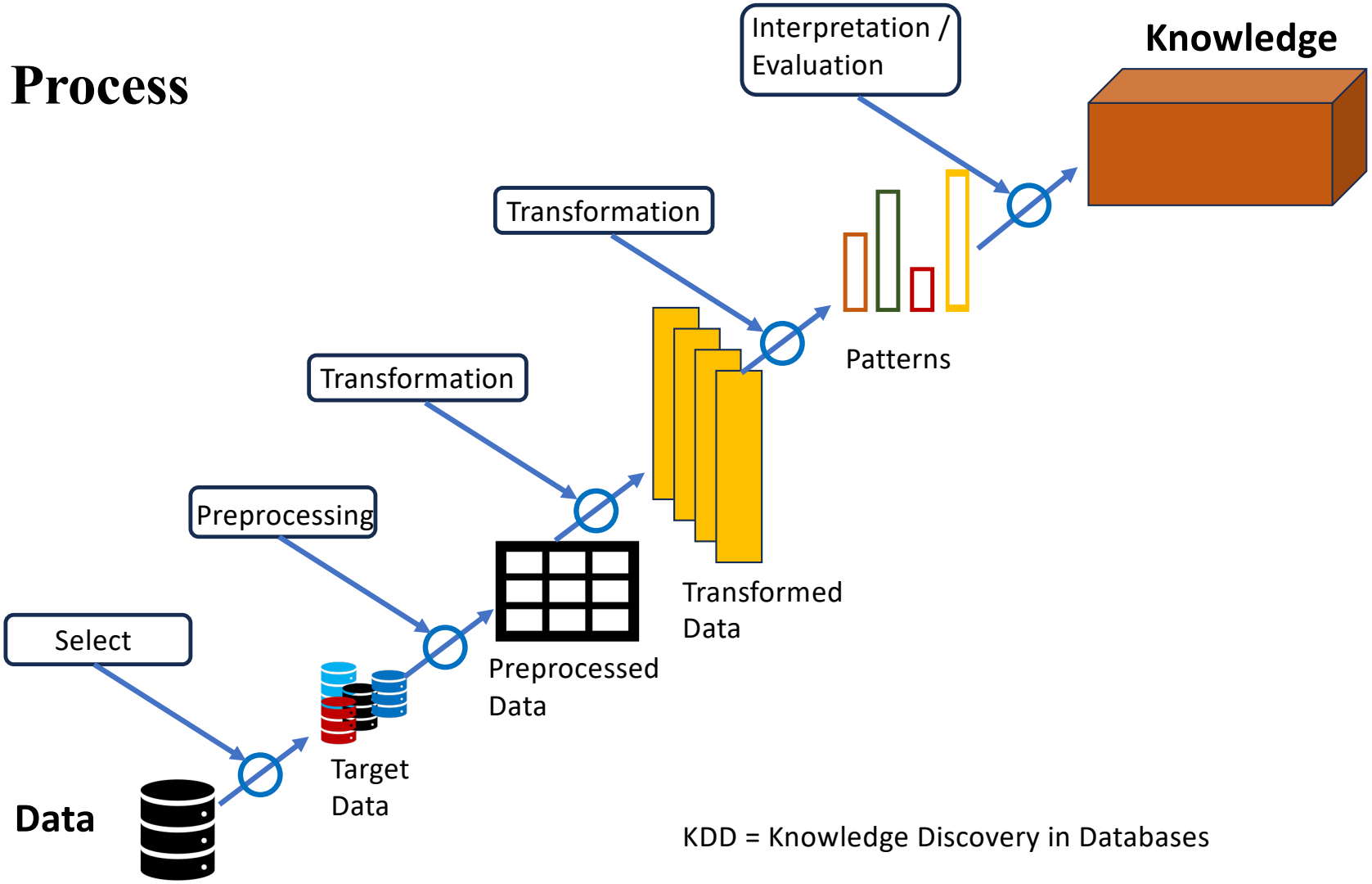


CRISP-DM



CRISP-DM = CRoss Industry Standard Process for Data Mining

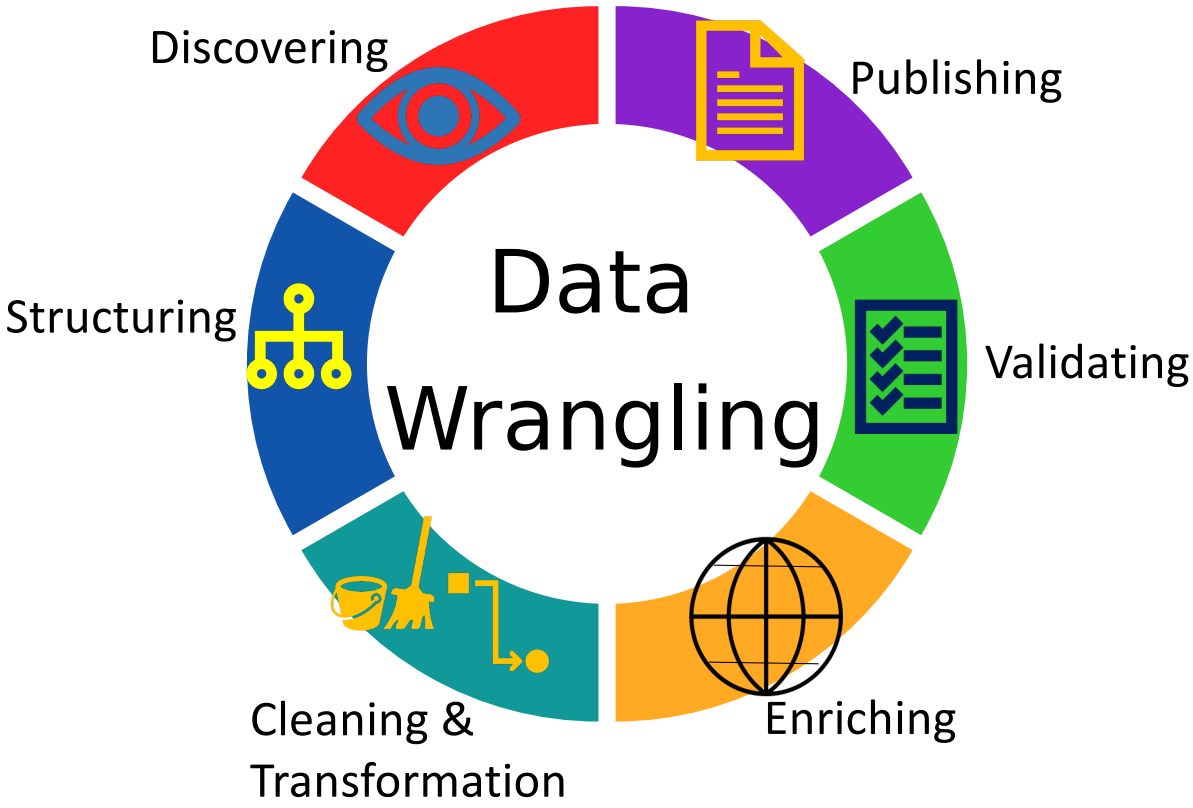
KDD Process



KDD = Knowledge Discovery in Databases

What is Data Wrangling?

- Data Wrangling (Munging): an iterative process to convert the raw data into more understandable format.



Data Wrangling

- Discovering
  - Understanding the data with respect to the application
  - Creating a plan for preparing the data (structuring, cleaning and normalizing the data).

- Structuring
  - Defining a structure that suits the analytical model
  - Various datasets should be in compatible formats
  - Data may come from different sources (Union, Join)

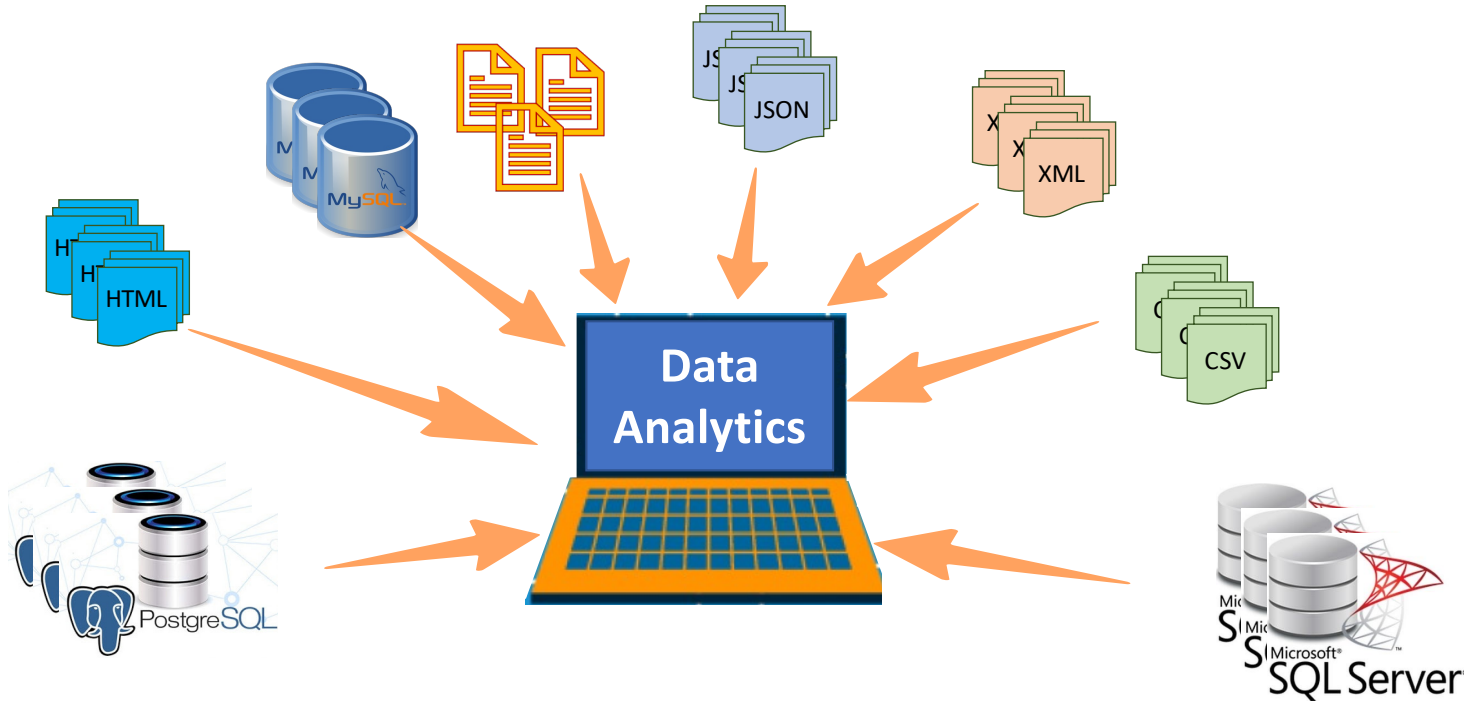
Data Wrangling

- Transformation and cleaning involve
  - Normalizing the data, removing redundant and irrelevant data, combining multiple datasets.
  - Removing errors, de-duplication, fixing inaccuracies and inconsistences, and handling data bias.
- Enriching
  - Do you have the required data to perform accurate analysis?
  - If the data is not enough, can we add more examples/features?
  - What about enriching with synthetic data?

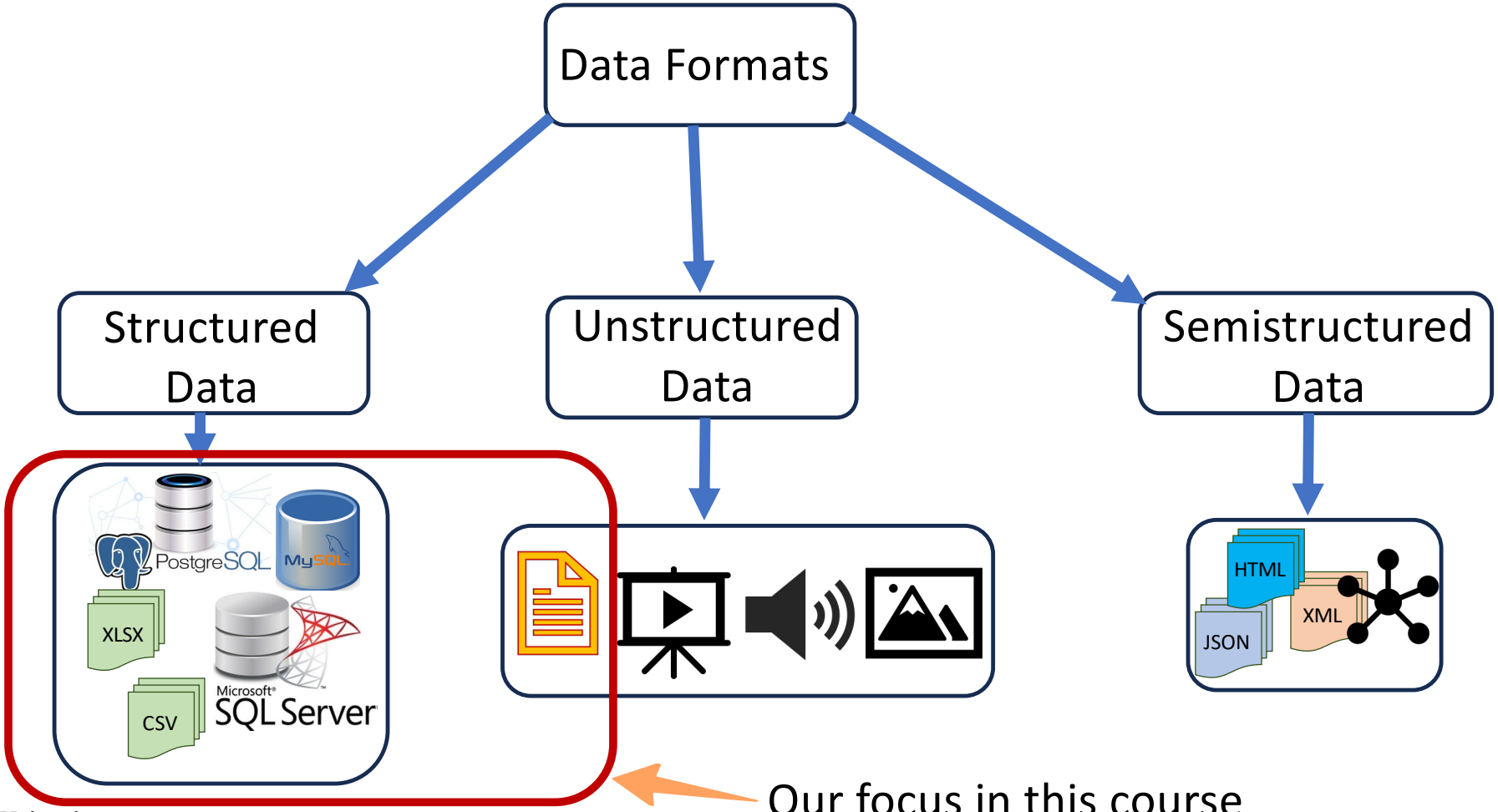
Data Wrangling

- Validating
  - Confirm whether the steps that were done during the structuring, transformation and cleaning are correct or not.
- Publishing
  - Choose the file format
  - Store the data in a place where it can be accessed by others
  - In many cases, data privacy should be considered before sharing the data (anonymize the data).

We Collect Huge Amount of Data



Data is stored in **different formats**


Data Formats
Structured Data
Unstructured Data
Semistructured Data
PostgreSQL
MySQL
XLSX
CSV
Microsoft SQL Server
HTML
XML
JSON
Our focus in this course

Metadata

- Data about data
  - Describes the context of the data, the attributes and their domains.
  - How the data was collected and where it is stored.
  - How the data can be used.

Data Models

Utrecht University

Data Models

- **Data model:** mathematical representation of the data**.**
  - Relational model: uses tables to represent both data and relationships
  - Entity-Relationship (E-R) model: Conceptual model
    - Data is represented as a collection of entities (real-world objects) and the relationships among the objects
  - Semistructured model: different objects may have different sets of attributes (e.g. JSON and XML)
  - Object-Based data model: entities (real-world objects) are represented using objects.
    - Currently, the concept of object is integrated into relational databases.

Data Models

- Data models can be viewed as a collection of tools describing:
  - Data
  - Data relationships
  - Data semantics
  - Data constraints
  - Data operations

Data Models

- Data models are said to contain three main components:
  - Structures: define the format
    - Rows and columns?
    - Nodes and edges?
    - Key-value pairs?
  - Constraints: define the relationships and semantics
    - All rows must have the same number of columns
    - Column A3 contains numerical values
    - The age must be positive
  - Operations: store and retrieve the data
    - Return the values of record x
    - Find the rows where the column "lastname" is "Miller"

Data Models

Structure – The Relational Model

Utrecht University

The Relational Model

- Relation (R): made up of 2 parts:
  - Instance: a table, with rows and columns.
    #Rows = cardinality, #fields = degree / arity.
  - Schema: specifies name of relation, plus name and type of each column.
    - E.G. Students(sid string, name string, login string, age integer, gpa real).
- Relation: a set of rows or tuples (i.e., all rows are distinct).
  - There should be a subset of the attributes where each tuple differs from the other in at least one value (the concept of Keys).

The Relational Model

- Data stored in different tables
- Example of tabular data

Columns (Attributes, fields or Features)
#columns = degree

Rows (Records or tuples)

#row = Cardinality

ID	name	dept_name	Salary
22322	Einstein	Physics	95000
21212	Wu	Finance	90000
32343	El Said	History	86000
43521	Katz	Comp. Sci.	75000
98531	Kim	Biology	78000
58763	Crick	Elec. Eng.	80000
52187	Mozart	Music	65000
33452	Katz	Physics	78000

Values of ID are different for each tuple

Relations are Unordered

- Order of tuples is not considered (tuples might be stored in an arbitrary order)
- The same is also true in the case of attributes
  - Example: putting “salary” before/after “dept_name” will have no effect on the relation (table)

ID	name	dept_name	salary
22322	Einstein	Physics	95000
21212	Wu	Finance	90000
32343	El Said	History	86000
43521	Katz	Comp. Sci.	75000
98531	Kim	Biology	78000
58763	Crick	Elec. Eng.	80000
52187	Mozart	Music	65000
33452	Gold	Physics	87000

The Relational Model

The *instructor* table

ID	name	dept_name	Salary
22322	Einstein	Physics	95000
21212	Wu	Finance	90000
32343	El Said	History	86000
43521	Katz	Comp. Sci.	75000
98531	Kim	Biology	78000
58763	Crick	Elec. Eng.	80000
52187	Mozart	Music	65000
33452	Katz	Physics	87000

Unique values

The *department* table

dept_name	building	Budget
Comp. Sci.	Turing	1000000
Physics	Watson	800000
Chemistry	Ibn-Hayyan	850000
Biology	AlRazi	800000
Elec. Eng.	Taylor	690000
Music	Packard	400000
Finance	Painter	1200000
History	Painter	500000

Unique values

Keys

- Let $K \subseteq R$
- K is a superkey of R if values for K are sufficient to identify a unique tuple of each possible relation $r(R)$
  - Example: *{ID}* and *{ID,name}* are both superkeys of instructor.
- Superkey K is a candidate key if K is minimal
  Example: *{ID}* is a candidate key for Instructor
- One of the candidate keys is selected to be the **primary key**.
  - Which one?
- **Foreign key**: Value in one relation must appear in another
  - **Referencing** relation
  - **Referenced** relation
  - Example – dept_name in instructor is a foreign key in the instructor relation referencing department

Attribute Types

- The set of allowed values for each attribute is called the **domain** of the attribute
- Attribute values are (normally) required to be **atomic**; that is, indivisible
- The special value ***null*** is a member of every domain. Indicating that the value is "unknown"
- The null value causes complications in the definition of many operations

Relation Schema and Instance

- Let $A_1, A_2, \ldots, A_n$ be the names of attributes
- $R = (A_1, A_2, \ldots, A_n)$ is a relation schema
- Given sets $D_1, D_2, \ldots, D_n$ from which $A_1, A_2, \ldots, A_n$ take their values, we call $r \subseteq D_1 \times D_2 \times \cdots \times D_n$ a relation instance or simply a relation
- An element t in r is a tuple and represented as a row in the table

Data Models

Constraints

Utrecht University

Constraints

- Guard against accidental damage to the data
- Ensure that authorized changes to the data do not violate data consistency.
  - An account must have a balance greater than $10.00
  - A salary of a bank employee must be at least $4.00 per hour
  - A customer must have a (non-null) phone number
- Constraints on a single relation
  - Not null
  - Primary key
  - Unique
  - Check (P), P is a predicate

Data Models

Operations

Utrecht University

Operations

- Define a set of operations for accessing and manipulating the data
  - Create a relation
  - Add more records/attributes (tuples/columns)
  - Delete or alter a set of records/attributes/values
  - Retrieve a set of records
  - …

How to store the data?

File system as data storage

- Drawbacks of using file systems to store data include:
  - Data redundancy and inconsistency
    - Multiple file formats, duplication of information in different files
  - Difficulty in accessing data
    - Need to write a new program to carry out each new task
  - Data isolation
    - Data are scattered in various files with (possibly) different formats
  - Integrity problems
    - Integrity constraints (e.g., account balance > 0) become “buried” in program code rather than being stated explicitly
    - Hard to add new constraints or change existing ones

File system as data storage (Cont.)

- Drawbacks of using file systems to store data include:
  - Atomicity of updates
    - Failures may leave database in an inconsistent state with partial updates carried out
    - Example: Transfer of funds from one account to another should either complete or not happen at all
  - Concurrent access by multiple users
    - Concurrent access needed for performance
    - Uncontrolled concurrent accesses can lead to inconsistencies
      - Example: Two people reading a balance (say 100) and updating it by withdrawing money (say 50 each) at the same time
  - Security problems
    - Hard to provide user access to some, but not all, data

Database systems offer solutions to all the above problems

10 minutes break!

Databases

Utrecht University

Databases

- Traditionally: systems that contain records about real world entities
  - Employee records, bank records, etc.
- Today, the field covers all the largest sources of data, with many new ideas.
  - Web search.
  - Data mining.
  - Scientific and medical databases.
  - Graph databases.
  - Integrating information.

Databases – More

- You may not notice it, but databases are behind almost everything you do on the Web.
  - Google searches.
  - Queries at Amazon, eBay, etc.
- DBMS contains information about a particular enterprise
  - Collection of interrelated data
  - Set of programs to access the data
  - An environment that is both *convenient* and *efficient* to use

Database – Example

- University Database Example:
  - Add new students, instructors, and courses
  - Register students for courses, and generate class rosters
  - Assign grades to students, compute grade point averages (GPA) and generate transcripts

Database levels of abstraction

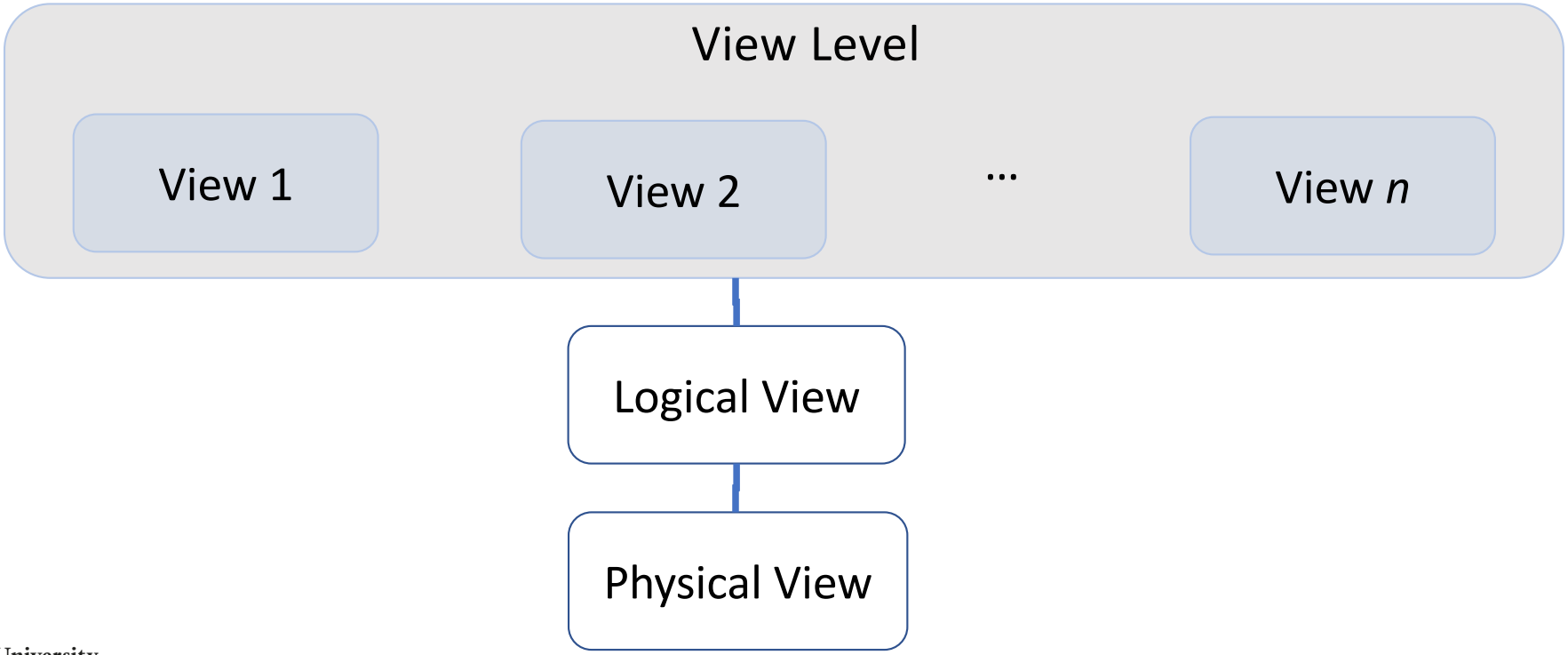
- **Physical level:** describes how a record (e.g., instructor) is stored.
- **Logical level:** describes data stored in database, and the relationships among the data.

```
type instructor = record
        ID : string;
        name : string;
        dept_name : string;
        salary : real;
        end;
```

- **View level:** application programs which hide details of data types. Views can also hide information for security/privacy purposes (such as an employee's salary).

Database levels of abstraction

- Architecture of a database system



Data Modeling in Databases

Utrecht University

Data Definition Language (DDL)

- Specification notation for defining the database schema
- DDL is used to create and modify database objects such as tables, indexes and users

  Example: create table instructor (
  ID **char**(5),
  name **varchar**(20),
  dept_name **varchar**(20),
  salary **real**);

- DDL include:
  - Commands for specifying integrity constraints
  - Commands for defining views
  - Commands for specifying access rights

Data Manipulation Language (DML)

- Language for accessing and manipulating the data organized by the appropriate data model
  - DML also known as query language to retrieve information from the database
  - DML allows for adding (inserting), deleting, and modifying (updating) the records in a database

DDL and DML are not two separate languages – they form parts of a

single database language (e.g. SQL)

SQL

Utrecht University

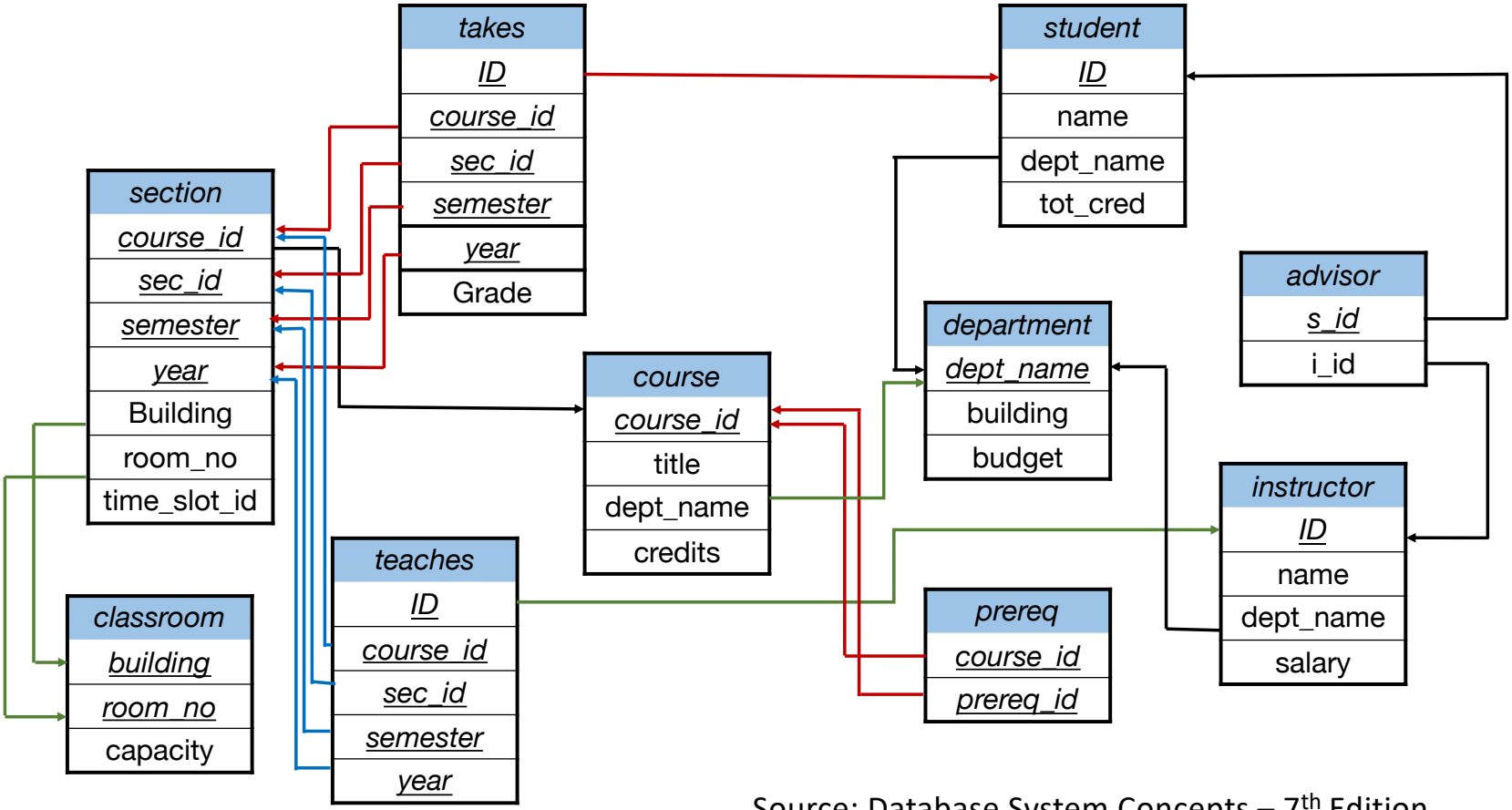
Structured Query Language (SQL)

- High level language to manipulate relational data
  - Say “what to do” rather than “how to do it.”
  - Avoid a lot of data-manipulation details needed in procedural languages like C++ or Java.
- Database management system figures out “best” way to execute query.
  - Called “query optimization.”
- SQL has three categories based on the functionality involved
  - DDL: Data Definition Language
  - DML: Data Manipulation Language
  - DCL: Data Control Language -- used to grant and revoke authorization.

Structured Query Language (SQL)

- SQL DDL is used to define:
  - The schema for each relation.
  - The types of values associated with each attribute.
  - The integrity constraints.
  - The security and authorization information for each relation.

Example: Schema Diagram for University Database



Source: Database System Concepts – 7th Edition

Creating Relations in SQL

- Creating relation in SQL can be done as:

```
CREATE TABLE tab_name (
        A1 D1,
        A2 D2,
        …,
        An Dn,
        ⟨IC1⟩,
        ⟨IC2⟩,
        …,
        ⟨ICn⟩
        );
```

- Di is the domain of the values in attribute (column) Ai
- IC = integrity constraint

Creating Relations in SQL – Examples

- Creating the Student relation
  - Observe that the type (domain) of each field is specified, and enforced by the DBMS whenever tuples are added or modified.

```
CREATE TABLE student (
        id CHAR(20),
        name CHAR(20),
        dept_name CHAR(20),
        tot_cred INTEGER   );
```

- Another example, the `takes' table holds information about courses that students take.

```
CREATE TABLE takes (
        id CHAR(20),
        course_id CHAR(20),
        sec_id CHAR(2),
        semester CHAR(3),
        year NUMERIC(4,0),
        grade REAL    );
```

Deleting and Altering Relations

```
DROP TABLE students
```

- Destroys the relation `student'. The schema information *and* the tuples are deleted.

```
ALTER TABLE students
        ADD COLUMN firstYear integer;
```

- The schema of `student' is altered by adding a new field; every tuple in the current instance is extended with a null value in the new field.

Insertion

- Can insert a single tuple using:

```
INSERT INTO student
VALUES (53688, `Smith', `ICS', 18);
```

```
INSERT INTO Student (name, id, tot_cred, dept_name)
VALUES (`Smith', 53688, 18, `ICS');
```

Deletion

- Can delete all tuples satisfying some condition (e.g., name = Smith):

```
DELETE
FROM student S
WHERE S.name = `Smith'
```

- Deleting all records from a relation

```
DELETE  FROM student
```

Update

- Change specific values of the tuples:

```
UPDATE  tab_name
SET Ai = vi
[WHERE Aj = vj  ]
```

- Example: change the department name from Math to Mathematics

```
UPDATE  department
SET dept_name = `Mathematics'
WHERE dept_name = `Math'
```

Wrap-Up

- Summerize what you learned today in 2-minutes

The information in this presentation has been compiled with the utmost care,
but no rights can be derived from its contents.