24/09/2023

# DATA MANAGEMENT

ITC / DEPARTMENT OF GEO-INFORMATION PROCESSING

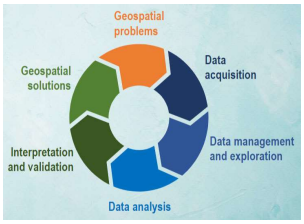
Presented by Lucas De Oto - September 2023

ITC

UNIVERSITY OF TWENTE.

1

## OVERVIEW




UNIVERSITY OF TWENTE.

2

## OVERVIEW

- Databases and DBMS
  - Definitions and basic terminology
- Relational data model
  - Concepts and terminology
  - Domain, Attribute, Relation, Schema, SQL
  - Constraints
- GIS and DBMS
- Spatial databases
- GDB design

UNIVERSITY OF TWENTE.

3

## DATABASE

DATA, DATASET, DATABASE, DBMS AND DB SYSTEM

**Data**

- Is a **resource** held on paper or in digital format that serves to record or administer some **facts and descriptions of phenomena of interest.**

**Data set** (or dataset):

- A homogeneous **collection of data** normally describing a single kind of phenomenon

**Database (DB)**

- A **collection of interrelated data sets properly structured** by means of, and stored through a DBMS (Data Base Management System)

UNIVERSITY OF TWENTE.

4



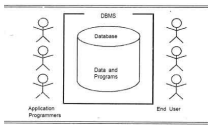
24/09/2023

## DATABASE

DATA, DATASET, DATABASE, DBMS AND DB SYSTEM

**Database management system (DBMS)**

- A **software package** that is designed for the purpose of managing databases**.** This means, DBMS allows to set-up, maintain and explore one or more databases**.**




**Database system**

- Combination of **a database and its DBMS.**

UNIVERSITY OF TWENTE.

5

## DATABASE MANAGEMENT SYSTEM (DBMS)

WHAT DOES A DBMS OFFER TO USERS?

- Supports proper storage and manipulation of ***very large data sets***.
- Can be instructed to guard over ***data correctness***.
- Allows the ***control of data redundancy.***
- Supports the ***concurrent use*** of the same data set by many users.
- Includes data ***backup and recovery functions*** to ensure data protection and availability at all times.

UNIVERSITY OF TWENTE.

6



24/09/2023

## TERMINOLOGY

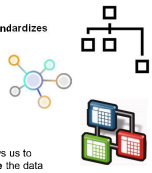
DATA MODEL

A DBMS works based on a ceratin **data model.**

A data model **organizes elements of data and standardizes how they relate to one another**

A data model is an integrated collection of:

- **Data structuring primitives,**
- **Rules of how to structure, and**
- **Mechanisms to handle the data**



In other words, a data model is a **toolbox** that allows us to **create/define** a database structure and **manipulate** the data stored in it**.**

UNIVERSITY OF TWENTE.

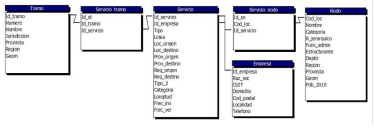
7

## RELATIONAL DATA MODEL

WHY STUDY RELATIONAL DATA MODEL?

There are different data models: hierarchical, network, object-oriented, etc. → DBMS linked to most GIS packages make use of the **relational data model**

The **relational data model** structures a DB as a **collection of inter-related tables (or *relations*)** → we model a piece of reality as linked tables



UNIVERSITY OF TWENTE.

8

24/09/2023

## TERMINOLOGY
THE LANGUAGE USED IN RELATIONAL DATA MODEL

**Structured Query Language *(SQL)*** – the relational database language:

- SQL is a DSL (*domain-specific language*) used in programming and **designed for managing data held in a relational DBMS.**
- **Natural language** → its syntax is easy to read
- Developped in the 70's → initially called SEQUEL (Structured English QUEry Language)
- **Particularly useful in handling structured data where there are relations between different entities/variables of the data.**

UNIVERSITY OF TWENTE.

9

## TERMINOLOGY
THE LANGUAGE USED IN RELATIONAL DATA MODEL

Find all cities (codes and names) that belong to the department called Tilcara

```
SELECT  Code, Name
FROM    A
WHERE   Dep = Tilcara
```

A

| Code | Name | Dep |
|---|---|---|
| 38077020 | Cerrega de Paicone | Santa Catalina |
| 38077080 | Paicone | Santa Catalina |
| 38084020 | Coranzuli | Susques |
| 38084090 | San Juan de Quillaques | Susques |
| 38084080 | Puesto Sey | Susques |
| 38084040 | Huancar | Susques |
| 38084070 | Pastos Chicos | Susques |
| 38084060 | Olaroz Chico | Susques |
| 38084050 | Mina Providencia | Susques |
| 38084030 | El Toro | Susques |
| 38084055 | Olacapato | Susques |
| 38084045 | Jama | Susques |
| 38084100 | Susques | Susques |
| 38084010 | Cinsa | Susques |
| 38094050 | Tilcara | Tilcara |
| 38094040 | Maimara | Tilcara |
| 38094010 | Colonia San Jose | Tilcara |
| 38094030 | Juella | Tilcara |
| 38094020 | Huacalera | Tilcara |
| 38098010 | Barcena | Tumbaya |
| 38098050 | Volcan | Tumbaya |

UNIVERSITY OF TWENTE.

10



## TERMINOLOGY
RELATIONAL DATA MODEL

The relational data model is an integrated collection of:

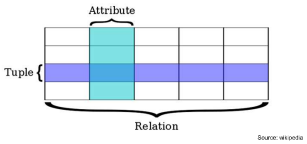
1. Data structuring primitives = **attributes, tuples,** and **relations**
2. Rules of how to structure = **data definition language**
3. Mechanisms to handle the data in a database = **data manipulation language.**

UNIVERSITY OF TWENTE.

11

## TERMINOLOGY
DATA STRUCTURING PRIMITIVES

- **Attribute**: properties that describe an entity
- **Tuple**: the entire row of attribute's values corresponding to a particular entity
- **Relation** *(or **table**)*: a collection of tuples that are similarly shaped




Source: wikipedia

UNIVERSITY OF TWENTE.

12



24/09/2023

24/09/2023

## TERMINOLOGY
ATTRIBUTE

**Attribute**

- In the relational data model, we represent **real-world objects** by **tuples** stored in **relations** (*tables*).
- These real-world objects have particular **properties** that describe them. These properties are called **attributes.**

The relation *Productions* has these attributes: *cid*, *crop*, *annum*, *score*, *quality*

Productions

| cid | crop | annum | score | quality |
|---|---|---|---|---|

UNIVERSITY OF TWENTE.

13

## TERMINOLOGY
DOMAIN

**Domain**

- A set of acceptable values that an attribute is allowed to contain
- Original set of **atomic values** used to model data.
- By atomic value, we mean that each value in the domain is **indivisible** as far as the relational model is concerned; e.g. the domain of the attribute *Day* has the set of all possible days: {Mon, Tue, Wed…}.
- An attribute domain belongs to a certain **data type (also called domain)**
- There are many system-defined data types/domain: Numeric (Integer, Floating-point); Character (Char, Varchar); Temporal (Date, Time); Boolean (true/false, y/n)

UNIVERSITY OF TWENTE.

14



## TERMINOLOGY
ATTRIBUTE & DOMAIN

Each attribute has an associated domain:

- *In the relation Productions:*
  - ✓ *cid: integer*
  - ✓ *crop: varchar(255)*
  - ✓ *annum: integer*
  - ✓ *score: integer*
  - ✓ *quality: varchar(255)*

Productions

| Field Name | Data Type | Description |
|---|---|---|
| cid | Number | country code |
| crop | Text | crop type |
| annum | Number | year in which produced |
| score | Number | production achieved, in tonnes |
| quality | Text | * = Unofficial figure, [ ] = Official data, F = FAO estimate, M = Data not available |

UNIVERSITY OF TWENTE.

15

## TERMINOLOGY
NULL VALUES

Some attributes may have **missing values.** An attribute value may be:

- '**unknown**'
- '**not applicable**'

} In SQL we use **NULL** in both cases

**NULL** *(or Null)*

- A special marker used in SQL to indicate that a data value does not exist in the DB.
- It enables the representation of missing and inapplicable information
- **Not to be confused with a value of 0**

UNIVERSITY OF TWENTE.

16

## TERMINOLOGY
TUPLE

**Tuple**

- A tuple is a **record**
- In the relational data model, tuples represent **real-world objects/phenomena** that have certain attributes
- A tuple can be defined as **a list of attribute values**

One tuple from the relation *Production* is: *(cid=3, crop="Rice,paddy", annum=2000, score=300, quality="F")*

| cid | crop | annum | score | quality |
|---|---|---|---|---|
| 3 | Rice, paddy | 2000 | 300 | F |

UNIVERSITY OF TWENTE.

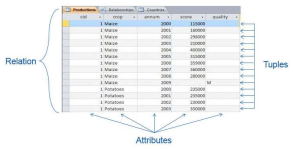
17

## TERMINOLOGY
RELATION (or TABLE)

**Relation (*Table*)**

- A **collection of tuples** that are **similarly shaped:** all tuples have the same attributes

The relation *Productions* in *FAOcrops.mdb*




UNIVERSITY OF TWENTE.

18



## TERMINOLOGY
RELATION SCHEMA AND RELATION INSTANCE

**Relation schema**

- Basic information describing a table or relation:
  - **Name** of the relation;
  - List of **attributes**;
  - **Domain** of each attribute

**Relation instance**

- Set of tuples that adheres to all the requirements that are formulated by the relation schema
- The **set of tuples in a relation at some point in time**

UNIVERSITY OF TWENTE.

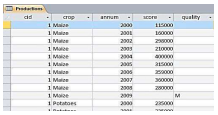
19

## TERMINOLOGY
RELATION SCHEMA AND RELATION INSTANCE

- **Relation schema**

*Productions (cid: integer, crop: varchar(255), annum: integer, score: integer, quality: varchar(5))*

- **Relation instance –**



on 18.09.2021 – 15:47 relation *Productions* had five attributes and 6930 tuples

UNIVERSITY OF TWENTE.

20





## TERMINOLOGY
RELATIONAL DATA MODEL

The relational data model is an integrated collection of:

1. Data structuring primitives = **attributes, tuples,** and **relations**
2. Rules of how to structure = **data definition language**.
3. Mechanisms to handle the data in a database = **data manipulation language.**

UNIVERSITY OF TWENTE.

21

## TERMINOLOGY
DATA DEFINITION LANGUAGE

**Data Definition Language (DDL)**

- A (subset of a) computer language used to create and modify the **structure** of database objects in a database:
- Set of commands that can be used to define the database **schema:** CREATE, ALTER, DROP

```
CREATE TABLE Productions(
                    cid single,
                    crop varchar(255), annum integer,
                    score integer, quality varchar(255)
                    )
```

| cid | crop | annum | score | quality |
|---|---|---|---|---|
| | | | | |
| | | | | |
| | | | | |

Productions

UNIVERSITY OF TWENTE.

22





## TERMINOLOGY
RELATIONAL DATA MODEL

The relational data model is an integrated collection of:

1. Data structuring primitives = **attributes, tuples,** and **relations**
2. Rules of how to structure = **data definition language**.
3. Mechanisms to handle the data in a database = **data manipulation language.**

UNIVERSITY OF TWENTE.

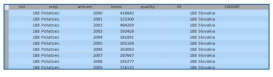
23

## TERMINOLOGY
DATA MANIPULATION LANGUAGE

**Data Manipulation Language (DML)**

- A (subset of a) computer language used for **adding** (*inserting*), **deleting**, and **modifying** (*updating*) **data** in a DB
- Set of commands that deals with the **manipulation of data:** SELECT, INSERT, UPDATE, DELETE

```
SELECT *
FROM Productions AS p, Countries AS c
WHERE c.ID=p.cid AND p.crop="Potatoes" AND
c.CNAME="Slovakia"
```



UNIVERSITY OF TWENTE.

24

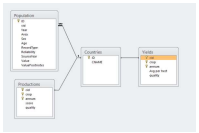
## TERMINOLOGY
RELATIONAL DATABASE

**Relational database**

- A DB based on the **relational data model**
- A collection of relations (*tables*) **structured to recognize relations** *(links/associations)* between stored items of information.

**Database schema**

- The DB **structure** described in a formal language
- A **collection** of relation schemas and the **associations** amongst them
- The **skeleton structure** that represents the **logical view** of the entire database



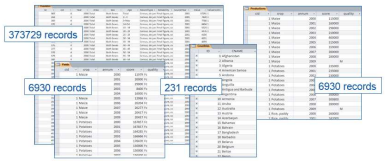
The GDB *FAOcrop.mdb* contains 4 relations: *(Productions, Yields, Countries, Population)*



## TERMINOLOGY
DATABASE INSTANCE

**Database instance**

- A **collection of relation instances**, one for each relation in the database schema




*FAOcrops.mdb* instance on 18.09.2021 at 15:47






## TERMINOLOGY
INTEGRITY CONSTRAINTS

**Integrity constraints**

- A **set of rules** that are used **to maintain the quality of information** in a DB
- **Ensure** that the data manipulation is performed in such a way that **data integrity is not affected.**
- Used to **guard against accidental damage/errors** to/in the DB
- Integrity constraints are **specified** when the schema is defined
- Also used to **establish links between relations/tables**





24/09/2023

## TERMINOLOGY
INTEGRITY CONSTRAINTS

**Integrity constraints**

- There are various types of integrity constraints that a DB provides

We will discuss 3 types:

1. **Domain** integrity constraint
2. Entity integrity constraint: **Primary key**
3. Referential integrity constraint: **Foreign key**



## INTEGRITY CONSTRAINTS
DOMAIN CONSTRAINTS

**Domain integrity constraints**

- Refers to the **definition of a valid set of values for an attribute**
- In order to add a new value, it needs to meet the criteria defined for that attribute. Otherwise, we get an error

```
CREATE TABLE Productions(
                cid single,
                crop varchar(255), annum integer,
                score integer, quality varchar(255)
                )
```

| cid | crop | annum | score | quality |
|---|---|---|---|---|
| | | | | |
| | | | | |
| | | | | |

Productions





24/09/2023

## INTEGRITY CONSTRAINTS
PRIMARY KEY

**Primary key**

- The primary key *K* of a relation *R* is one of (or a combination of) R's attributes, such that:
  1. It is **unique** – there are no two distinct tuples of R that have the same attribute value.
  2. It is **minimal** – there is no proper subset of K that is unique.

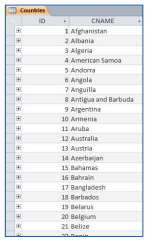


## INTEGRITY CONSTRAINTS
PRIMARY KEY

Example – For the relation *Countries* in *FAOcrops.mdb* we can think of three candidate keys:

*a. ID*
*b. CNAME*
*c. ID+CNAME*

*Which one would you choose as the primary key? Why?*



## INTEGRITY CONSTRAINTS
PRIMARY KEY

**Golden rule of keys:**

- Check suitability carefully → Whether one (or a set) of the attributes can become a primary key or not can never be judged from looking at a relation instance.
- Each stored relation in a DB **must have** a defined primary key.
- Primary keys are conceived and defined (or even created!) when designing the database schema

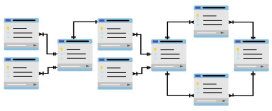
UNIVERSITY OF TWENTE.

33

## INTEGRITY CONSTRAINTS
REFERENTIAL INTEGRITY

**Foreign key**

- A set of attributes that is used to **refer to** a tuple in another relation.
- It must correspond with a primary key value in the second relation.
- A foreign key behaves like a 'logical pointer'.
- Links between **primary** and **foreign** keys determine the relationships amongst tables in a DB



UNIVERSITY OF TWENTE.

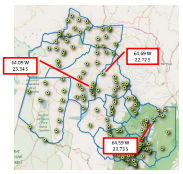
34





## GIS AND DATABASES
LINKING SPATIAL AND NON-SPATIAL DATA

**Spatial data** **Non-spatial data**

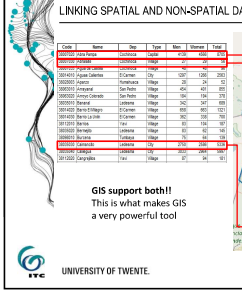


| Code | Name | Dep | Type | Men | Women | Total |
|---|---|---|---|---|---|---|
| 33007320 | Abra Pampa | Cochinoca | Capital | 4139 | 4080 | 8765 |
| 33007330 | Abralaite | Cochinoca | Village | 27 | 29 | 56 |
| 33007370 | Agua de Castilla | Cochinoca | Village | 48 | 42 | 90 |
| 33014810 | Aguas Calientes | El Carmen | City | 1297 | 1288 | 2585 |
| 33020003 | Aparzo | Humahuaca | Village | 28 | 24 | 52 |
| 33053010 | Arrayanal | San Pedro | Village | 454 | 401 | 855 |
| 33061020 | Arroyo Colorado | San Pedro | Village | 184 | 194 | 378 |
| 33033010 | Barcenal | Ledesma | Village | 342 | 347 | 689 |
| 33014420 | Barrio El Milagro | El Carmen | Village | 858 | 663 | 1521 |
| 33014430 | Barrio La Unión | El Carmen | Village | 362 | 330 | 692 |
| 33012310 | Barrios | Yavi | Village | 83 | 104 | 187 |
| 33033020 | Barrealito | Ledesma | Village | 83 | 62 | 145 |
| 33090010 | Burcena | Tumbaya | Village | 75 | 64 | 139 |
| 33033030 | Calilegua | Ledesma | City | 2734 | 2598 | 5332 |
| 33033040 | Calilegua | Ledesma | City | 3033 | 2864 | 5897 |
| 33012320 | Campanilla | Yavi | Village | 97 | 94 | 191 |

UNIVERSITY OF TWENTE.

37

## GIS AND DATABASES
LINKING SPATIAL AND NON-SPATIAL DATA



**GIS support both!!**
This is what makes GIS a very powerful tool

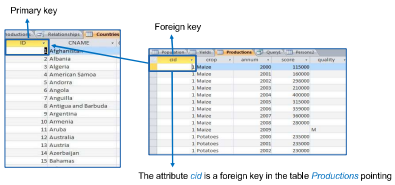
UNIVERSITY OF TWENTE.

38





## INTEGRITY CONSTRAINTS
REFERENTIAL INTEGRITY




The attribute *cid* is a foreign key in the table *Productions* pointing at the attribute *ID* of the table *Countries*. It only allows:

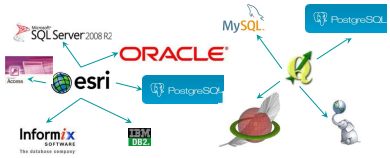
- ✓ Existing values in *ID*
- ✓ NULL values

UNIVERSITY OF TWENTE.

35

## DBMS AND GIS

A GIS **is not** a DBMS
Main GIS packages **can be linked** to a DBMS to store and manage data




UNIVERSITY OF TWENTE.

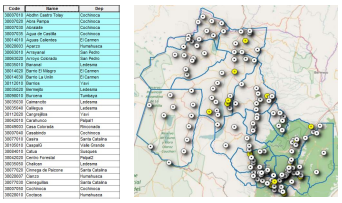
36





## GIS AND DATABASES
LINKING SPATIAL AND NON-SPATIAL DATA

Spatial and non-spatial data in a GIS environment are stored separately (.shp, .dbf) but dynamically linked

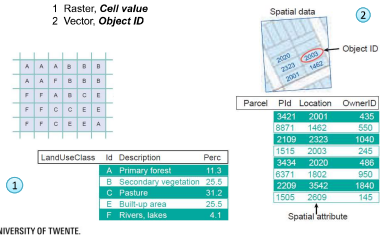


UNIVERSITY OF TWENTE.

39

## GIS AND DATABASES
LINKING SPATIAL AND NON-SPATIAL DATA

1 Raster, ***Cell value***
2 Vector, ***Object ID***




| LandUseClass | Id | Description | Perc |
|---|---|---|---|
| | A | Primary forest | 11.3 |
| | B | Secondary vegetation | 26.5 |
| | C | Pasture | 31.2 |
| | E | Built-up area | 25.5 |
| | F | Rivers, lakes | 4.1 |

| Parcel | Pld | Location | OwnerID |
|---|---|---|---|
| | 3421 | 2001 | 435 |
| | 8671 | 1462 | 550 |
| | 2109 | 2323 | 1040 |
| | 1515 | 2002 | 245 |
| | 3434 | 2020 | 486 |
| | 6371 | 1802 | 950 |
| | 2209 | 3542 | 1840 |
| | 1505 | 2609 | 145 |

UNIVERSITY OF TWENTE.

40

## SPATIAL DATABASES

Databases have evolved over the last 20 years towards supporting more complex data, such as **spatial data**

A **spatial database** allows users to store, query and manipulate collections of **spatial data by means of a new** data type: ***geometry***

| Parcel | Pid | Geometry | OwnerID |
|---|---|---|---|
| 3421 | "MULTIPOLYGON(((257462.704079333 464780.750851061,257463.89798 ))))" | 435 |
| 8871 | "MULTIPOLYGON(((257409.813950544 464789.915850449,257407.898903 ))))" | 550 |
| 2109 | "MULTIPOLYGON(((257765.714911912 464796.839972167,257782.59784 ))))" | 1040 |
| 1515 | "MULTIPOLYGON(((257790.672100448 464807.137925585,257788.609078 ))))" | 245 |
| 3434 | "MULTIPOLYGON(((257435.527950478 464803.92857633,257439.254882 ))))" | 498 |
| 6371 | "MULTIPOLYGON(((257432.476077854 464813.848852072,257433.147910...))))" | 950 |
| 2209 | "MULTIPOLYGON(((257444.888027332 464826.555046319,257446.43201 ))))" | 1840 |
| 1505 | "MULTIPOLYGON(((256293.760107491 464935.203846095,256292. 00881 ))))" | 145 |

UNIVERSITY OF TWENTE.

41

## SPATIAL DATABASE MANAGEMENT SYSTEM (SDBMS)

A S(patial)DBMS is a software package that:

- offers **all benefits of a DBMS** for data storage/management
- supports the **storage of spatial data** models – e.g. point, line, or polygon,
- can **manage coordinate systems** and transformations
- **extends querying** and manipulating capabilities of traditional SQL by adding special commands **aimed at spatial data** – e.g. `SELECT p.* FROM parcels AS p WHERE Area(p.geom)>1000`
- provide storage of the relationships between features, including the **creation and storage of topological relationships.**

UNIVERSITY OF TWENTE.

42

## GIS & SDBMS
A GOOD TEAM

**GIS:**

- Built-in 'understanding' of geographic space,
- Functions for spatial analysis of (almost) any kind, and
- Equipment for efficient map production
- **BUT** lack of fully developed query language to operate on tabular data.

**DBMS**

- Specifically designed for handling attribute data (i.e. administrative, non-spatial, tabular, thematic)
- Long tradition in multi-user concurrent management of large amounts of data
- **BUT** not support provided for spatial data storage and representation

UNIVERSITY OF TWENTE.

43

## GIS & SDBMS
A GOOD TEAM

**SDBMS:**

- Focuses on storage, querying and sharing large **spatial data sets**,
- Understand geometry and topology
- Provide fully-fledged querying language (i.e. SQL)
- Spatial querying functions
- Basic visualization

UNIVERSITY OF TWENTE.

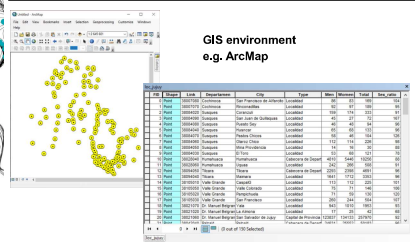
44





24/09/2023

## GIS & SDBMS
SAME DATA DIFFERENT VIEWS

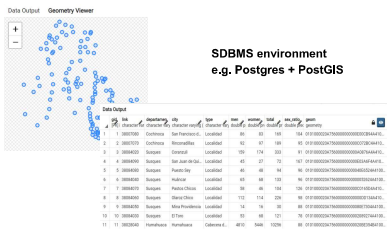


**GIS environment**
**e.g. ArcMap**

UNIVERSITY OF TWENTE.

45

## GIS & SDBMS
SAME DATA DIFFERENT VIEWS



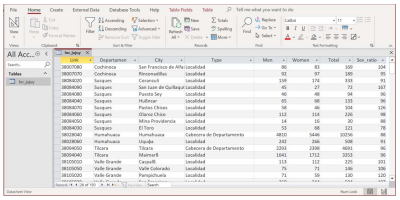
**SDBMS environment**
**e.g. Postgres + PostGIS**

UNIVERSITY OF TWENTE.

47

## GIS & SDBMS
SAME DATA DIFFERENT VIEWS

**DBMS environment - e.g. MS Access**



UNIVERSITY OF TWENTE.

46

## TERMINOLOGY
UNIVERSE OF DISCOURSE

We always aim at representing **only a part of the real world.**

**Universe of discourse**

- A **part** of the real world that is **of interest.**

We use data models for representing the universe of discourse and storing that representation in a database.

A.K.A. = the **database miniworld**

*It must be well understood by the designers of the DB in order to be able to properly represent it!*

UNIVERSITY OF TWENTE.

48

## TERMINOLOGY
UNIVERSE OF DISCOURSE – *EXAMPLE*

*Example:*

- *We are interested in understanding if and how **crop production** relates to **population size and structure** in different **countries** of the world.*

Part of the real world that constitutes the *universe of discourse*:

- *Crop types*
- *Crop production*
- *Population (size and structure)*
- *Country*

UNIVERSITY OF TWENTE.

49

## DATABASE CREATION
PROCESS OF DATABASE DESIGN

Relational **database design** involves:

- Deciding which **relations** will be present in the database, and
- Defining **their schemas**, which in turn implies:
  - Deciding which **attributes** relations will have, and
  - What are the **domains** of these attributes.

Database design also includes:

- Defining **relationships** between relations by **integrity constraints**
- Defining desirable **queries**

UNIVERSITY OF TWENTE.

50





## DATABASE CREATION
STAGES

Relational **database creation** involves 4 stages:

1. **Data investigation**
   - Type, quantity and quality of the data to be included in the DB
   - Entities and attributes are decided
2. **Data modelling**
   - Create a conceptual model of the data based on relationships between entities
3. **Database design**
   - Create a practical design for the DB
4. **Database implementation**
   - Create DB structure and populate it

UNIVERSITY OF TWENTE.

51

## DATA MODELLING
STAGES

**Data modelling or entity relationship modelling** involves 4 steps:

1. Identification of entities
2. Identification of relationships
   - One to one
   - One to many
   - Many to many
3. Identification of attributes of entities
4. Design of tables

**The process should be led by:**

- **Purpose** of the DB
- **Questions** that the DB should be able to answer

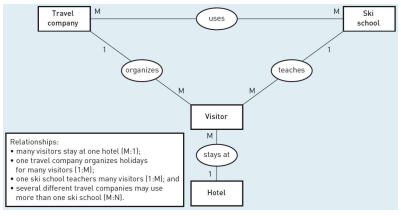
UNIVERSITY OF TWENTE.

52





## DATA MODELLING
RELATIONSHIPS




Source: Heywood et al, 2011

UNIVERSITY OF TWENTE.

53

## DATA MODELLING
TABLES

Table 1: Hotel

| Hotel ID | Name | Other attributes... |
|---|---|---|
| 001 | Mountain View | .... |
| 002 | Palace Deluxe | .... |
| 003 | Ski Lodge | .... |
| .... | .... | .... |

Table 2: Travel company

| Travel co. ID | Travel Co. Name | Other attributes... |
|---|---|---|
| T01 | Ski Tours | .... |
| T02 | Snow Breaks | .... |
| .... | .... | .... |

Table 3: Visitors

| Visitor ID | Visitor Name | Hotel ID | Travel Co. ID | Ski School ID | Other attributes... |
|---|---|---|---|---|---|
| V001 | Smith J | 002 | T01 | S02 | .... |
| V002 | Schmidt S | 001 | T02 | S02 | .... |
| .... | .... | .... | .... | .... | .... |

Table 4: Ski schools

| Ski School ID | Ski School Name | Other attributes... |
|---|---|---|
| S01 | Snow Fun | .... |
| S02 | Bert's Ski School | .... |
| S03 | Ski School Professional | .... |
| .... | .... | .... |

Table 5: Link table

| Travel Co. ID | Ski School ID |
|---|---|
| T01 | S01 |
| T01 | S02 |
| T02 | S02 |
| .... | .... |

Source: Heywood et al, 2011

UNIVERSITY OF TWENTE.

54





## Thanks for joining!

**Coming up**

**Practical: Today 11:00 – 15:30**
**Wrap up session: Tomorrow 16:00 – 17:30**

**Questions?**

*Discussions > Lesson 9: Data Management*

55