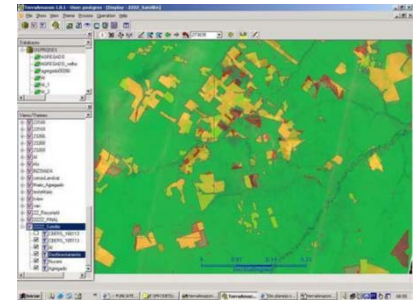
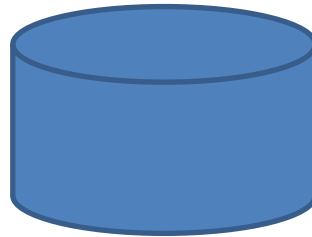
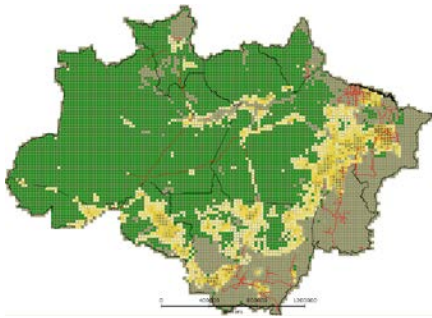


Spatial Databases: Lecture 2

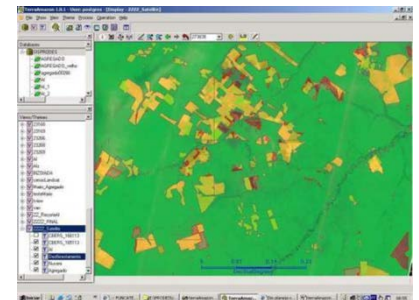
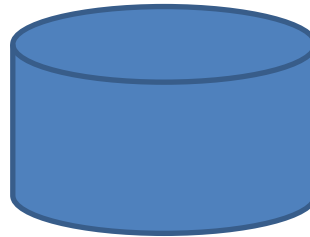
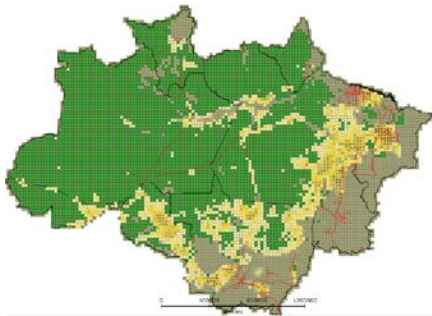
Institute for Geoinformatics
Winter Semester 2014



Malumbo Chipofya: room 109

Recap

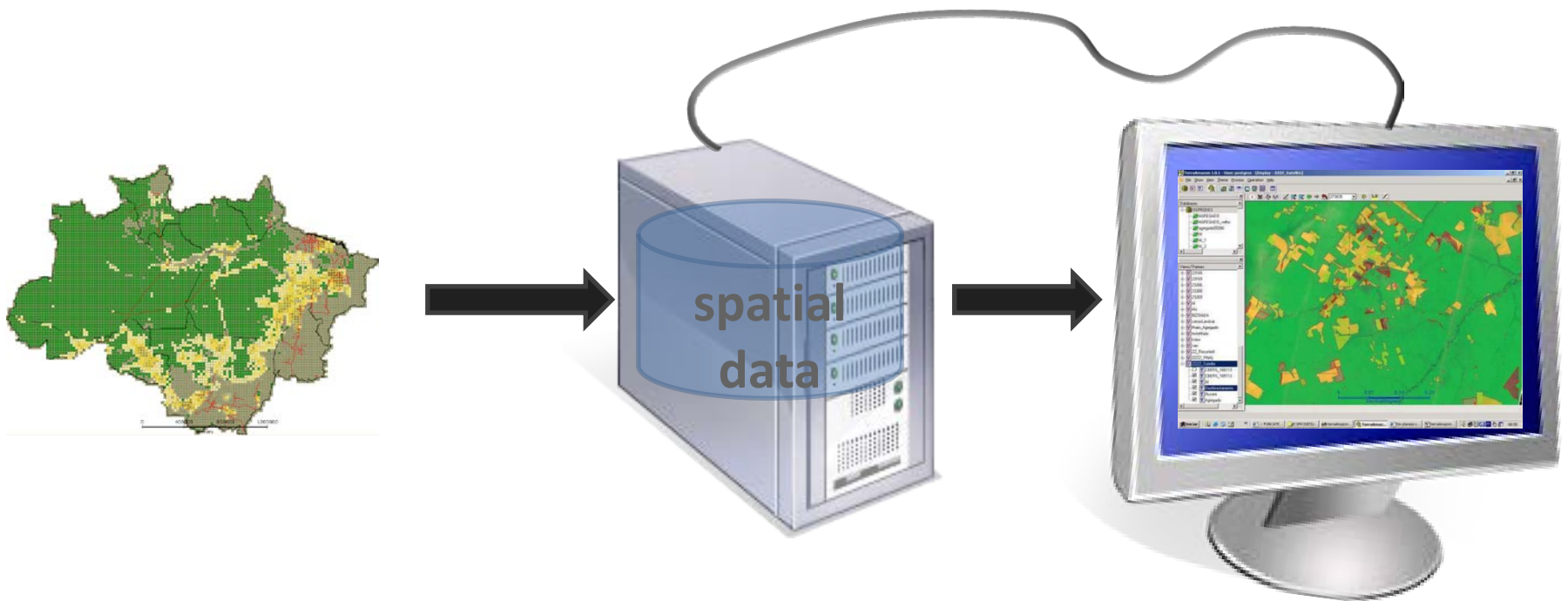
- What is a Spatial Database?



Recap

- What is a Spatial Database?

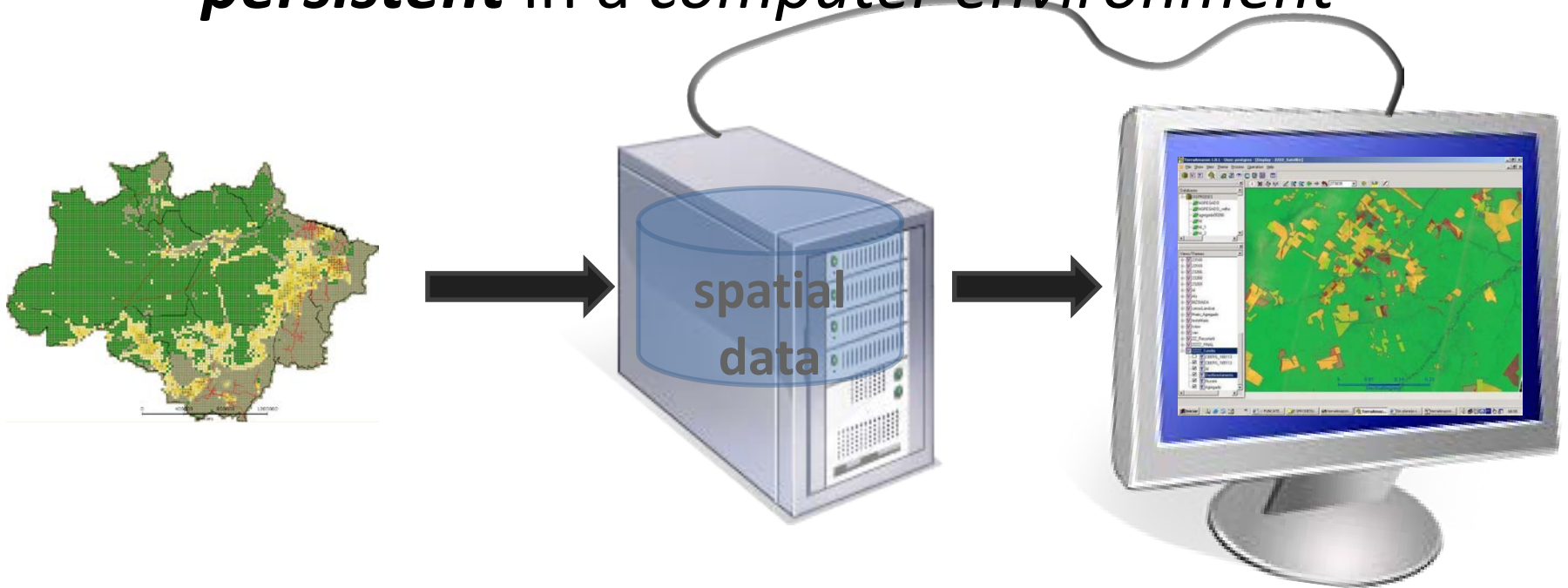
In a nutshell



Recap

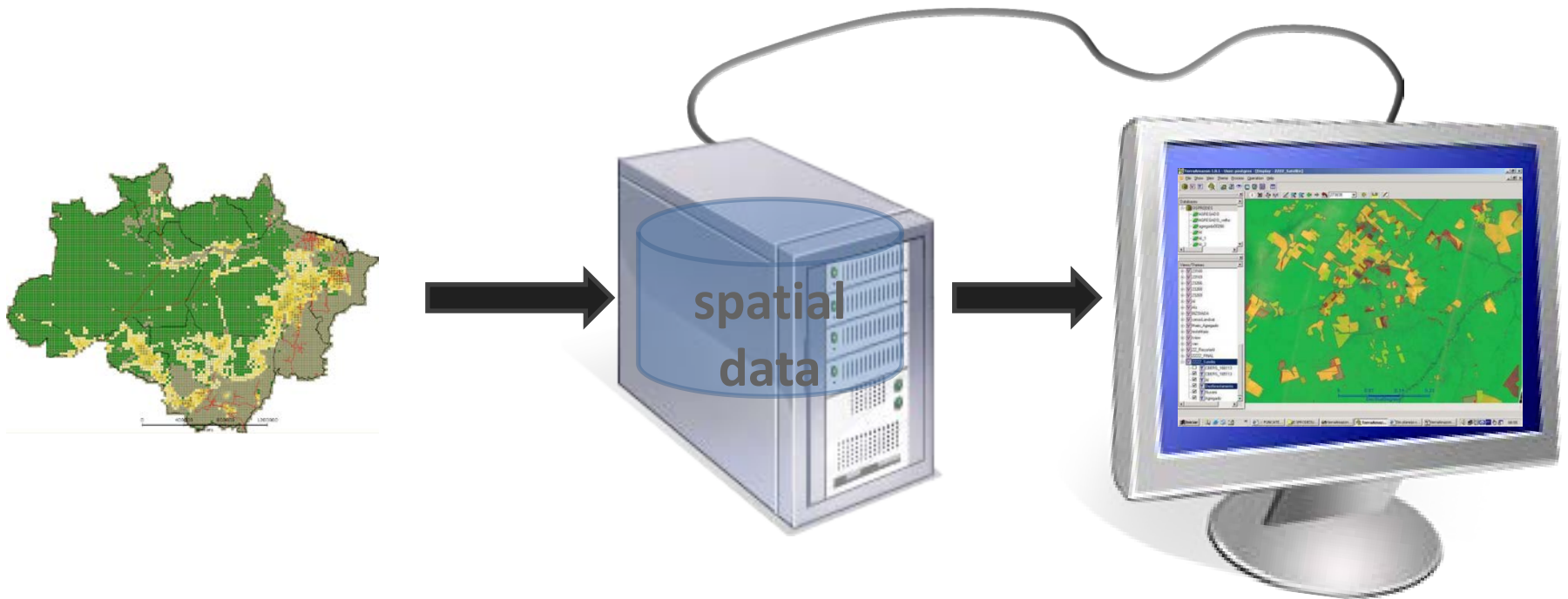
- What is a Spatial Database?

In a nutshell: a set a spatial data values
persistent in a *computer environment*



Recap

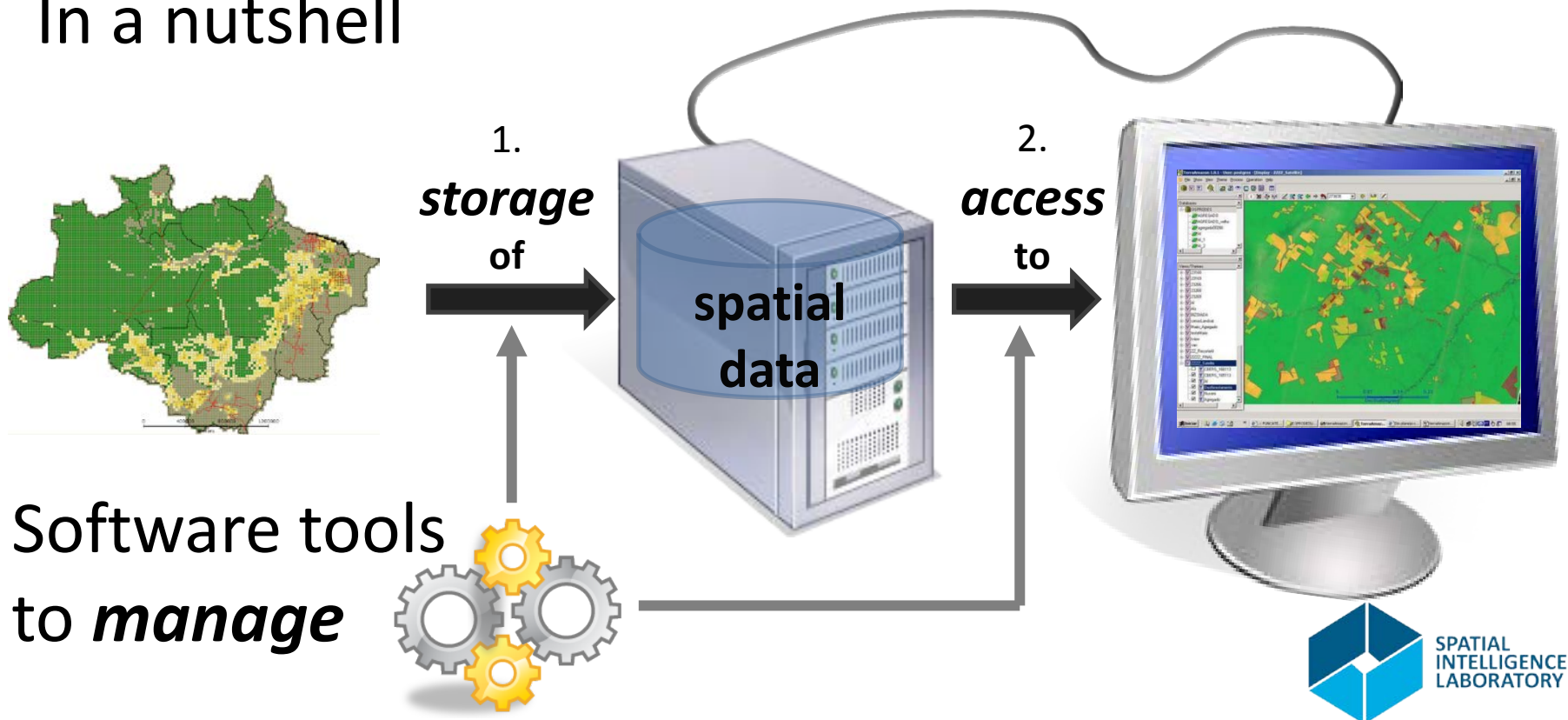
- What about a SDBMS - Spatial Database Management System?



Recap

- What about a SDBMS - Spatial Database Management System?

In a nutshell



Topic Overview

1. Prelude: Data and problem solving in science and applications
- 2. The Relational Database model**
3. Interacting with relational databases
4. Spatial Relational Database Management Systems
5. Applications: Terraview and Terralib: Prof. Dr. Gilberto Camara
6. A sample of Nosql Databases: brief introductions + example applications
 - a. Array databases: SciDB
 - b. Document databases: MongoDB
 - c. Graph databases: Neo4J
7. Summary of all lectures given.

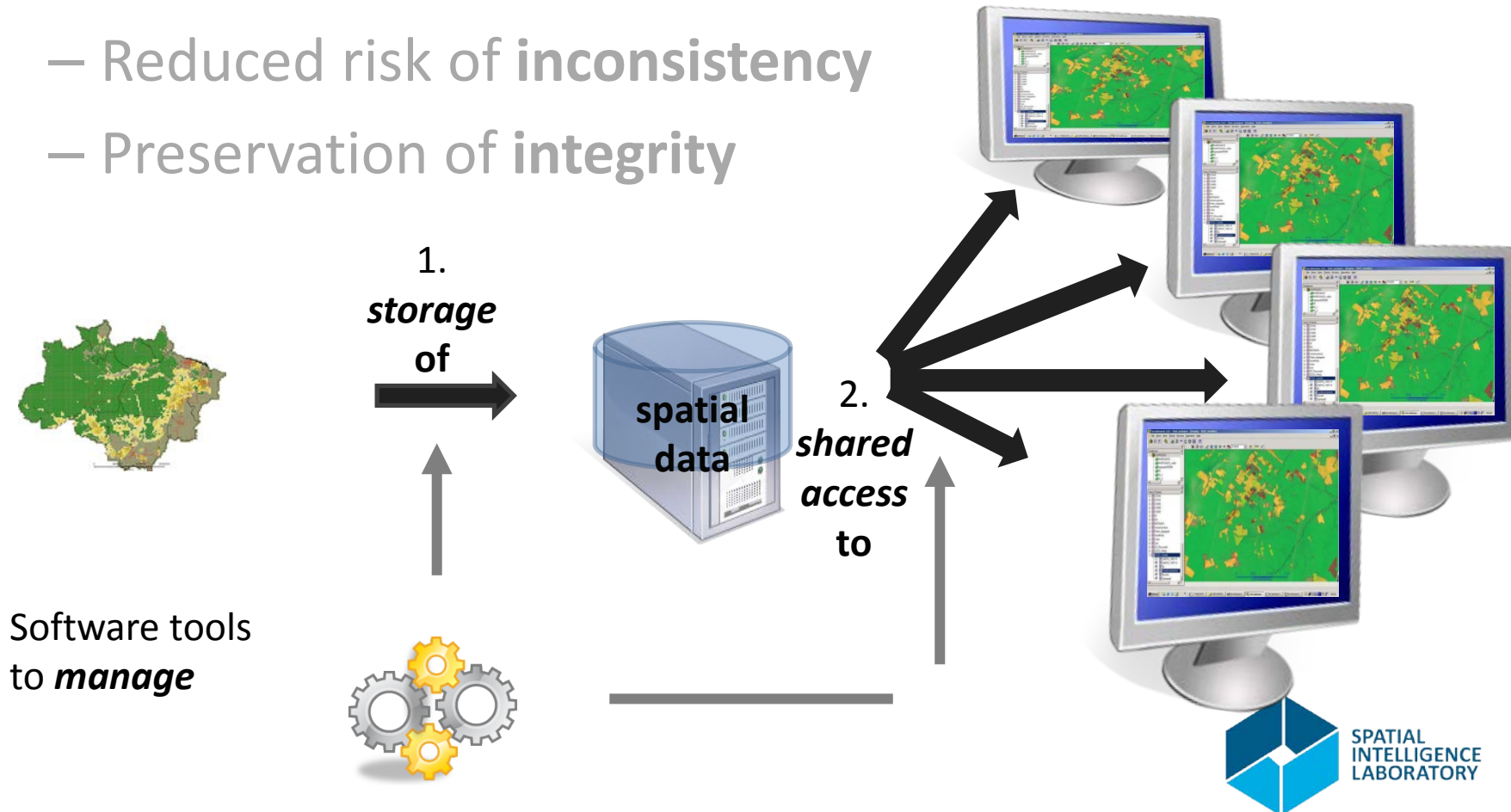
Why do we need DBMSs?

Why do we need DBMSs?

- **Shared** data access
- Reduce data **redundancy**
- Reduced risk of **inconsistency**
- Preservation of **integrity**
- **Transaction support**
- **Security**
- **Standardization**

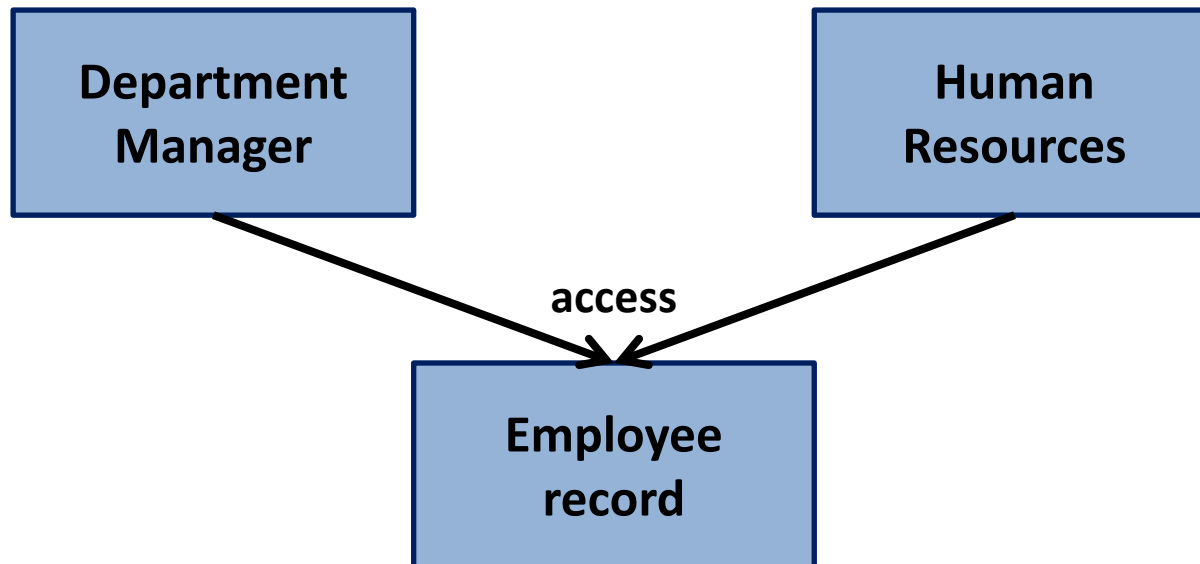
Why do we need DBMSs?

- **Shared** data access
- Reduce data **redundancy**
- Reduced risk of **inconsistency**
- Preservation of **integrity**



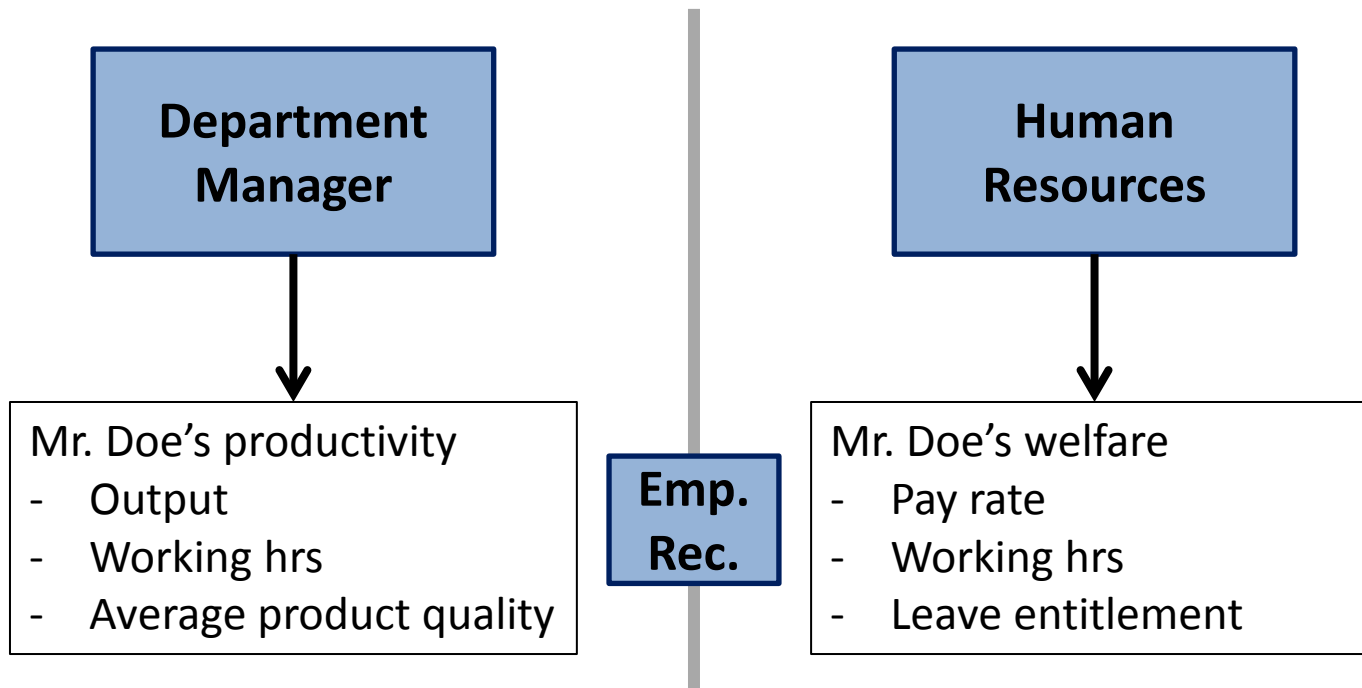
Why do we need DBMSs?

- **Shared** data access
- Reduce data **redundancy**
- Reduced risk of **inconsistency**
- Preservation of **integrity**



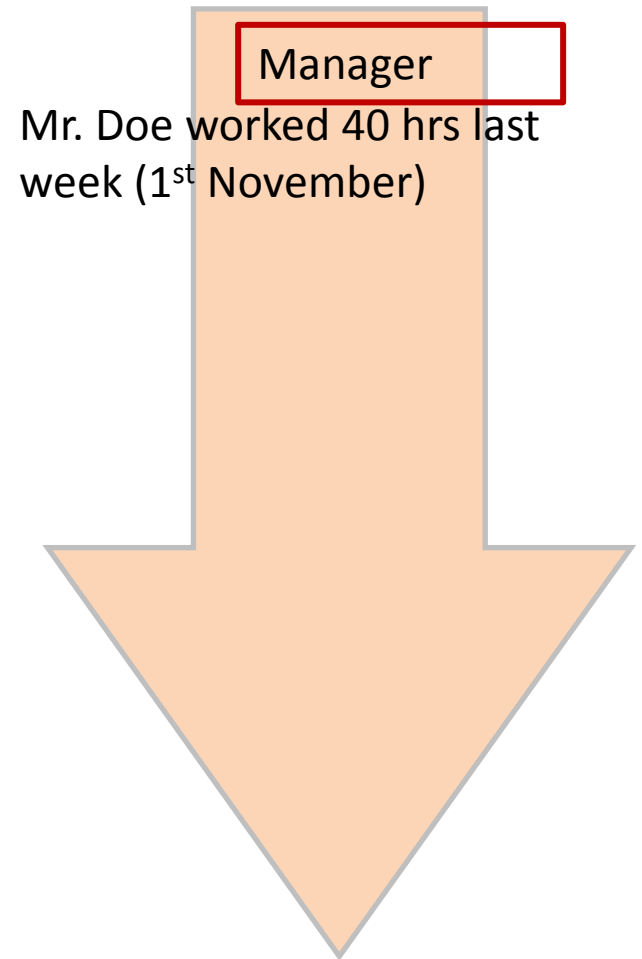
Why do we need DBMSs?

- Shared data access
- Reduce data **redundancy**
- Reduced risk of **inconsistency**
- Preservation of **integrity**



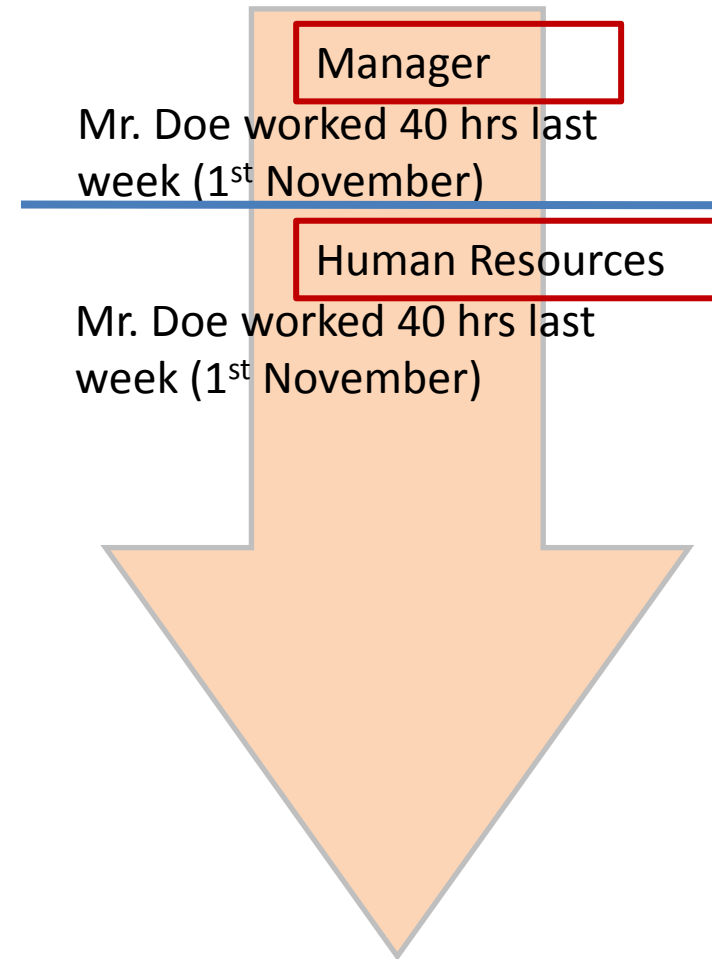
Why do we need DBMSs?

- Shared data access
- Reduce data **redundancy**
- Reduced risk of **inconsistency**
- Preservation of **integrity**



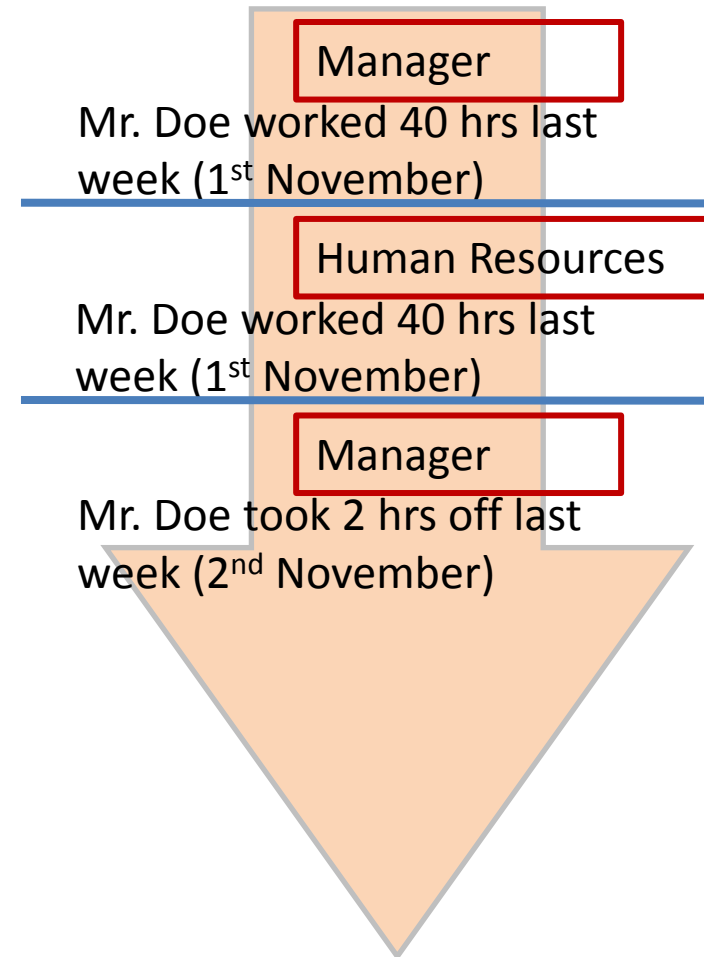
Why do we need DBMSs?

- Shared data access
- Reduce data **redundancy**
- Reduced risk of **inconsistency**
- Preservation of **integrity**



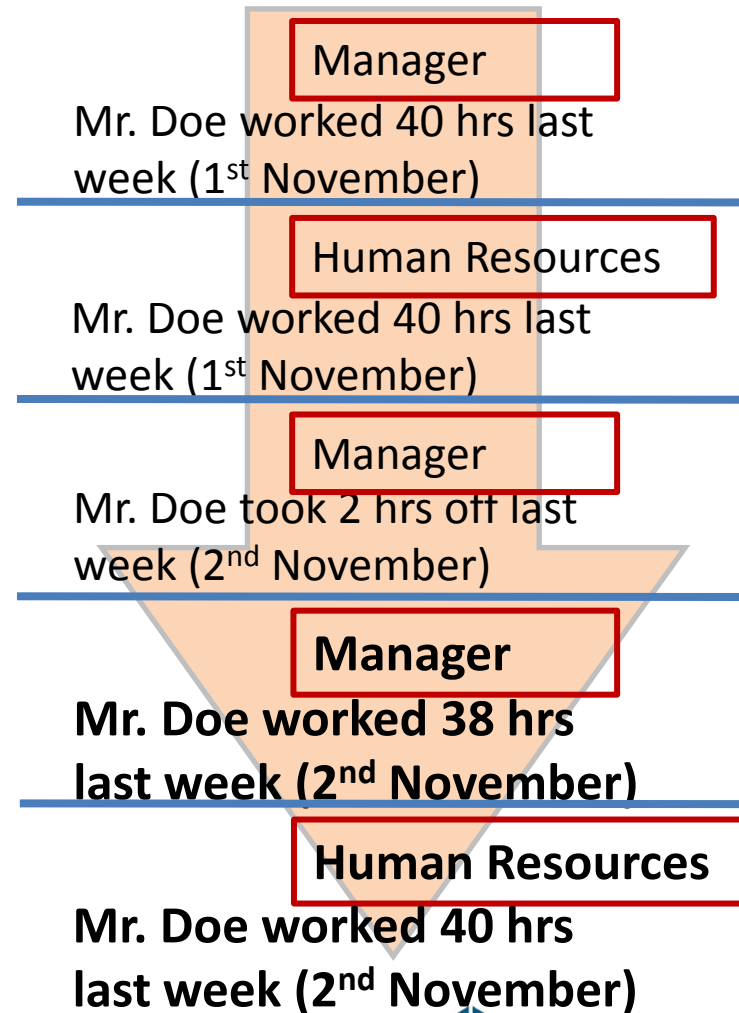
Why do we need DBMSs?

- Shared data access
- Reduce data **redundancy**
- Reduced risk of **inconsistency**
- Preservation of **integrity**



Why do we need DBMSs?

- Shared data access
- Reduce data **redundancy**
- Reduced risk of **inconsistency**
- Preservation of **integrity**



Why do we need DBMSs?

- **Shared** data access
- Reduce data **redundancy**
- Reduced risk of **inconsistency**
- Preservation of **integrity**

Manager

Mr. Doe worked **400 hrs**
last **week**

Why do we need DBMSs?

- **Shared** data access
- Reduce data **redundancy**
- Reduced risk of **inconsistency**
- Preservation of **integrity**
- **Transaction support**
- **Security**
- **Standardization**

Database Systems Architecture

- There are typically three levels (K.a. ANSI/SPARC architecture)

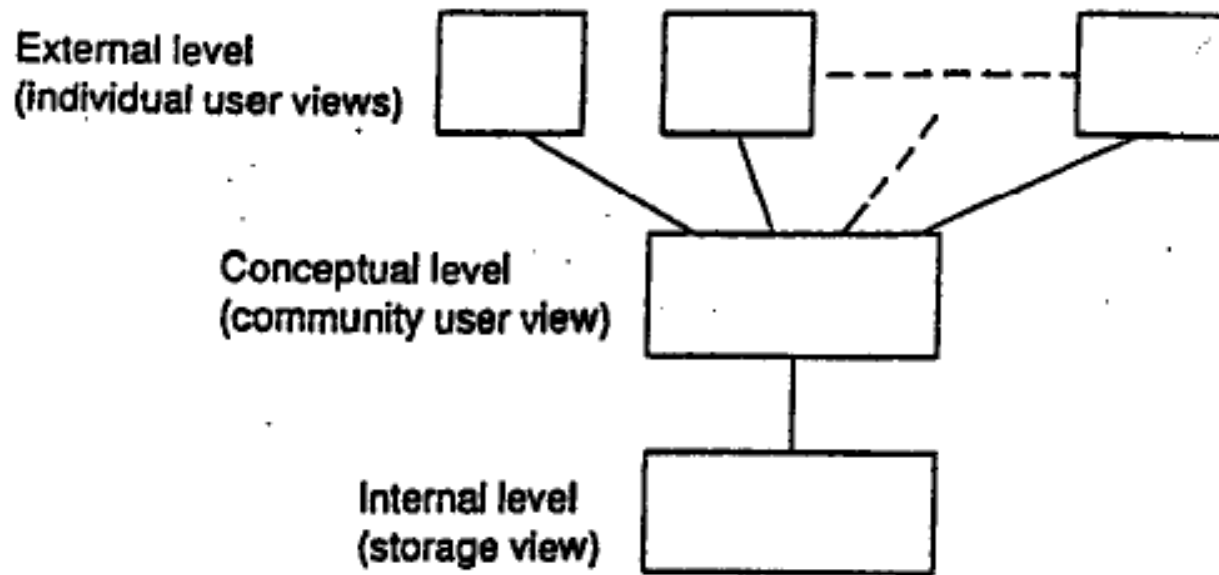


Fig. 2.1 The three levels of the architecture

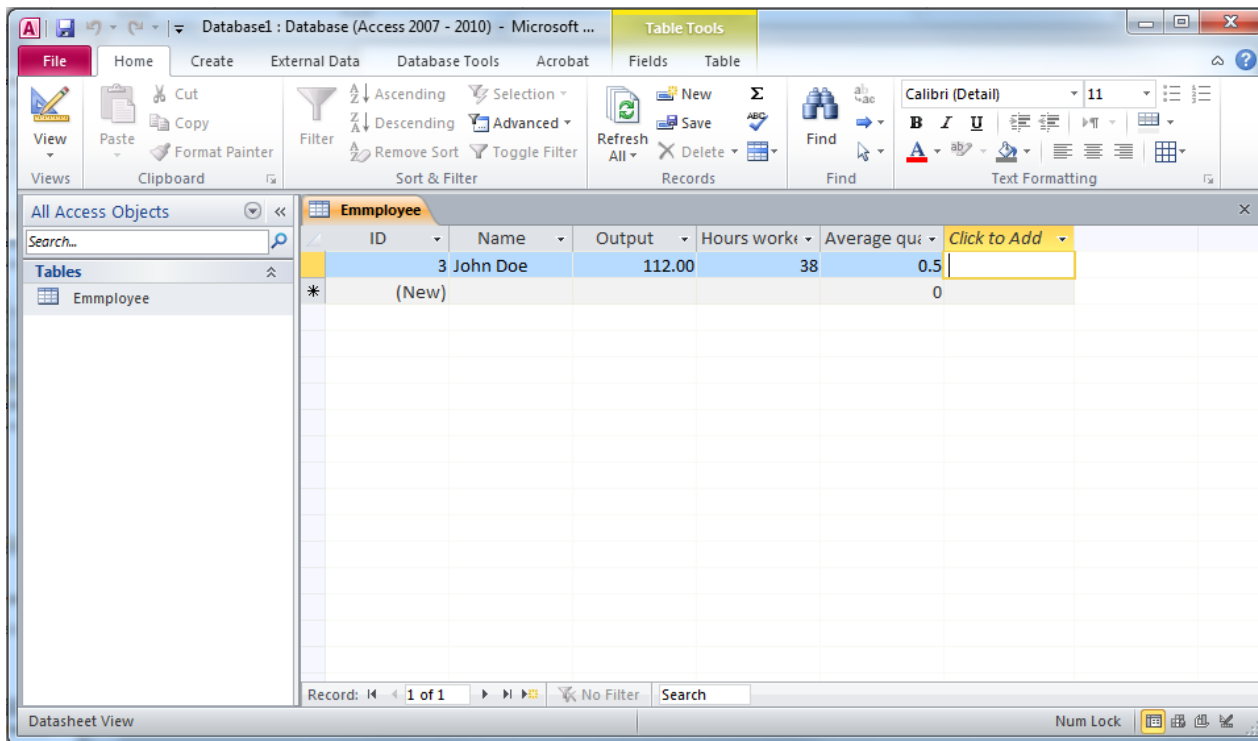
From: C.J. Date, *An Introduction to Database Systems, 8th Edition*. Pearson Education Inc., 2004, page 34.

Database Systems Architecture

- External level: Individual user view
 - Programmers
 - End users
- Allows users to
 - Implement their database designs in a DBMS
 - Add data to the database
 - Process the data in the database
 - Query the database for data

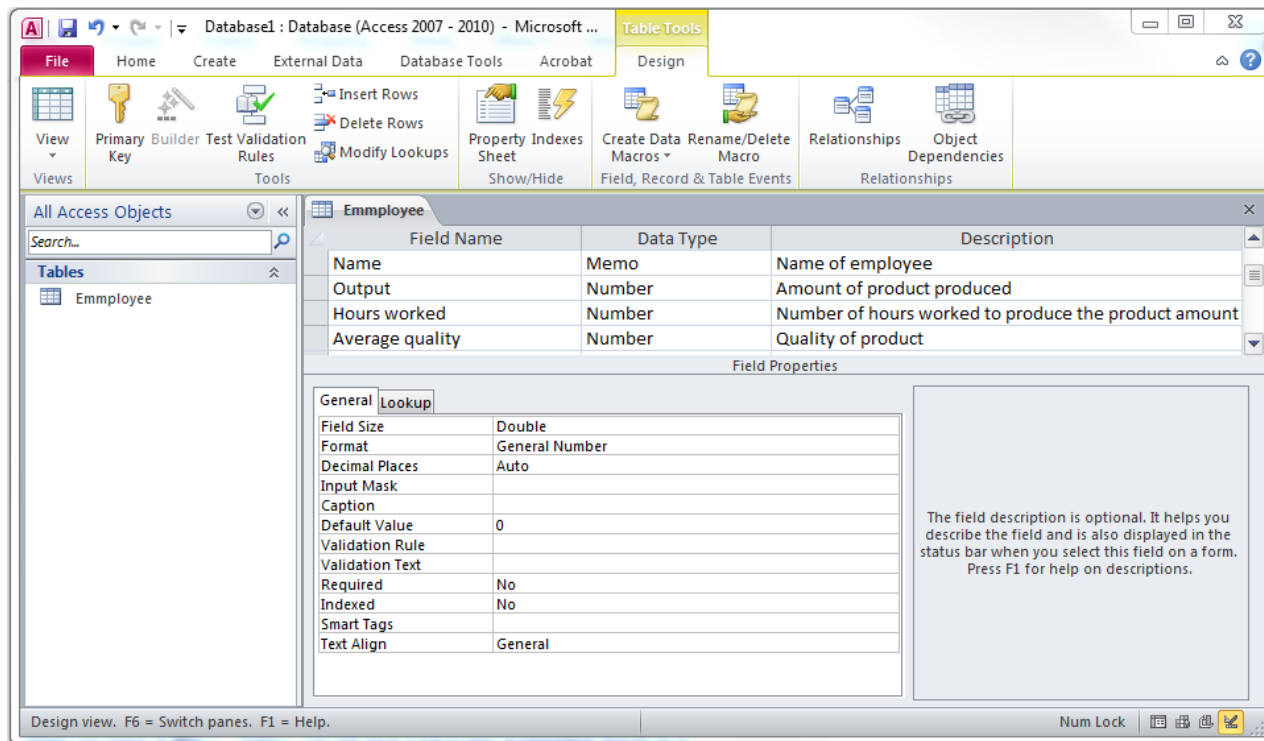
Database Systems Architecture

- External level: Individual user view
 - Programmers
 - End users (E.g. MS Access, GIS systems)



Database Systems Architecture

- External level: Individual user view
 - Programmers
 - End users (E.g. MS Access, GIS systems)



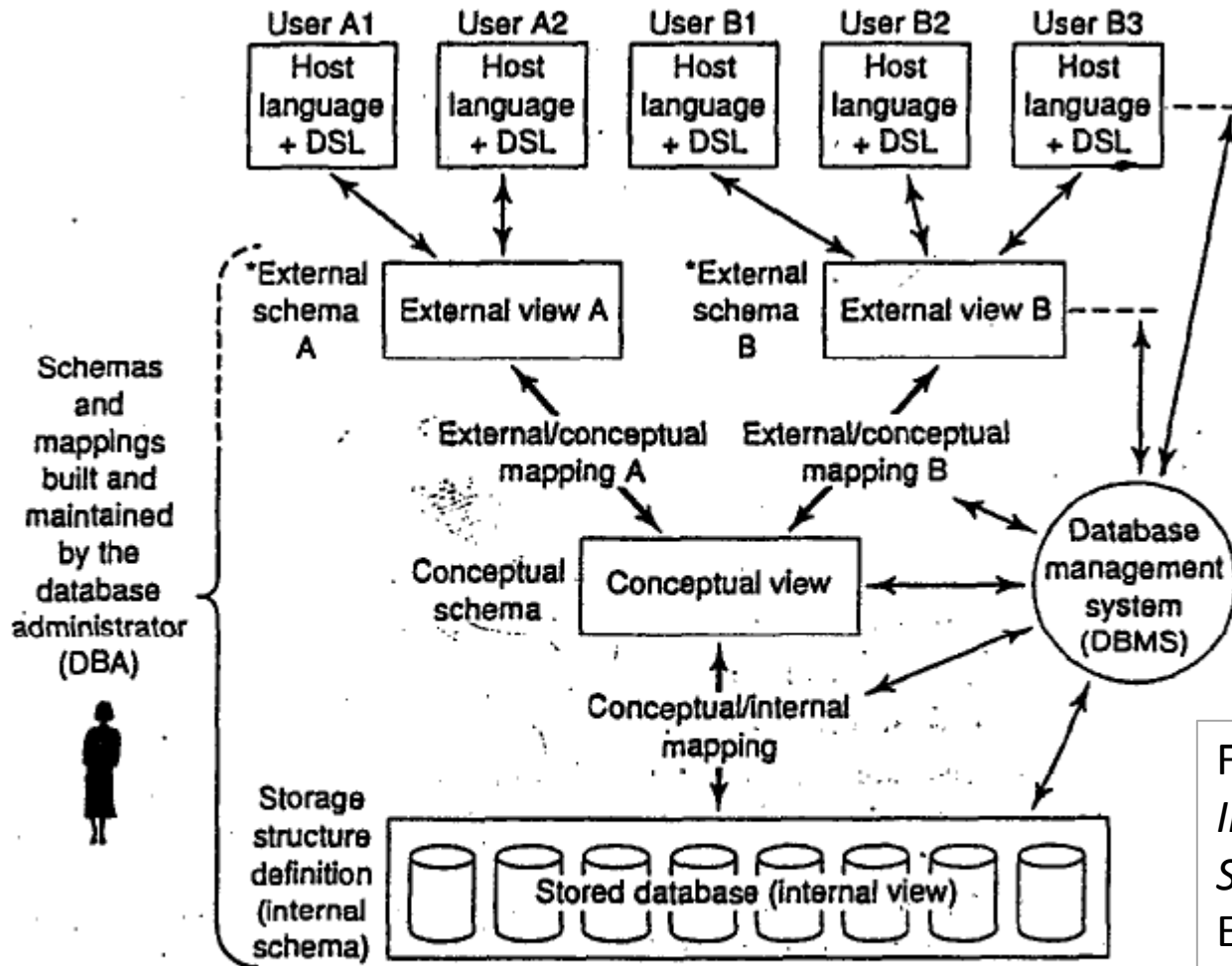
Database Systems Architecture

- Conceptual level: conceptual view
 - A representation of the entire information content of the database
 - A view of the data **as it really is**
 - The domain of database design
(what we are interested in)
- Internal level: physical storage
 - A view of the data as it is stored on physical storage

Database Systems Architecture

- Data languages
 - Data definition language
 - Data manipulation language
- Mappings from External to Conceptual level
 - Translate between individual external views (schemas) and the conceptual view (schema)
- Mappings from Conceptual to Internal level
 - Translate between conceptual views and the internal view

Database Systems Architecture



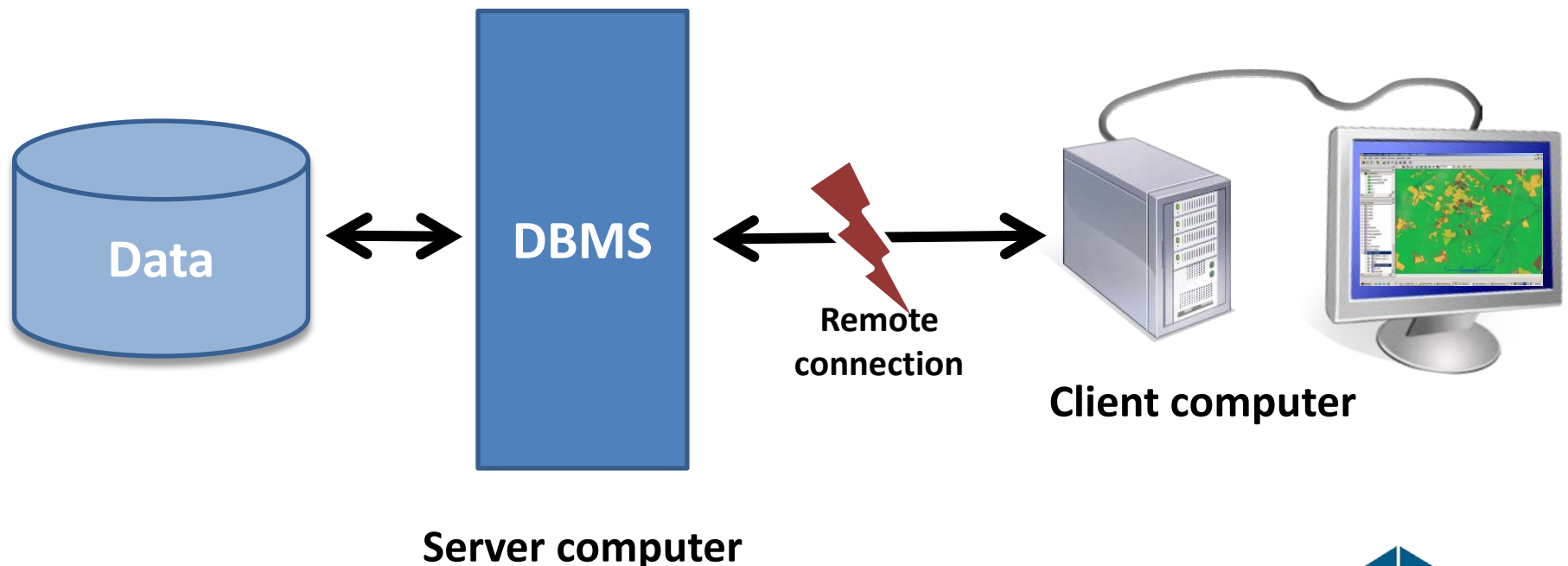
From: C.J. Date, *An Introduction to Database Systems, 8th Edition*. Pearson Education Inc., 2004, page 67.

*User interface

Fig. 2.3 Detailed system architecture

Database Systems Architecture

- Most common architecture of DBMS/user application setup is the Client-server model



~~Spatial~~ Database Systems

Relational Databases

What is a Relational Database

- A relational database system consists of
 - A set of tables
 - Operators that derive new tables from others

Department Manager's record

Name	Output (Kg)	Hours worked	Average quality
John Doe	112	38	0.5
Mary Jane	117	40	0.8
Chikondi Uyo	116	40	0.9

What is a Relational Database

- A relational database system consists of
 - A set of tables
 - Operators that derive new tables from others

Operator: **project**

Name	Output (Kg)	Hours worked	Average quality
John Doe	112	38	0.5
Mary Jane	117	40	0.8
Chikondi Uyo	116	40	0.9

**Manager inspects
qualitative of output**

What is a Relational Database

- A relational database system consists of
 - A set of tables
 - Operators that derive new tables from others

Operator: **restrict**

Name	Output (Kg)	Hours worked	Average quality
John Doe	112	38	0.5
Mary Jane	117	40	0.8
Chikondi Uyo	116	40	0.9

**Manager inspects
John Doe's record**

What is a Relational Database

- Tables are formally known as “Relations”
- A row in table is called a “Tuple”
- A column in table is called an “Attribute”
 - Attributes have associated “Types”
 - E.g. **Name** can be the type of the column with heading “Name”
 - Qn.: What is the actual type of the columns
 - **Name**
 - **Output (Kg)**
 - **Hours worked**
 - **Average quality**

What is a Relational Database

- Tables are formally known as “Relations”
- A row in table is called a “Tuple”
- A column in table is called an “Attribute”
 - Attributes have associated “Types”
 - E.g. **Name** can be the type of the column with heading “Name”
 - Qn.: What is the actual type of the columns
 - **Name** - **Text/String**
 - **Output (Kg)**
 - **Hours worked**
 - **Average quality**

What is a Relational Database

- Tables are formally known as “Relations”
- A row in table is called a “Tuple”
- A column in table is called an “Attribute”
 - Attributes have associated “Types”
 - E.g. **Name** can be the type of the column with heading “Name”
 - Qn.: What is the actual type of the columns
 - **Name** - **Text/String**
 - **Output (Kg)** - **Integer**
 - **Hours worked**
 - **Average quality**

What is a Relational Database

- Tables are formally known as “Relations”
- A row in table is called a “Tuple”
- A column in table is called an “Attribute”
 - Attributes have associated “Types”
 - E.g. **Name** can be the type of the column with heading “Name”
 - Qn.: What is the actual type of the columns
 - **Name** - **Text/String**
 - **Output (Kg)** - **Integer**
 - **Hours worked** - **Integer**
 - **Average quality**

What is a Relational Database

- Tables are formally known as “Relations”
- A row in table is called a “Tuple”
- A column in table is called an “Attribute”
 - Attributes have associated “Types”
 - E.g. **Name** can be the type of the column with heading “Name”
 - Qn.: What is the actual type of the columns
 - **Name** - **Text/String**
 - **Output (Kg)** - **Integer**
 - **Hours worked** - **Integer**
 - **Average quality** - **Float/Decimal**

What is a Relational Database

- Tables are formally known as “Relations”
- A row in table is called a “Tuple”
- A column in table is called an “Attribute”
 - Attributes have associated “Types”

- According to C.J. Date
 - Type are sets of things we can talk about
 - Relations are sets of things we can say about types

Tuples

- A tuple represents a statement of the form

$(X_1, type_1)$ and $(X_2, type_2) \dots$ and $(X_k, type_k)$

Where the relation has k attributes and X_i is value of type $type_i$

Exercise 1

A manager in a tobacco manufacturing company has the following information for the Tobacco Rollers Department:

1. Employee ID of each Roller
2. Names of Each Roller
3. Tobacco sticks rolled for each day
4. Total weight of tobacco rolled per day
5. Number of hours worked each day



Exercise 1

1. Employee ID of each Roller
2. Names of Each Roller
3. Tobacco sticks rolled for each day
4. Total weight of tobacco rolled per day
5. Number of hours worked each day



- Qn. 1: What are the possible attributes and their types?
- Qn. 2: What relations can be formed?

Exercise 1

1. Employee ID of each Roller
2. Names of Each Roller
3. Tobacco sticks rolled for each day
4. Total weight of tobacco rolled per day
5. Number of hours worked each day



Qn. 1: What are the possible attributes and their types?

Qn. 2: What relations can be formed?

1) a. Employee b. Productivity, 2) EmployeeProductivity

Relations, Attributes, and Tuples

- The set of all attributes of a relation must satisfy the **Uniqueness** property

At any given time no two tuples are equivalent

- This leads to the definition of keys

Examples

- Which of these is a valid relation

Passport #	Name	Country
MW-4321	Malumbo	Malawi
DE-678910	Jan	Germany
IN-765432	Pradeep	India

H	L	W	M	O	D
3	13	7	0.2	6	1.9

ID	Name	Pop	Buses	Comment
89	Gievenbeck	15	3	Could use more
72	Kinderhaus	23	8	Has enough
89	Gievenbeck	15	5	Could use more

Special Attributes: Keys

- A **Candidate Key** for a relation, say R , is a collection of attributes, say K satisfying (and being satisfied by, the following 2 properties:
 - **Uniqueness**: at **no** time does R have two distinct tuples whose projections to K have the same values
 - **Irreducibility**: no proper subset of K has the uniqueness property

Examples

- Which combinations of attributes are candidate keys?

Passport #	Name	Country
MW-4321	Malumbo	Malawi
DE-678910	Jan	Germany
IN-765432	Pradeep	India

H	L	W	M	O	D
3	13	7	0.2	6	1.9

ID	Name	Pop	Buses	Comment
89	Gievenbeck	15	3	Could use more
72	Kinderhaus	23	8	Has enough
89	Gievenbeck	15	5	Could use more

Special Attributes: Keys

- Primary keys
 - Consider the Employee relation: Attribute **ID** is a choice of primary key.
 - Primary keys uniquely identify each tuple in a relation – they are *candidate keys*
 - Primary keys may consist of more than one attribute

Special Attributes: Keys

- Foreign keys
 - Consider the Productivity relation: Attribute **ID** is a choice of foreign key to the Employee relation.
 - Foreign keys are attribute that must contain values which are contained in some candidate key in another relation.
 - Foreign keys are **not necessarily** unique to a tuple in a relation – they are **not *candidate keys***

Examples

- Which combinations of attributes are primary or foreign keys?

Passport #	Name	Country
MW-4321	Malumbo	Malawi
DE-678910	Jan	Germany
IN-765432	Pradeep	India

H	L	W	M	O	D
3	13	7	0.2	6	1.9

ID	Name	Wt	Ht	Passport
89	Malumbo	60	1.8	MW-4321
72	Walid	80	2	UA-678910
89	John	95	2.1	UK-765432

Exercise 2

The human resources department keeps track of the welfare of Tobacco Rollers and Tobacco Packers who belong to different departments:

1. Employee's income
2. Monthly health reports
 - a. Skin condition
 - b. Lung condition
 - c. Eye condition (left and right eye separately)
3. Marital status
4. Number of children
5. Years of service

Qn. 1: Define the relations and specify the type for each attribute

Qn. 2: For each relation above plus those from exercise 1 (update the relations if necessary) specify

- a. **some** candidate keys
- b. a primary key
- c. the foreign keys



Relational Operations

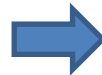
- Restrict
- Project
- Join

Passport #	Name	Country
MW-4321	Malumbo	Malawi
DE-678910	Jan	Germany
IN-765432	Pradeep	India

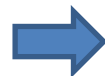
ID	Name	Wt	Ht	Passport
72	Walid	80	2	UA-678910
89	Malumbo	60	1.8	MW-4321
89	John	95	2.1	UK-765432

Relational Operations

- **Restrict**
- Project
- Join



Passport #	Name	Country
MW-4321	Malumbo	Malawi
DE-678910	Jan	Germany
IN-765432	Pradeep	India



ID	Name	Wt	Ht	Passport
72	Walid	80	2	UA-678910
89	Malumbo	60	1.8	MW-4321
89	John	95	2.1	UK-765432

Relational Operations

- Restrict
- **Project**
- Join

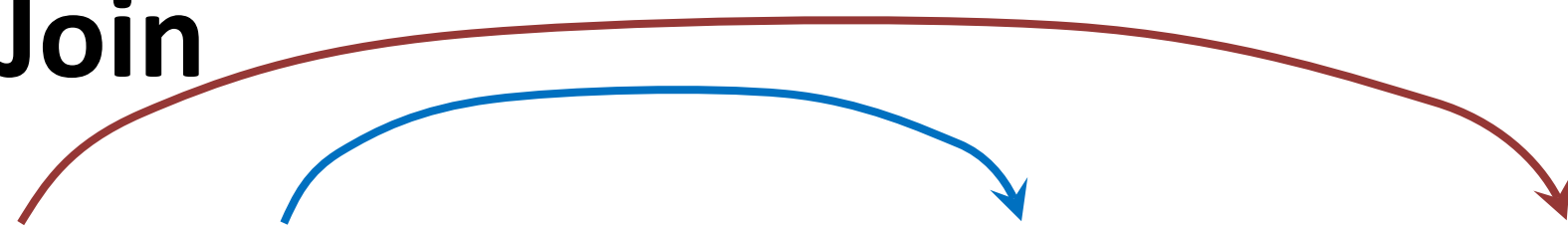
Passport #	Name	Country
MW-4321	Malumbo	Malawi
DE-678910	Jan	Germany
IN-765432	Pradeep	India



ID	Name	Wt	Ht	Passport
89	Malumbo	60	1.8	MW-4321
72	Walid	80	2	UA-678910
89	John	95	2.1	UK-765432

Relational Operations

- Restrict
- Project
- **Join**



Passport #	Name	Country
MW-4321	Malumbo	Malawi
DE-678910	Jan	Germany
IN-765432	Pradeep	India

ID	Name	Wt	Ht	Passport
72	Walid	80	2	UA-678910
89	Malumbo	60	1.8	MW-4321
89	John	95	2.1	UK-765432

Relational Operations

- Restrict
- Project
- **Join**

Passport #	Name	Country
MW-4321	Malumbo	Malawi
DE-678910	Jan	Germany
IN-765432	Pradeep	India

ID	Name	Wt	Ht	Passport
72	Walid	80	2	UA-678910
89	Malumbo	60	1.8	MW-4321
89	John	95	2.1	UK-765432

Relational Operations

- Restrict
- Project
- **Join**

Passport #	Name	Country
MW-4321	Malumbo	Malawi
DE-678910	Jan	Germany
IN-765432	Pradeep	India

ID	Name	Wt	Ht	Passport
72	Walid	80	2	UA-678910
89	Malumbo	60	1.8	MW-4321
89	John	95	2.1	UK-765432

Passport #	Name	ID	Wt	Ht	Country
MW-4321	Malumbo	89	60	1.8	Malawi

Exercise 3



For the relations you have defined in exercises 1 and 2, which (how) of the operators can be used to answer the following *queries*

Q1. What are the names of the employees with bad left eye condition

Q2. Which department has an employee earning less than 100 monies

Q3. What are names of all tobacco rollers

Schemas

- A Schema is a logical specification of the **structure** and **procedures** of a particular database
 - Structure: It defines Relations, Views and their constraints and relationships
 - Procedure: it defines procedures that operate on relations

References

- C.J. Date, *An Introduction to Database Systems, 8th Edition*. Pearson Education Inc., 2004.
- See www.geoinformatic.cc

Reading

- C.J. Date, *An Introduction to Database Systems, 8th Edition*. Pearson Education Inc., 2004.
 - Chapters 11, 12, and 14

That's all for today

Thank you!

Questions?