Databases
-
Chapter I: Introduction

Rui Oliveira © rui.oliveira@ipleiria.pt

References

Further study required

- "Fundamentals of Database Systems", Elmasri & Navathe, 6th Edition, Adison Wesley, 2011, *Chapter 1: Databases and Database Users*

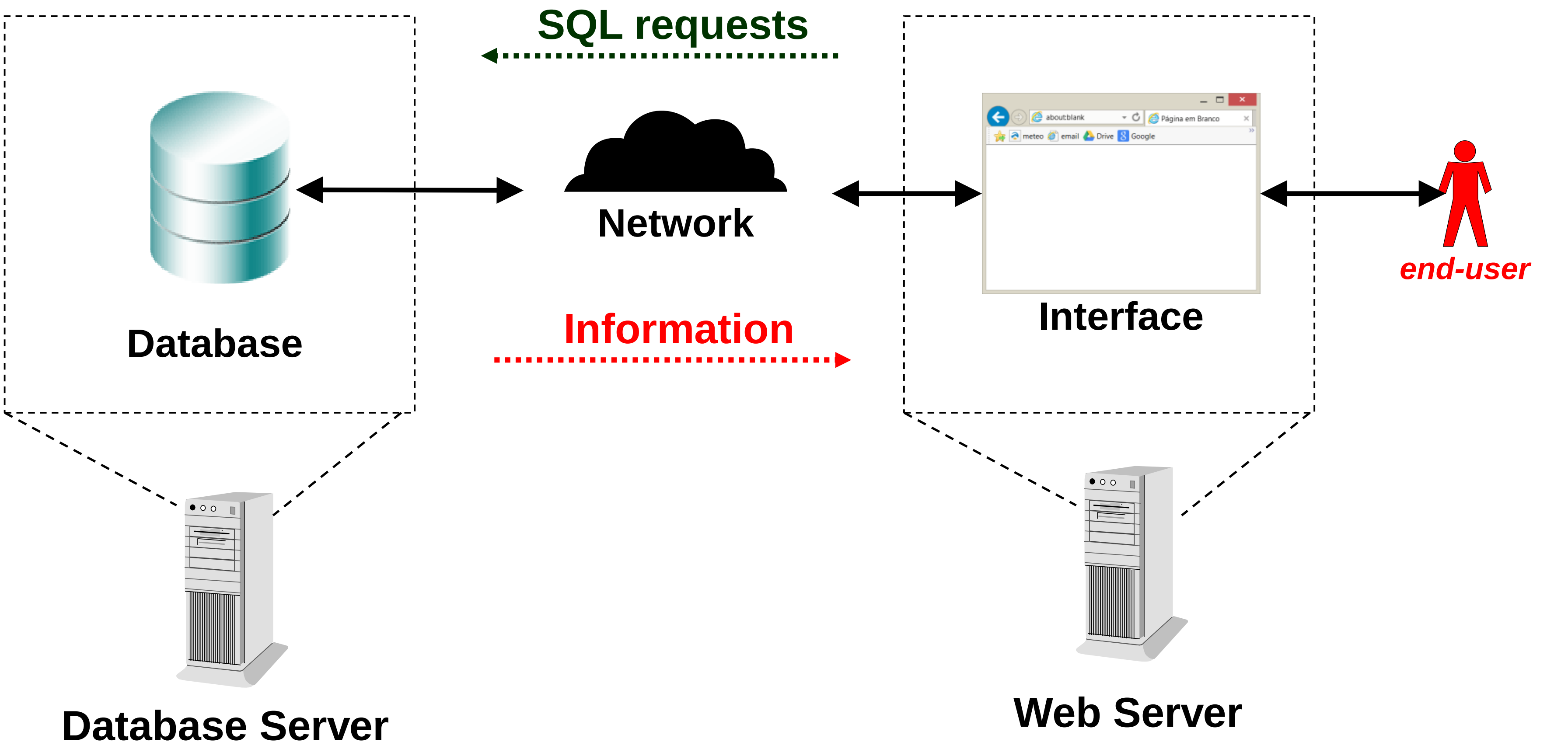
Some concepts

- Information system
- Data vs Information
- Database

> “A database is a collection of related data”
>
> (Elmasri & Navathe, 2009)

> “A shared collection of logically related data, and a description of this data, designed to meet the information needs of an organization.”
>
> (Connolly & Begg, 2005)

Information Systems: a typical configuration



SQL requests

Database

Network

Information

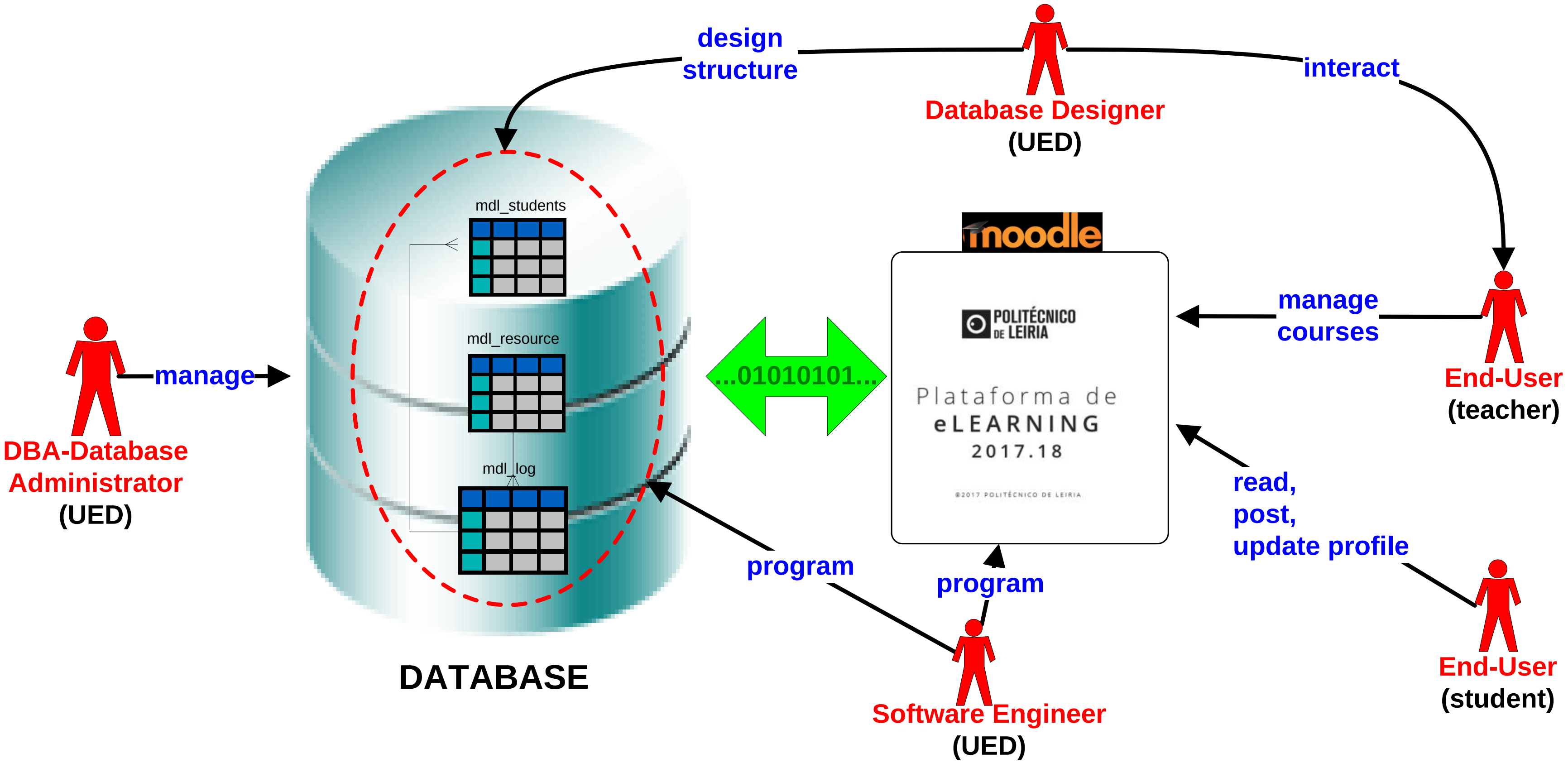
Interface

end-user

Database Server

Web Server

Information Systems: an example



design structure

Database Designer (UED)

interact

mdl_students

mdl_resource

mdl_log

DBA-Database Administrator (UED)

manage

...01010101...

moodle

POLITÉCNICO DE LEIRIA

Plataforma de eLEARNING 2017.18

©2017 POLITÉCNICO DE LEIRIA

manage courses

End-User (teacher)

read, post, update profile

End-User (student)

program

program

Software Engineer (UED)

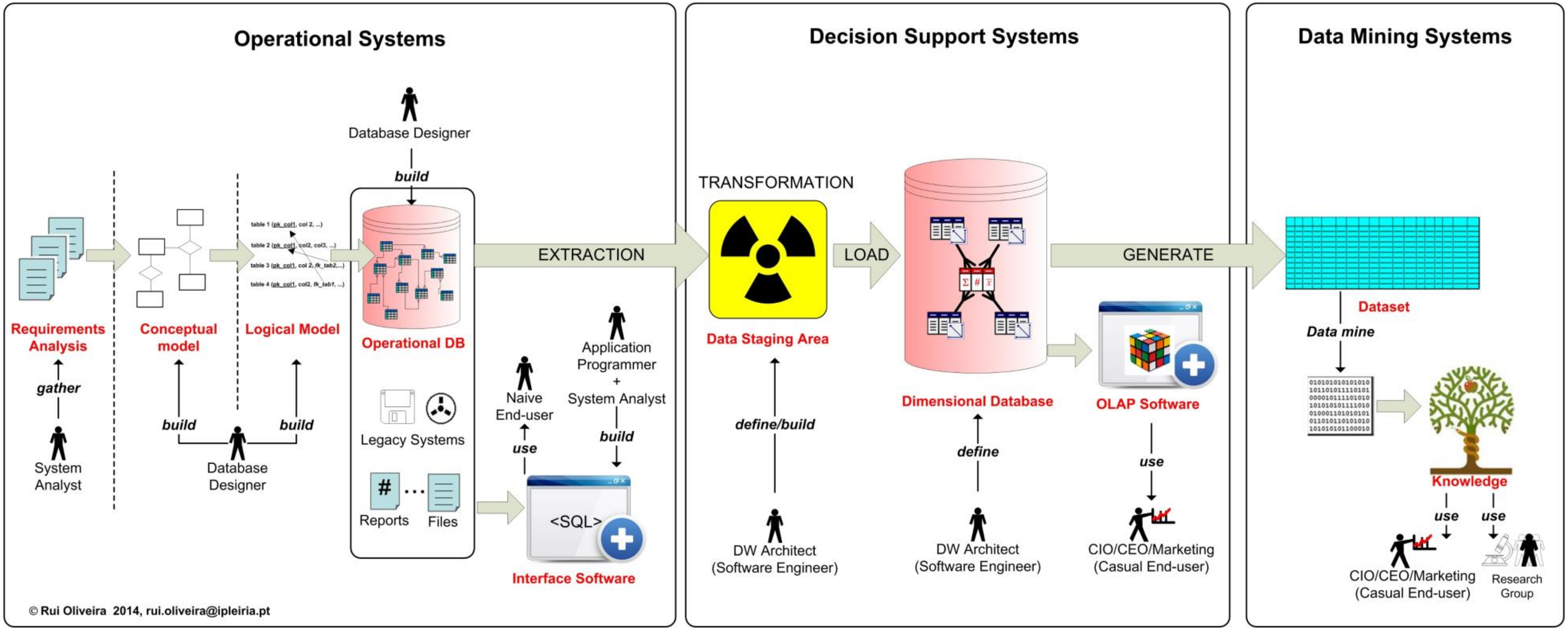
DATABASE

type of actor

action

UED=Unidade de Ensino à Distância

Information Systems: main areas



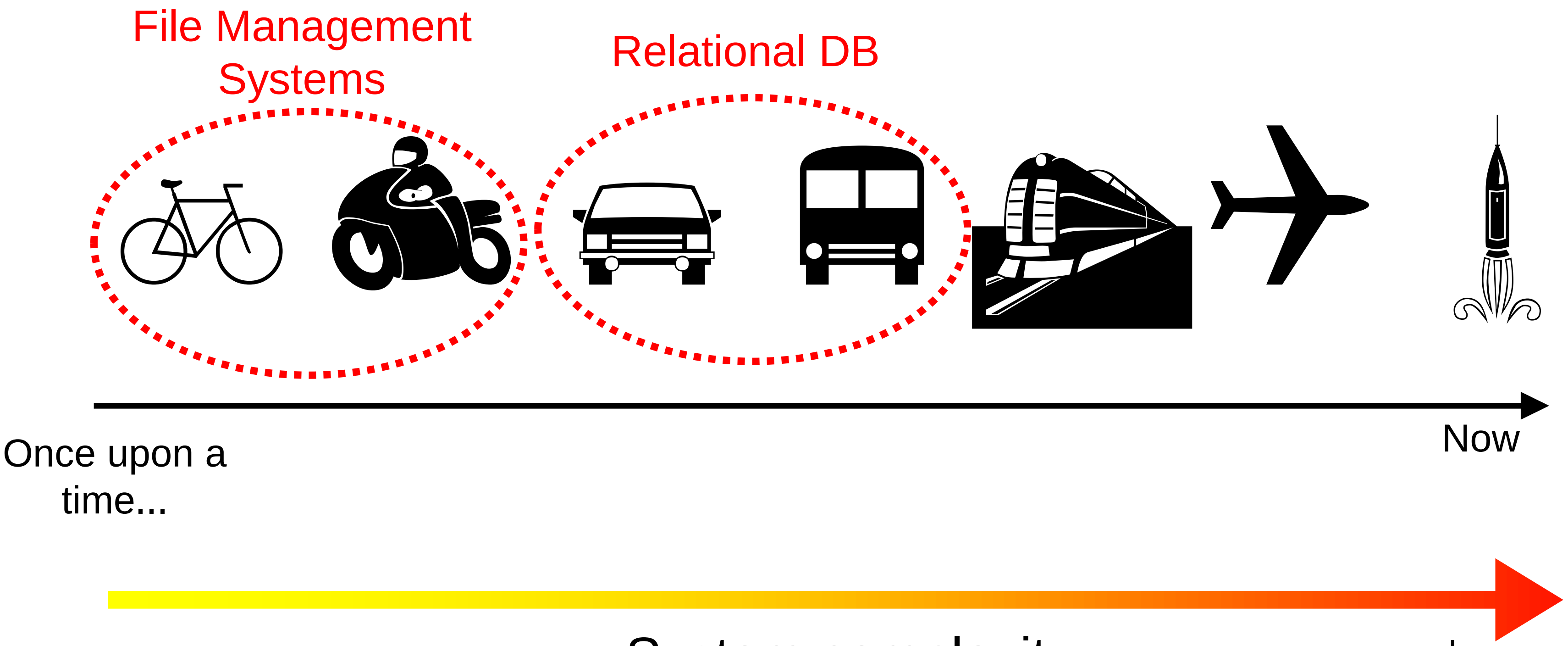
© Rui Oliveira 2014, rui.oliveira@ipleiria.pt

focus: ***data***

focus: ***information***

focus: ***knowledge***

Databases: historical evolution



System complexity
Processing capacity
Cost
Data volume and complexity
Information flexibility and complexity

Why relational Databases were born?

Relational Databases:

- Are based on the *Relational Model*
- Use bidimensional tables as data storage structures, using keys to **relate** data
- Use a simple, powerful and universally adopted language (SQL) to store data, to share and produce information
- Managed by a specific software, the Relational Database Management System (RDBMS or *Sistema de Gestão de Bases de Dados*, in PT). Examples: Oracle, Microsoft SQL Server, MySql, MariaDB, PostgreSQL, DBase, DB2,...

Common scenarios:

- Continente, Worten, Portal da Saúde, Moodle, GISEM,...

“Do I need a Relational Database?”

First, consider:

- What is the data volume and complexity?
- What security level is required at data access?
- How many users exist per type of user?
- Will data be simultaneously accessible by many?
- How reliable must the storage mechanism be?

Then, ask: “*Isn’t a File Management System enough?*”

- XML, text files, shared files

Do I need a database?

Data is everywhere, everything is data

First ask

- How much data do I have?
- How complex is my data?
- How secure must the data be?
- Can I use simple file management? (XML, text files,...)
- How many users will access the data?

Examples near you

- Continente, Worten, SNS, Decathlon, Inscrições nos turnos, Moodle, IPMA, …

Will the world still need databases?

You will live,
your kids will will live,
your grandchildren will live,
your grand-grandchildren will have lived
*and your data will still be **stored** and **relevant***
to someone...

Accessing Databases

Universal programming language

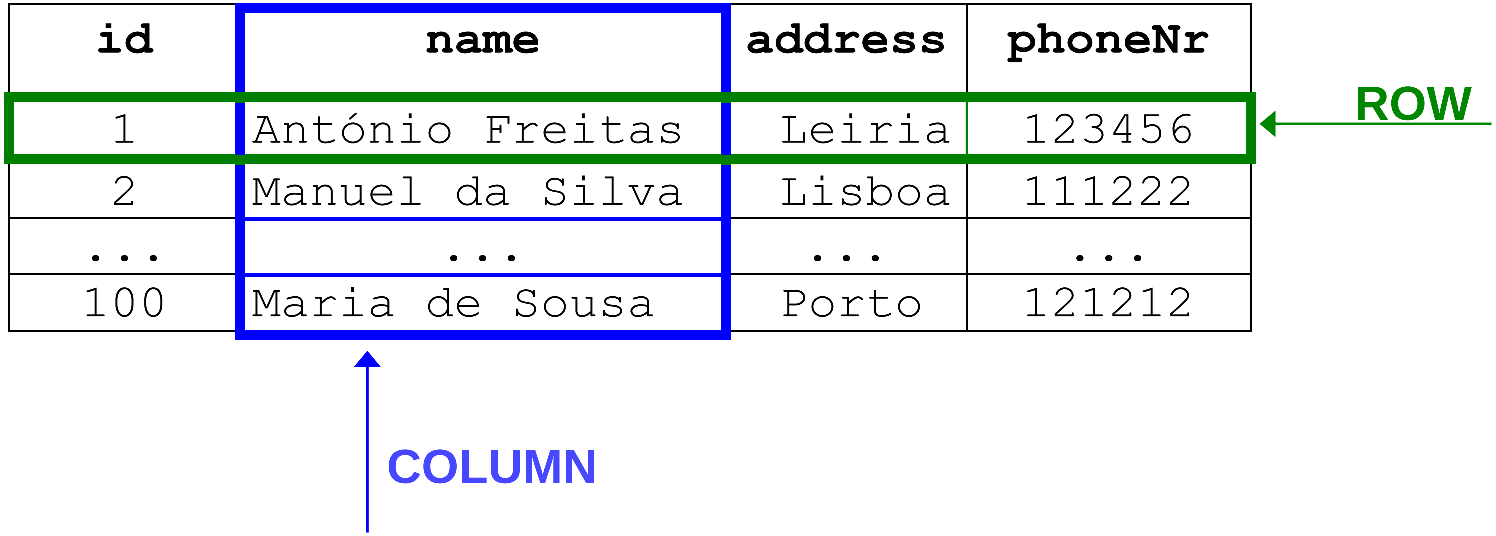
- SQL, *Structured Query Language*

Available basic data operations

- Insert new rows
- Update old data
- Delete old data
- Select existing data

Data storage in a database

TABLE clients



id	name	address	phoneNr
1	António Freitas	Leiria	123456
2	Manuel da Silva	Lisboa	111222
...	...	...	...
100	Maria de Sousa	Porto	121212

ROW

COLUMN

What do users expect from a database?

Different scenarios => different expectations

- Is data safe from attacks?
- Will data be safely stored?
- Will information retrieval be fast? (what is fast?)
- Will GUI programmers easily build interfaces?
- Is the data scalable?

≠

What do users expect from a database?

Different scenarios => different expectations

≠

Bad ways of looking at databases :(

Don’t think like this…

- “I’m a Web programmer”
- “The database is self-managed by the operating system” (the ***black box approach***)
- “What I really like is to code”

… just because you think or heard that

- “Databases are complex”
- “Databases are inflexible”
- “Databases are boring”
- “Databases will end ***sooooooon***” (old Inca prophecy?)
- “Databases *this*, databases *that*...”

No excuses allowed...

[you] “I only like programming”

[reality] Databases require heavy programming in SQL and PL/SQL (e.g., Oracle)

=> become a Database Programmer

[you] “I only like systems' management”

[reality] Databases need an optimized and always running database engine software and server

=> become a Database Administrator

[you] ““I’m *dig* idealizing solutions””

[reality] Databases require professionals with strong modeling and abstraction skills to keep it scalable

=> become a Database Designer/Analyst

No excuses allowed...

[you] “I’m a security freak”

[reality] Databases require data protection and strong programming skills to keep data private and attack-safe

=> become a Database Administrator

[you] “I only like building interfaces”

[reality] Nowadays, interfaces require the access to data, so if you don’t understand how the database works, your interfaces will perform poorly

=> learn strong SQL skills

=> learn the basics about database servers

No excuses allowed...

[you] “I’m shy...”

[reality] So am I! But data makes an excellent conversation topic.

=> learn databases

[you] “I’m too tall”

[reality] Good for you, you will reach the top faster!

=> learn databases

[you] “No, you don’t understand! I’m really tall!”

[reality] Good! You will see the world’s problems from above and better discover how data can solve them.

=> learn databases

Common excuses to do things the wrong way

"*I* have *always* programmed this way and it works just fine!"

"In the real world *everybody* does it like this!"

"These concepts are *obsolete*!"

Skipping database learning in a *data based* world: *utopia*

One often meets his destiny on the road he takes to avoid it.

Grand Master Oogway,
in **Kung-fu Panda**,
DreamWorks