CHALMERS
UNIVERSITY OF TECHNOLOGY

UNIVERSITY OF GOTHENBURG

Introduction to Databases

LECTURE 1

Dr. Philipp Leitner

philipp.leitner@chalmers.se

@xLeitix

GENERAL COURSE INFORMATION

Usual Time: 13:15 - 14:45

Lecture TUE, THU

Supervision WED, FRI

Language: entirely in English

(please also send mails etc. in English)

Credits: 7.5 HEC

1 HEC ~ 30 hours of effort (225 hours total)

Web Page

All information etc. is on GUL:

https://gul.gu.se/courseId/82750/content.do?id=39332486

Lecture list:

https://gul.gu.se/courseId/82750/content.do?id=39765276

Discussion Forum:

https://gul.gu.se/courseId/82750/courseDiscussListForums.do

Teachers

philipp.leitner@chalmers.se

Primary contact point for assignment questions

scheuner@chalmers.se

TAs

Justinas Stirbys
gusstirju@student.gu.se

Andres De Biase
gusdebana@student.gu.se

Renjith Sebastian
renjith@student.chalmers.se

Weekly Schedule

Lectures two times a week

(almost) every week TUE, THU (13:15 - 14:45)

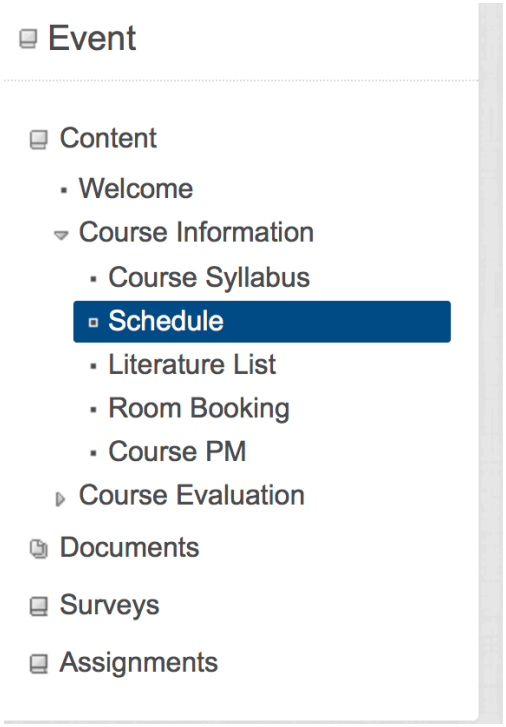
Alfons

Supervision twice a week

Every week while the assignments are running WED, FRI (13:15 - 14:45)

Mållgan

Regularly Check Lecture List on GUL



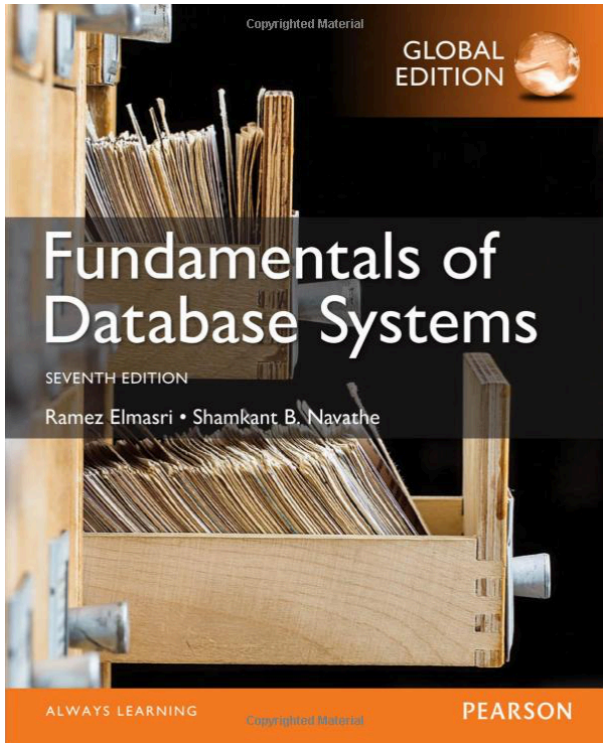
Previous Next Search Maximize the content Save as PDF

Schedule

Below is the **Class Schedule** for this course. This is in addition to the official Room Booking as published by the university. Consider the b
schedule the authoritive source for information, as we may have to cancel classes (which does not show up in the room reservation list):

Week	Date & Time	Event	Teacher	Room	Resources
Week 1					
	January 16th, 13:15 - 14:45	Lecture 1 - Kick-Off and Introduction to Databases	Philipp Leitner	Alfons	TBA
	January 17th, 13:15 - 14:45	*No supervision*			
	January 18th, 13:15 - 14:45	Lecture 2 - Entity Relationship Diagrams	Philipp Leitner	Alfons	TBA
	January 19th, 13:15 - 14:45	*No supervision*			
Week 2					

Text Book



Fundamentals of Database Systems
by Ramez Elmasri and Shamkant B. Navathe
Pearson Global Edition
7th Edition

(older editions are fine for most of the lecture, but the NOSQL and Map/Reduce part is not covered)

https://www.adlibris.com/se/bok/fundamentals-of-database-systems-global-edition-9781292097619

Grading

Written Exam:

4.5 hec

Pass with Distinction **(VG),** Pass **(G)** and Fail **(U)**

3-hour written exam in the week of March 12th

Assignments:

3 hec

Pass **(G)** and Fail **(U)**

Detailed Grading Scheme

Written Exam:

- **VG: >= 90% of exam points**
- **G: >= 50% of exam points**
- **U: < 50% of exam points**

Assignments:

- **G: >= 70% of assignment points in all three assignments**
- **U: < 70% of assignment points in at least one assignment**

Assignments

All assignments are done in **teams of two**. Choose your own partner and register like that in GUL.

Assignment 1:

Entity Relationship Diagrams and Relational Algebra

Due Feb. 5th, Midnight

Assignment 2:

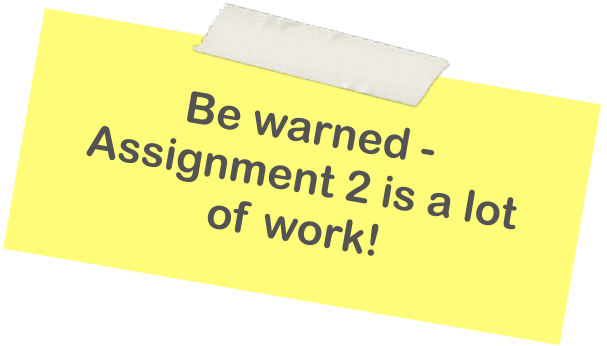
SQL and JDBC

Due Feb. 19th, Midnight

Assignment 3:

MongoDB and Map/Reduce

Due Mar. 5th, Midnight



Next Written Exam Dates

Main exam date: **March 13th, 2018**

(after that: June 7th and August 22nd, 2018)

Forum

We will be operating a discussion forum on GUL:

https://gul.gu.se/courseId/82750/courseDiscussListForums.do

This will be the best way to get answers to your questions

We will check the forum frequently, and so should you.

Students are encouraged to answer questions as well!

Self- and Peer-Assessment Forms

There's a form on GUL that **every student** should submit for **every assignment**

- Three questions:
  - What did you do?
  - What did your partner do?
  - How much time did you invest (in hours)?
    - The last question is mainly to improve our assignments for the next years - this has no impact on grading.

Print and sign the form, and bring it into the lecture, supervision, or my office hour.

Office Hour

I hold office hours

THU, 09:00 - 10:00

in my office (Jupiter, 4th floor, room 411).

Outside of office hours **you will need an appointment**. Drop me an email if something urgent comes up.

Student Representatives

We need 2 - 5 student representatives

- **Feedback during 2 meetings (one midway, one after the lecture)**
- **Can raise issues at any time**

Volunteers - please contact me after the lecture

Bug Bounty Programme

You can get a small number of extra points for the **next assignment grading** for each unique bug you report

Bugs can be:

- Errors in the slides
- Errors in assignment texts
- Errors in assignment solutions

Bug Bounty Programme - Guidelines

We'll reserve the right to assign points based on the severity of the bug:

Roughly:

Typos, grammar errors, other inconsequential stuff: *0 - 1 points*

Inelegant / inefficient solutions: *2 - 3 points*

Proposed solution does not work at all: *4 - 5 points*

Bug reports should come with an error case and/or proposed improvement.

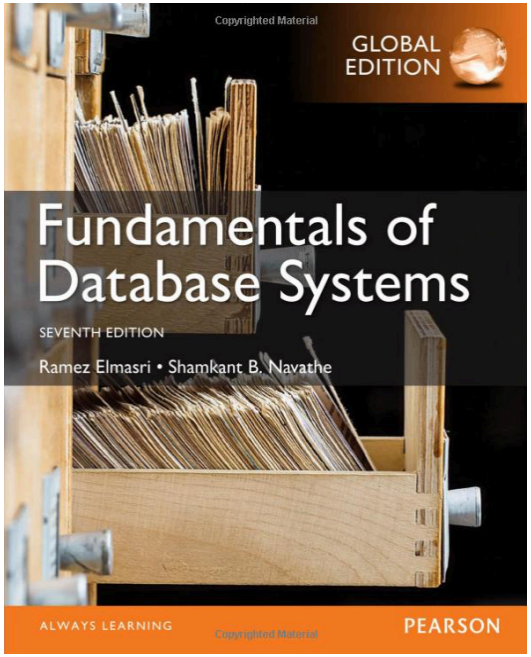
Submit bugs via forum:

First come / first serve (each bug will only be awarded a bounty once)

LECTURE 1

Covers …

Chapter 1 and Chapter 2
(+ short information theory detour)



What we will be covering

About data, information, and knowledge

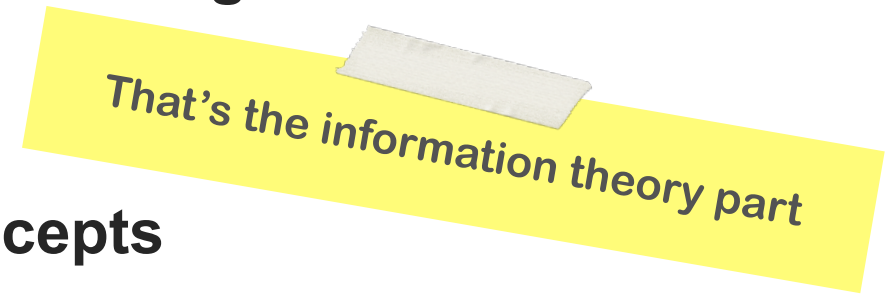
Types of databases

What's a database, really?

Database architecture and concepts

What we will be covering

- About data, information, and knowledge
- Types of databases
- What's a database, really?
- Database architecture and concepts

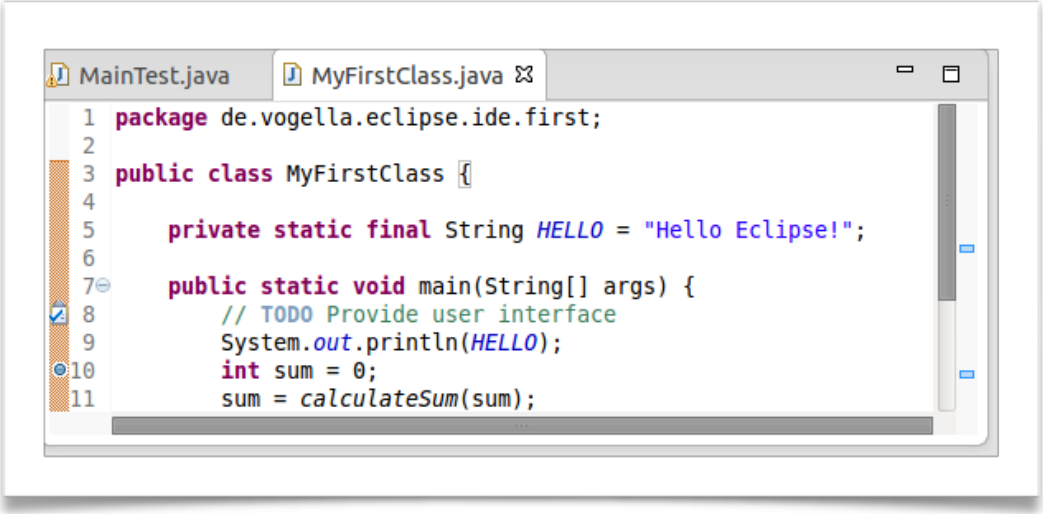


Most programming is about *data*

Simplest cases:

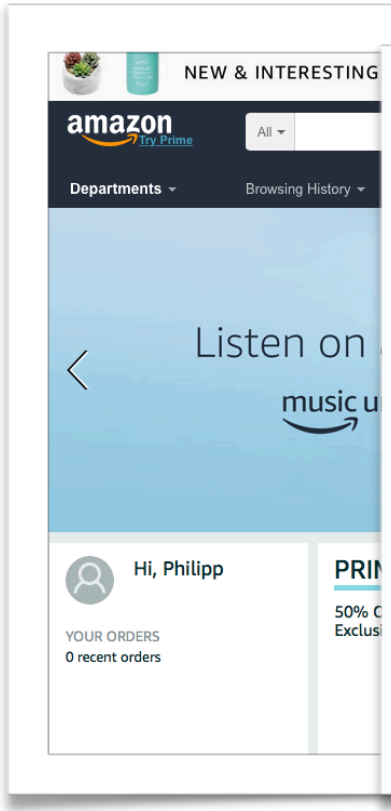
Variables

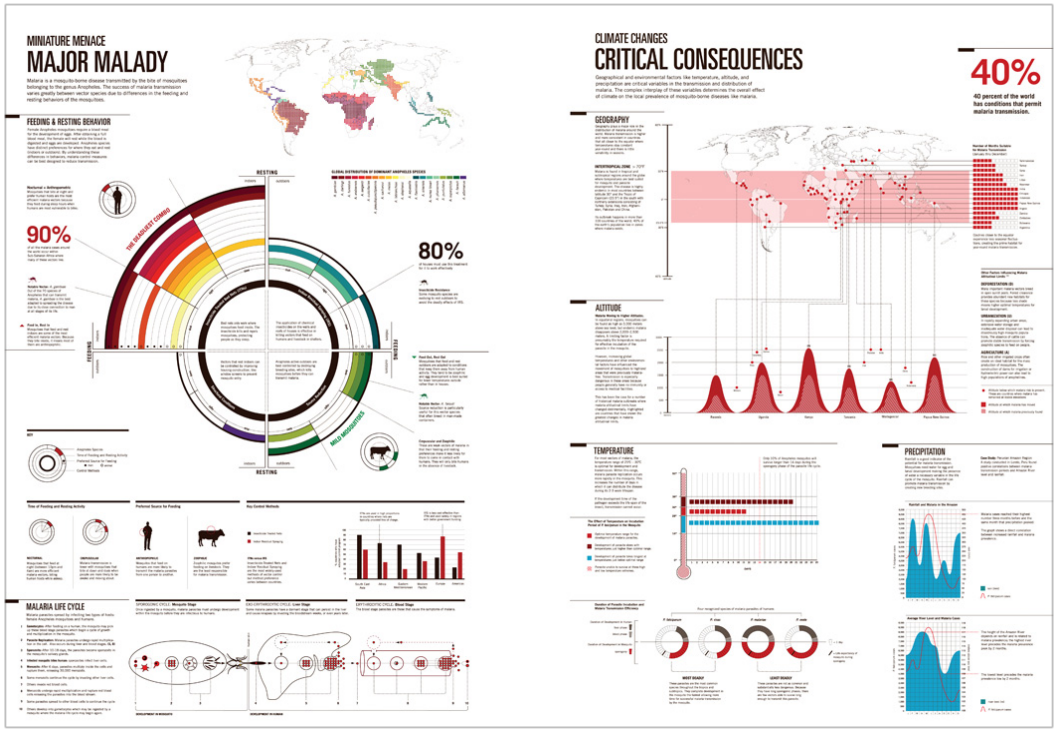
Text files



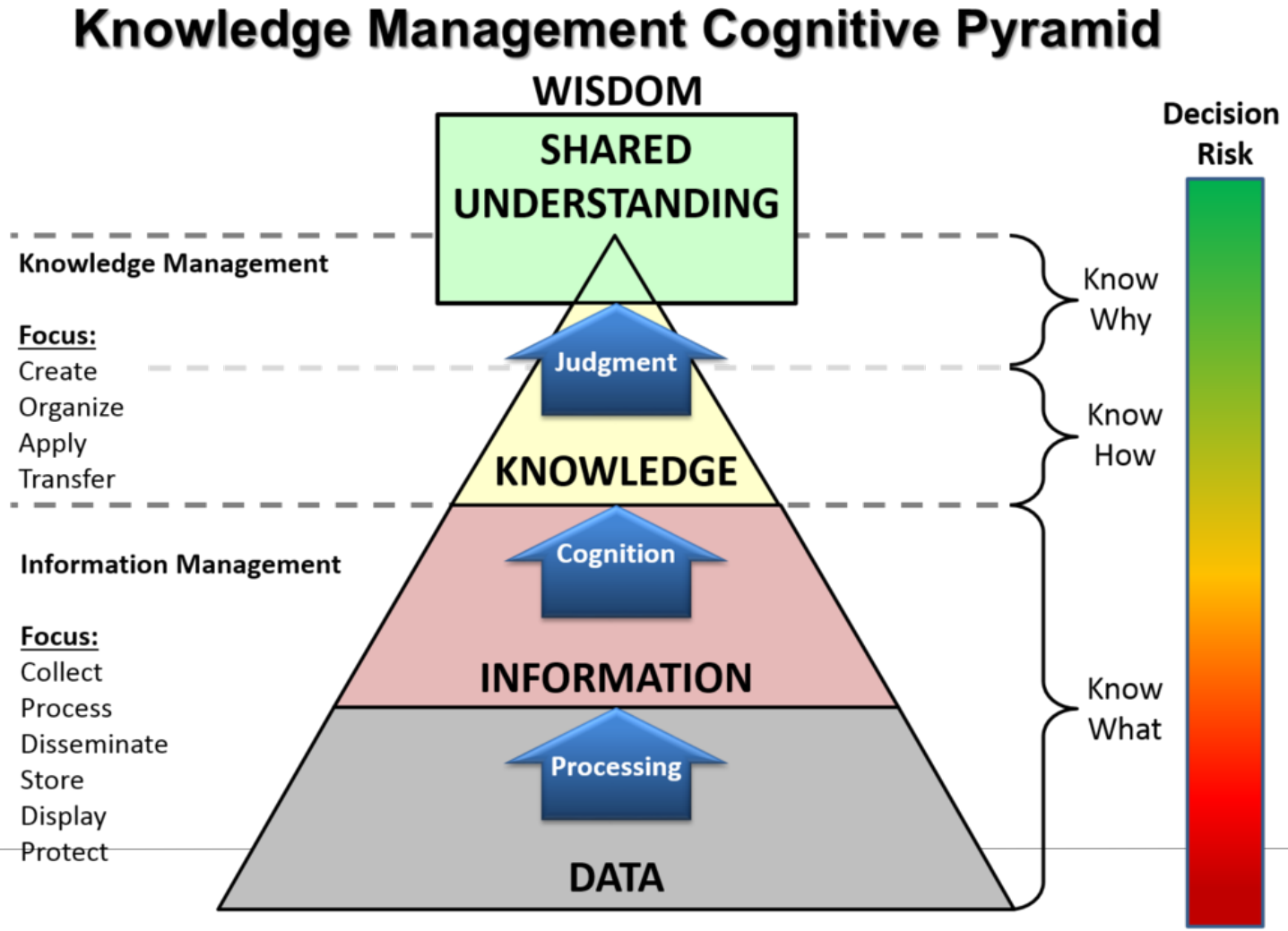
… **not sufficient** for many use cases

Most programming is about *data*





https://en.wikipedia.org/wiki/DIKW_pyramid



Data

Signals that are stored (on disk, in memory, …)

No “**meaning**” in itself

May or may not be **correct** (“noise”)

May or may not be **redundant**

Information

Data with **labels** (== some sort of description)

- Can be a **schema**
- Or just implicit labels

Generally **assumed to be** not redundant nor wrong

Data and information is what we primarily deal with!

(but the meaning will need to come from your programs)

Knowledge / Wisdom

Less well-defined concepts

Generally:

Information + meaning

Typically **not the realm of computer programmes**

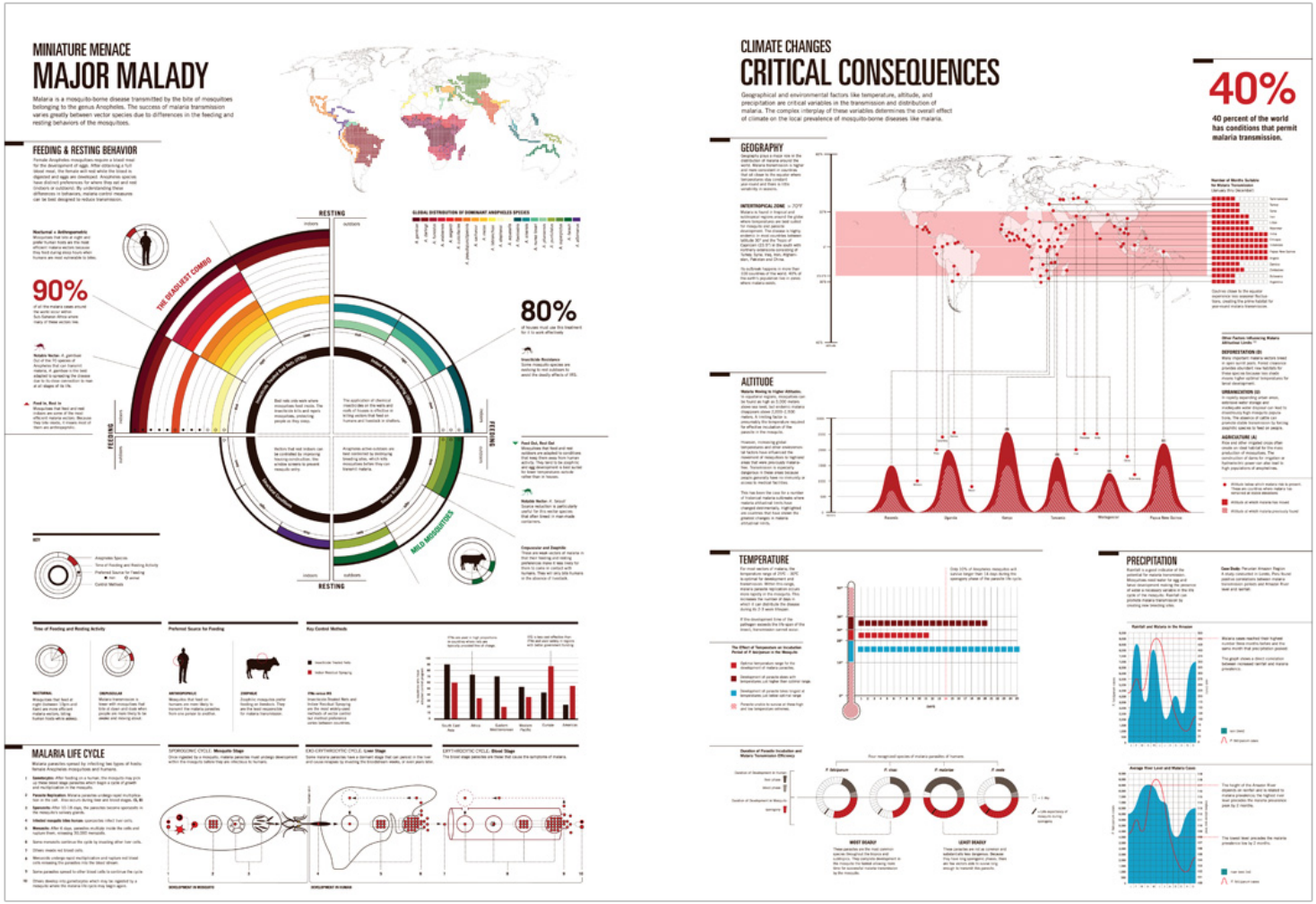
But AI (artificial intelligence) is stretching the boundaries

Data Visualization

Common (human) way to **go from information to knowledge**

Good visualisation helps understanding

- Quick overview of lots of data
- Easier to establish relationships
- Easier to compare data


MINIATURE MENACE
MAJOR MALADY
Malaria is a mosquito-borne disease transmitted by the bite of mosquitoes belonging to the genus Anopheles. The success of malaria transmission varies greatly between vector species due to differences in the feeding and resting behaviors of the mosquitoes.
FEEDING & RESTING BEHAVIOR
90%
80%
THE DEADLIEST COMBO
MILD MOSQUITOES
RESTING
FEEDING
MALARIA LIFE CYCLE
CLIMATE CHANGES
CRITICAL CONSEQUENCES
Geographical and environmental factors like temperature, altitude, and precipitation are critical variables in the transmission and distribution of malaria. The complex interplay of these variables determines the overall effect of climate on the local prevalence of mosquito-borne diseases like malaria.
40%
40 percent of the world has conditions that permit malaria transmission.
GEOGRAPHY
ALTITUDE
TEMPERATURE
PRECIPITATION

Types of Visualizations

Textual (e.g., printing to `System.out`)

Tables

MS Excel style visualization

Plots

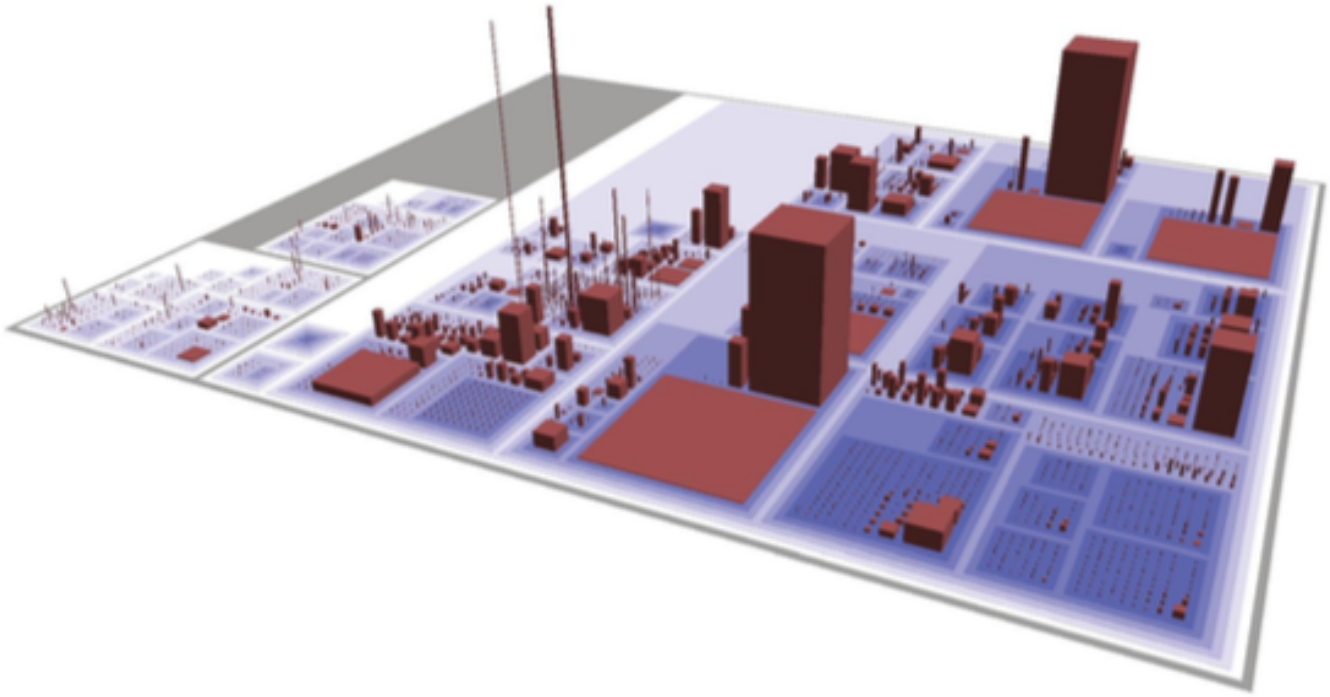
Line charts, bar charts, boxplots, pie charts, …

Custom

E.g., the “city metaphor” for visualization of source code

City Metaphor

Wettel and Lanza. Software systems as cities. VISSOFT 2007.



Types of Visualizations

Textual (e.g., printing to `System.out`)

Tables

MS Excel style visualization

Plots

Line charts, bar charts, boxplots, pie charts, …

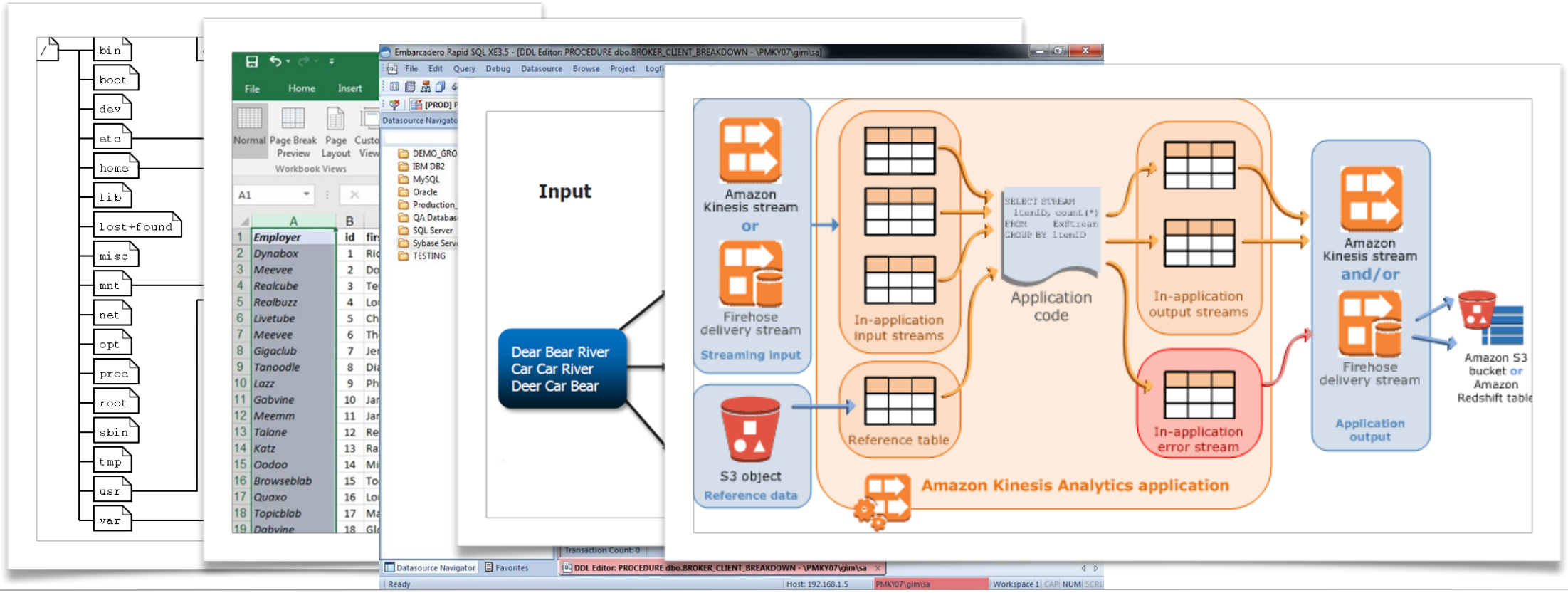
Custom

E.g., the “city metaphor” for visualization of source code

What we will be covering

- About data, information, and knowledge
- Types of databases
- What's a database, really?
- Database architecture and concepts

Types of databases



Types of databases

Files and File Systems

Spreadsheets

SQL

NOSQL

Cloud services

Basic data operations - CRUD

Create
Read
Update
Delete

Querying

Querying is in reality where the “meat” in most real database systems is

Special form of “read”

Whereas “read” delivers exactly one entry, a query is used to **retrieve**:

- More than one entry
- Parts of one or more entries (e.g., only the IDs)
- Based on complex constraints

Examples:

- Give me all files that end with “*exe”
- Give me the last names of all students enrolled in DIT032

What we will be covering

- About data, information, and knowledge
- Types of databases
- What's a database, really?
- Database architecture and concepts

Database

Collection of related data

Miniworld or **universe of discourse** (UoD)

Represents **some aspect** of the real world

Logically coherent collection of data with inherent meaning

Built for a **specific purpose**

An Example

UNIVERSITY database

Information concerning students, courses, and grades in a university environment

Data records

STUDENT

COURSE

SECTION

GRADE_REPORT

PREREQUISITE

An Example

Database > Records > Data Items

‘chalmers’ > ‘course’ > course_name: ‘Database’

‘chalmers’ > ‘student’ > student_number: 17

An Example

Database > Records > Data Items

‘chalmers’ > ‘course’ > course_name: ‘Database’
‘chalmers’ > ‘student’ > student_number: 17

Meta-data

Data

An Example

Database > Records > Data Items

string

‘chalmers’ > ‘course’ > course_name: ‘Database’

‘chalmers’ > ‘student’ > student_number: 17

integer

An Example

Database > Records > Data Items

‘chalmers’ > ‘course’ > course_name: ‘Database’

‘chalmers’ > ‘student’ > student_number: 17  *Enrolled In*

STUDENT

Name	Student_number	Class	Major
Smith	17	1	CS
Brown	8	2	CS

COURSE

Course_name	Course_number	Credit_hours	Department
Intro to Computer Science	CS1310	4	CS
Data Structures	CS3320	4	CS
Discrete Mathematics	MATH2410	3	MATH
Database	CS3380	3	CS

SECTION

Section_identifier	Course_number	Semester	Year	Instructor
85	MATH2410	Fall	07	King
92	CS1310	Fall	07	Anderson
102	CS3320	Spring	08	Knuth
112	MATH2410	Fall	08	Chang
119	CS1310	Fall	08	Anderson
135	CS3380	Fall	08	Stone

GRADE_REPORT

Student_number	Section_identifier	Grade
17	112	B
17	119	C
8	85	A
8	92	A
8	102	B
8	135	A

PREREQUISITE

Course_number	Prerequisite_number
CS3380	CS3320
CS3380	MATH2410
CS3320	CS1310

Figure 1.2
A database that stores student and course information.

Meta-data

“Data about data”

Data that is used to describe what the “actual” data is allowed to look like

Think Java classes and how they relate to object instances

Example Queries

Get a student

List the names of students who took the section of the ‘Database’ course offered in fall 2008

List the prerequisites of the ‘Database’ course

Example Updates

Change the class of ‘Smith’ to sophomore

Enter a grade of ‘A’ for ‘Smith’ in the database section of last semester

Delete the grade of ‘Smith’ in last semester’s database section

Database Management Systems

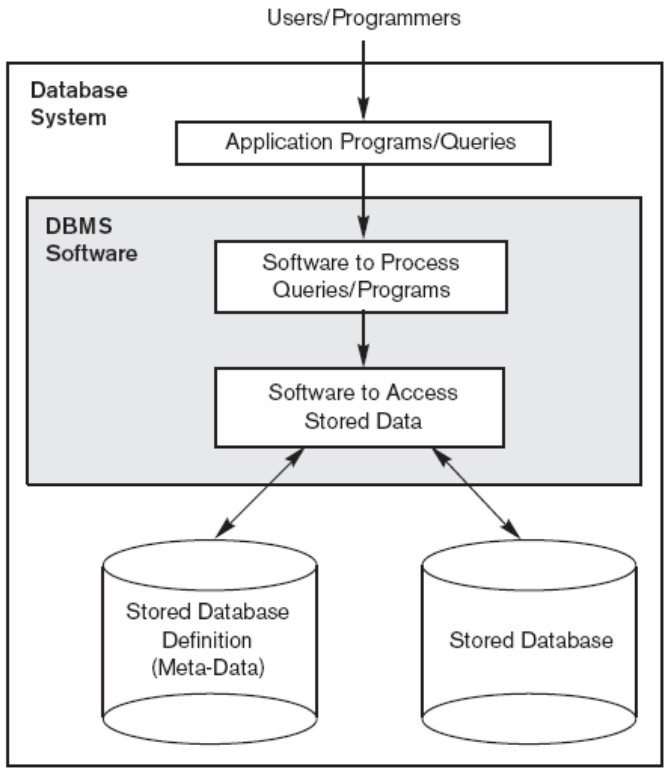


Figure 1.1
A simplified database system environment.

Advantages of (many) DBMS

- Providing **persistent storage** for program objects
  - But: Impedance **mismatch problem**
- Providing **efficient querying** capabilities (search)
  - Indexing / optimized data structures
  - Query optimization
  - Caching
- Controlling **redundancy**
  - Data normalization
  - Denormalization

Advantages of (many) DBMS

- Ensuring **data integrity**
  - Data types
  - Referential integrity
  - Uniqueness constraints
- Ensuring **transactional properties**
  - **“ACID”** properties
- Restricting **unauthorized access**
  - Access control
- **Backup and recovery**

ACID

Atomicity

All changes are applied, or nothing is applied

Consistency

Database is always in a consistent state (no “in-between” time)

Isolation

Concurrency control - guarantees that concurrent transactions are not interfering

Durability

Once a change is applied, it remains even through failures

What we will be covering

About data, information, and knowledge

Types of databases

What's a database, really?

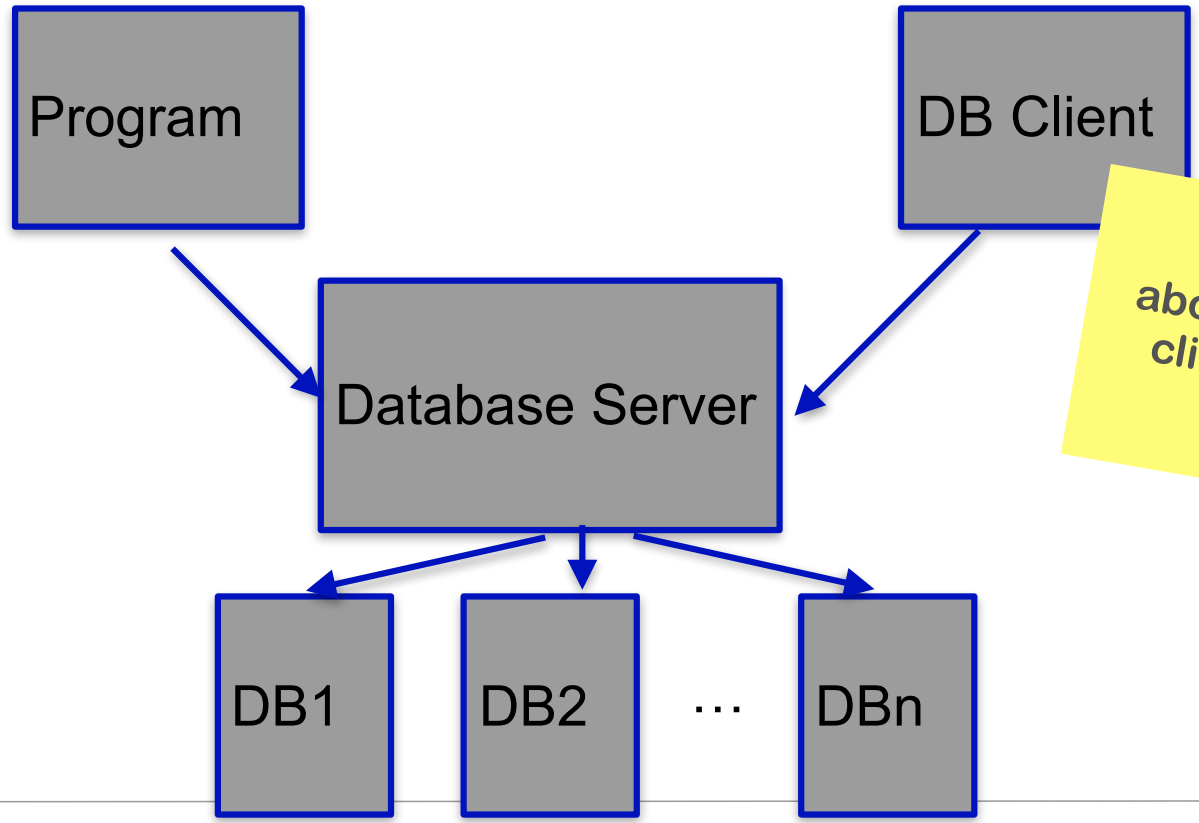
Database architecture and concepts

Client/Server Architecture

(diff. kinds of) **Clients**

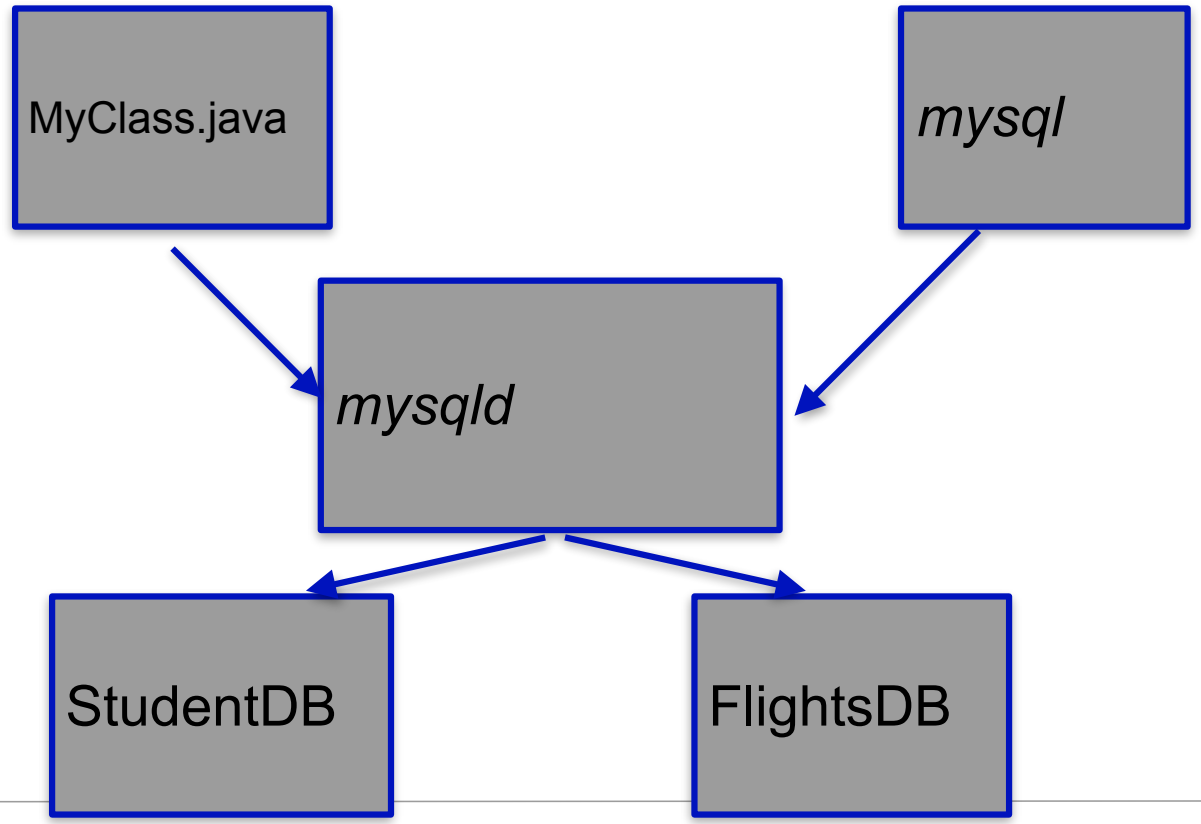
actual **DBMS**

different **Databases**



nothing magical about the cmdline DB client - just another program

Example: MySQL



Roles in a Database World - User Side

Database **Administrator**

- Manages the DBMS itself

Database **Designer**

- Manages the meta-data (schema), but not the data itself

Database **User**

- Developers writing programs using DBs
- “Naive” end users (use pre-canned queries)
- Sophisticated end users (write custom, interactive queries)

Roles in a Database World - Backend Side

Database **Developers**

- Build the DBMS itself

Database **Tool Developers**

- Responsible for building dev. tools, e.g., ORM libraries

Operators and Maintenance

- Manage hardware and software that the DBMS is running on

Some Basic Concepts and Terminology

Database schema

Description of a database

Entities

"Things" that exist in the schema (students, courses, …)

Attributes

Describe entities

Instances

Concrete objects matching an entity

Cp.: instances in OOP

Some Basic Concepts and Terminology

Schema diagram

Displays selected aspects of schema

Schema construct

Each object in the schema

Database state or snapshot

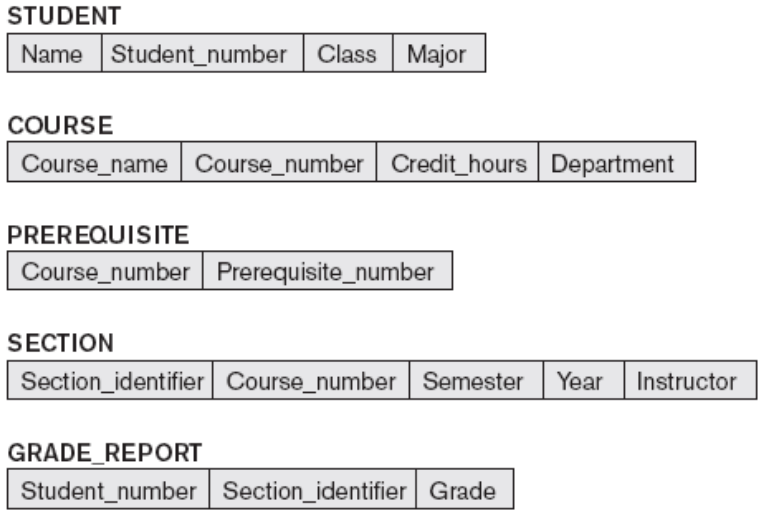
Data in database at a particular moment in time

Change in the data represented by series of snapshots

Schema Evolution

Some Basic Concepts and Terminology

Figure 2.1
Schema diagram for the database in Figure 1.2.



[6]Schema changes are usually needed as the requirements of the database applications change. Newer database systems include operations for allowing schema changes, although the schema change process is more involved than simple database updates.

[7]It is customary in database parlance to use *schemas* as the plural for *schema*, even though *schemata* is the proper plural form. The word *scheme* is also sometimes used to refer to a schema.

Schema Evolution

In practice, not only the data but also the **schema** changes over time

E.g., a new version of your program uses some extra attributes

Schema Evolution

Difficult because old data needs to be **migrated**

Three-Schema Architecture

ANSI/SPARC Standard

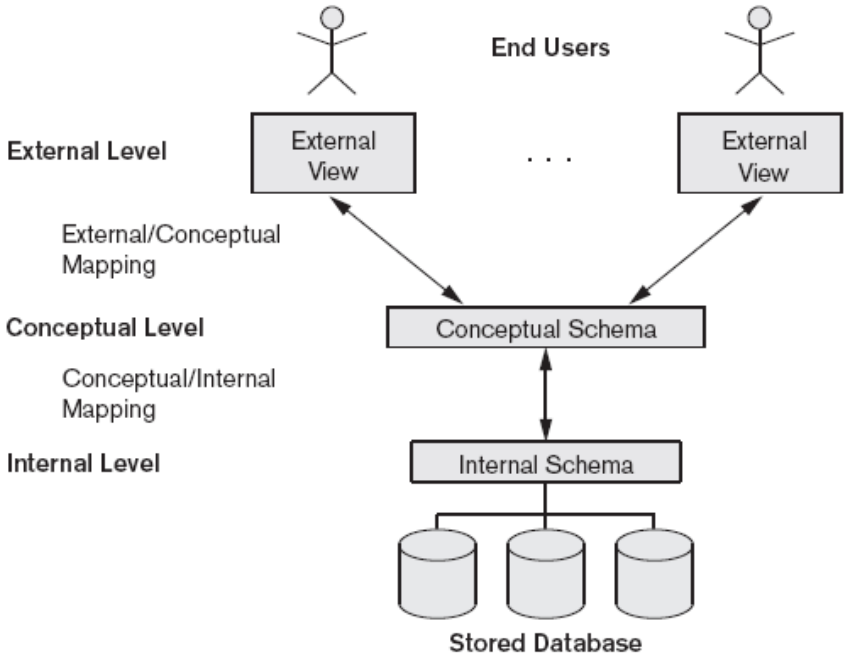


Figure 2.2
The three-schema architecture.

A More Comprehensive View on a Relational Database

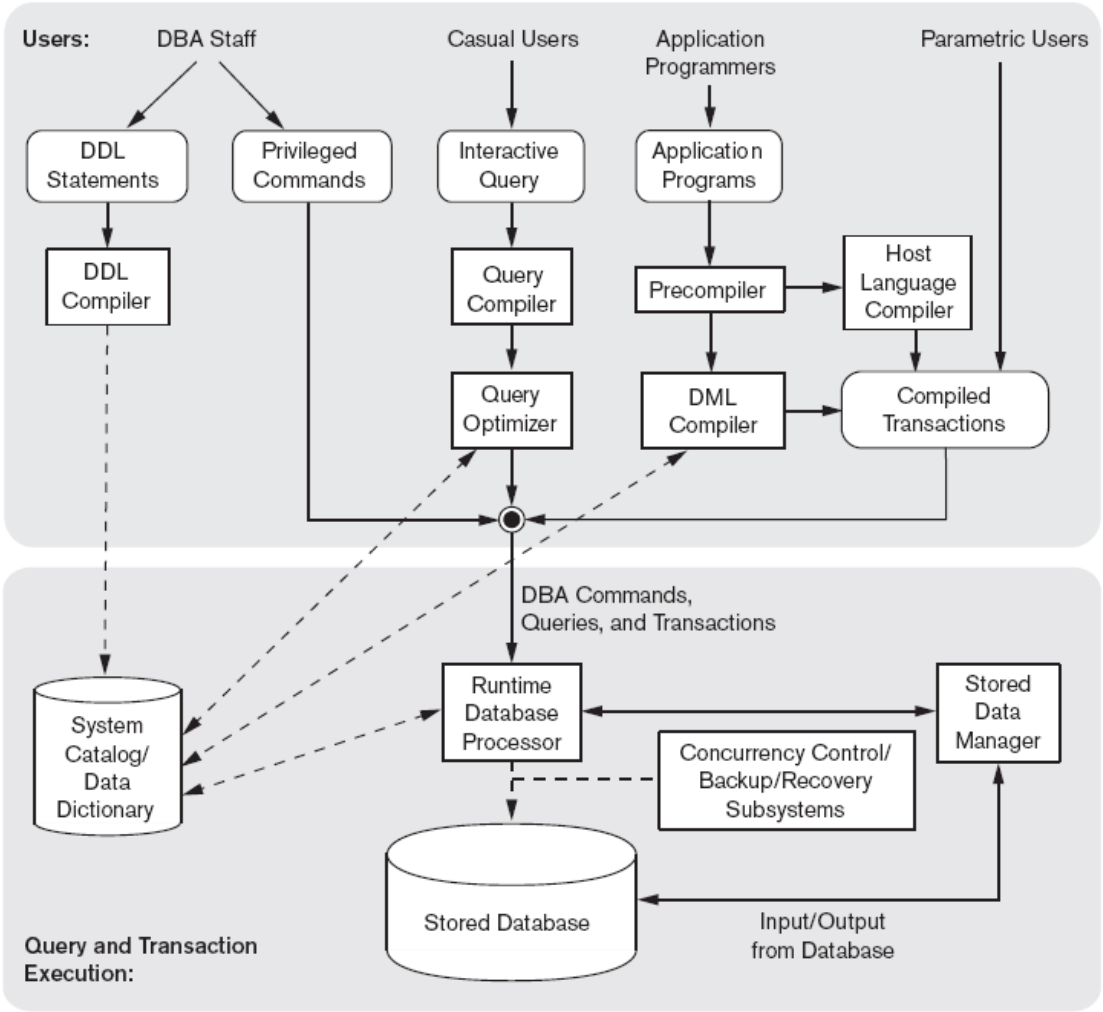


Figure 2.3
Component modules of a DBMS and their interactions.

Different tools for different users

A More Comprehensive View on a Relational Database

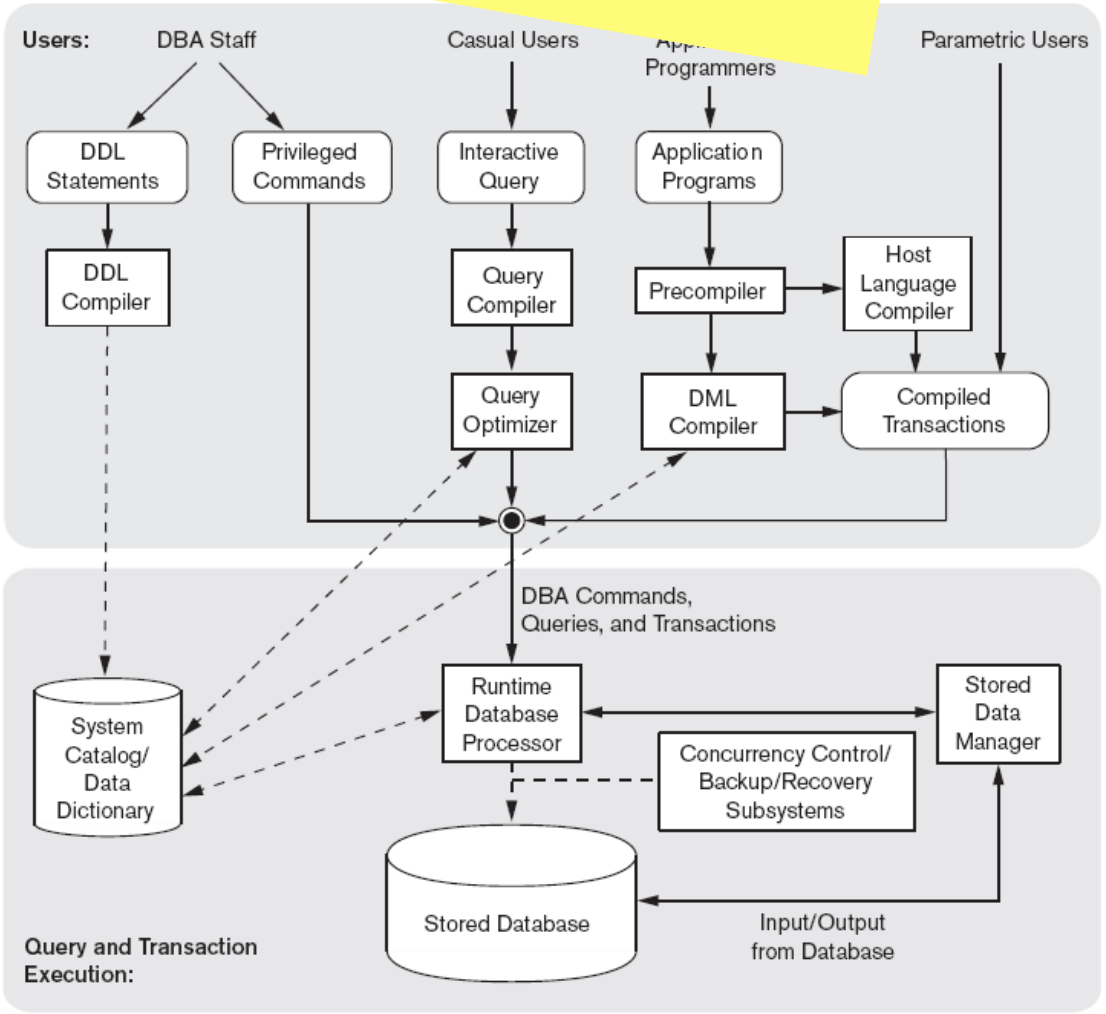


Figure 2.3
Component modules of a DBMS and their interactions.

A More Comprehensive View on a Relational Database

Client/Server Architecture

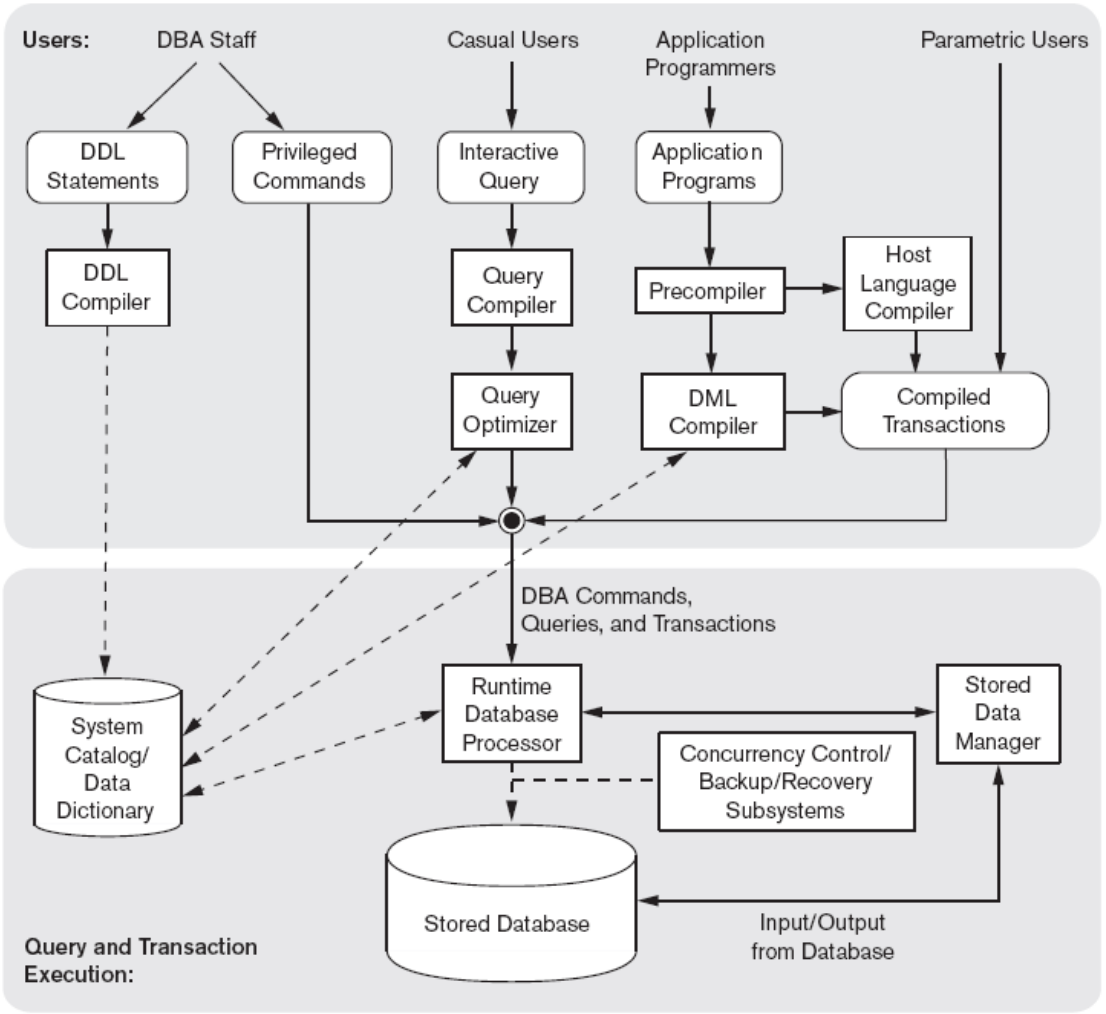


Figure 2.3
Component modules of a DBMS and their interactions.

A More Comprehensive View on a Relational Database

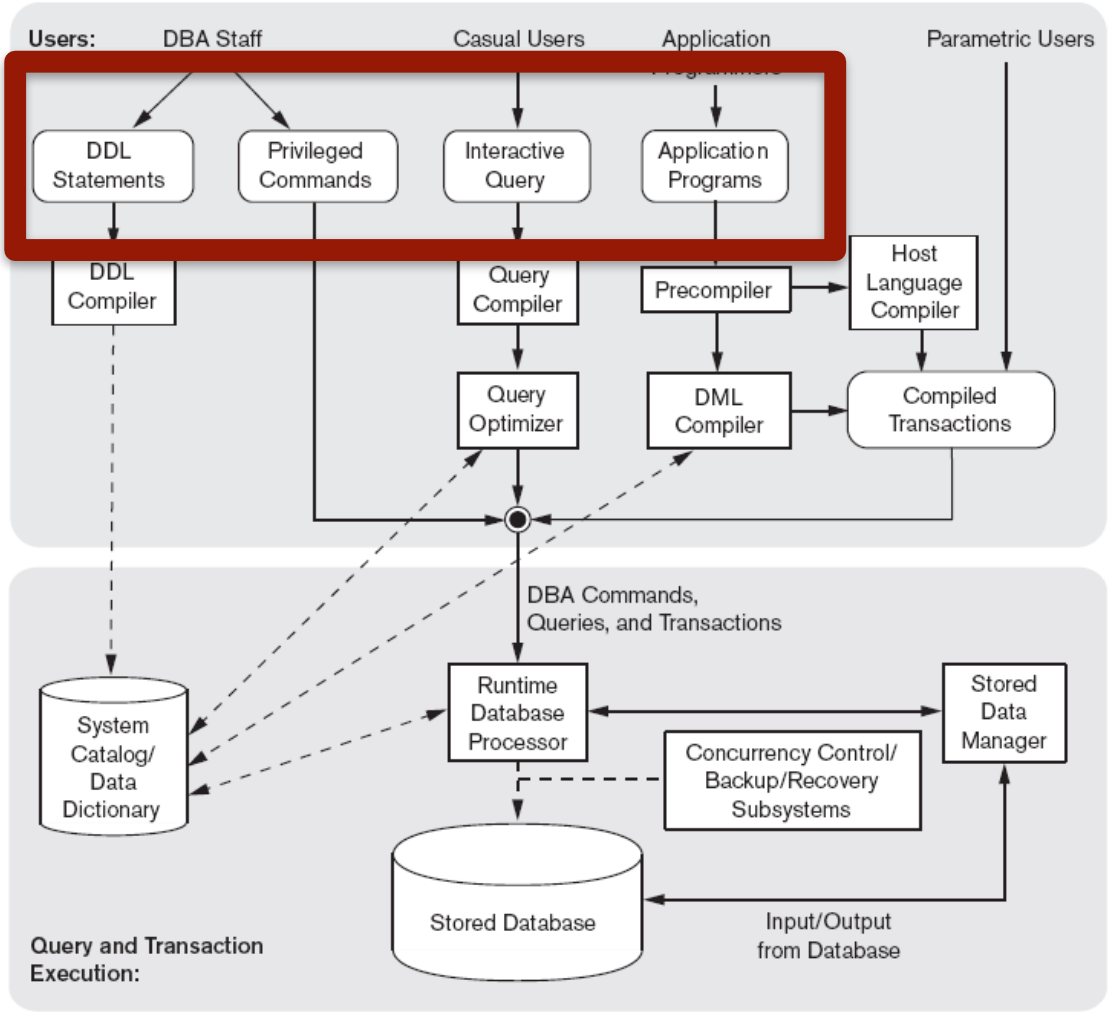


Figure 2.3
Component modules of a DBMS and their interactions.

Interaction with the DBMS

Menu-based interfaces for Web clients or browsing

Forms-based interfaces

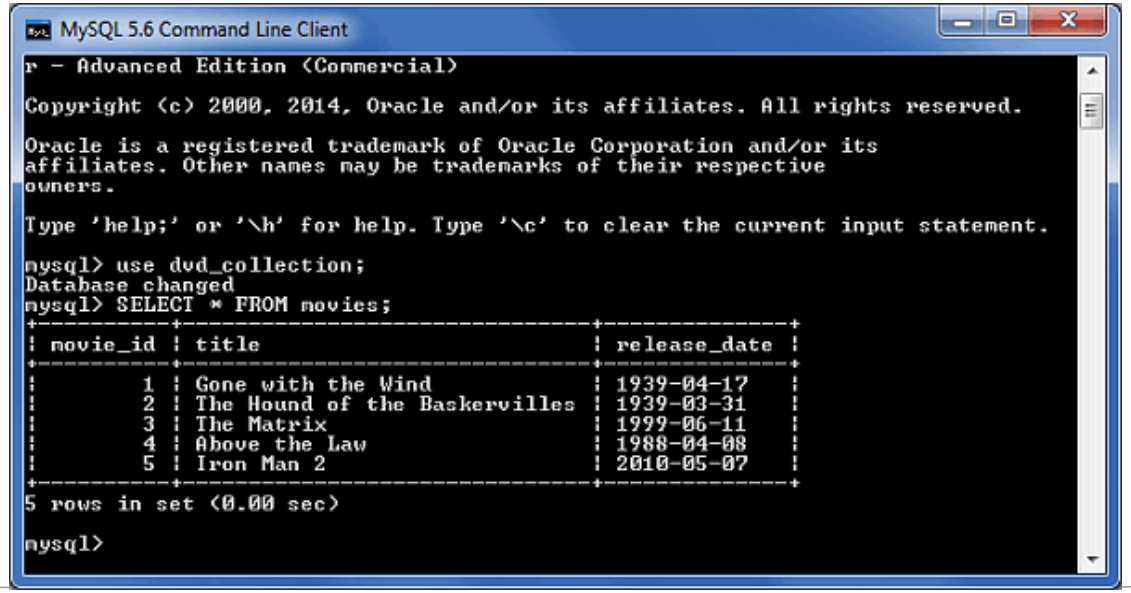
Graphical user interfaces (GUI)

Natural language interfaces

Speech input and output

Interfaces for parametric users

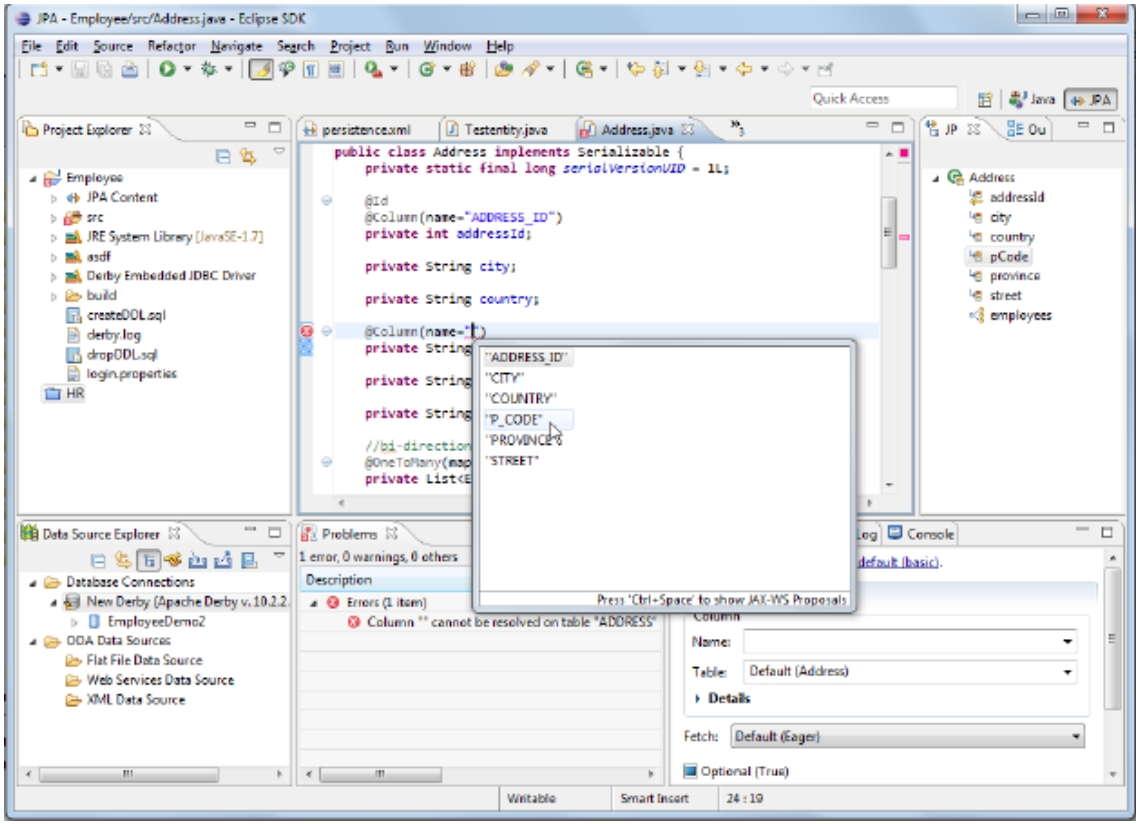
Interfaces for the DBA



This what we will use

Interaction with the DBMS

Over an **API** (e.g., Java)



Key Takeaways

Data, information, and knowledge

- What ways to we have to visualize data?

Basic database operations

- CRUD - create, read, update, delete
- The importance of querying

Databases as **collections of meta-data and data**

The concept of **“Database Management Systems”**

Key Takeaways

Advantages of using a DBMS

Incl. relational integrity, ACID properties, powerful queries, …

Basic database operations

CRUD - create, read, update, delete

The importance of querying

Roles in a database application

The ANSI/SPARC **Three-Schema Architecture**