INF115 Lecture 1: *Introduction to Databases*

Adriaan Ludl

Department of Informatics
University of Bergen

Spring Semester

INF115 course content

An introduction to ***methods for organizing, structuring, representing and storing large amounts of information***.

Emphasis on: - *techniques for **data modeling**,*

- *theory of **relational databases**.*

Important topics: *relational algebra, query language, storage media and storage methods.*

INF115 course content

An introduction to ***methods for organizing, structuring, representing and storing large amounts of information***.

Emphasis on:

- *techniques for **data modeling**,*
- *theory of **relational databases**.*

Important topics: *relational algebra, query language, storage media and storage methods.*

INF115 course content

An introduction to ***methods for organizing, structuring, representing and storing large amounts of information***.

Emphasis on: - *techniques for* ***data modeling,***

- *theory of* ***relational databases.***

Important topics: ***relational algebra***, ***query language***, ***storage media*** *and* ***storage methods.***

Practical information about INF115

- Course books

- Lecture schedule
- Group sessions

- Mandatory assignments
- Exam

- Software used in the course

Practical information about INF115

- Course books
- Lecture schedule
- Group sessions
- Mandatory assignments
- Exam
- Software used in the course

INF115 Databases and Modelling

Course page updated regularly!

https://mitt.uib.no/courses/33533

MITT UIB

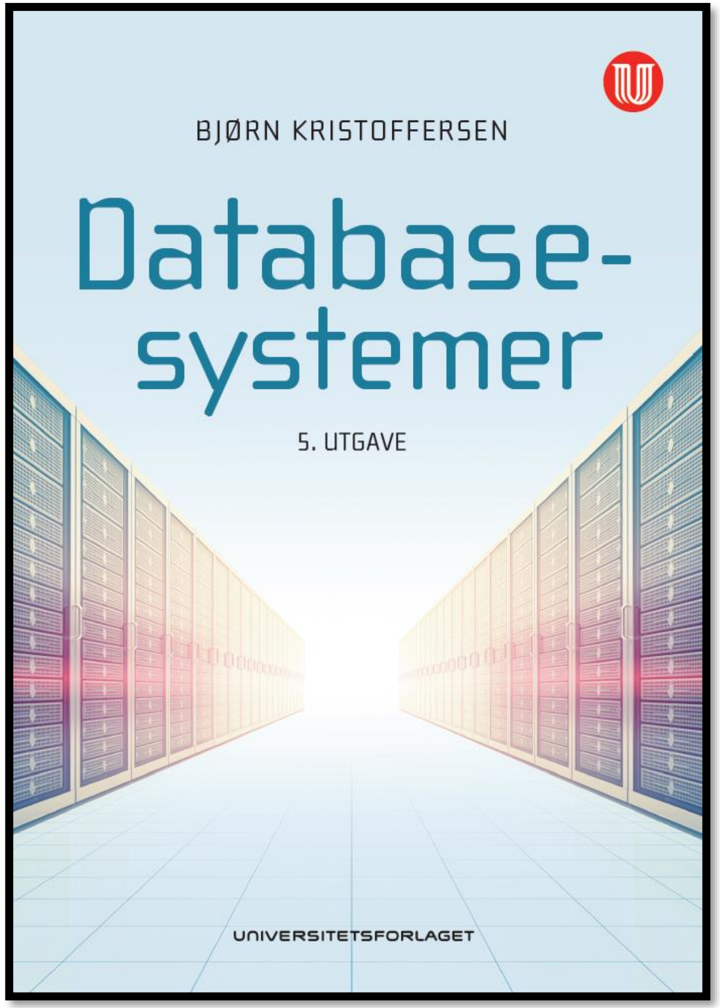
Course Books

- **Both** cover the INF115 course material

- Students can choose the one they prefer

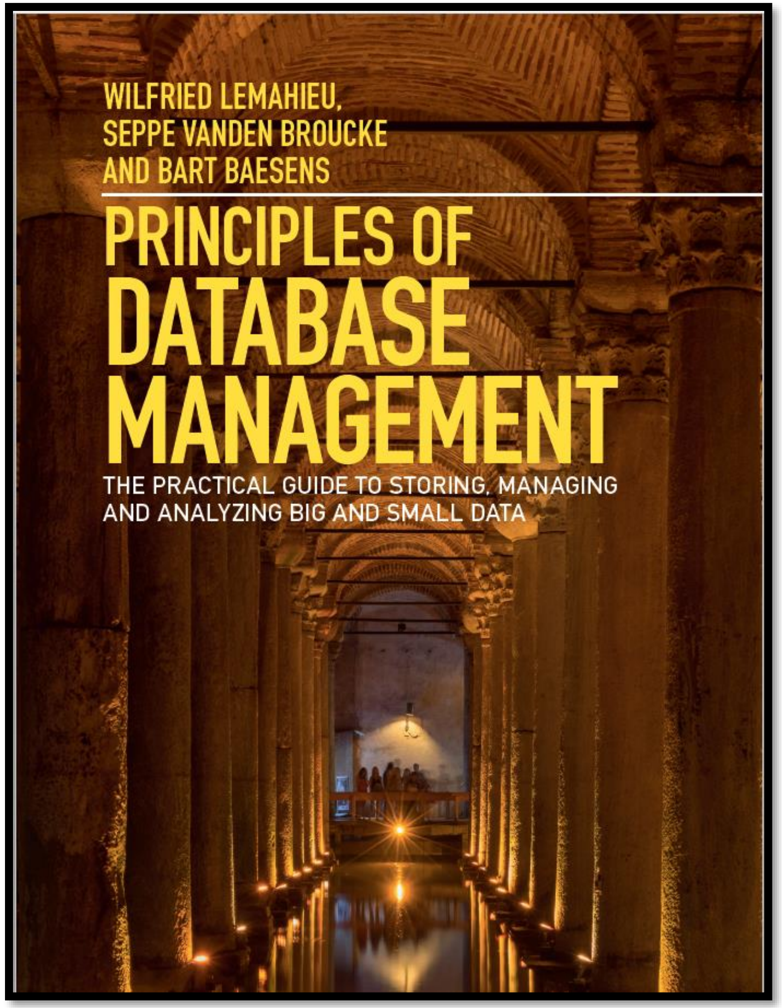
In Norwegian:

https://www.dbsys.info/Databasesystemer/



In English:

https://www.pdbmbook.com/


WILFRIED LEMAHIEU, SEPPE VANDEN BROUCKE AND BART BAESENS
PRINCIPLES OF DATABASE MANAGEMENT
THE PRACTICAL GUIDE TO STORING, MANAGING AND ANALYZING BIG AND SMALL DATA

Lecture Schedule (preliminary)

MITT UIB

Lectures on Tuesdays (14:15 – 16:00)
will be recorded and Fridays (10:15 to 12:00)

Week	Day	Nr.	*Topic*
3	18.01.2022	1	*Introduksjon*
3	21.01.2022	2	*Tabeller og enklespørringer*
4	25.01.2022	3	*Tabelldefinisjon og Datamanipulering*
4	28.01.2022	4	*Spørringer mot flere tabeller*
5	01.02.2022	5	*Avanserte spørreteknikker*
5	04.02.2022	6	*Relasjonsmodellen*
6	08.02.2022	7	*Datamodellering med E/R*
6	11.02.2022	8	*Datamodellering med E/R*
7	15.02.2022	9	*E/R-diagrammer til tabeller*
7	18.02.2022	10	*Normalisering*
8	22.02.2022	11	*Filer og indekser*
8	25.02.2022	12	*Transaksjoner*
9	01.03.2022		*(vinterferie)*
9	04.03.2022		*(vinterferie)*
10	08.03.2022	13	*Databaseadministrasjon*
10	11.03.2022	14	*Web-applikasjoner*
11	15.03.2022	x	*No lecture - På vei uke*
11	18.03.2022	15	*XML og JSON*
12	22.03.2022	16	*XML og JSON*
12	25.03.2022	17	*Via Objekter to NoSQL*
13	29.03.2022	18	*Via Objekter to NoSQL*
13	01.04.2022		
14	05.04.2022	IL	*Invited Lecture*
14	08.04.2022	IL	*Invited Lecture*
15	12.04.2022		*(Paskeferie)*
15	15.04.2022		*(Paskeferie)*
16	19.04.2022	IL	*Invited Lecture*
16	22.04.2022	IL	*Invited Lecture*

Group session schedule

Sessions start on 24th of January:

- 7 groups to happen physically,
- 1 online group.

You can only attend the group for which you are registered !

<u>Self-registration from today</u>

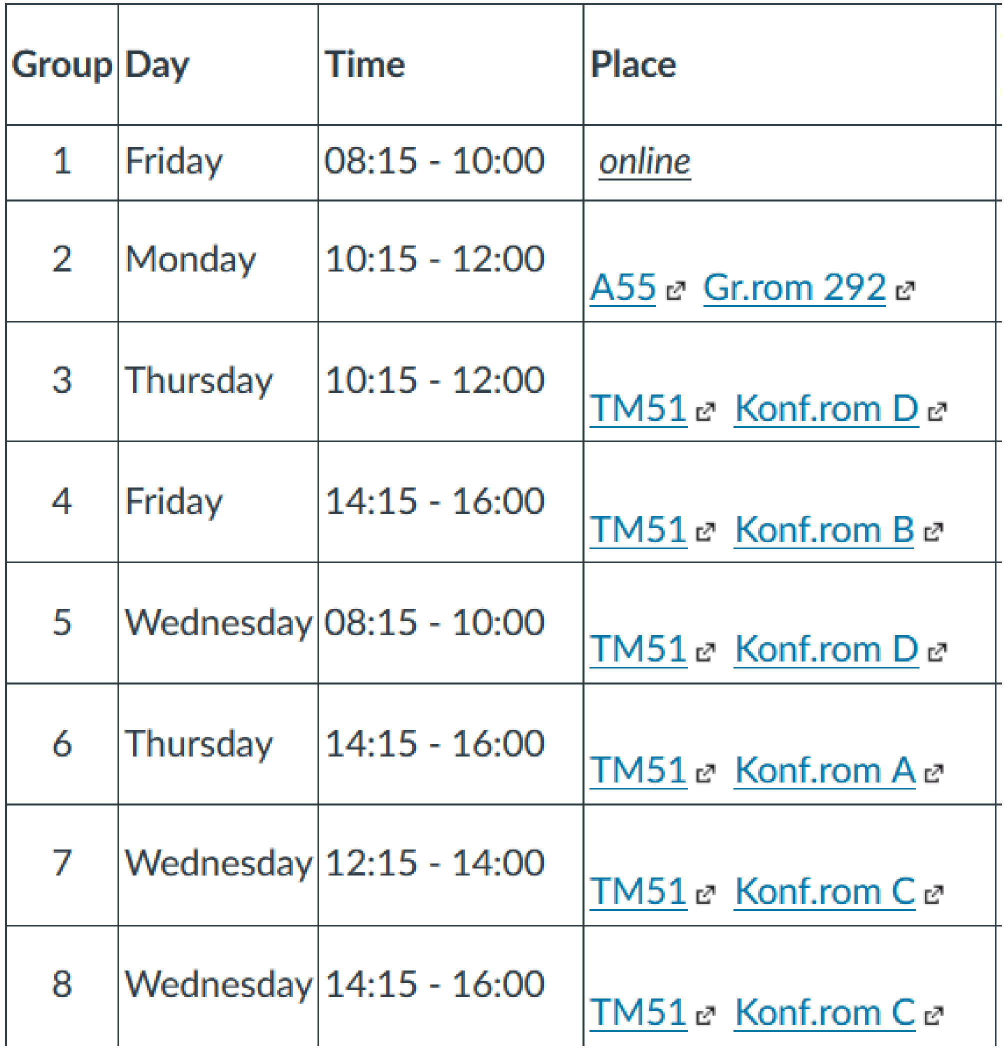
on mitt.uib.no

MITT UIB

- Teaching Assistant:

  Thomas Stautland

  Thomas.Stautland@uib.no



Group	Day	Time	Place
1	Friday	08:15 - 10:00	*online*
2	Monday	10:15 - 12:00	A55 Gr.rom 292
3	Thursday	10:15 - 12:00	TM51 Konf.rom D
4	Friday	14:15 - 16:00	TM51 Konf.rom B
5	Wednesday	08:15 - 10:00	TM51 Konf.rom D
6	Thursday	14:15 - 16:00	TM51 Konf.rom A
7	Wednesday	12:15 - 14:00	TM51 Konf.rom C
8	Wednesday	14:15 - 16:00	TM51 Konf.rom C

Mandatory assignments

- **Three mandatory hand-in assignments**,
- **Due dates** will be announced on https://mitt.uib.no/courses/33533/announcements

- Each assignment counts for 10% of the final grade.
- Each assignment must be passed (50%) to be allowed to take the exam.

- You can discuss the assignments in groups,
- but **each student submits their solution individually.**

Exam INF115

3-hour written exam on 30th of May, 30.05.2022.

Note: exam form will depend on regulations.

Exact location and time can be found (here) at a later date.

Be aware that location and time may change up to a few days before the exam!

Software used in the course

Install it now before the first group sessions ☺

- **MySQL Workbench**: visual database design tool : https://dev.mysql.com/downloads/workbench/
- **PHP**: webpage script language
- **XAMPP**: web server stack, **contains MariaDB and PHP**: https://www.apachefriends.org/download.html

- Recommended **E/R Diagram Editors**:
  - LucidChart
  - or MySQL Workbench Diagrams

- Recommended **SQL Stacks**:
  - phpMyAdmin
  - MySQL Workbench

For questions about the software please contact the TA or the student group leaders:

Teaching Assistant: Thomas Stautland
Thomas.Stautland@uib.no

How to study INF115 successfully ☺

- **Attend the lectures live online !**
  - ***Ask questions***
  - ***Discuss with your colleagues:*** *IRL or on discord ☺*
  - ***Read*** *the* ***chapters*** *corresponding to the lectures*

- **Attend the study groups !**
  - *Try to* ***answer each question by yourself,***
  - *Be* ***active*** *during group sessions,*
  - *attend the* ***consultancies*** *for the* ***mandatory assignments*** *!*

- **Study all the topics and problems again during exam preparation**
- **Reach out to us if you have questions !**

Questions on course organization ?

INF115 Databases and Modelling

Course page updated regularly!

https://mitt.uib.no/courses/33533

MITT UIB

First Quizz

Please answer the practice quizz on mitt.uib now ☺

(you can take it again later if you want)

Link:

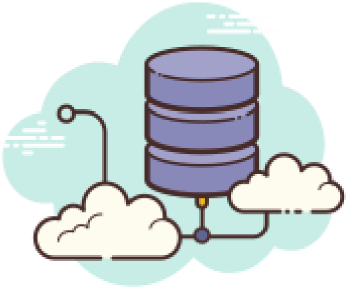
➢ https://mitt.uib.no/courses/33533/quizzes/22618

Chapter 1: Learning goals

- **Important uses** of database systems
- **Tasks** of a database system
- How to **develop** a database system
- **Important concepts** of database systems
- Understand the principles of **digital representation of data**

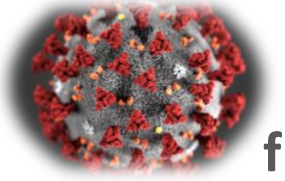
Databases are **behind** many software applications

- **Account management**
- **Cloud computing**, Email, Github …
- **Health**: Statistics, *Coronavirus test results*, Hospitals, *Genetics* …
- Banks, **Payment systems**
- **Universities**, Research, Weather forecast, **Machine learning** …
- Media, **Wikipedia**
- Online services: **government** (tax), banks, stock market, shopping, travel, videos, games …

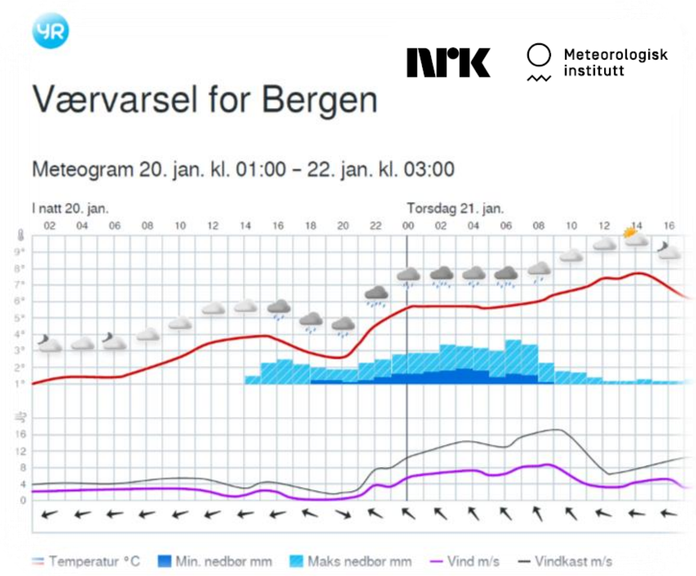


CoV3D

CORONAVIRUS 3D STRUCTURE DATABASE



fhi.no



YR

NRK

Meteorologisk institutt

Værvarsel for Bergen

Meteogram 20. jan. kl. 01:00 – 22. jan. kl. 03:00

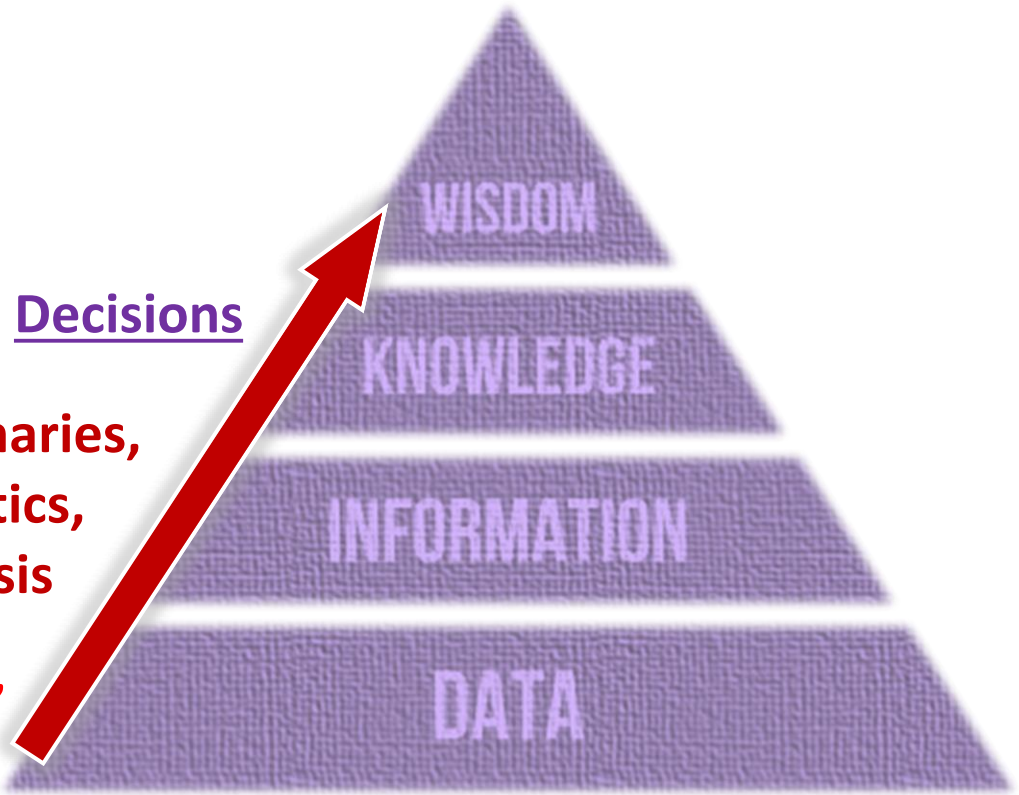
DIKW Pyramid (or hierarchy)

DIKW: **Data, Information, Knowledge, Wisdom**

“scientia potestas est” (*= knowledge is power*)
- Francis Bacon, philosopher (1561–1626)

Reference*:*
Henry, Nicholas L. (May–June ***1974****).*
*"****Knowledge Management****:*
A New Concern for Public Administration*".*
Public Administration Review. ***34*** *(3): 189–196.*
doi:10.2307/974902. JSTOR 974902.

Rowley, Jennifer (2007). "The wisdom hierarchy:
representations of the DIKW hierarchy".
Journal of Information and Communication Science. ***33*** *(2): 163–180.*
doi:10.1177/0165551506070706



Decisions

Summaries,
Statistics,
Analysis

Registration,
Collecting

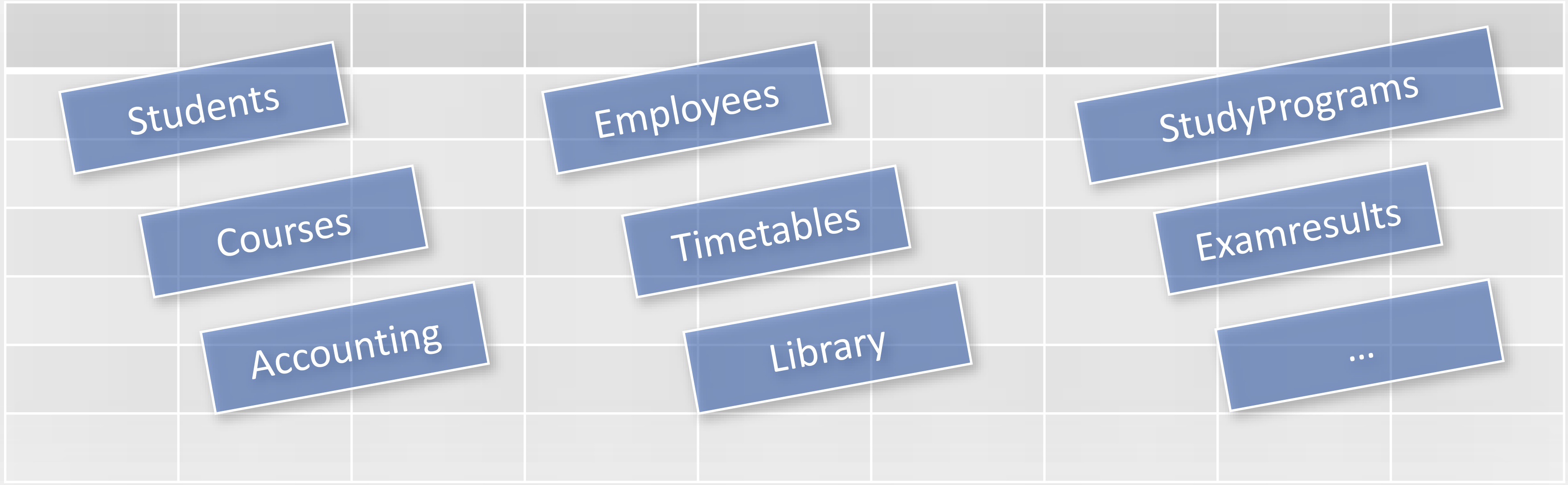
https://en.wikipedia.org/wiki/DIKW_pyramid

What is a database ?

Database (DB) = a **logically** organized **collection** of **data** (*information*)

Examples: A university keeps many databases

➢ **every** database can contain **many tables:**



Students

Courses

Accounting

Employees

Timetables

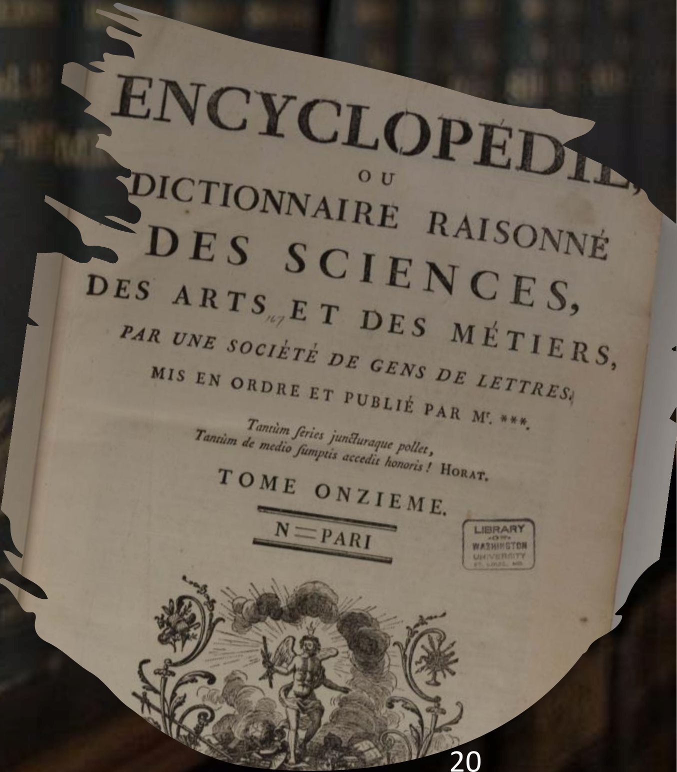
Library

StudyPrograms

Examresults

...

Why use databases ?



ENCYCLOPÉDIE,

OU

DICTIONNAIRE RAISONNÉ

DES SCIENCES,

DES ARTS ET DES MÉTIERS,

PAR UNE SOCIÉTÉ DE GENS DE LETTRES.

MIS EN ORDRE ET PUBLIÉ PAR Mr. ***.

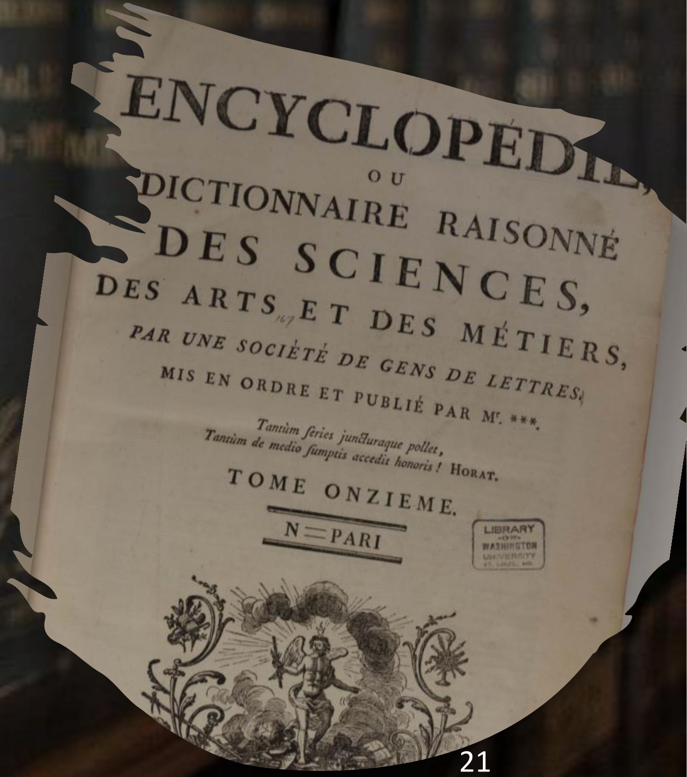
Tantùm ſeries juncturaque pollet,
Tantùm de medio ſumptis accedit honoris! HORAT.

TOME ONZIEME.

N — PARI

Why use databases ?

- Allows to collect & organize data from many sources in one system
  - **Create**, **update**, **delete** tables and entries
  - Streamline data **curation** and **administration**
- Store & access data efficiently (even big data: e.g. Wikipedia)
  - **Queries, indexing**, search, references …
- Serve the ***same*** *data to* ***many*** *users* ***simultaneously***
  - Allow users to **choose an appropriate view** of the data
  - **Security**: control access rights of users
  - Allow access over a **network** (internet or internal)
- Want to guarantee **consistency & integrity** of the data
  - Transaction management enables **reliable backups**



ENCYCLOPEDIE,

OU

DICTIONNAIRE RAISONNÉ

DES SCIENCES,

DES ARTS ET DES MÉTIERS,

PAR UNE SOCIÉTÉ DE GENS DE LETTRES.

MIS EN ORDRE ET PUBLIÉ PAR M. ***.

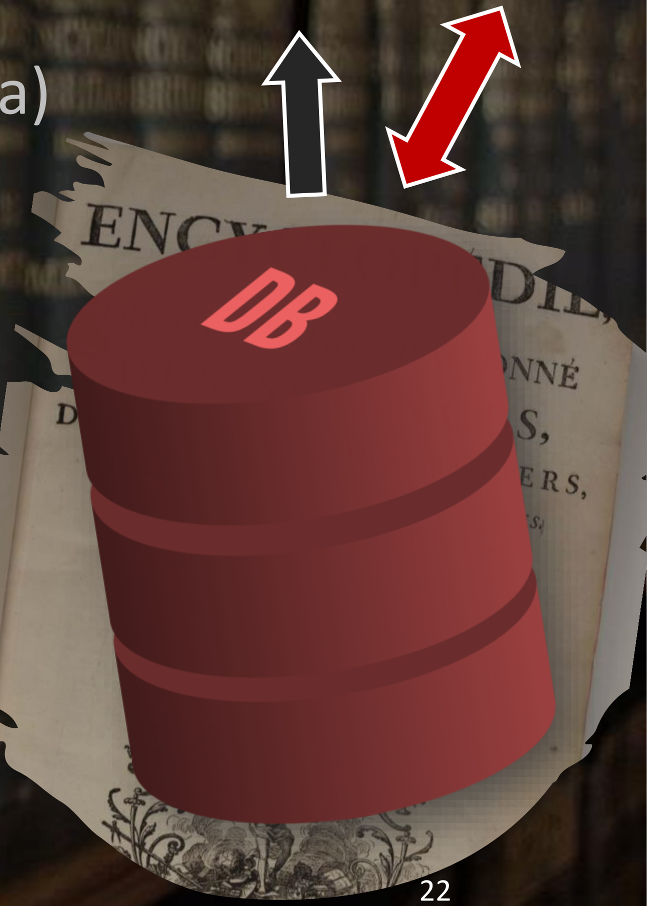
Tantùm series juncturaque pollet,
Tantùm de medio sumptis accedit honoris! HORAT.

TOME ONZIEME.

N — PARI

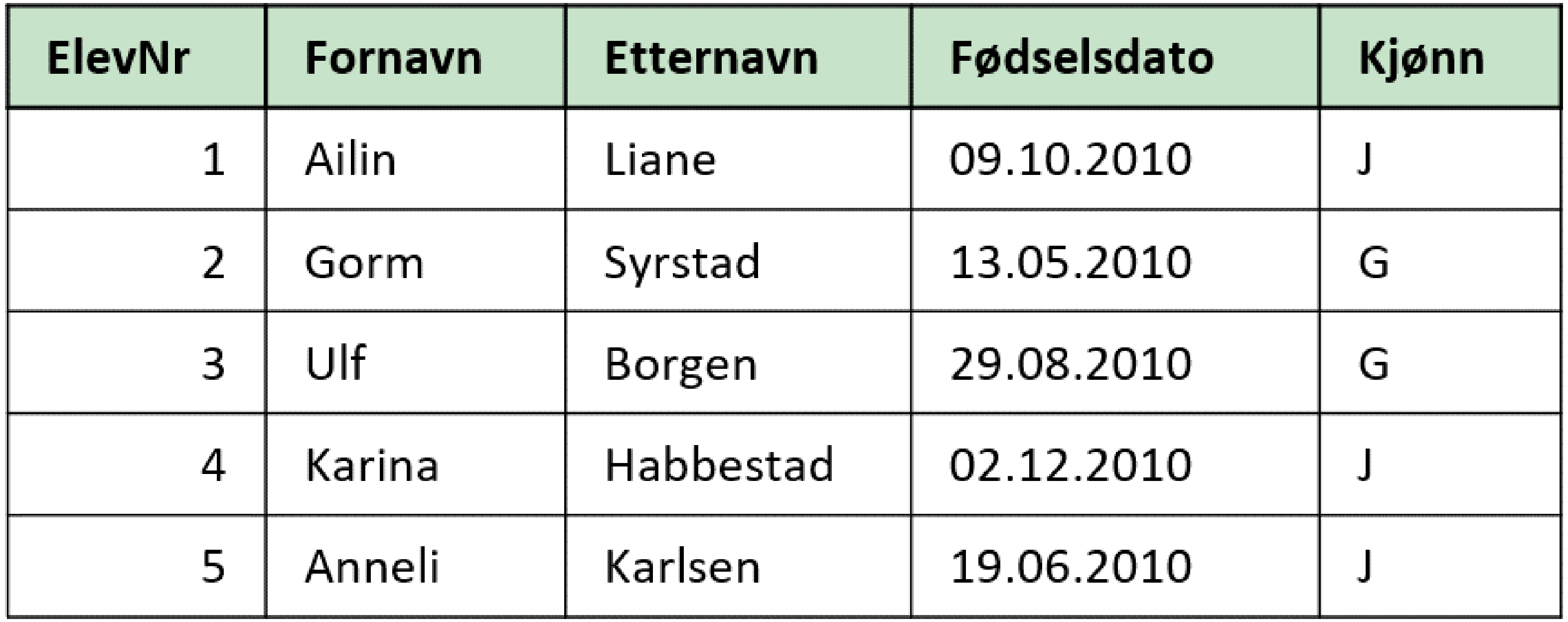
Why use databases ?

- Allows to collect & organize data from **many sources** in one system
  - **Create**, **update**, **delete** tables and entries
  - Streamline data **curation** and **administration**
- Store & access data **efficiently** (even big data: e.g. Wikipedia)
  - **Queries, indexing**, search, references …
- Serve the ***same*** *data to* ***many*** *users* ***simultaneously***
  - Allow users to **choose an appropriate view** of the data
  - **Security**: control access rights of users
  - Allow access over a **network** (internet or internal)
- Want to guarantee **consistency & integrity** of the data
  - Transaction management enables **reliable backups**



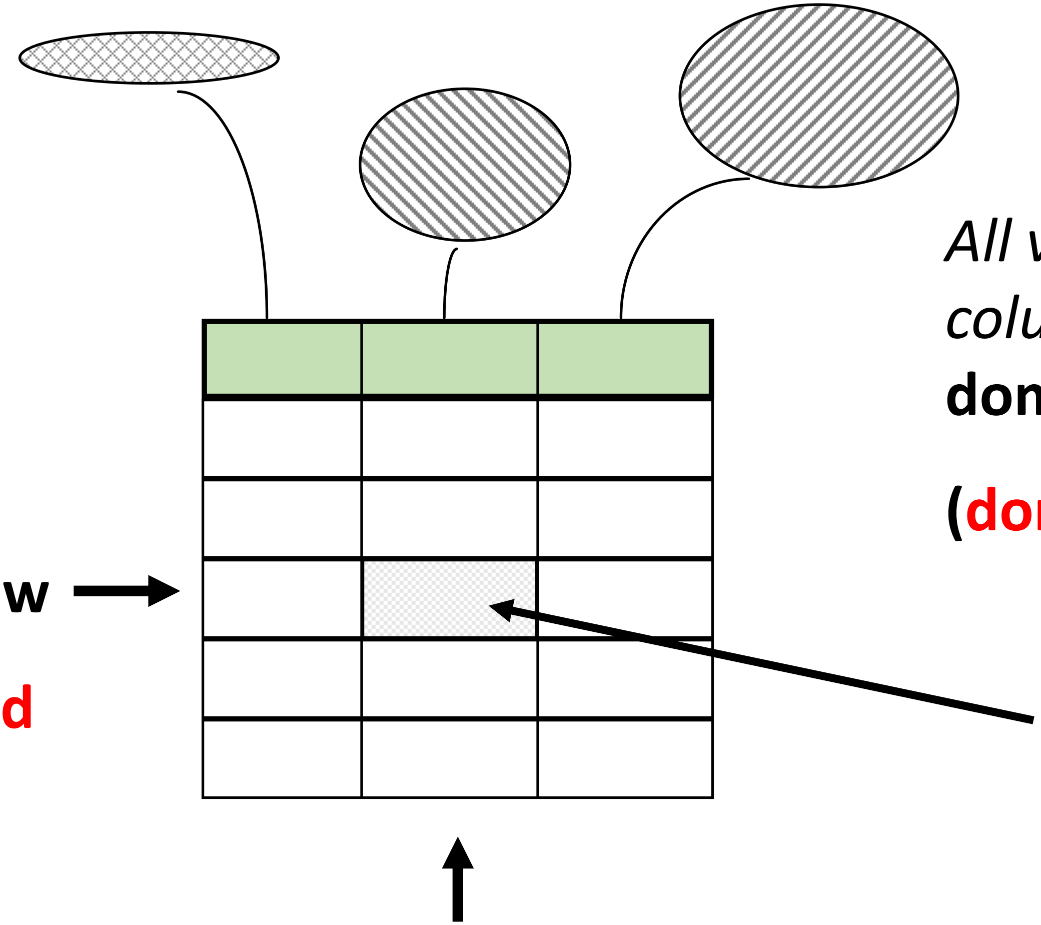
A database table is like a spreadsheet, but:

- All **columns** have a (logical) **name**
- All ***values in a given column*** are of the **same type**
- *And a few more restrictions* that we will describe later ...



ElevNr	Fornavn	Etternavn	Fødselsdato	Kjønn
1	Ailin	Liane	09.10.2010	J
2	Gorm	Syrstad	13.05.2010	G
3	Ulf	Borgen	29.08.2010	G
4	Karina	Habbestad	02.12.2010	J
5	Anneli	Karlsen	19.06.2010	J

Database tables



Row

Rad

Columns

Kolonne

Values

Verdi

All values in a given column are in one **domain / datatype**.

(domene / datatype)

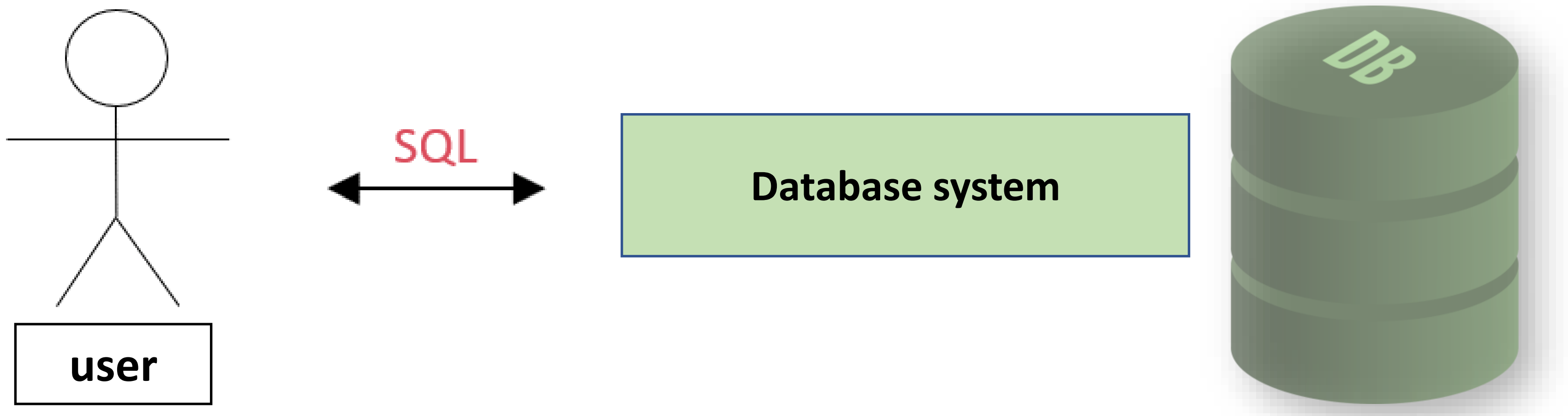
Structured Query Language: SQL queries

SELECT Fornavn, Etternavn

FROM Elev

WHERE Kjønn = 'J'

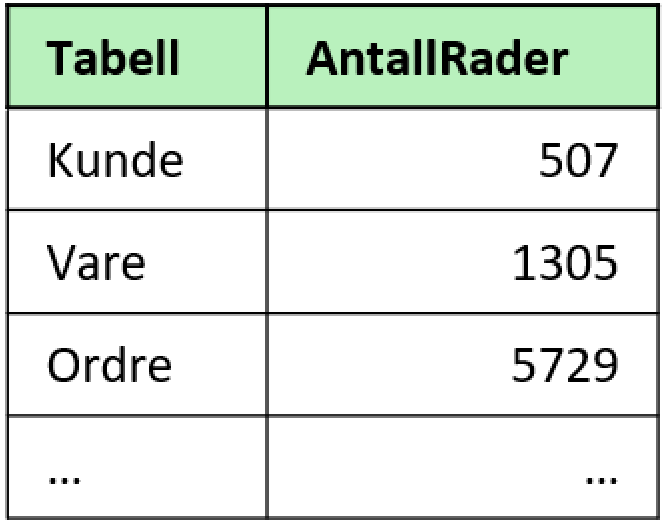
ORDER BY Etternavn



Metadata

- **Metadata** are «*data about data*».
- **Metadata** are also **stored in tables**.
- They can be obtained using SQL.

In MySQL metadata is in the *INFORMATION_SCHEMA* .
In other DB systems this is often called *System Catalog*.

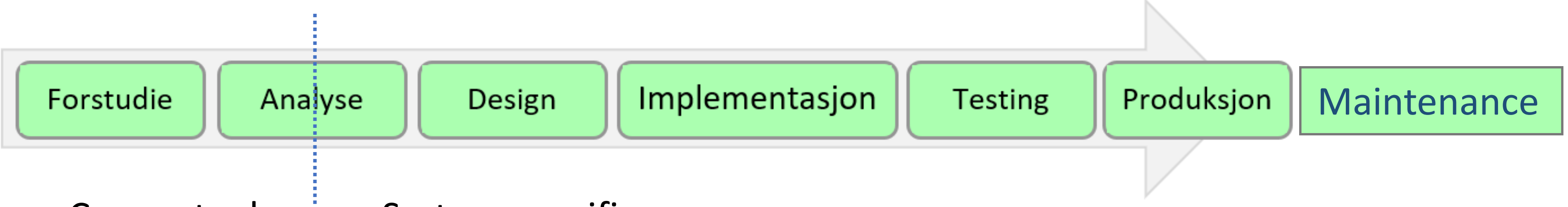


Tabell	AntallRader
Kunde	507
Vare	1305
Ordre	5729
…	…

Metadata
=
data om data

Life cycle of a database system

1. **Preliminary study** (Forstudie): Which **need** for the new system?
2. **Analysis**: **What** shall the system do?
3. **Design**: **How** should the system be **built**?
4. **Implementation**: Make (**programming**) the system!
5. **Testing**: Does it **work** as **expected**?
6. **Production**: **Use** the system!
7. **Maintenance**: Keep it running ***for years***!

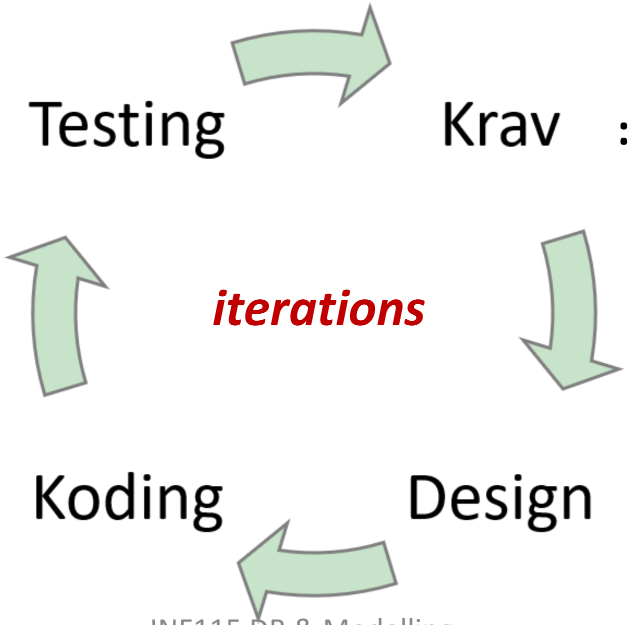


Conceptual Phase

System specific Phase

Agile (smidig) method

- **Not strict** on planning and documentation **at the beginning**
- **Iterative** process with many small deliverables
- **Testing** & **feedback** from users while developing
- Example: ***Scrum framework***



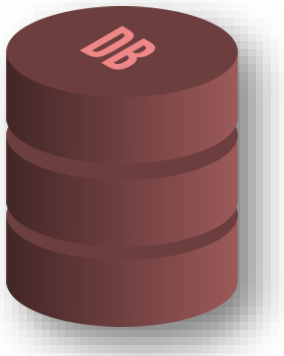
: **Requirement Analysis**

Now: 15 minute break

Lecture resumes at 15:10

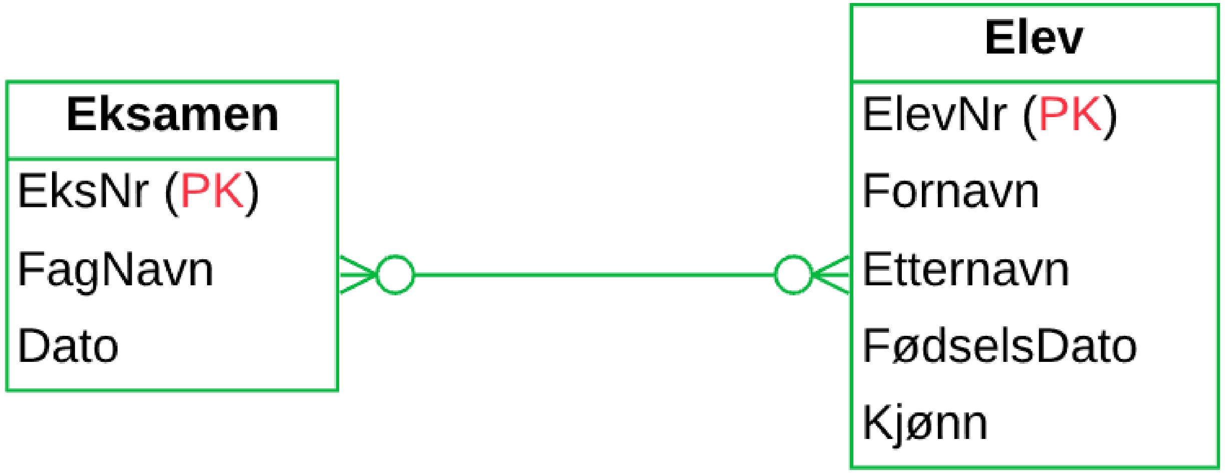
Summary: Chapter 1 – Introduction (Part 1)

- Databases are **behind many software applications**.
- A Database (DB) is a **logically organized collection of data** (*information*).
- DB systems (DBS) allow to **collect & organize data from many sources.**
- DBS enable us to **store & access data efficiently and reliably**.
- DB Tables and SQL
- **Life cycle of a DB** system development



Data models

- **(DB) System development** needs **good planning.**
- Making **models** helps **to understand the system.**
- A **data model** represents the **structure of the database.**



Eksamen

EksNr (PK)

FagNavn

Dato

Elev

ElevNr (PK)

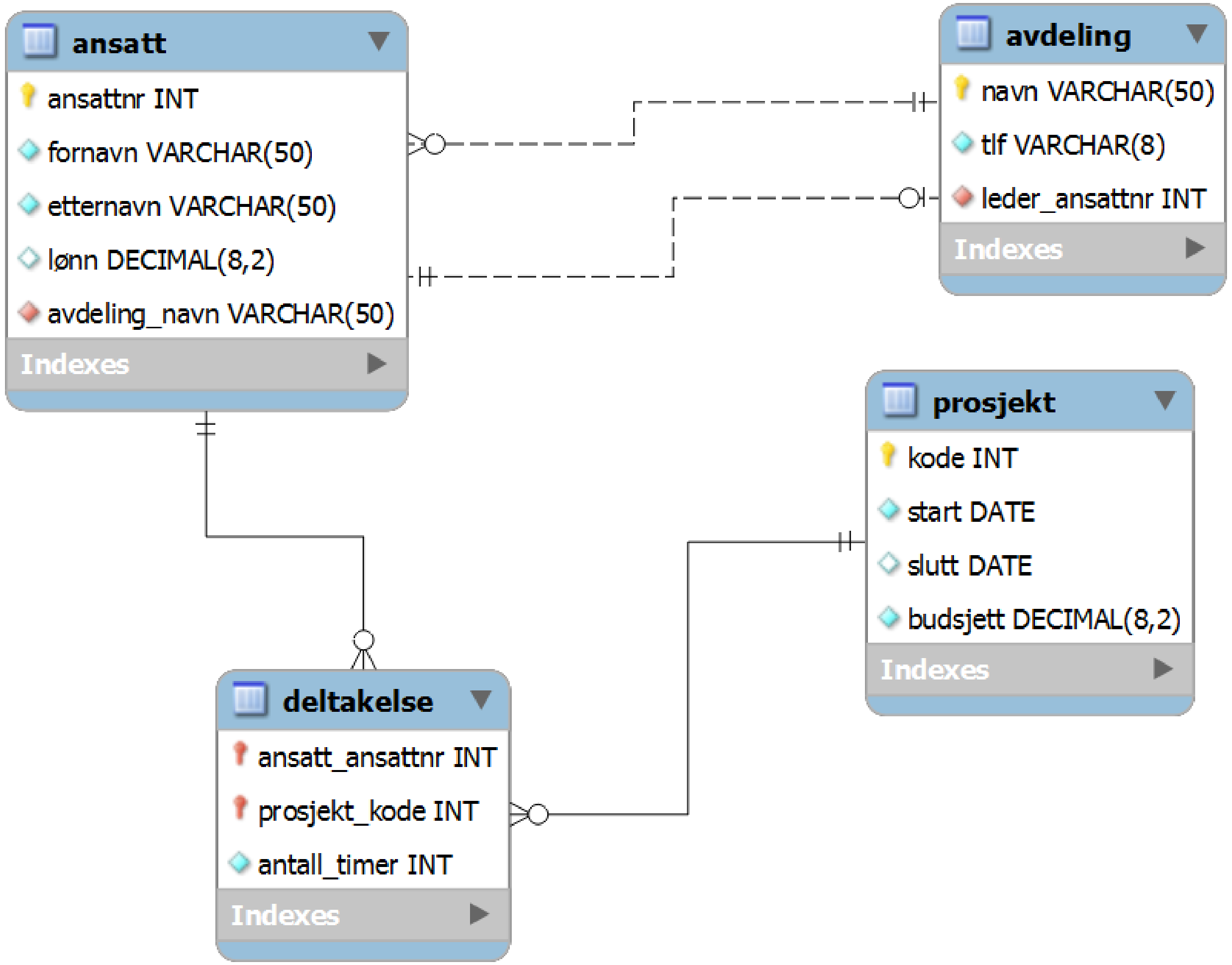
Fornavn

Etternavn

FødselsDato

Kjønn

Example from MySQL Workbench



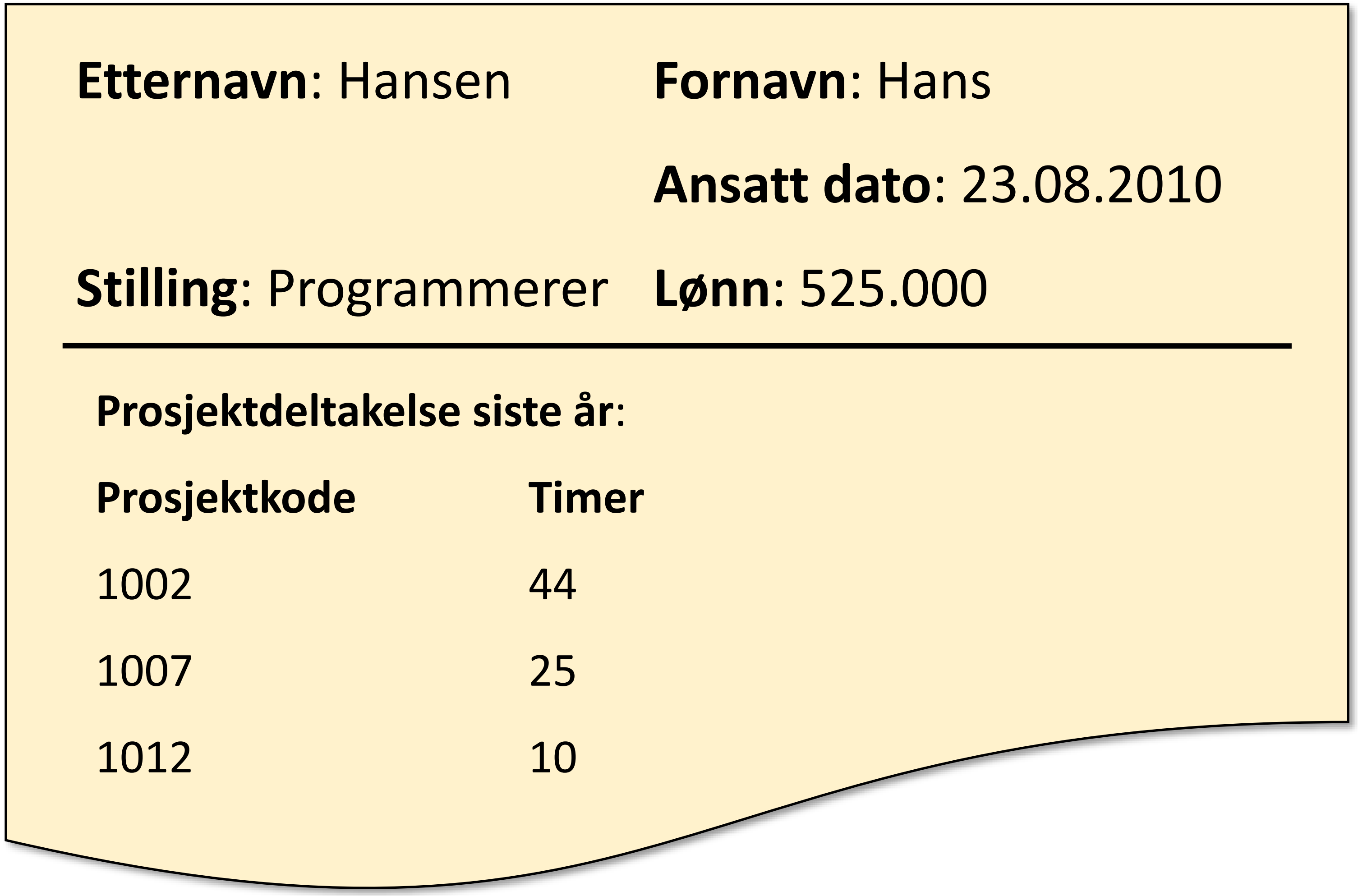
ansatt
- ansattnr INT
- fornavn VARCHAR(50)
- etternavn VARCHAR(50)
- lønn DECIMAL(8,2)
- avdeling_navn VARCHAR(50)
- Indexes

avdeling
- navn VARCHAR(50)
- tlf VARCHAR(8)
- leder_ansattnr INT
- Indexes

prosjekt
- kode INT
- start DATE
- slutt DATE
- budsjett DECIMAL(8,2)
- Indexes

deltakelse
- ansatt_ansattnr INT
- prosjekt_kode INT
- antall_timer INT
- Indexes

A printout (papirskjema)



Etternavn: Hansen **Fornavn**: Hans

Ansatt dato: 23.08.2010

Stilling: Programmerer **Lønn**: 525.000

Prosjektdeltakelse siste år:

Prosjektkode	**Timer**
1002	44
1007	25
1012	10

Quizz on Database Design

Please answer the practice quizz on mitt.uib now ☺

(you can take it again later if you want)

Link:

- ➢ https://mitt.uib.no/courses/33533/quizzes/

A Few Examples of Database Use Cases

- **Print out** of a table (on paper or on a screen),
- Calculate the **invoice** for an order (e.g. online shopping),
- **Search for results** of hockey matches (in a local league, or for the last week),
- **Find a list** of *Wikipedia* articles containing search terms,
- **Management of access rights** to the database,
- And many more …

More Examples: Databases and Tables

- **Cloud computing**: UserAccounts, Passwords (encrypted), UserData, …
- **Email**: Contacts, Inbox, Sent, Labels, Spam …
- **Health**: Corona test results, Patients, Medication, Hospitals, Doctors, Appointments, …
- **Payment systems**: eID, Cardnumber, PaymentOperations, …
- **Wikipedia**: Articles, Pictures, Media, Users, …
- **Libraries**: Books, Journals, Articles, Loans, …

- (Almost) All **businesses** use databases
- Many systems must be **operational 24/7** (*critical services*)
- Databases are part of larger **information processing systems**:
  - Examples: **Accounting** systems, Project **planning** systems, …

Employees Table (Ansatt)

AnsattNr	Etternavn	Fornavn	AnsattDato	Stilling	Lønn
1	Veum	Varg	01.01.1996	Løpegutt	383 000.00
2	Stein	Trude	10.10.2004	DBA	470 700.00
3	Dudal	Inger-Lise	24.12.2012	Sekretær	499 000.00
4	Hansen	Hans	23.08.2010	Programmerer	525 000.00
5	Bjørnsen	Henrik	01.01.2014	Tekstforfatter	575 000.00
6	Gredelin	Sofie	18.05.2012	Underdirektør	825 850.00
7	Zimmermann	Robert	17.05.1999	Regnskapsfører	575 000.00
8	Nilsen				
11	Fosheim				
13	Lovløs				
16	Ibsen				
17	Fleksnes				
20	Felgen				
23	Karius				
29	Wirkola				

1 rad = 1 ansatt

AnsattNr = 4

Etternavn = Hansen

Fornavn = Hans

AnsattDato = 23.08.2010

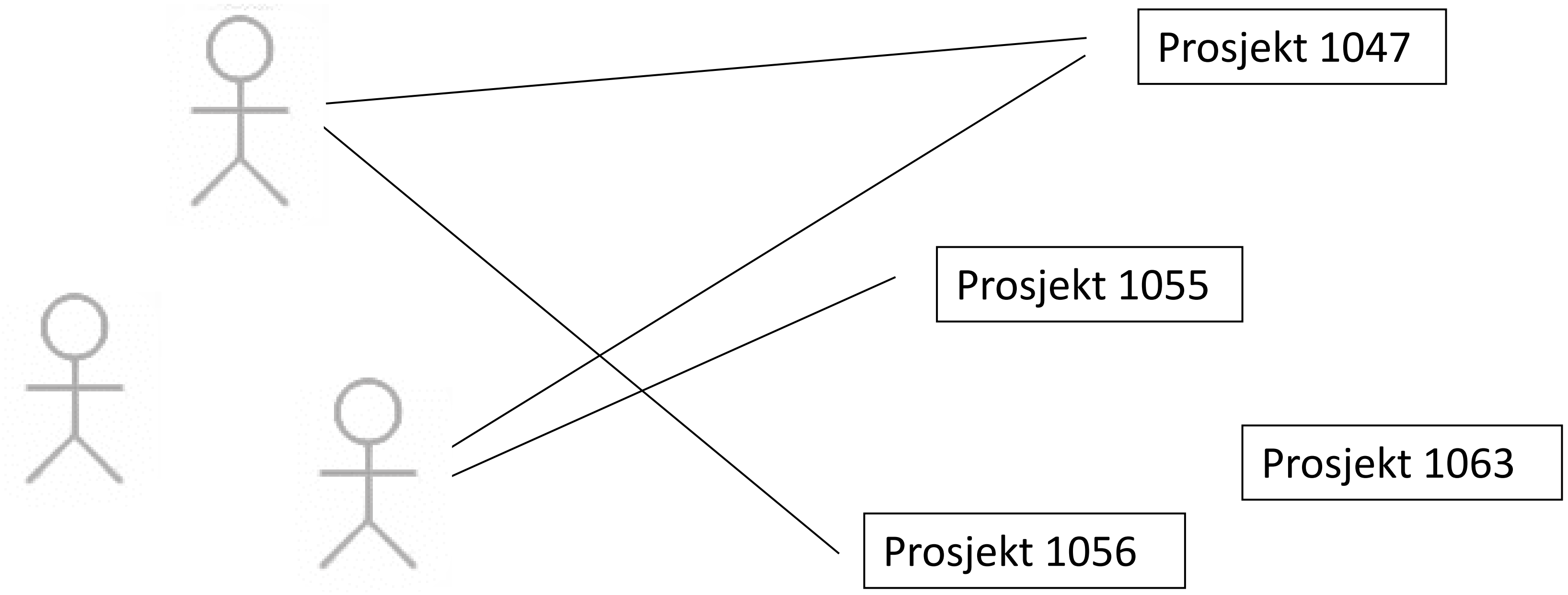
Stilling = Programmerer

Lønn = 525 000

Project Table

ProsjektNr	Budsjett	Leder	Start	Slutt
1001	kr 15 000.00	20	12.01.2019	12.03.2019
1002	kr 750 000.00	8	23.06.2019	23.07.2019
1007	kr 125 000.00	2	12.06.2020	
1009	kr 500 000.00	20	01.01.2020	
1012	kr 10 000.00	4	10.07.2020	
1020	kr 900 000.00	8	23.07.2019	01.09.2019

Employee and Projects

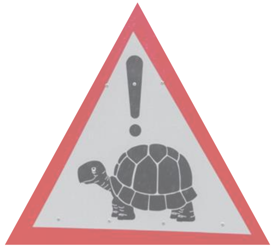


- One employee can participate in **0**, **1** or **many** projects.
- A project can have **0**, **1** or **many** participants.
- Need another table to describe *who* works *where* !

ProjectParticipation Table

Which employees have worked on *which project* – and *how many hours* ?

- ➢ The table represents a **relationship.**



(norsk: forhold)

- ➢ *ProjectParticipation* is a **coupling table.**

(*ProsjektDeltakelse* er en **koblingstabell**.)

ProsjektNr	AnsattNr	AntTimer
1001	1	12
1002	4	44
1002	8	20
1002	13	125
1002	20	2
1007	4	25
1007	11	20
1009	2	5
1009	17	10
1009	20	23
1012	4	10
1020	1	20
1020	8	35
1020	17	125

Missing values

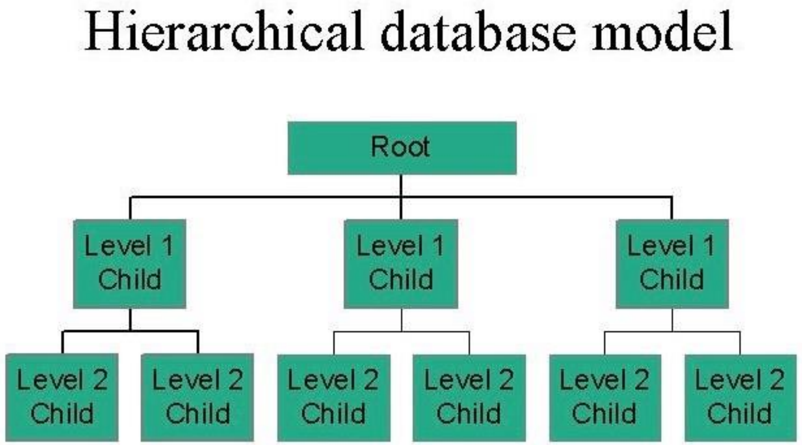
- **NULL** can be used to indicate that some values are **missing** in a table (e.g. the *Project* table).
- They occur **because**:
  - We forgot to register data.
  - We do not know the correct value.
  - It is not meaningful to register a value in this field
- **NULL** is **not** a value.
- *NULL can cause problems* ! So be careful !

Relational database = « Table database »

- A **table** can be seen as a mathematical **relation** (**relasjon**).
- A **relational database** (**relasjonsdatabase**)

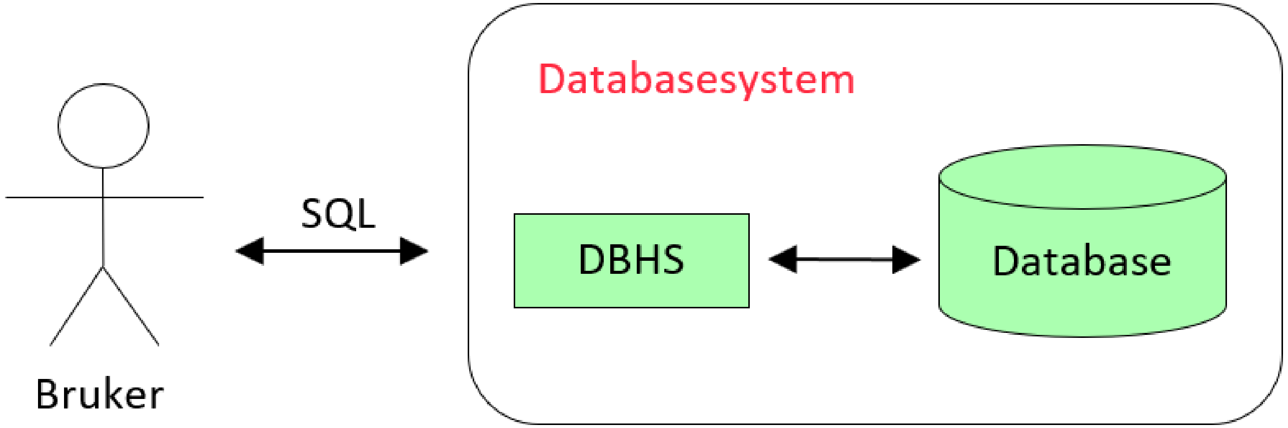
  is a collection of tables (relations).

- Other types of databases:
  - Hierarchical databases
  - Network databases
  - Object oriented databases
  - **Object-relational databases**
  - Logical databases
  - **NoSQL databases**



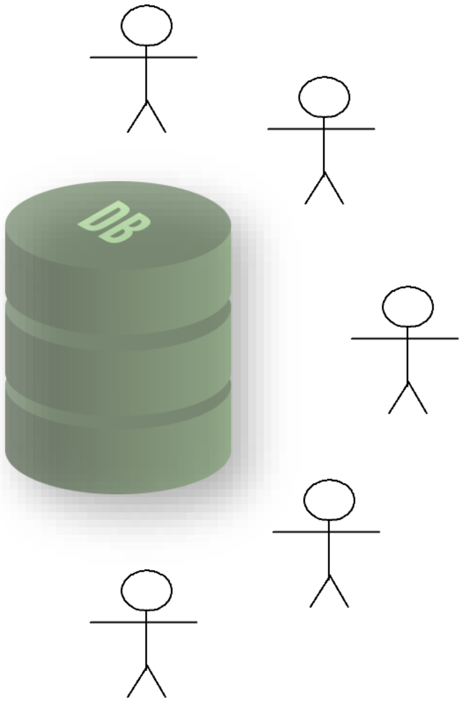
Database system = *DBMS* + database

- DBMS = **D**ata **B**ase **M**anagement **S**ystem
- DBHS = **D**ata**B**ase**H**åndterings**S**ystem
- Examples: **MySQL**, **MariaDB**, Access, Oracle, PostgreSQL, SQL Server, …
- **SQL** = language to interact with a database via a DBMS



Tasks for a database system

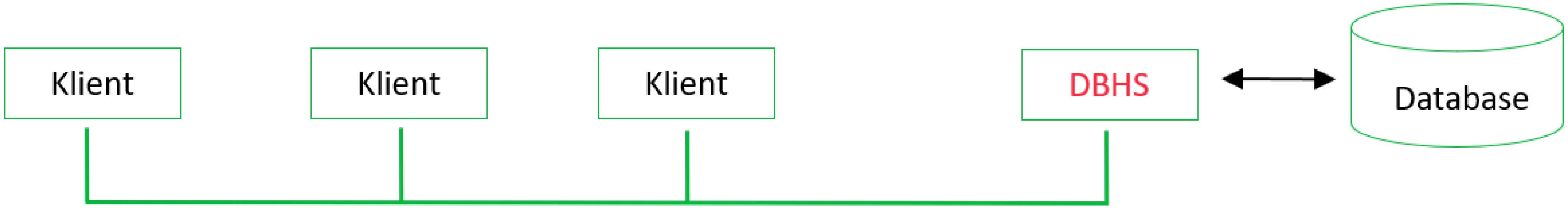
- **Store large** amounts of **structured data** over **long time** in a **secure** manner.
- **Find** data **quickly** and **accurately.**
- Serve **many** users **simultaneously:**
  - Define user access rights (read or write).
  - Allow access over a network.
- **Robustly** manage **failures** such as crashed disks and electricity cuts.
- **Communicate** with other programs.



Database clients in a network

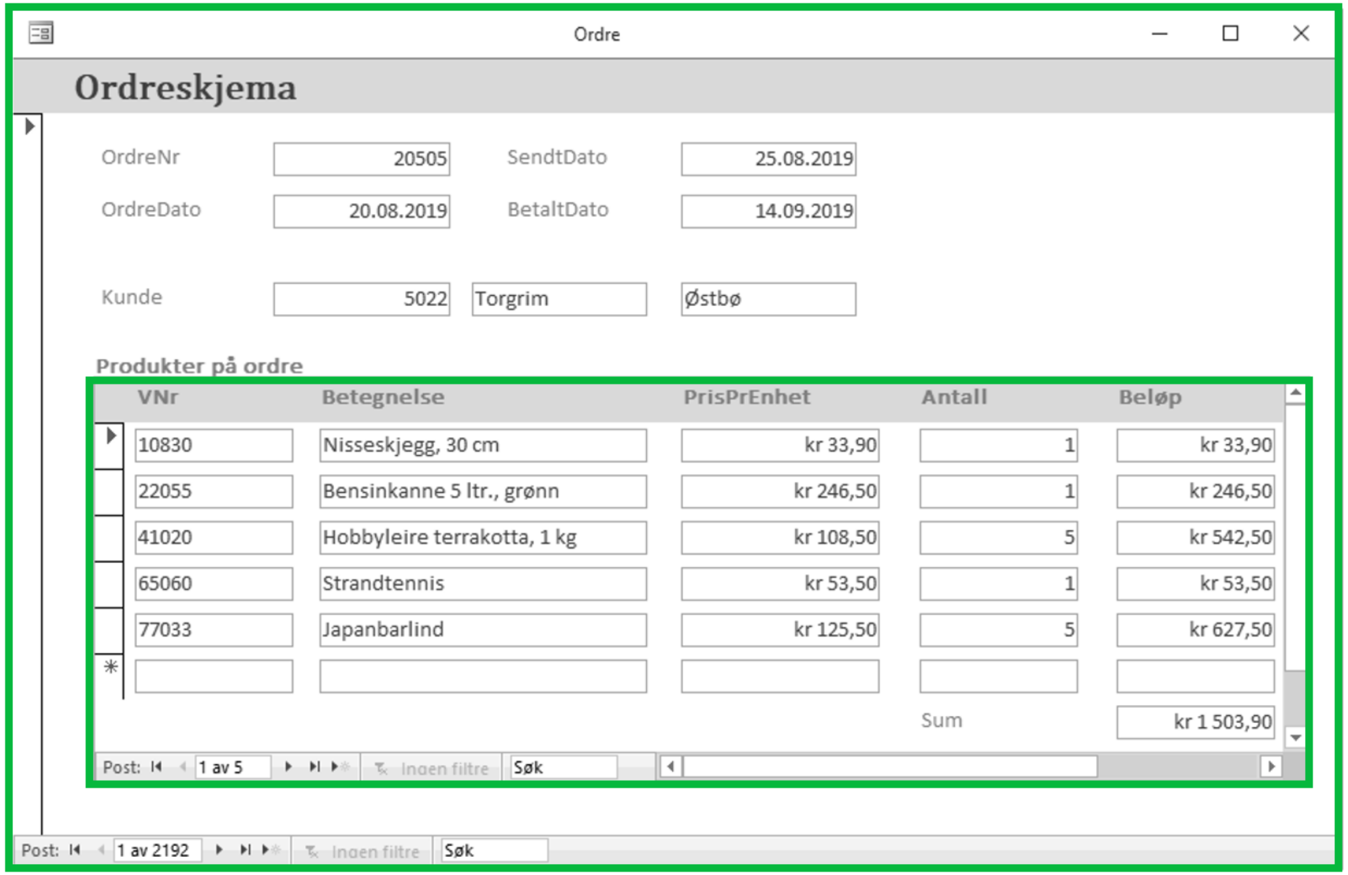
- A database is a **shared** resource.

- Client/server architecture: **many clients** are connected to **one server** via the network.



Database Applications

- Many interact with databases via an application.



Ordreskjema

OrdreNr	20505	SendtDato	25.08.2019
OrdreDato	20.08.2019	BetaltDato	14.09.2019

Kunde	5022	Torgrim	Østbø

Produkter på ordre

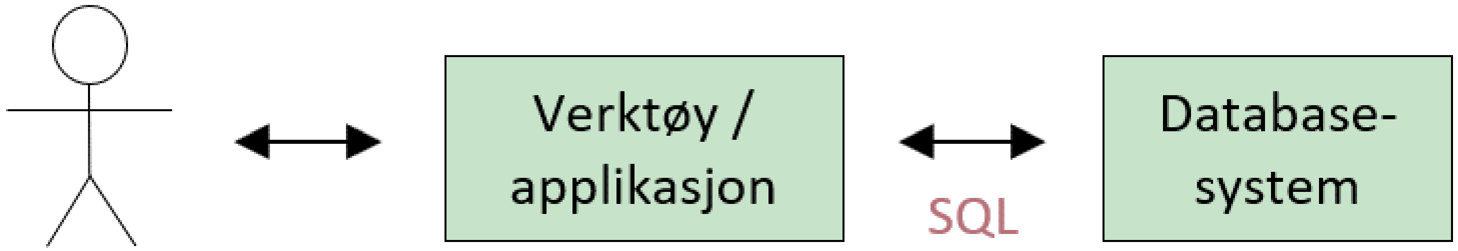
VNr	Betegnelse	PrisPrEnhet	Antall	Beløp
10830	Nisseskjegg, 30 cm	kr 33,90	1	kr 33,90
22055	Bensinkanne 5 ltr., grønn	kr 246,50	1	kr 246,50
41020	Hobbyleire terrakotta, 1 kg	kr 108,50	5	kr 542,50
65060	Strandtennis	kr 53,50	1	kr 53,50
77033	Japanbarlind	kr 125,50	5	kr 627,50
			Sum	kr 1 503,90

Database Users and Tools / Applications

- **Applications & tools** form an additional **layer** **between** the **user** and the **database.**
- Applications & tools ***communicate*** with the DB **using SQL**.

➢ The **enduser** does not need to know SQL.

➢ DBA: database **administrator** can be different from the **developer**.



5 minutes breakout rooms:

How would you design an application to manage read and write access rights of the users of a database?

5 minutes breakout rooms:

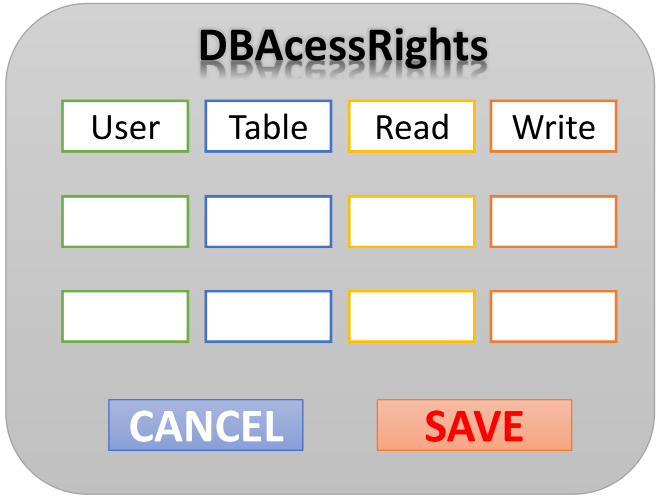
How would you design an application to manage read and write access rights of the users of a database?

- A table **Users** with columns:

  UserNumber, UserName, TableName, ReadAccess, WriteAccess.

- A GUI where one can add users and set values.

User Number	User Name	Table Name	Read Access	Write Access



Representation of Data in Computers

1	1	0	0	1	1	0	1	0
2	0	0	1	1	1	0	1	1
3	1	1	1	0	1	1	1	0
4	0	0	0	0	0	1	1	0
5	1	0	0	0	0	0	1	1

… … … … … … … … … . . .

- **All values in a table** are represented as **numbers** which can be **represented using 0s and 1s**.

- A **row** of a database table can be stored as a ***sequence of values.***

- A **database table** can be stored as a ***sequence of rows.***

AnsNr	Fornavn	Etternavn
11	Varg	Veum
12	Ada	Lovløs
13	Jon	Nymann

Quizz on Data Representation

Please answer the practice quizz on mitt.uib now ☺

(you can take it again later if you want)

Link:

- ➢ https://mitt.uib.no/courses/33533/quizzes

Summary: Chapter 1 – Introduction (Part 1)

- Databases are **behind** **many software applications**.
- A Database (DB) is a **logically organized** **collection of data** (*information*).
- DB systems (DBS) allow to **collect & organize data from** **many sources.**
- DBS enable us to **store & access data** **efficiently and reliably**.
- DBS are part of **information processing systems**.
- A DB **may contain** **incomplete data**, **some values may be missing**.
- A DB **should** **not** **contain errors or contradictions**.

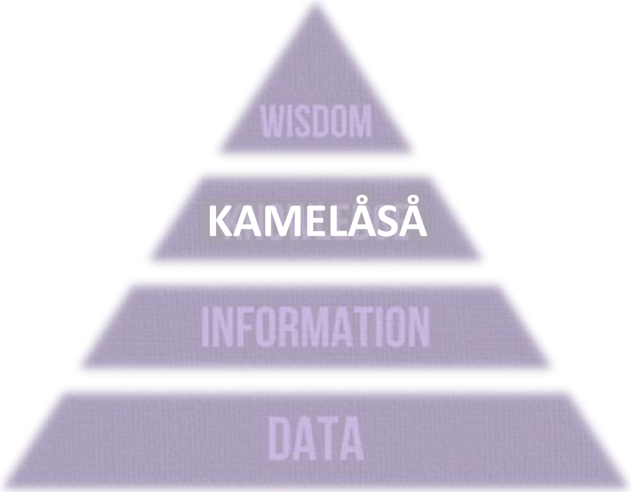
Summary: Chapter 1 – Introduction (Part 2)

- Databases are **behind many software applications and tools**.
- DBMS = **D**ata **B**ase **M**anagement **S**ystem
- **SQL** = language to interact with a database via a DBMS

- A **data model** represents the **structure of the database.**
- A DB may contain **incomplete data**, because some values may be missing.

- **Data representation**:
  - **Memory** (RAM and disk) store **sequences of numbers**.
  - A **computer stores values as numbers** encoded as 0s and 1s.
  - A **table** is stored as a **sequence of values**.

Questions ?



- ***Next*** **lecture on Friday 21.01.2022 (10:15 – 12:00)**

INF115 Lecture 1: *Representation of Data*

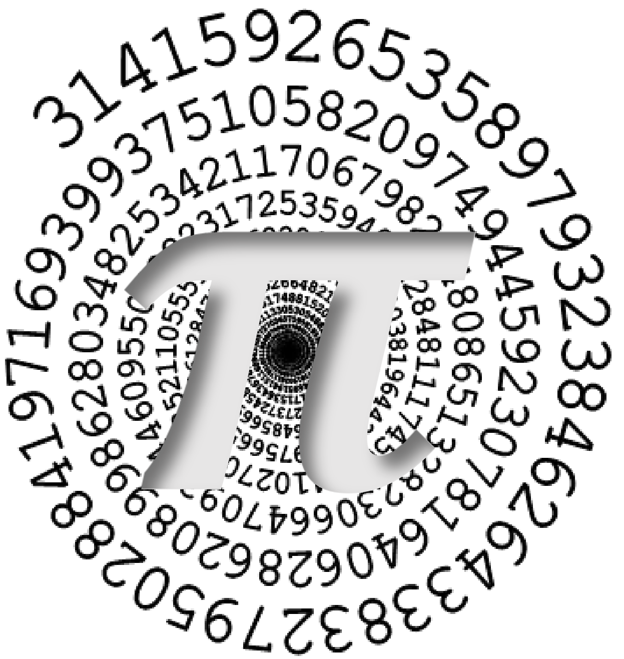
Adriaan Ludl

Department of Informatics
University of Bergen

Spring Semester

Representation of Data in Computers

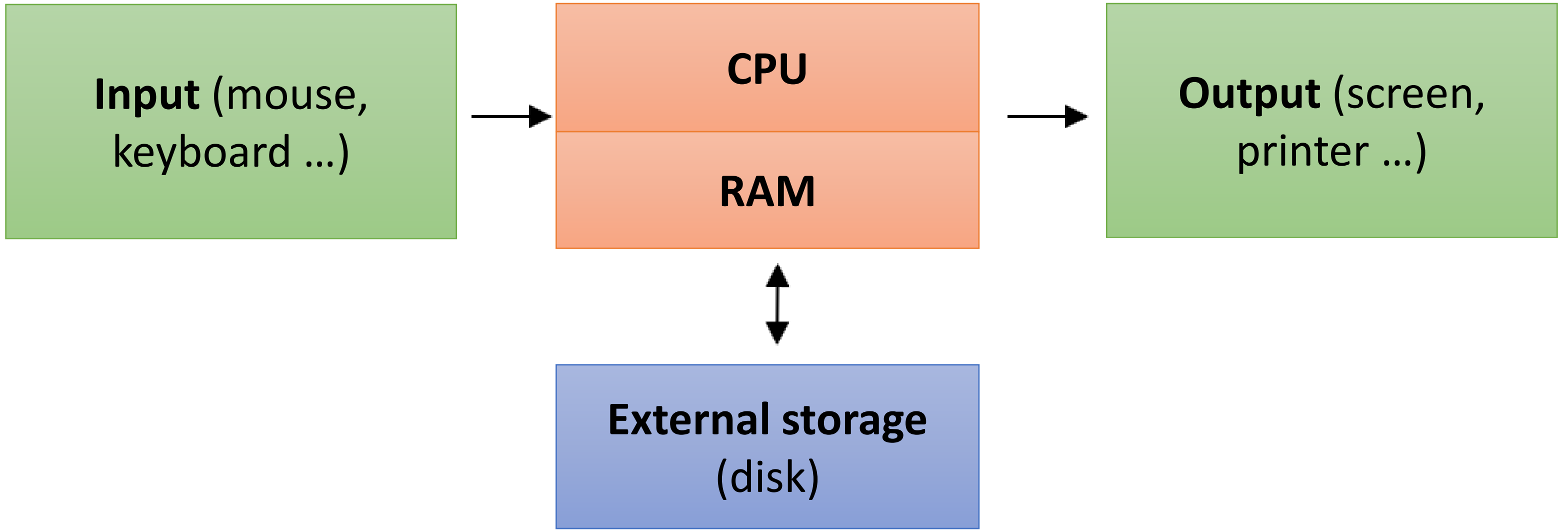
1. How do **computers** work ?
2. **Represenation** of **numbers**
3. **Representation** of **text**
4. **Representation** of **tables**



How do computers work ?

The physical **details** of a **computer** are **complex** (gates, transistors…). Let’s consider **a simplified but useful model**.

➢A computer can **store data** and **execute programs**.



Input (mouse, keyboard …)

CPU

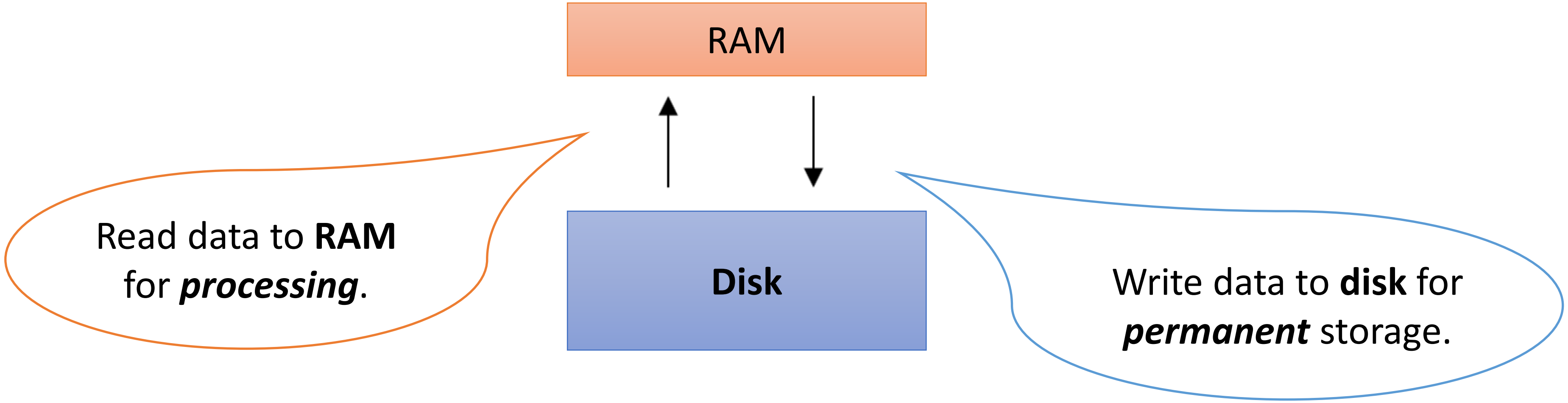
RAM

Output (screen, printer …)

External storage (disk)

Disk and RAM

- **Random access memory** (RAM, hurtigminnet) is **much faster** than the disk.
- The **data in the RAM is lost *when*** the machine is ***turned off*** !



Read data to **RAM** for ***processing***.

Write data to **disk** for ***permanent*** storage.

Bits and Bytes

A simplified model of *storage media:*

Both **disk** and **RAM** contain **numbered sequences of bytes**.

- 1 **byte** = 8 **bits** in either 0 or 1 state.
- *Every bit can store 2 different values.*
- 2 bits can store 2^2 = 4 values.

  …

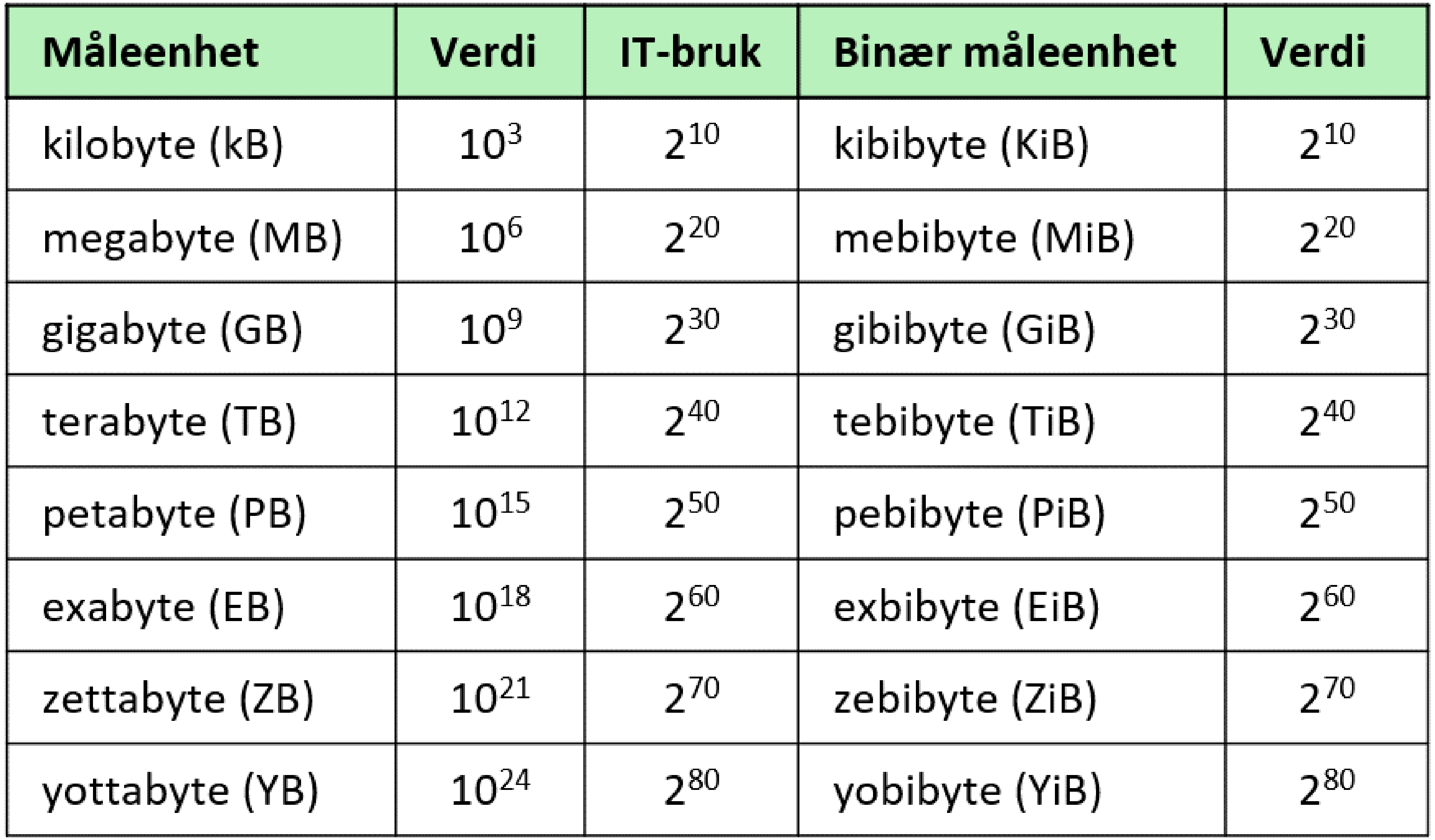
- **8 bits** can store 2^8 = **256 values**.

1	1	0	0	1	1	0	1	0
2	0	0	1	1	1	0	1	1
3	1	1	1	0	1	1	1	0
4	0	0	0	0	0	1	1	0
5	1	0	0	0	0	0	1	1

… … … … … … … … … . . .

Units to measure data size

SI Unit: **Value:** **Binary Unit:** **Value:**



Måleenhet	Verdi	IT-bruk	Binær måleenhet	Verdi
kilobyte (kB)	10^3	2^{10}	kibibyte (KiB)	2^{10}
megabyte (MB)	10^6	2^{20}	mebibyte (MiB)	2^{20}
gigabyte (GB)	10^9	2^{30}	gibibyte (GiB)	2^{30}
terabyte (TB)	10^{12}	2^{40}	tebibyte (TiB)	2^{40}
petabyte (PB)	10^{15}	2^{50}	pebibyte (PiB)	2^{50}
exabyte (EB)	10^{18}	2^{60}	exbibyte (EiB)	2^{60}
zettabyte (ZB)	10^{21}	2^{70}	zebibyte (ZiB)	2^{70}
yottabyte (YB)	10^{24}	2^{80}	yobibyte (YiB)	2^{80}

Representing Numbers

1	1	0	0	1	1	0	1	0
2	0	0	1	1	1	0	1	1
3	1	1	1	0	1	1	1	0
4	0	0	0	0	0	1	1	0
5	1	0	0	0	0	0	1	1
…	…	…	…	…	…	…	…	… . . .

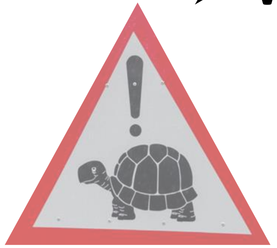
Integer numbers (or integers):

➢1 byte = **256 *different* patterns of bits**

- 1 byte can be interpreted as an **integer** in [0..255]
- 2 bytes can be interpreted as a **integer** in [0.. 65 535]

➢When using the first bit as the **sign bit we can represent:**

integer numbers in [-32 768..+32 767]



Representing Numbers

1	1	0	0	1	1	0	1	0
2	0	0	1	1	1	0	1	1
3	1	1	1	0	1	1	1	0
4	0	0	0	0	0	1	1	0
5	1	0	0	0	0	0	1	1
…	…	…	…	…	…	…	…	… . . .

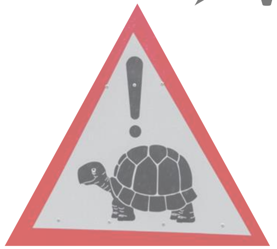
Integer numbers (or integers):

➢1 byte = **256 *different* patterns of bits**

- 1 byte can be interpreted as an **integer** in [0..255]
- 2 bytes can be interpreted as a **integer** in [0.. 65 535]

➢When using the first bit as the **sign bit we can represent:**

integer numbers in [-32 768..+32 767]



Every **decimal number** (or float) can be **represented as two integers**:

- **For example**: the number 486.229 can be written as 0.486229×10$^{3.}$
- Which can be represented as the integers 486229 and 3.

➢The **same method** can be used **for all decimal numbers**.

Representing Text

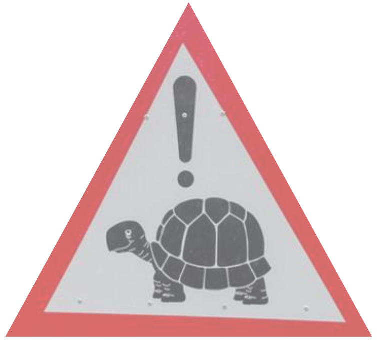
A **character encoding** assigns one number to every symbol.

➢***Letters, digits, special symbols*** can thus be **represented as numbers** !

With 2 bytes we can represent 65 536 symbols.

➢A **string of characters** is a **sequence of symbols**.

ASCII and **Unicode** are two examples of **character encodings.**



Tegn	Kode	Tegn	Kode
A	65	æ	145
Z	90	Æ	146
a	97	!	33
z	122	=	61
0	60	?	63
9	71	@	64
…	…	…	…

From 0 and 1 to database tables

Every cell in a database table contains a **value**:

• **Integer** numbers:	1, 2, 3 …
• **Decimal** numbers (floats):	such as 0.0, 0.101 …
• **Text** (string of characters):	«Gjelder hele Svalbard» …
• **Boolean** truth values:	**True** or **False**
• **Date** / **Time** (as numbers):	26.01.2021, 10:15:00.000

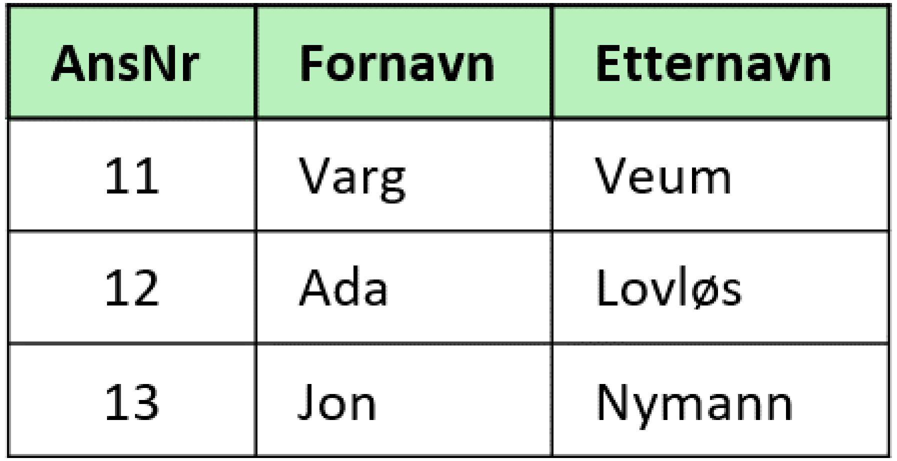
From 0 and 1 to database tables

Every cell in a database table contains a **value**:

• **Integer** numbers:	1, 2, 3 …
• **Decimal** numbers (floats):	such as 0.0, 0.101 …
• **Text** (string of characters):	«Gjelder hele Svalbard» …
• **Boolean** truth values:	**True** or **False**
• **Date** / **Time** (as numbers):	26.01.2021, 10:15:00.000

- ➢**All values are represented as numbers** which can be represented using 0s and 1s (as we just saw).
- ➢A **row** of a database table can be stored as a ***sequence of values.***
- ➢A **database table** can be stored as a ***sequence of rows*** …

A database table:



AnsNr	Fornavn	Etternavn
11	Varg	Veum
12	Ada	Lovløs
13	Jon	Nymann

A possible physical organization in memory:

Memory address	value
1024	11
1025	V
1026	A
1027	r
1028	g
…	…
1045	V
1046	e
1047	u
1048	m
…	…
1085	12
…	…

Rad 1 starter på adresse 1024.

AnsNr	Fornavn	Etternavn
11	Varg	Veum
12	Ada	Lovløs
13	Jon	Nymann

Rad 2 starter på adresse 1085.

Quizz on Data Representation

Please answer the practice quizz on mitt.uib now ☺

(you can take it again later if you want)

Link:

➢ https://mitt.uib.no/courses/33533/quizzes