INF115 Lecture 1:
Introduction to Databases

Adriaan Ludl

Department of Informatics

University of Bergen

Spring Semester 2021

26.01.2021

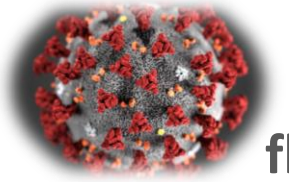
Chapter 1: Learning goals

- **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**

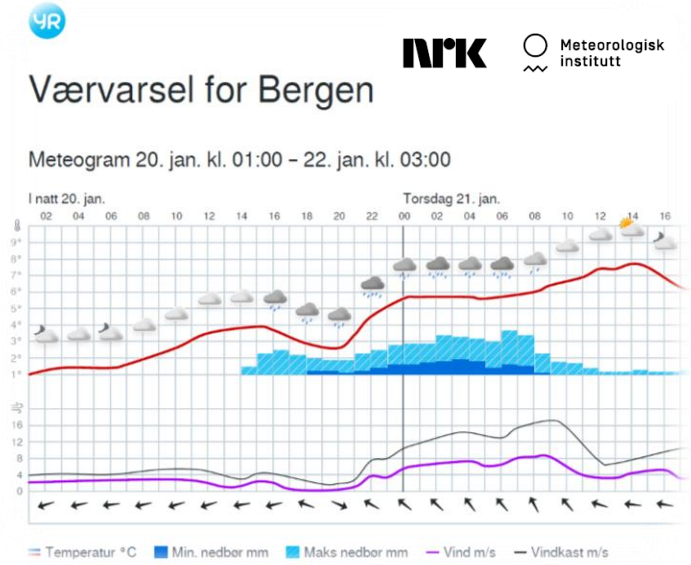
Databases are **behind** many software applications

- **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



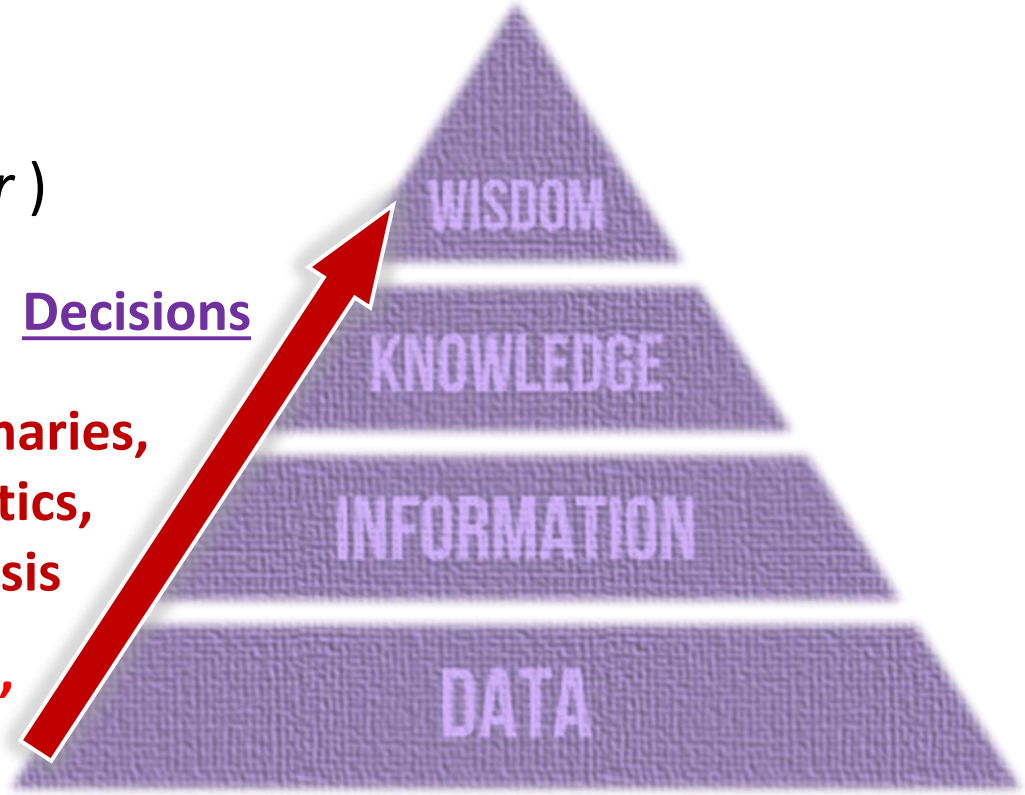
fhi.no



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

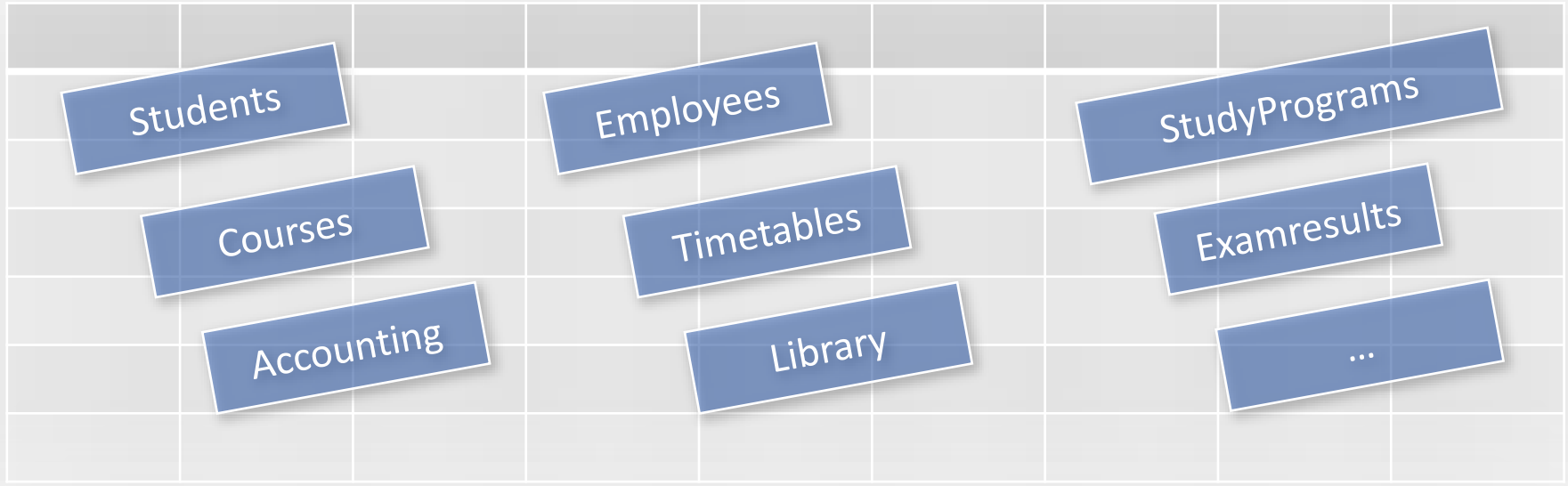
https://en.wikipedia.org/wiki/DIKW_pyramid

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:**

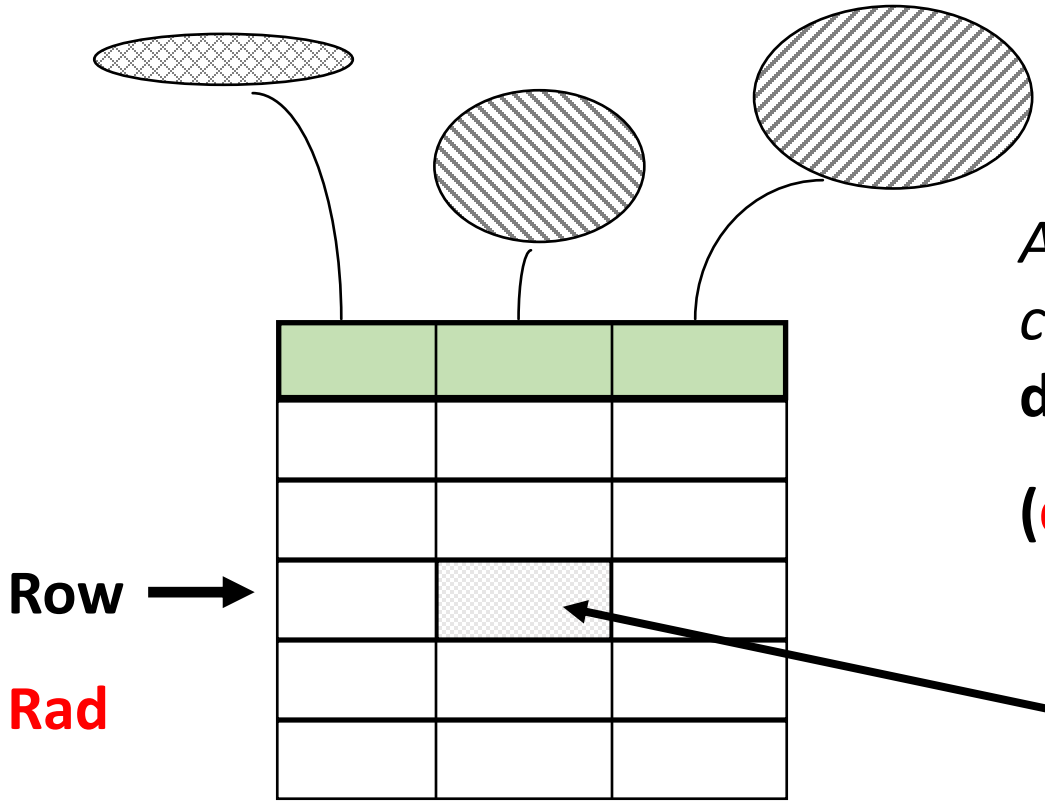


A database table is like a spreadsheet, but:

- 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



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

(domene / datatype)

Values

Verdi

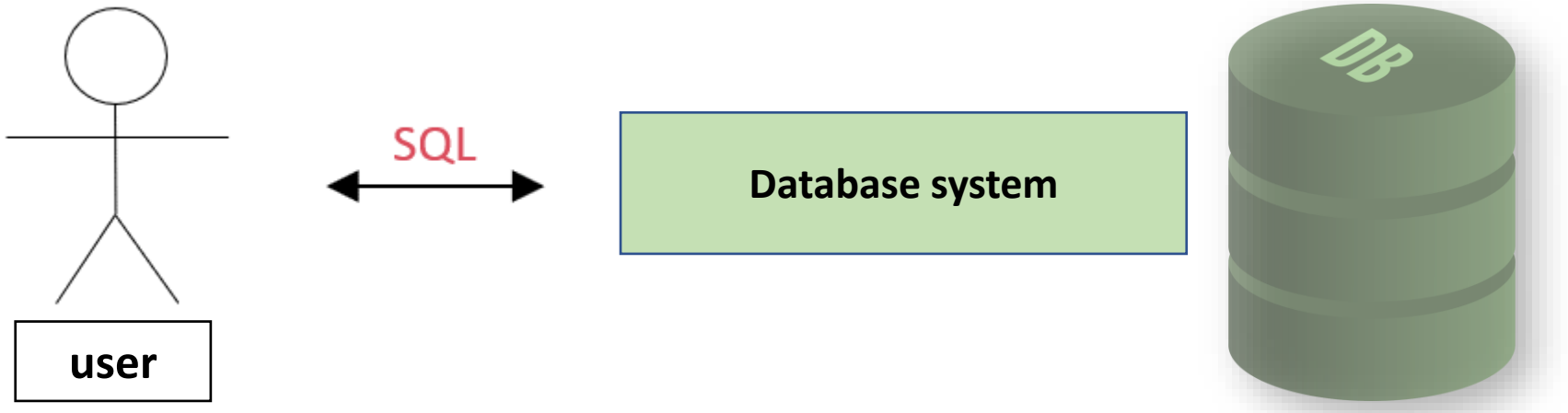
Structured Query Language: SQL queries

SELECT Fornavn, Etternavn

FROM Elev

WHERE Kjønn = 'J'

ORDER BY Etternavn



Metadata

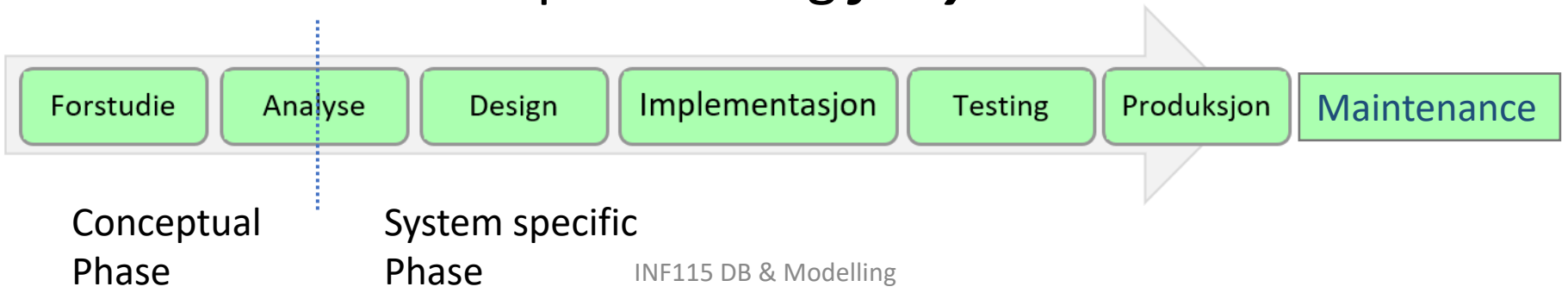
- **Metadata** are «*data about data*».
- **Metadata** are also **stored in tables**.
- They can be obtained using SQL.

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
...	...

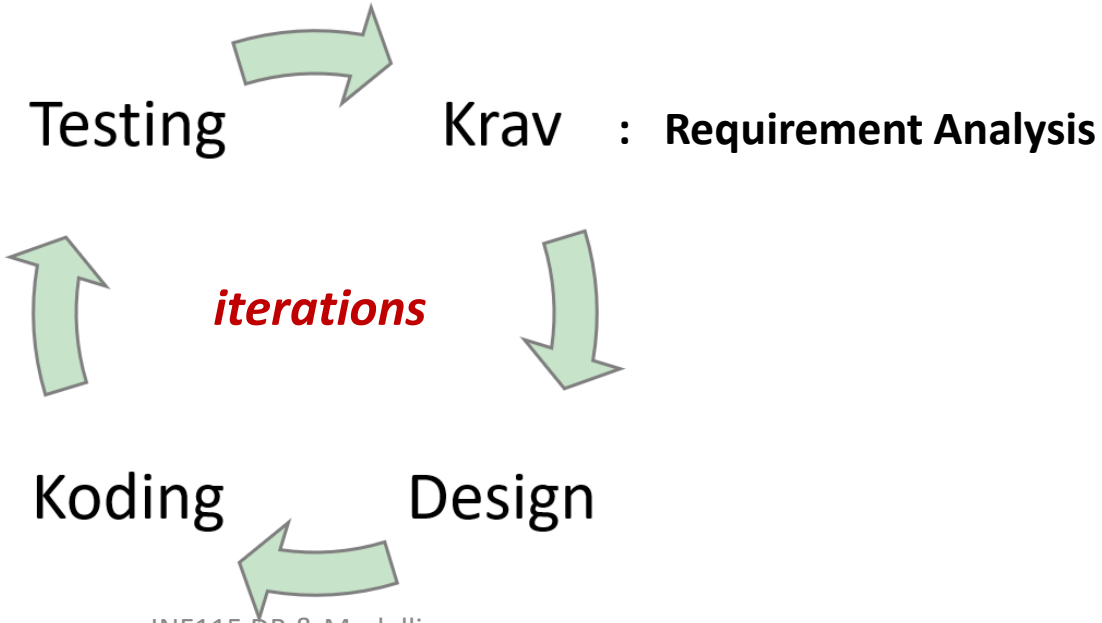
Life cycle of a database system

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***!



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***

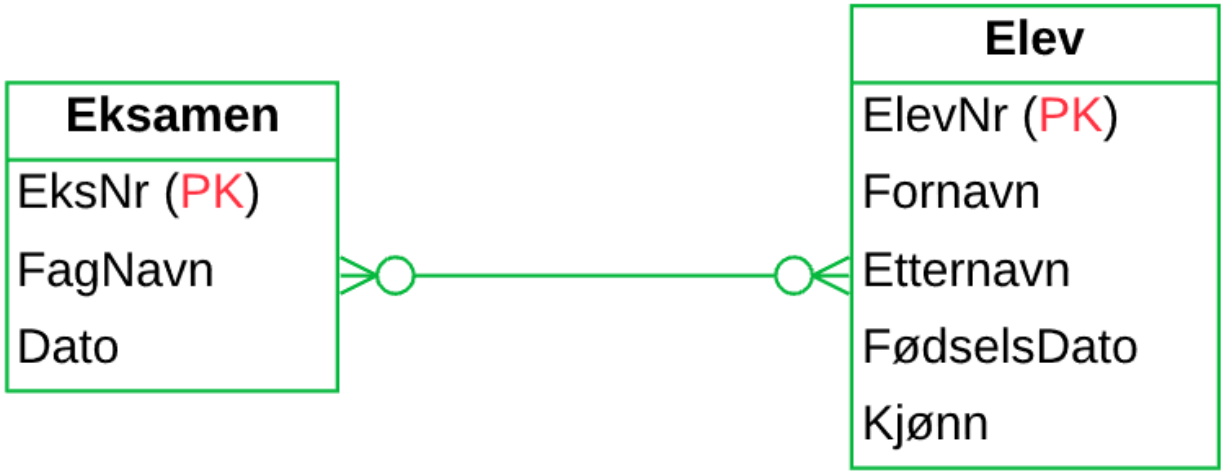


5 minutes breakout rooms:

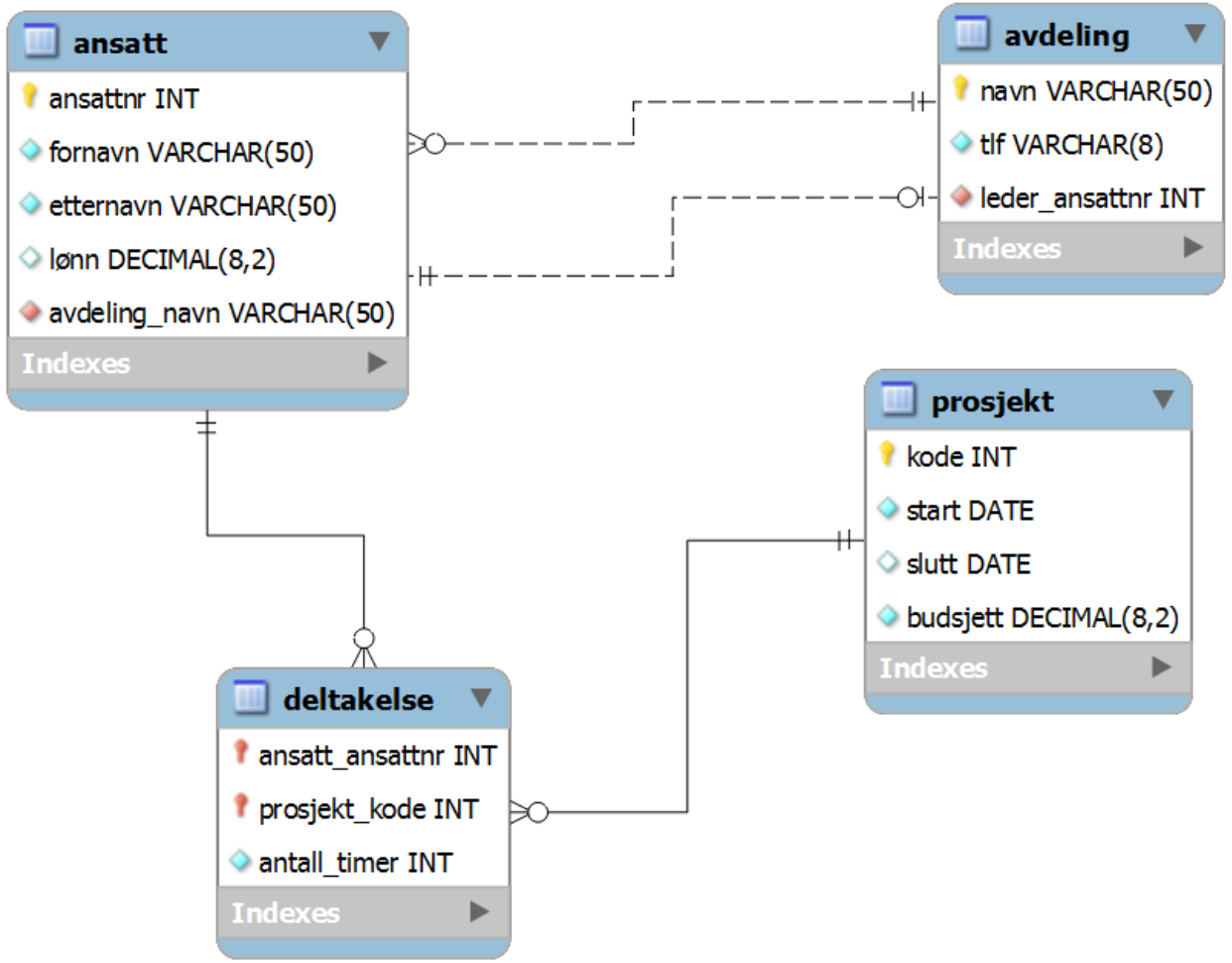
Discuss advantages of the different development strategies.

Data models

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



Example from MySQL Workbench



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:

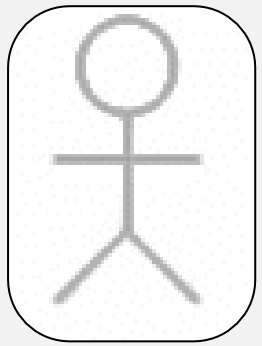
- https://mitt.uib.no/courses/27455/quizzes

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 …

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				



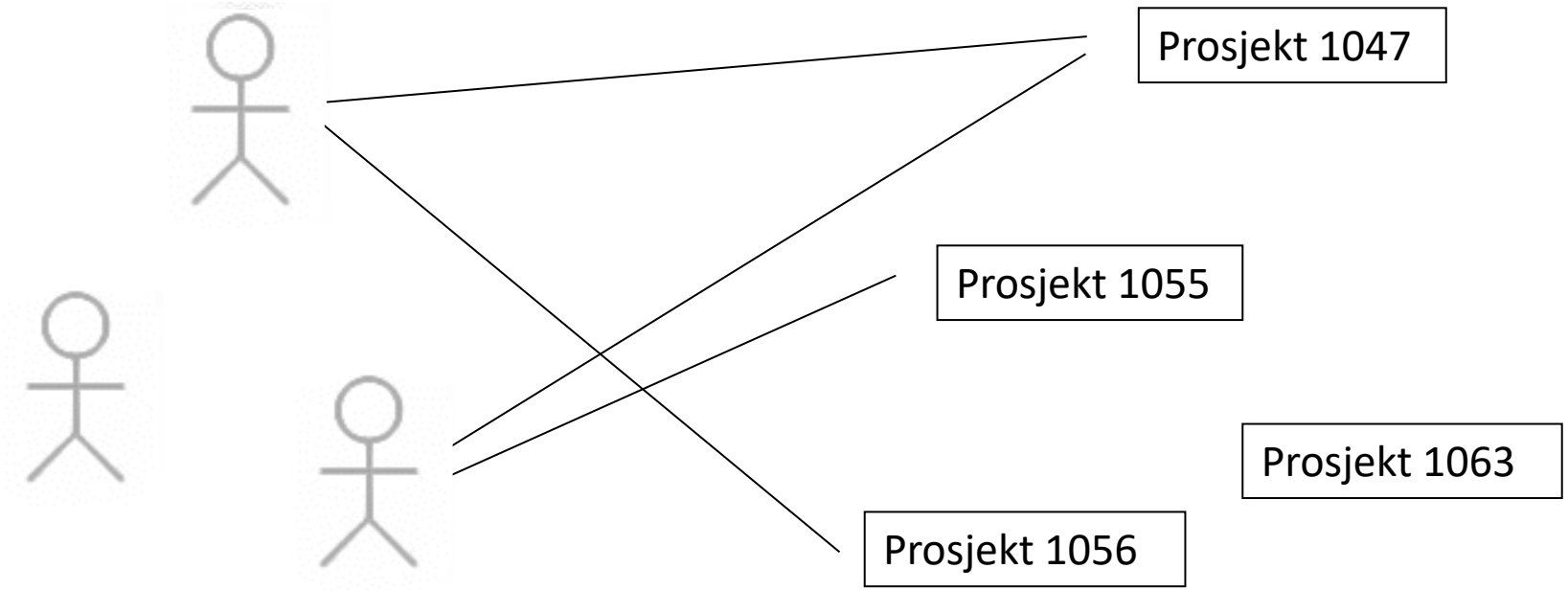
1 rad = 1 ansatt

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

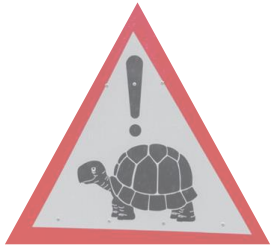
Employee and Projects



- One employee can participage 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 !

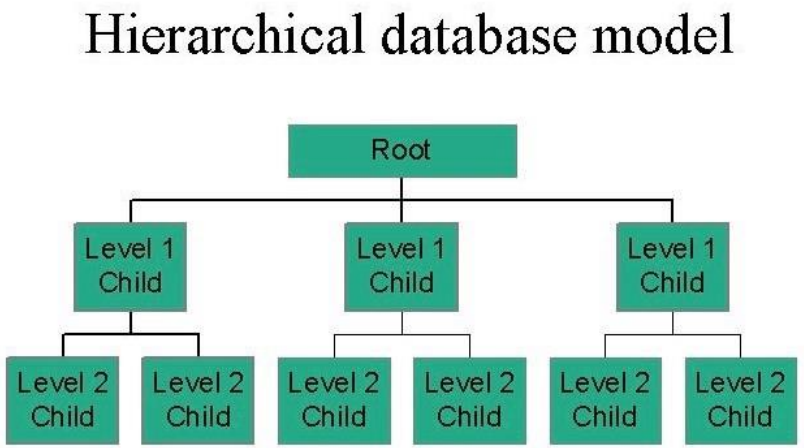
Now: 15 minute break
Lecture resumes at 11:15

Relational database = « Table database »

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

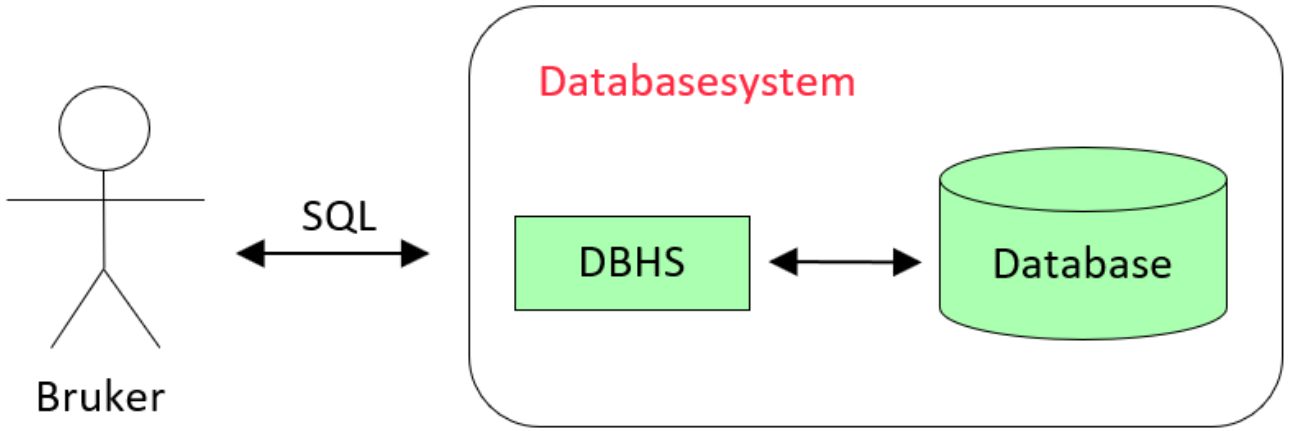
  is a collection of tables (relations).

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



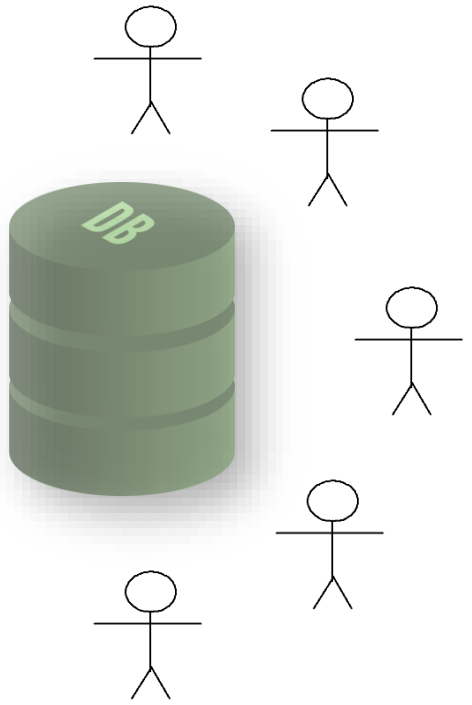
Database system = *DBMS* + database

- 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.

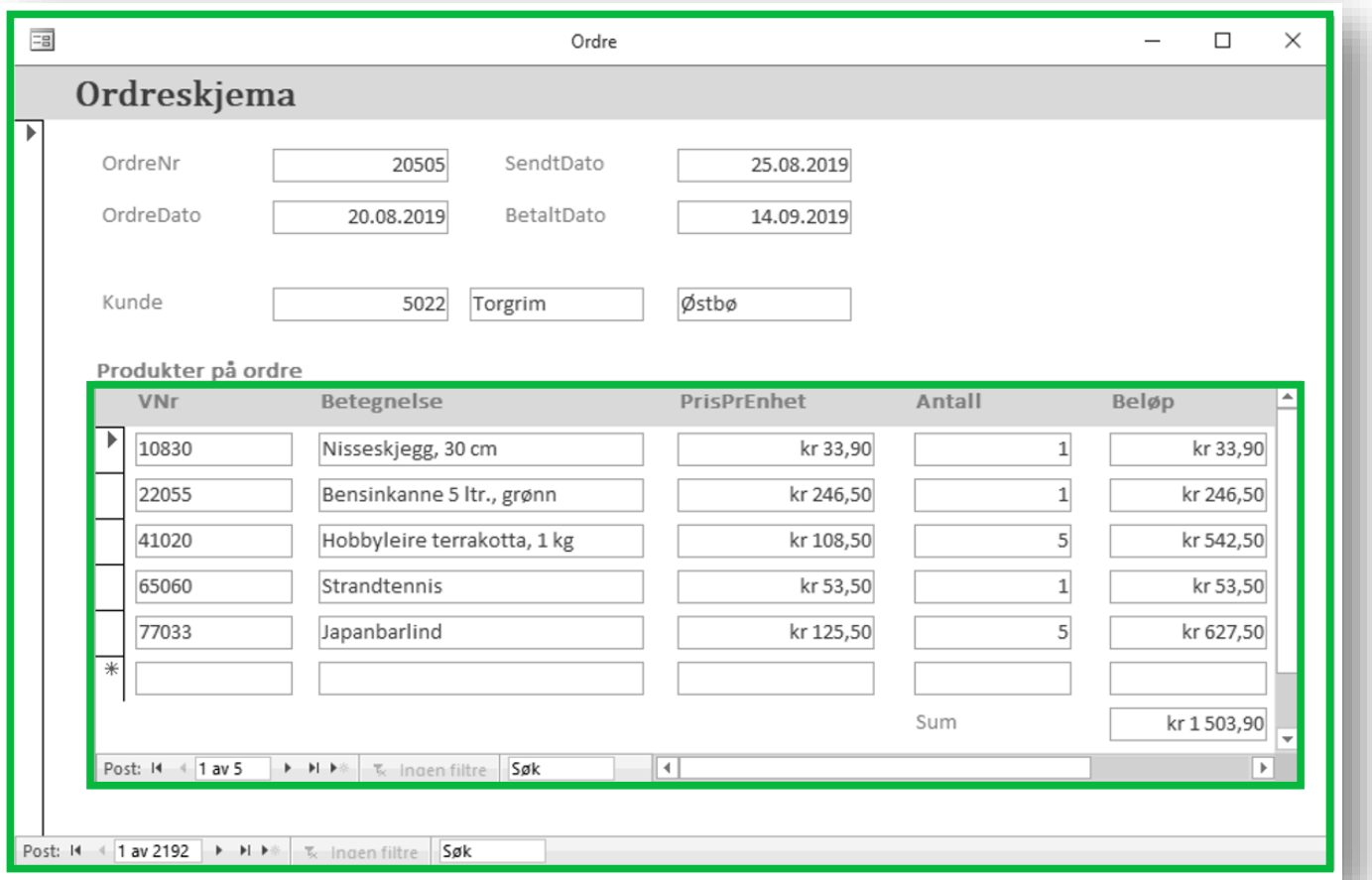


Database clients in a network

- 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.



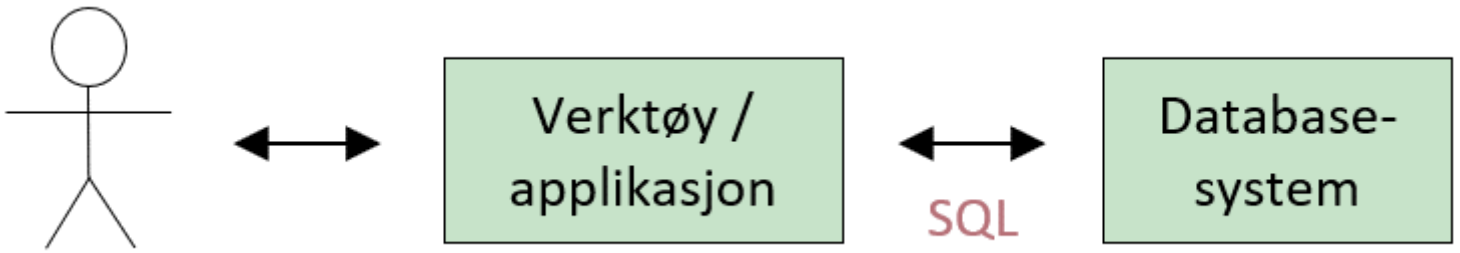
Database Users and Tools / Applications

- **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:

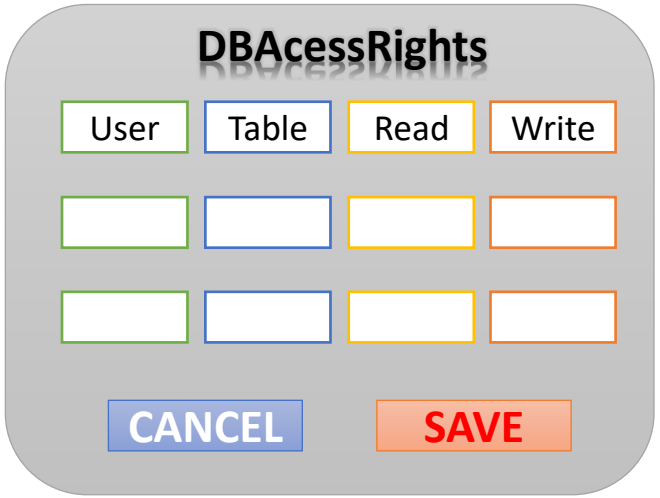
How would you design an application to manage read and write access rights of the users of a database?

- A table **Users** with columns:

  U*serNumber, UserName, TableName, ReadAccess, WriteAccess.*

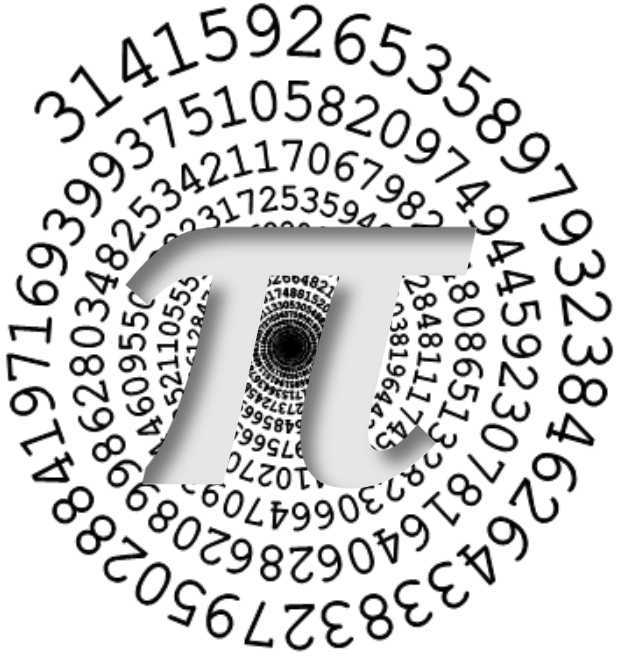
- A GUI where one can add users and set values.

User Number	User Name	Table Name	Read Access	Write Access



Representation of Data in Computers

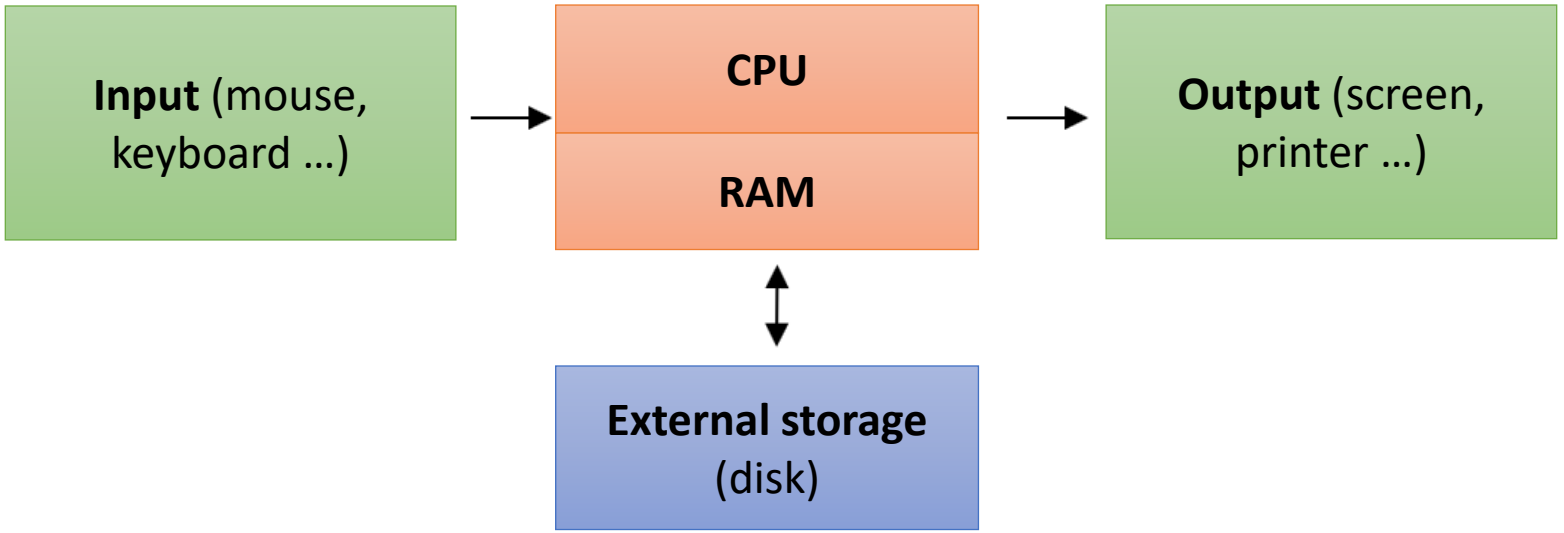
1. How do **computers** work ?
2. **Represenation** of **numbers**
3. **Representation** of **text**
4. **Representation** of **tables**



How do computers work ?

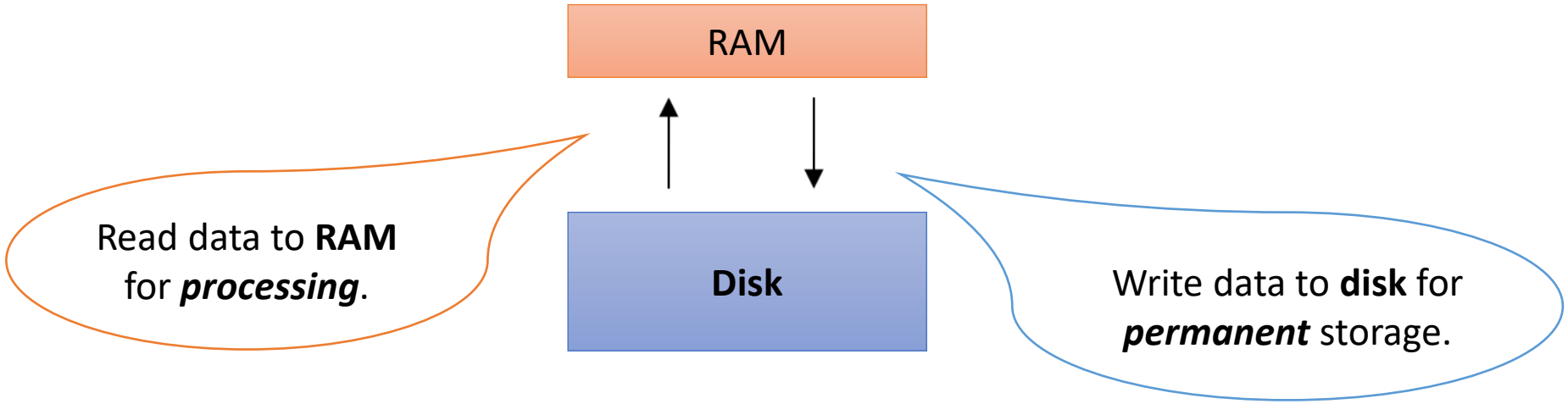
The physical **details** of a **computer** are **complex** (gates, transistors…). Let's consider **a simplified but useful model**.

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



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*** !



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
…	…	…	…	…	…	…	…	…

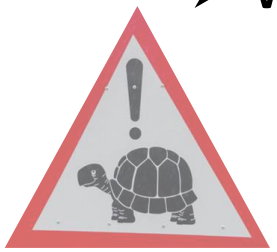
Integer numbers (or integers):

➢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
…	…	…	…	…	…	…	…	…

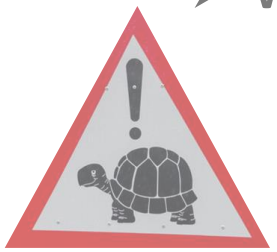
Integer numbers (or integers):

➢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 $0.486229 \times 10^{3.}$
- Which can be represented as the integers 486229 and 3.

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

Representing Text

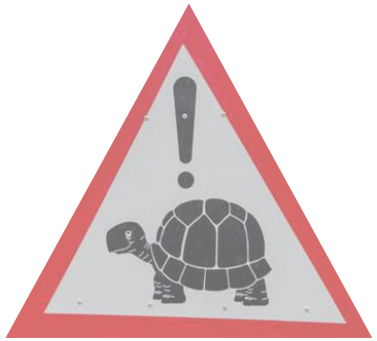
A **character encoding** assigns one number to every symbol.

➢***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**:

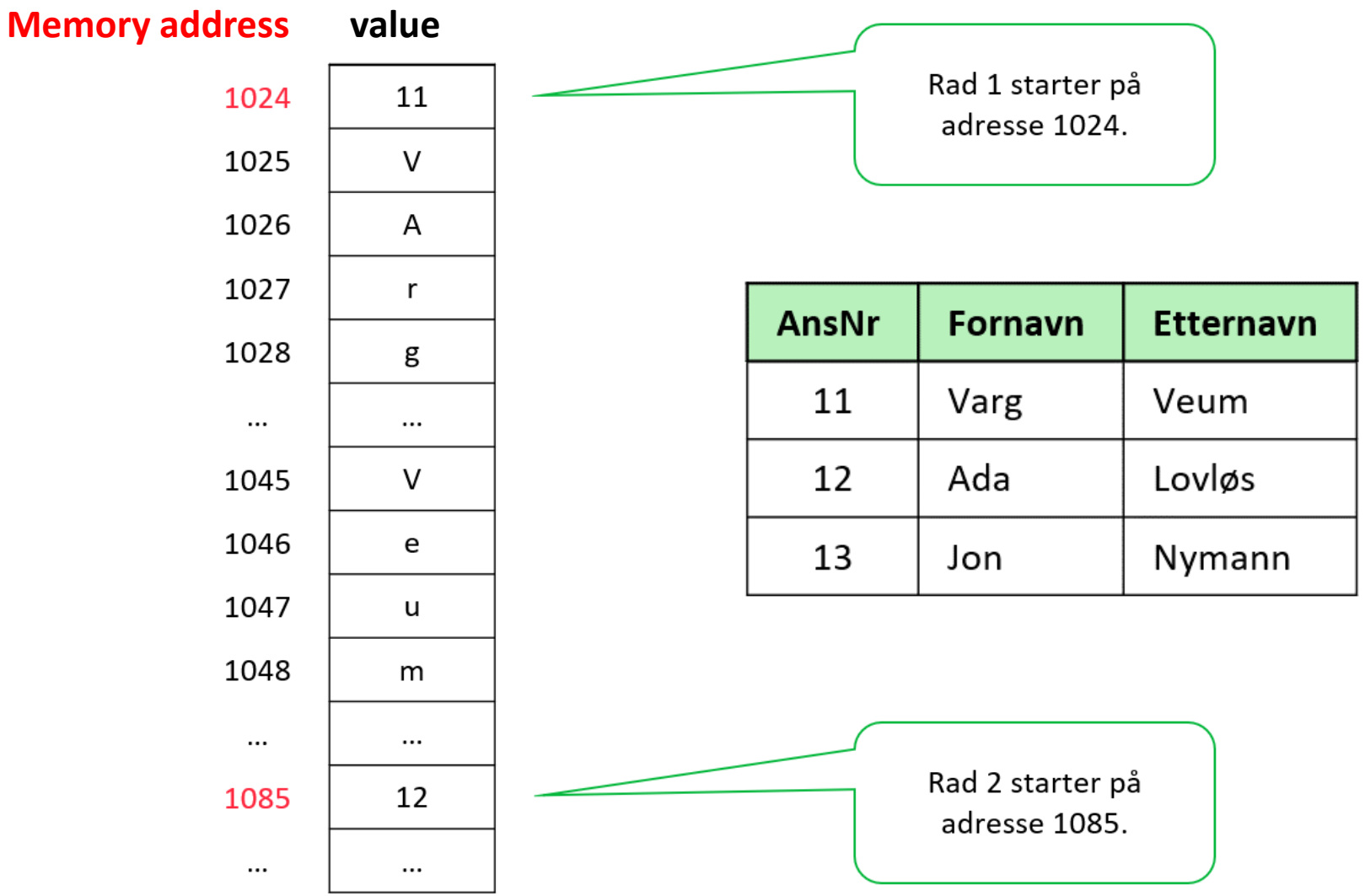
- **Integer** numbers: 1, 2, 3 …
- **Decimal** numbers (floats): such as 0.0, 0.101 …
- **Text** (string of characters): «Gjelder hele Svalbard» …
- **Boolean** truth values: **True** or **False**
- **Date** / **Time** (as numbers): 26.01.2021, 10:15:00.000

- **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:



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/27455/quizzes

Summary: Chapter 1 – Introduction to Databases

- 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**.