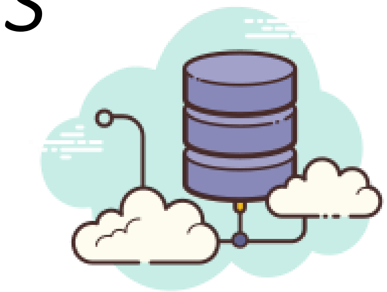
INF115 Lecture 3: *Creating and Managing Tables*

Adriaan Ludl
Department of Informatics
University of Bergen

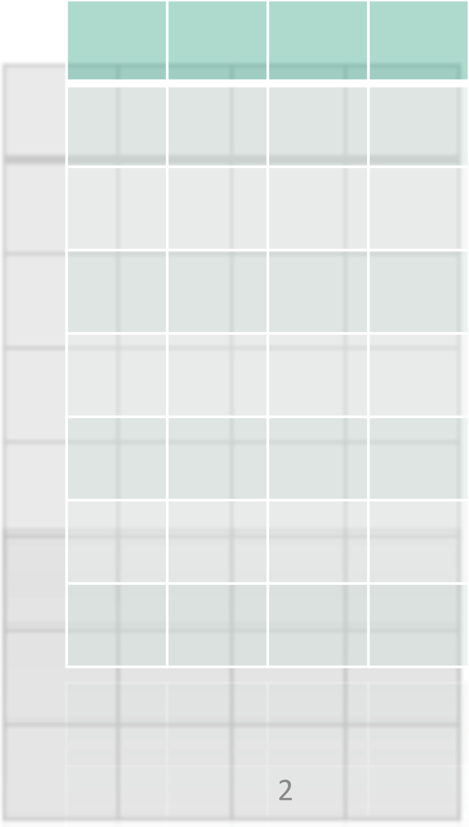
Spring Semester

Chapter 3: *Creating and Managing Tables*



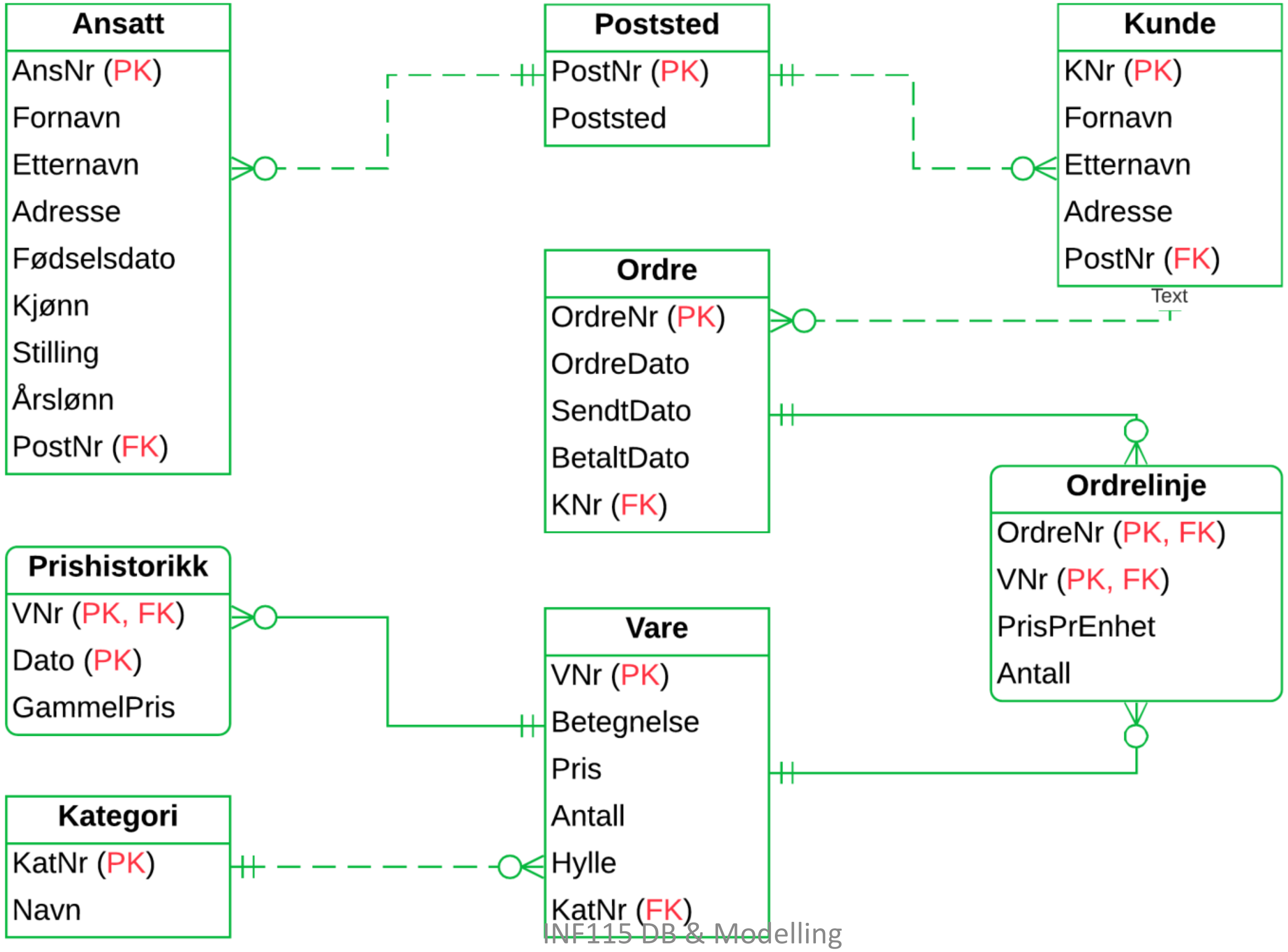
Learning Goals:

- ➢ **Understand** how tables are built from rows and columns
- ➢ Be able to **create tables using SQL**:
  - ❖ **Primary keys, *foreign keys***,
  - ❖ Declare **Data types**,
  - ❖ Validation or Business **rules** (forretningsregler)
- ➢ Use SQL to **manage table contents**: insert, update, delete
- ➢ Write **SQL-scripts that create tables** and fill in data



A Database Contains Many Tables

- The diagram shows a **<u>logical</u> data model** of «*Hobbyhuset*»
- Every box represents one table. Further modeling details will be explained in chapters 7 and 8.



Ansatt
AnsNr (PK)
Fornavn
Etternavn
Adresse
Fødselsdato
Kjønn
Stilling
Årslønn
PostNr (FK)

Poststed
PostNr (PK)
Poststed

Kunde
KNr (PK)
Fornavn
Etternavn
Adresse
PostNr (FK)

Text

Ordre
OrdreNr (PK)
OrdreDato
SendtDato
BetaltDato
KNr (FK)

Ordrelinje
OrdreNr (PK, FK)
VNr (PK, FK)
PrisPrEnhet
Antall

Prishistorikk
VNr (PK, FK)
Dato (PK)
GammelPris

Vare
VNr (PK)
Betegnelse
Pris
Antall
Hylle
KatNr (FK)

Kategori
KatNr (PK)
Navn

The Database Table *Products* (*Vare*)

VNr	Betegnelse	Pris	Kategori	Antall	Hylle
90693	Marsipantang	57.00	Konditor	0	B17
44939	Malerskrin, 6 farger	115.00	Hobbymaling	2	B02
10830	Nisseskjegg, 30 cm	57.50	Dukker	42	
64551	Hengebegonia, 10 stk	118.00	Blomster	206	E05
15217	Kram tørrfluekorker	32.00	Fiske	213	B42
90164	Lakrisekstrakt, 100g	75.50	Konditor	104	B06
15207	Antron garn, hvit	24.50	Fiske	21	B41
13001	Glasskuler, 100 gr	38.00	Dukker	0	E11
15211	Tubeflueverktøy	209.00	Fiske	39	B42
33045	Blomkarse	17.50	Blomster	206	E05
55130	Moro med marsipan	298.50	Bøker	140	C20
42615	Gipsform marihøner	106.00	Keramikk	124	B03

- Row, column, value, data type, **primary key** (primærnøkkel), Null values.
- Can we avoid writing the same **category** **text** multiple times ?

Tables *Category* and *Products*

KatNr	Navn
1	Blomster
2	Bøker
3	Konditor
4	Hobbymaling
5	Keramikk
6	Dukker
7	Fiske

VNr	...	KatNr	...
90693	...	3	...
44939	...	4	...
10830	...	6	...
64551	...	1	...
15217	...	7	...
90164	...	3	...
15207	...	7	...
13001	...	6	...
15211	...	7	...
33045	...	1	...
55130	...	2	...
42615	...	5	...

- Look-up table
- **Foreign key** (fremmednøkkel)
- **Advantages:**
  - Avoid typing mistakes
  - Save memory space
  - Can recreate the original table.

Students

SNr	**FNr**	**Fornavn**	**Etternavn**	**Studium**	**Start**
071234	12017911111	Per	Lie	Informatikk	2019
072341	29020022222	Kari	Hansen	Økonomi	2019
067234	18059744444	Lise	Andersen	Informatikk	2018
048231	03119333333	Ola	Klausen	Historie	2018
079021	10109966666	Hanne	Jensen	Idrett	2019
063288	24099788888	Gro	Aslaksen	Innovasjon	2019
031253	31039555555	Åge	Hansen	Revisjon	2016
075199	04090100000	Kari	Larsen	Innovasjon	2019

- **SNr** (studentnr) and **FNr** (fødselsnr) are both **candidate keys**.

➢ **Choose one** candidate key **as primary key**.

Vehicles (Kjøretøy)

RegNr	Merke	Modell	År	Km
LY 12345	Toyota	Corolla	2010	225501
NV 54321	Toyota	Avensis	2019	15238
DA 98765	Ford	Focus	2016	47624
NV 67890	Nissan	Micra	2017	31092

- For some objects there is an established **primary key** in the «real world»

→ a **natural key**.

  - E.g.: ISBN for books, FNr for people

- For other objects it is **useful** to introduce a **surrogate key** (serial number = «løpenr»).

Primary keys and foreign keys

- A **primary key** (primærnøkkel) is composed of *one or several columns* which are used to **uniquely identify rows** in a table.
  - No repetitions !
  - No Null values allowed !

- A **foreign key** (fremmednøkkel) is a reference to a primary key of another table.
  - All values in the foreign key must be found in the respective primary key.
  - *A foreign key* ***can contain Null*** *values.*

Foreign keys (Fremmednøkler)

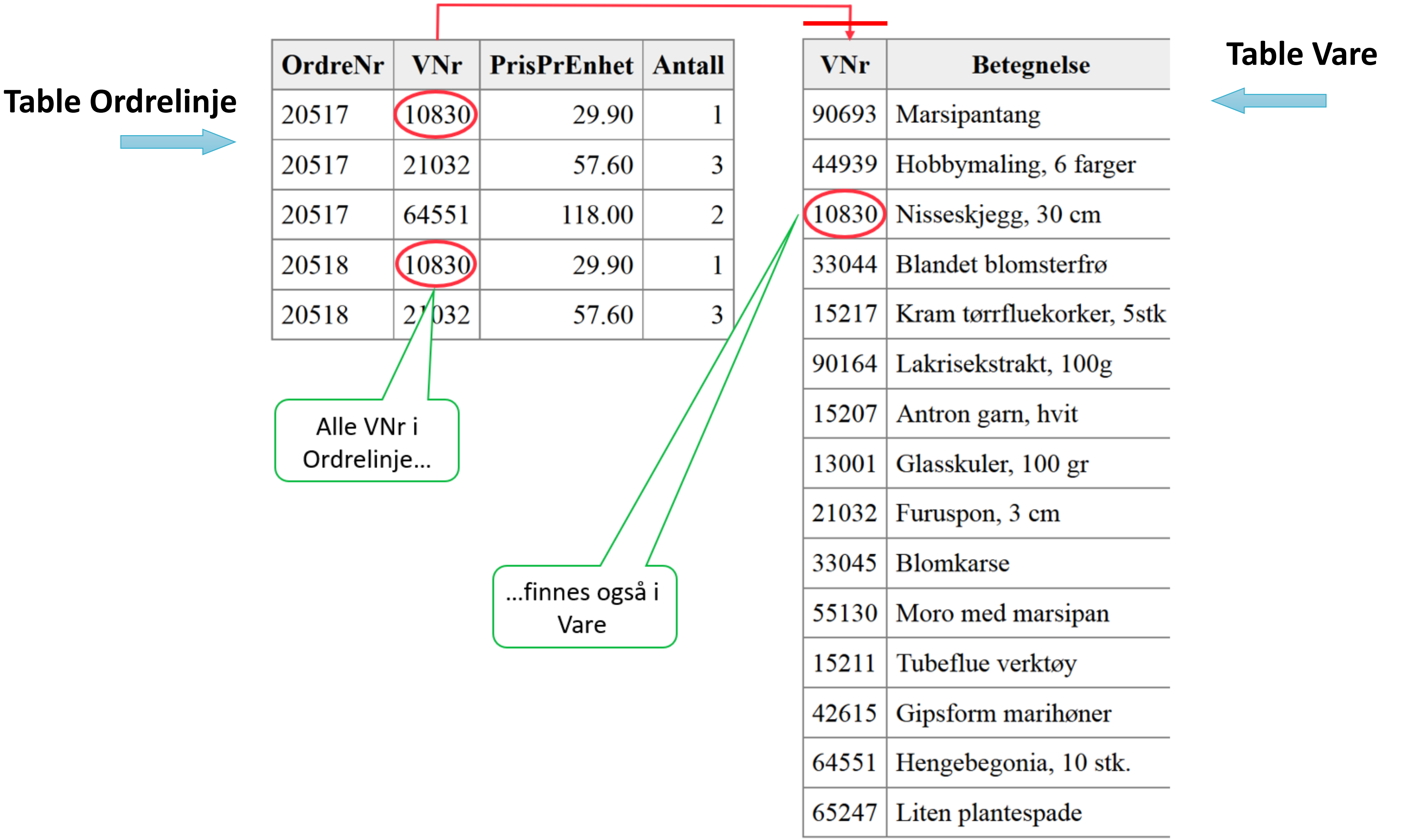


Table Ordrelinje

OrdreNr	VNr	PrisPrEnhet	Antall
20517	10830	29.90	1
20517	21032	57.60	3
20517	64551	118.00	2
20518	10830	29.90	1
20518	21032	57.60	3

Table Vare

VNr	Betegnelse
90693	Marsipantang
44939	Hobbymaling, 6 farger
10830	Nisseskjegg, 30 cm
33044	Blandet blomsterfrø
15217	Kram tørrfluekorker, 5stk
90164	Lakrisekstrakt, 100g
15207	Antron garn, hvit
13001	Glasskuler, 100 gr
21032	Furuspon, 3 cm
33045	Blomkarse
55130	Moro med marsipan
15211	Tubeflue verktøy
42615	Gipsform marihøner
64551	Hengebegonia, 10 stk.
65247	Liten plantespade

Alle VNr i Ordrelinje…

…finnes også i Vare

Quizz on Managing Tables (1)

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

Composed objects: Order and Orderline

Think of an order form, which consists of

- a header: date, client number (= kundenr) …
- and several «**lines**» (varenummer, antall, …)

OrdreNr	Ordredato	KNr
20505	20.8.2019	5022
20506	20.8.2019	5009
20507	20.9.2019	5188

There is a **one-to-many relationship**
between orders and order lines.

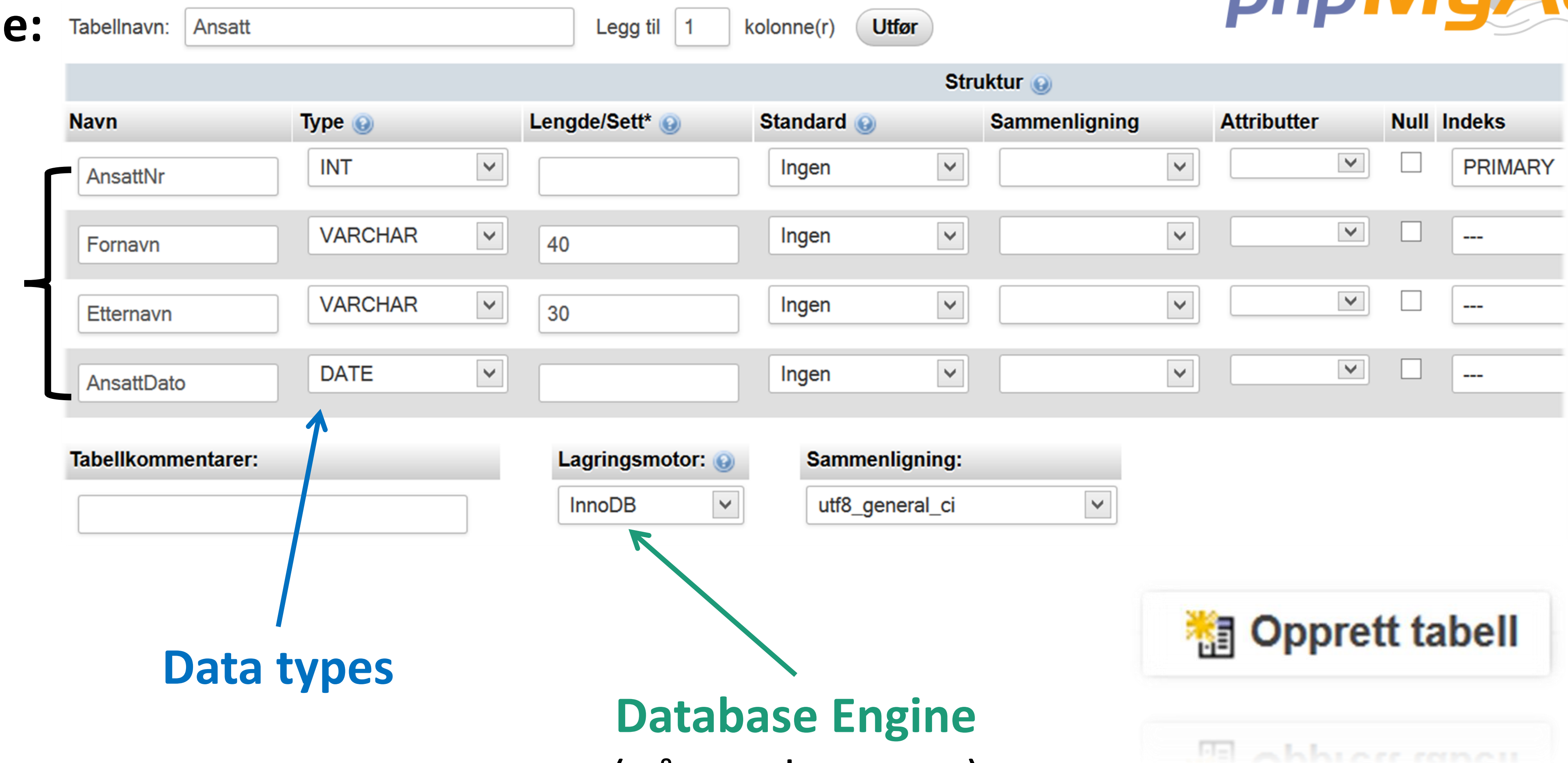
- Which are the primary keys ?
- Which are the foreign keys ?

Repetitions !

OrdreNr	VNr	Pris	Antall
20505	10830	29.90	1
20505	77033	109.50	5
20506	10830	29.90	1
20506	44939	57.60	3
20506	65081	109.50	5
20507	12088	109.50	2

How to create tables in phpMyAdmin

Table Name:



Tabellnavn: Ansatt Legg til 1 kolonne(r) Utfør

Struktur

Navn	Type	Lengde/Sett*	Standard	Sammenligning	Attributter	Null	Indeks
AnsattNr	INT		Ingen			□	PRIMARY
Fornavn	VARCHAR	40	Ingen			□	---
Etternavn	VARCHAR	30	Ingen			□	---
AnsattDato	DATE		Ingen			□	---

Tabellkommentarer:

Lagringsmotor: InnoDB

Sammenligning: utf8_general_ci

Primary Key

Column Names

Data types

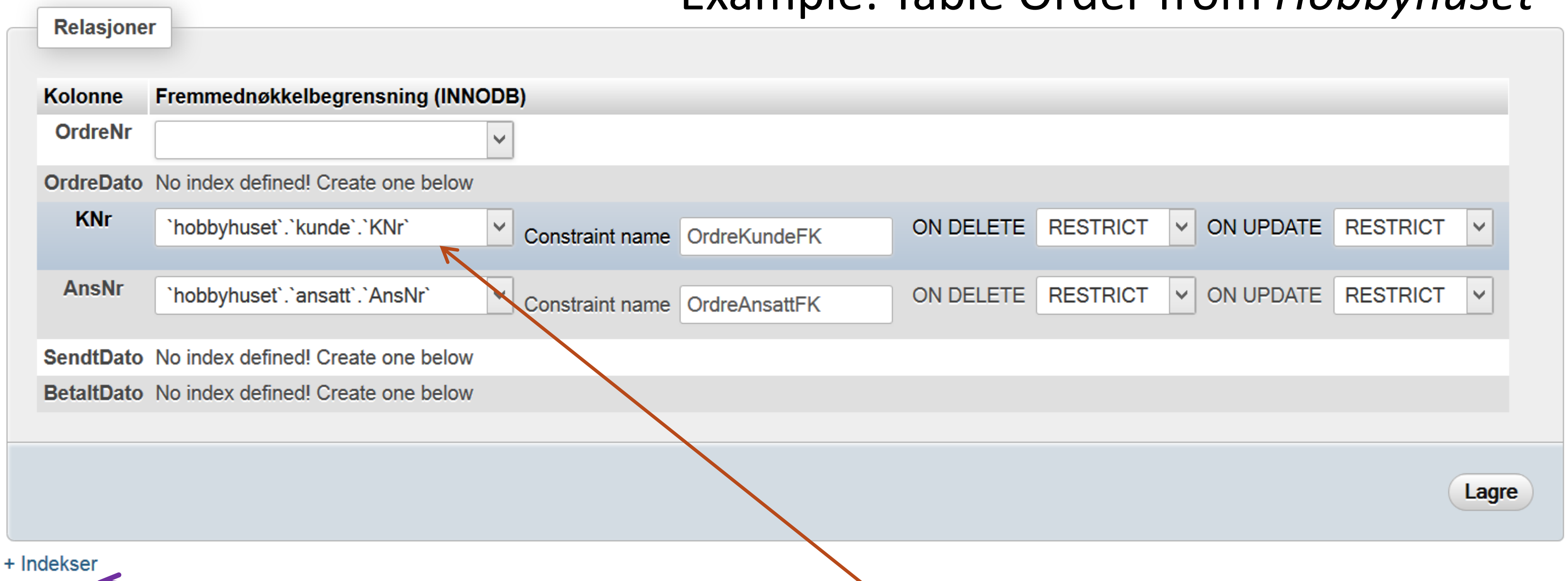
Database Engine

(på norsk: motor)

Creating foreign keys in phpMyAdmin

The **relational view** helps to understand the **structure.**

Example: Table Order from *Hobbyhuset*



Kolonne	Fremmednøkkelbegrensning (INNODB)						
OrdreNr							
OrdreDato	No index defined! Create one below						
KNr	\`hobbyhuset\`.\`kunde\`.\`KNr\`	Constraint name	OrdreKundeFK	ON DELETE	RESTRICT	ON UPDATE	RESTRICT
AnsNr	\`hobbyhuset\`.\`ansatt\`.\`AnsNr\`	Constraint name	OrdreAnsattFK	ON DELETE	RESTRICT	ON UPDATE	RESTRICT
SendtDato	No index defined! Create one below						
BetaltDato	No index defined! Create one below						

Set up indexes

Foreign keys «point to» columns in other tables

Creating Tables with SQL

The **structure** of a table must be **defined before** we can put data in it or read it out.

Commands for **table structure** i.e. columns :

- Define (opprette): `CREATE TABLE Students …`
- Change structure: `ALTER TABLE Students …`
- Delete: `DROP TABLE Students`

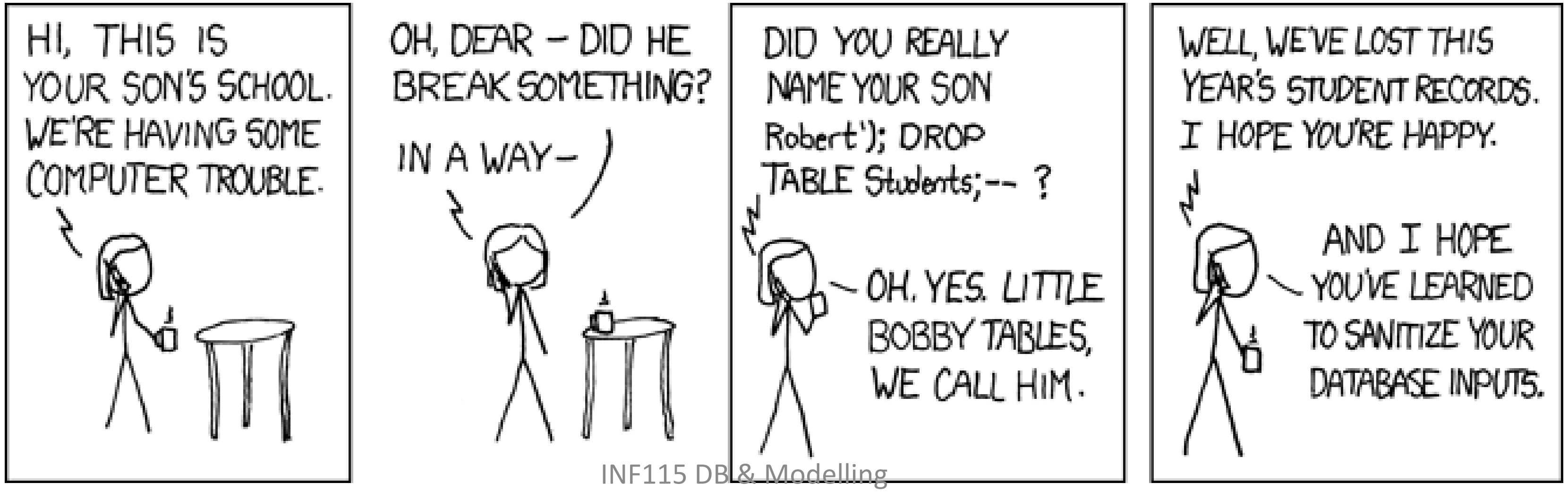
Creating Tables with SQL

The **structure** of a table must be **defined before** we can put data in it or read it out.

Commands for **table structure** i.e. columns :

- Define (opprette): `CREATE TABLE Students …`
- Change structure: `ALTER TABLE Students …`
- Delete: `DROP TABLE Students`

Exploits of a Mom (from xkcd.com/327)



HI, THIS IS YOUR SON'S SCHOOL. WE'RE HAVING SOME COMPUTER TROUBLE.
OH, DEAR – DID HE BREAK SOMETHING?
IN A WAY–
DID YOU REALLY NAME YOUR SON Robert'); DROP TABLE Students;-- ?
OH. YES. LITTLE BOBBY TABLES, WE CALL HIM.
WELL, WE'VE LOST THIS YEAR'S STUDENT RECORDS. I HOPE YOU'RE HAPPY.
AND I HOPE YOU'VE LEARNED TO SANITIZE YOUR DATABASE INPUTS.

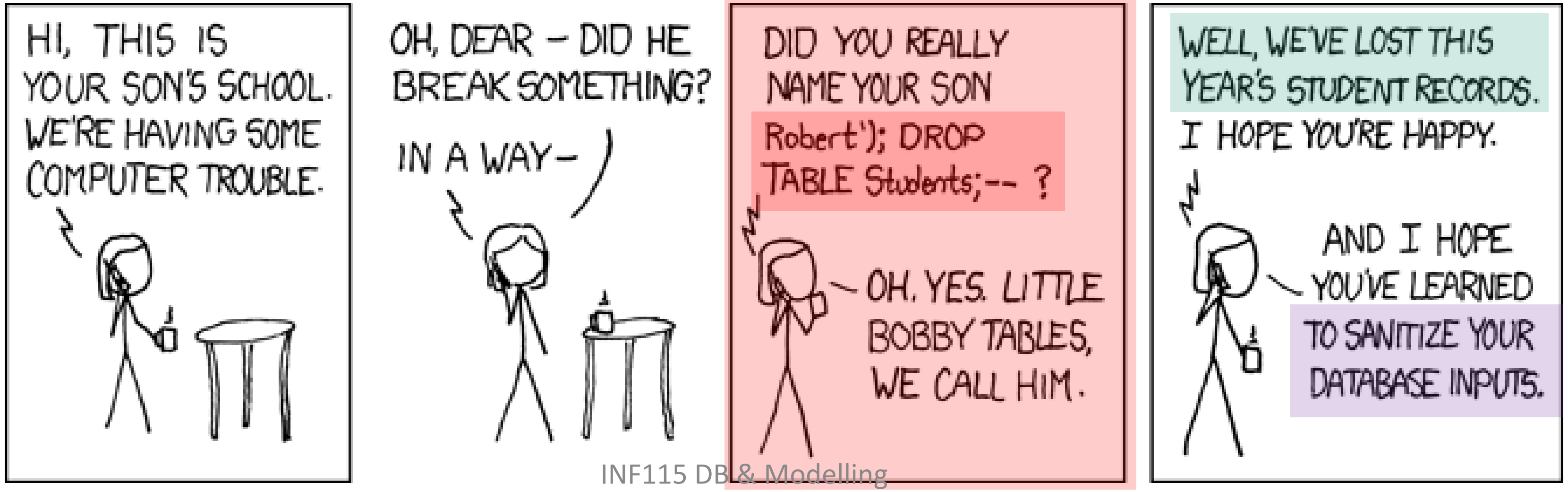
Creating Tables with SQL

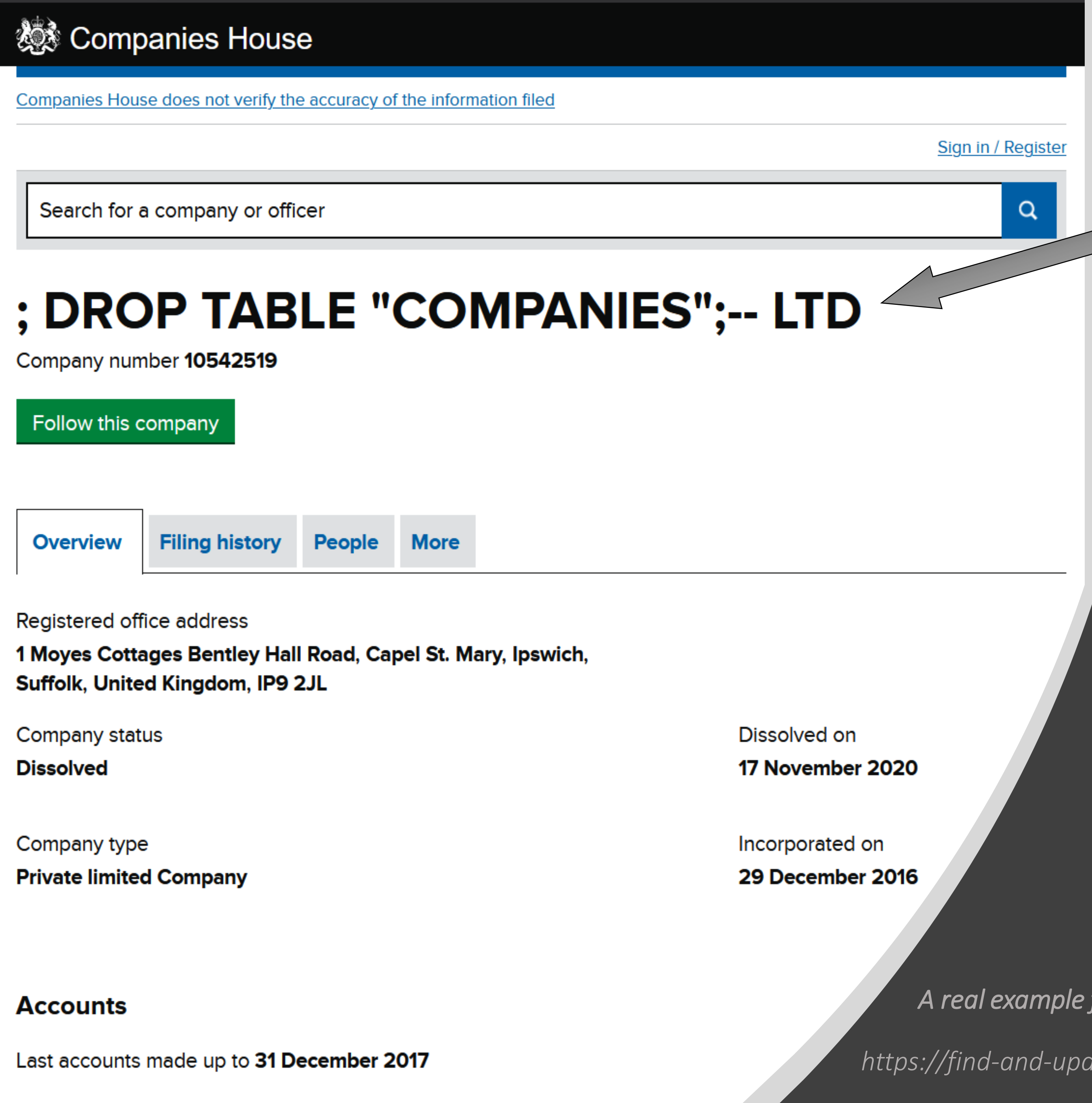
The **structure** of a table must be **defined before** we can put data in it or read it out.

Commands for **table structure** i.e. columns :

- Define (opprette): `CREATE TABLE Students …`
- Change structure: `ALTER TABLE Students …`
- Delete: `DROP TABLE Students`

Exploits of a Mom (from xkcd.com/327)


HI, THIS IS YOUR SON'S SCHOOL. WE'RE HAVING SOME COMPUTER TROUBLE.
OH, DEAR – DID HE BREAK SOMETHING?
IN A WAY–
DID YOU REALLY NAME YOUR SON Robert'); DROP TABLE Students;-- ?
OH. YES. LITTLE BOBBY TABLES, WE CALL HIM.
WELL, WE'VE LOST THIS YEAR'S STUDENT RECORDS. I HOPE YOU'RE HAPPY.
AND I HOPE YOU'VE LEARNED TO SANITIZE YOUR DATABASE INPUTS.



Companies House does not verify the accuracy of the information filed

; DROP TABLE "COMPANIES";-- LTD

Company number **10542519**

Registered office address

**1 Moyes Cottages Bentley Hall Road, Capel St. Mary, Ipswich,
Suffolk, United Kingdom, IP9 2JL**

Company status

Dissolved

Dissolved on

17 November 2020

Company type

Private limited Company

Incorporated on

29 December 2016

Accounts

Last accounts made up to **31 December 2017**

A ***real*** *company name …*

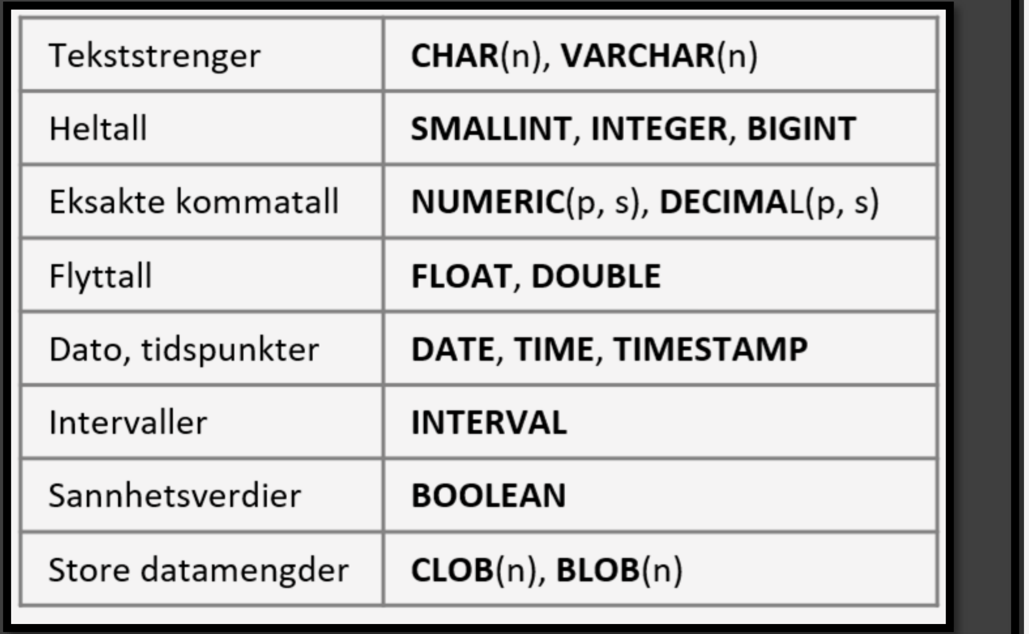
15 minute break

Lecture resumes at 11:00

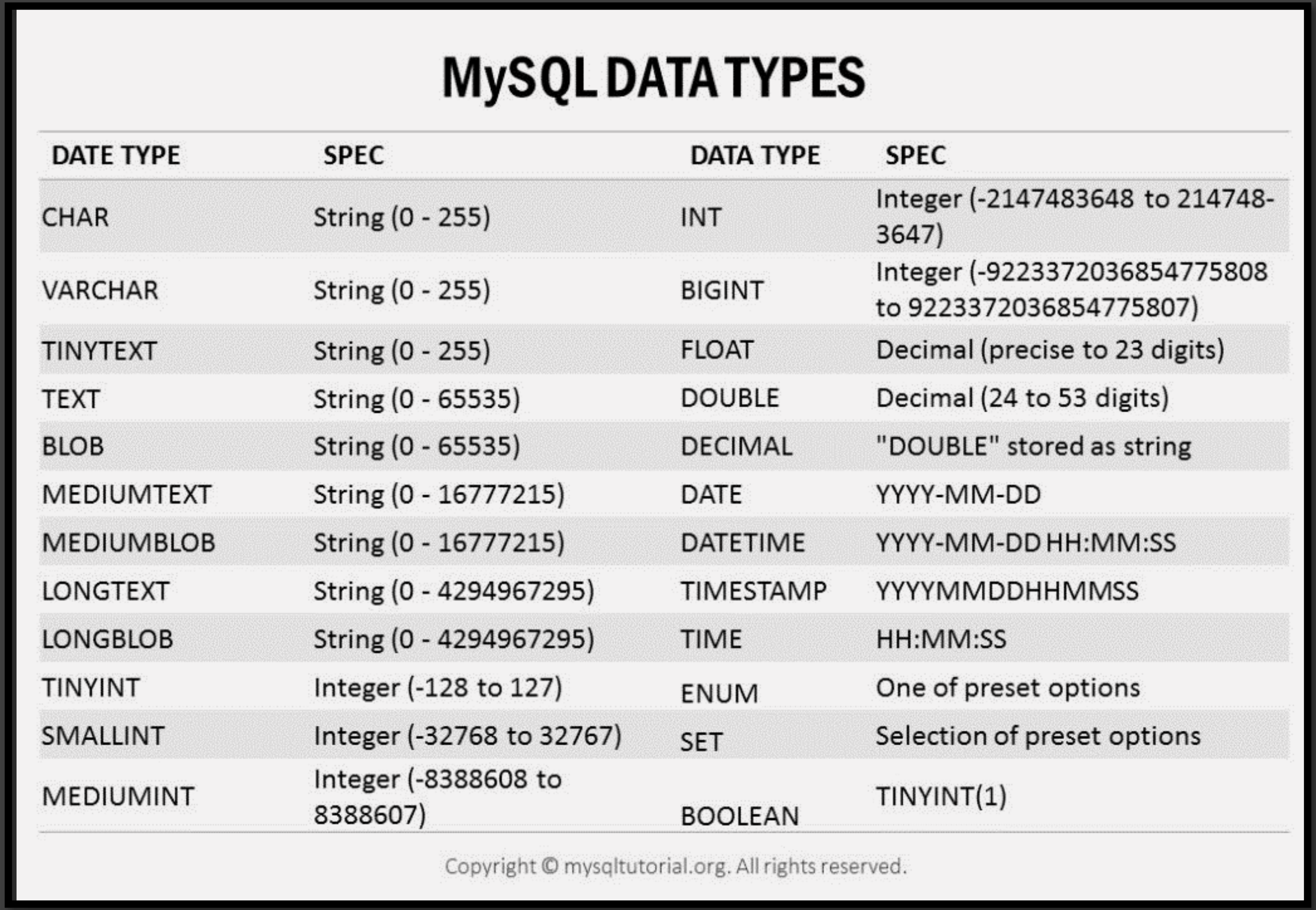
A real example from:

https://find-and-update.company-information.service.gov.uk/company/10542519

Data types in SQL



Tekststrenger	**CHAR**(n), **VARCHAR**(n)
Heltall	**SMALLINT**, **INTEGER**, **BIGINT**
Eksakte kommatall	**NUMERIC**(p, s), **DECIMA**L(p, s)
Flyttall	**FLOAT**, **DOUBLE**
Dato, tidspunkter	**DATE**, **TIME**, **TIMESTAMP**
Intervaller	**INTERVAL**
Sannhetsverdier	**BOOLEAN**
Store datamengder	**CLOB**(n), **BLOB**(n)



MySQL DATA TYPES

DATE TYPE	SPEC	DATA TYPE	SPEC
CHAR	String (0 - 255)	INT	Integer (-2147483648 to 2147483647)
VARCHAR	String (0 - 255)	BIGINT	Integer (-9223372036854775808 to 9223372036854775807)
TINYTEXT	String (0 - 255)	FLOAT	Decimal (precise to 23 digits)
TEXT	String (0 - 65535)	DOUBLE	Decimal (24 to 53 digits)
BLOB	String (0 - 65535)	DECIMAL	"DOUBLE" stored as string
MEDIUMTEXT	String (0 - 16777215)	DATE	YYYY-MM-DD
MEDIUMBLOB	String (0 - 16777215)	DATETIME	YYYY-MM-DD HH:MM:SS
LONGTEXT	String (0 - 4294967295)	TIMESTAMP	YYYYMMDDHHMMSS
LONGBLOB	String (0 - 4294967295)	TIME	HH:MM:SS
TINYINT	Integer (-128 to 127)	ENUM	One of preset options
SMALLINT	Integer (-32768 to 32767)	SET	Selection of preset options
MEDIUMINT	Integer (-8388608 to 8388607)	BOOLEAN	TINYINT(1)

Data types in SQL

SQL defines a set of **built-in data types**, amongst others:

- `CHAR(n)` - e.g. `CHAR(30)`
- `VARCHAR(n)` - e.g. `VARCHAR(30)`
- `INTEGER`
- `SMALLINT`
- `DECIMAL(p,s)` - e.g. `DECIMAL(5,2)`
- `FLOAT`
- `DATE`
- `BOOLEAN`

Some systems offer more data types.

For example: `CURRENCY` in Access

Creating tables with SQL:

Column definition + primary key + foreign keys:

```
CREATE TABLE Vare
(
  VNr        CHAR(5),
  Betegnelse VARCHAR(30),
  Pris       DECIMAL(8,2),
  KatNr      SMALLINT,
  Antall     INTEGER,
  Hylle      CHAR(3),
  CONSTRAINT VarePN PRIMARY KEY (VNr),
  CONSTRAINT VareKategoriFN FOREIGN KEY (KatNr)
    REFERENCES Kategori (KatNr)
);
```

Creating tables with SQL: simplified version

The reserved word CONSTRAINT and the following name **can be skipped.**
Then we get a simpler version:

```
CREATE TABLE Vare
(
  VNr        CHAR(5),
  Betegnelse VARCHAR(30),
  Pris       DECIMAL(8,2),
  KatNr      SMALLINT,
  Antall     INTEGER,
  Hylle      CHAR(3),
  PRIMARY KEY (VNr),
  FOREIGN KEY (KatNr)
    REFERENCES Kategori (KatNr)
);
```

Null values, repetitions and default values

- NOT NULL indicates that this column **must be filled in.**
- UNIQUE *prevents two or more identical values*.
- DEFAULT is used to define **default values** (standardverdier).

```
CREATE TABLE Student
(
  StudNr CHAR( 6 ),
  FodselsNr CHAR( 11 ) UNIQUE NOT NULL,
  Fornavn VARCHAR(15) NOT NULL,
  Etternavn VARCHAR(40) NOT NULL,
  PrivatEpost VARCHAR( 40 ) UNIQUE,
  Status VARCHAR(15) DEFAULT 'Aktiv',
  CONSTRAINT StudentPK PRIMARY KEY ( StudNr )
)
```

Validation / Business rules (forretningsregler)

- **Validation rules** aim to increase data quality, e.g.:
  - `Lønn < 1000000`
  - `Fødselsdato < AnsattDato`

*The **DBMS ensures** that **validation rules** are **respected**.*

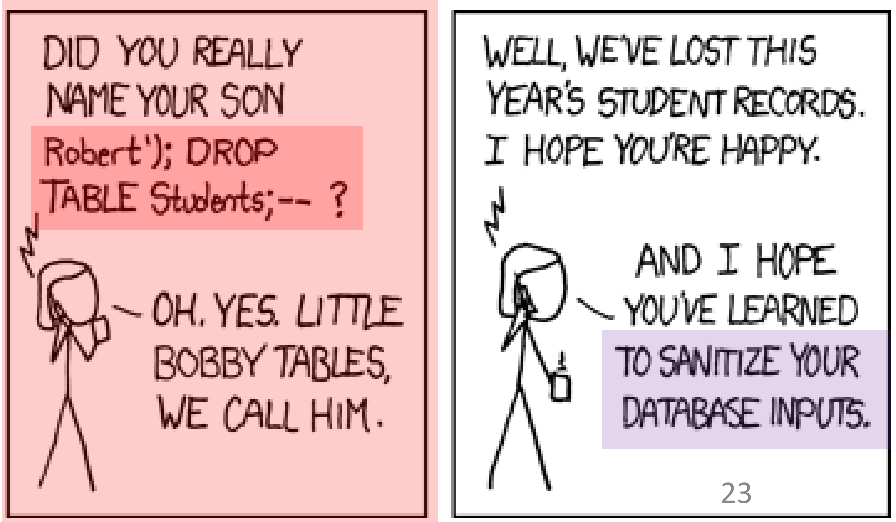
- Can be implemented with **CHECK** in CREATE TABLE

- **More complicated** Business rules

  can be checked with so-called **triggers**.

- We will come back to this topic later …

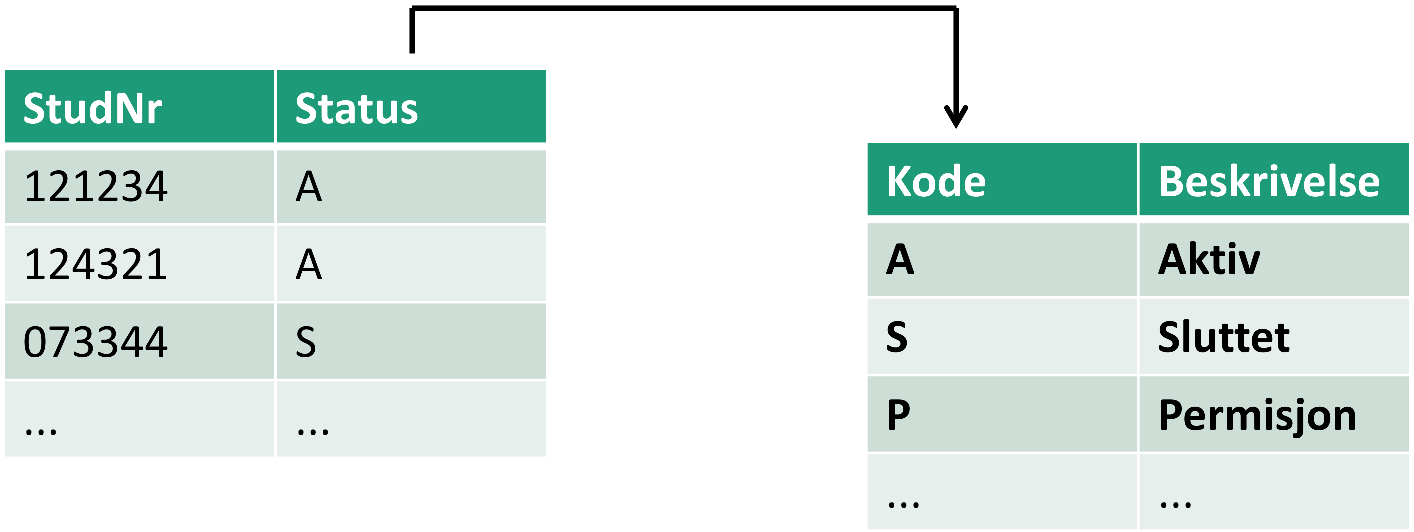
Exploits of a Mom (from xkcd.com/327)


DID YOU REALLY NAME YOUR SON Robert'); DROP TABLE Students;-- ?
OH. YES. LITTLE BOBBY TABLES, WE CALL HIM.
WELL, WE'VE LOST THIS YEAR'S STUDENT RECORDS. I HOPE YOU'RE HAPPY.
AND I HOPE YOU'VE LEARNED TO SANITIZE YOUR DATABASE INPUTS.

Foreign keys and validation rules

Example: allowed values for the column **Status** in the table **Student** must be «Aktiv», «Permisjon» og «Sluttet».

- We could add it with CHECK in CREATE TABLE.
- **But we can also use a «code table» and *foreign keys*:**



StudNr	Status
121234	A
124321	A
073344	S
…	…

Kode	Beskrivelse
A	**Aktiv**
S	**Sluttet**
P	**Permisjon**
…	…

Variants of Validation Rules

- When **deleting** a client: set **NULL** in Ordre.KNr for the client's order.
- When **changing** a ClientNumber in Client (Kunde):

  Update ClientNumber in the respective Order.KNr .

```
CONSTRAINT ClientOrderFK
  FOREIGN KEY (KNr)
   REFERENCES Client (KNr)
  ON DELETE SET NULL,
  ON UPDATE CASCADE
```

- Other possibilities:
  - `RESTRICT` `protect the parent from deletion` (gives an error message)
  - `SET DEFAULT` (define default values of your choosing)

Managing Table Contents:

INSERT	= insert new rows,
UPDATE	= change existing rows,
DELETE	= remove selected rows.

- INSERT has two variants: add 1 row / add many rows !
- UPDATE and DELETE **affect all rows by default**

  and ***must be <u>restricted</u> with a <u>condition</u>.***
- These commands are *more frequently used by programmers* than by users.

Useful Insert Queries

- When we employ a new seller (selger)

```
INSERT INTO Ansatt
       (AnsattNr, Etternavn, Stilling)
VALUES ( 14, 'Hansen', 'Selger’)
```

- For **column** names that are **omitted in this list NULL or default values will used.**
- **Automatically numbered columns** («løpenr») **must not be included in the list.**

- Copy old projects to a *helper table*:

```
INSERT INTO GamleProsjekter
  SELECT *
  FROM Prosjekt
  WHERE SluttDato <= '31/12/2018'
```

Update Queries

We increase the salaries of all employees by 10% (very Norwegian ;-)

```
UPDATE Ansatt
SET Lønn=Lønn*1.1
```

We increase the salary and change the title of the secretaries (sekretærene):

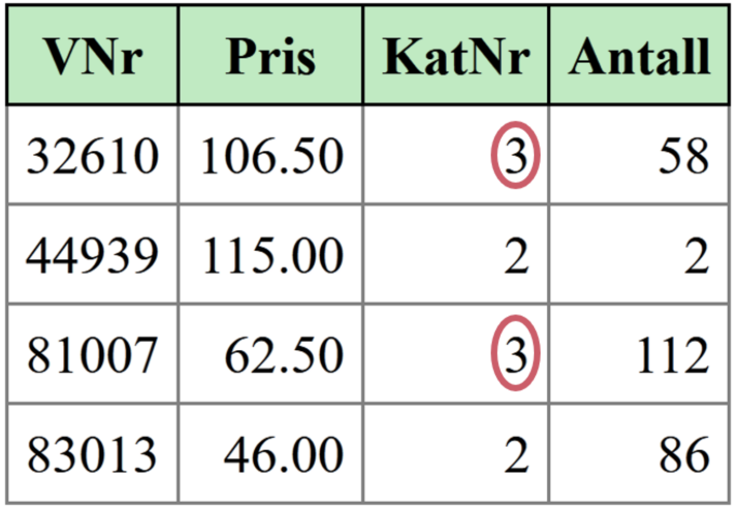
```
UPDATE Ansatt
SET Lønn=Lønn*1.5,
    Stilling='Sjefssekretær'
WHERE Stilling='Sekretær’
```

- **UPDATE** without **WHERE** affects all rows!

Visualisation of updates

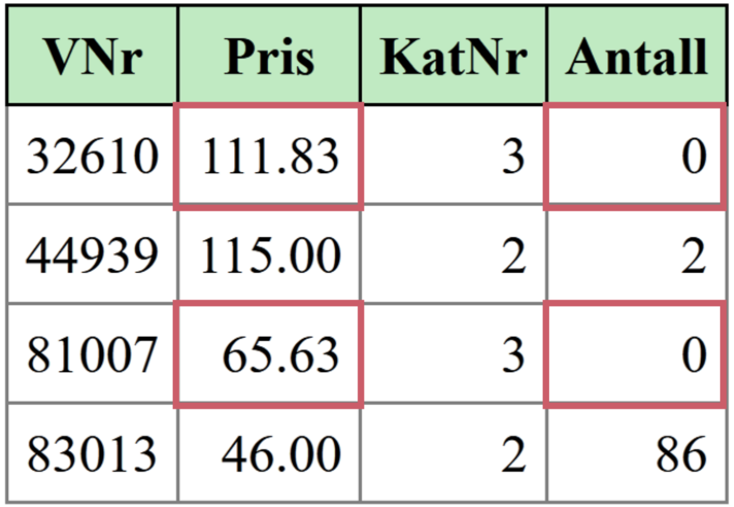
```
UPDATE Vare
SET Antall = 0, Pris = Pris*1.05
WHERE KatNr = 3
```

Original table Vare



VNr	Pris	KatNr	Antall
32610	106.50	3	58
44939	115.00	2	2
81007	62.50	3	112
83013	46.00	2	86

Updated table Vare



VNr	Pris	KatNr	Antall
32610	111.83	3	0
44939	115.00	2	2
81007	65.63	3	0
83013	46.00	2	86

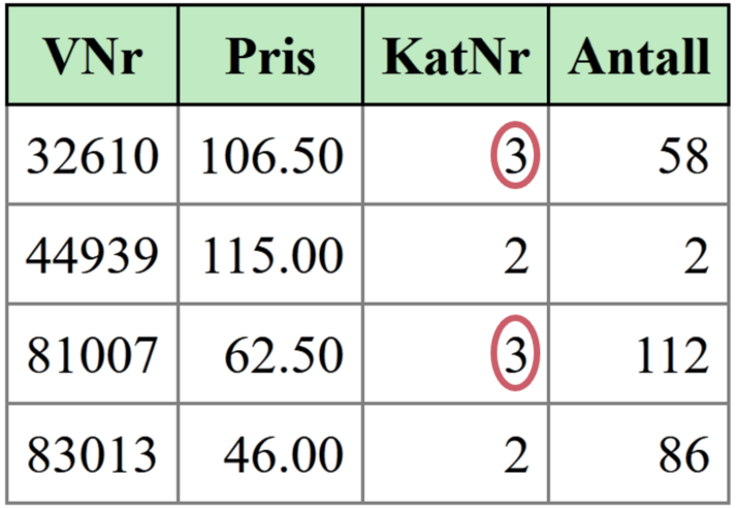
Delete Queries

- Remove all products in category 2 (books):

```
DELETE FROM Vare
WHERE KatNr = 2
```

- What does this do ?

```
DELETE FROM Vare
```



VNr	Pris	KatNr	Antall
32610	106.50	3	58
44939	115.00	2	2
81007	62.50	3	112
83013	46.00	2	86

❖ If we want **to delete from several tables,**

then we have to **write one DELETE command for each table.**

Quizz on Managing Tables (2)

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

Script files: *.sql

An **SQL script** is a sequence of SQL commands (setninger) saved in a file.

- ***SQL scripts are very useful*** when developing new databases

```
CREATE TABLE Vare …
CREATE TABLE Kunde …
CREATE TABLE Ordre …
INSERT INTO Vare …
INSERT INTO Vare …
…
```

*Access **cannot** read SQL script files !*

Try MySQL !

When **starting the script by dropping the tables**, we can run the script several times:

- make changes and run it again!

```
DROP TABLE Vare
DROP TABLE Kunde
…
```

Practical SQL Scripting

- **MySQL Workbench** enables easy scripting (one click)
  - It works as IDE for DB development (e.g. PyCharm): code and run scripts

- **phpMyAdmin** in XAMPP:
  - You write the code in a text editor (e.g. notepad),
  - Then select the import tab and select the file from your filesystem,
  - Then run the script.

- Both tools can do all we need for this course.

Data and Metadata («data about data»)

A database system contains *both data and metadata*.

A metadatabase describes the structure of a database:

- What are the tables called ?
- Which columns do the tables contain ?
- What are the data types of the columns ?
- What are the primary keys of each table ?
- Which columns are foreign keys pointing to which table ?

Data and Metadata («data about data»)

A database system contains *both data and metadata*.

A metadatabase describes the structure of a database:

- What are the tables called ?
- Which columns do the tables contain ?
- What are the data types of the columns ?
- What are the primary keys of each table ?
- Which columns are foreign keys pointing to which table ?

Metadata can be saved in tables:

- ➢ The **DBMS** *updates* the tables in the **system catalogue** (systemkatalogen).
- ➢ ***Can write SQL queries on these system tables.***

Summary (1/2): Creating and Managing Tables

CREATE TABLE using SQL:

- Declare columns and their **data types:**

  e.g. `Betegnelse VARCHAR(30), Pris  DECIMAL(8,2),`

- **Primary keys, foreign keys:** e.g. `CONSTRAINT StudentPK PRIMARY KEY (StudNr)`

- **Validation or Business rules** (forretningsregler): e.g. `ON DELETE SET NULL`

Summary: Example Create A Table

```
CREATE TABLE Student
(
  StudNr CHAR( 6 ),
  FodselsNr CHAR( 11 ) UNIQUE NOT NULL,
  Status VARCHAR(15) DEFAULT 'Aktiv',
  CONSTRAINT StudentPK PRIMARY KEY ( StudNr )
)
```

Summary (2/2): Creating and Managing Tables

- Manage table contents with **INSERT**, **UPDATE**, **DELETE**

```
INSERT INTO Student
    (StudNr, FodselsNr)
    VALUES ( 314159, ‘01010133333’ )
UPDATE Student
    SET Status = ‘Aktiv’ WHERE StudNr = 314159
DELETE FROM Student
    WHERE StudNr = 314159
```

- Write **SQL-scripts** to create tables **when developing** a DB
  - **Practice scripting** in MySQL Workbench or phpMyAdmin **now** ☺