Teknisk kursus for RN, 2024

Databaser

Agenda

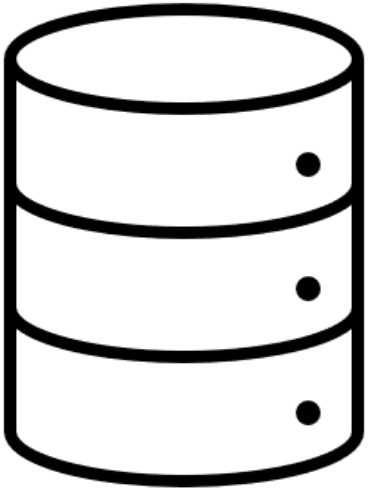
- Hvad er en database?
- Forskellige typer af databaser
- Databaser med tabeller
- Hvordan taler vi med en database?
- Hvordan sikre vi vores data (backup)?
- Hvad med sikkerhed?

Struktureret vs. ikke-struktureret data

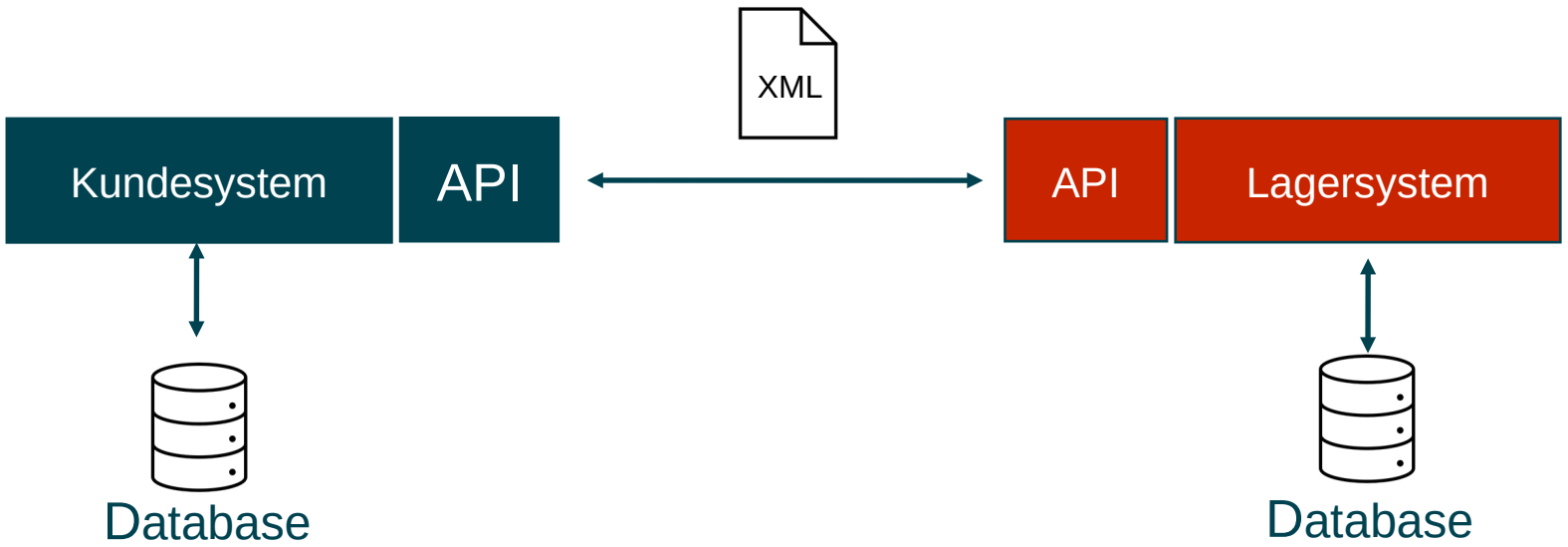
- **Struktureret data** er data, der er organiseret og formateret på en foruddefineret og konsistent måde. Det følger et specifikt skema eller en model, der definerer dataens struktur, herunder typen af data og deres relationer.
  - Kan eks. være XML, JSON, Excel eller ”relationelle databaser”.
- **Ikke-struktureret data** er data, der ikke har en fast eller foruddefineret struktur.
  - Kan eks. være tekstfiler, e-mails, billeder, videoer eller lydfiler.

Hvad er en database?

- En (relationel) database er en struktureret samling af data, der er organiseret, gemt og administreret på en systematisk måde.
- Den fungerer som en elektronisk opbevaringsplads, hvor data kan gemmes, opdateres, slettes og hentes efter behov.
- En (relationel) database muliggør effektiv lagring og håndtering af store mængder data, hvilket giver mulighed for at opretholde datakonsistens, adgangskontrol og støtte til komplekse forespørgsler og transaktioner.
- Typisk består en (relationel) database af en eller flere tabeller, som er struktureret som rækker og kolonner.



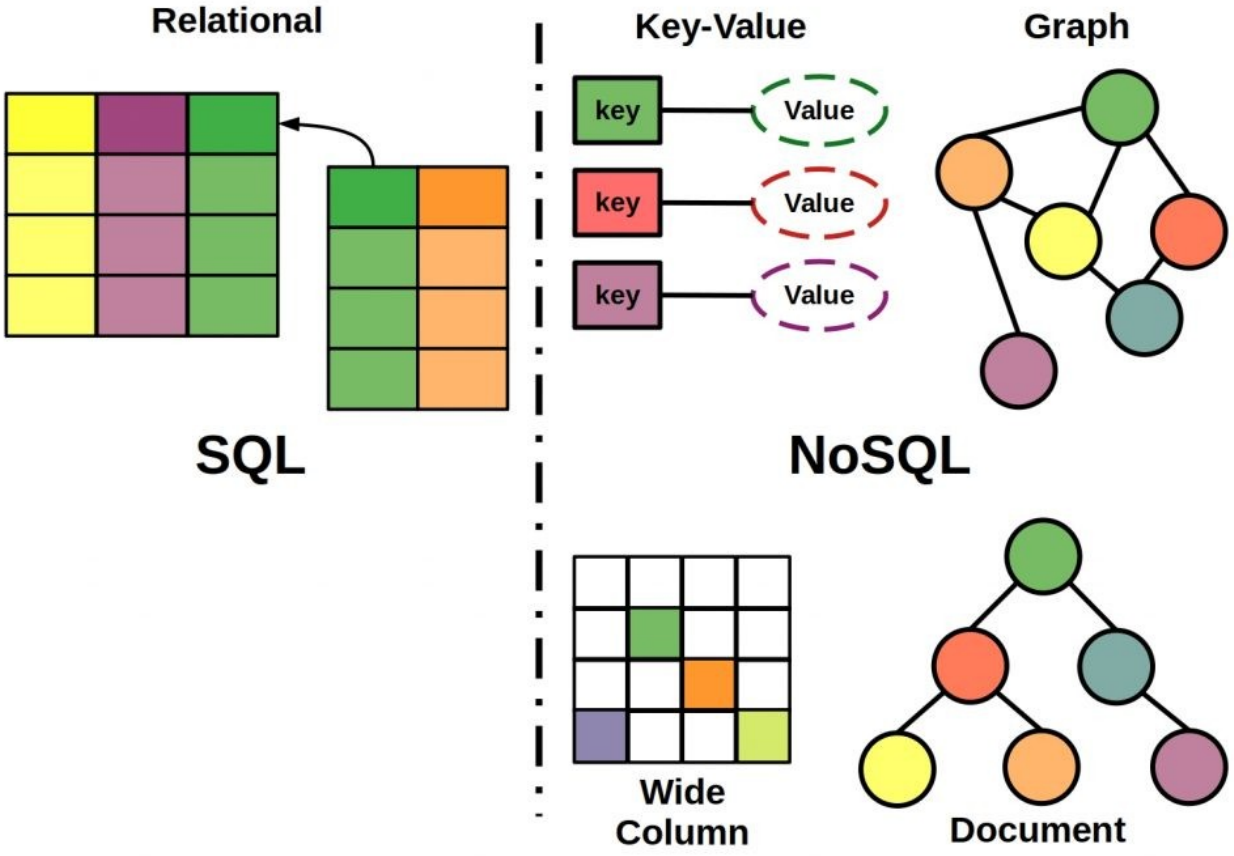
Databaser i arkitekturen



Typer af databaser

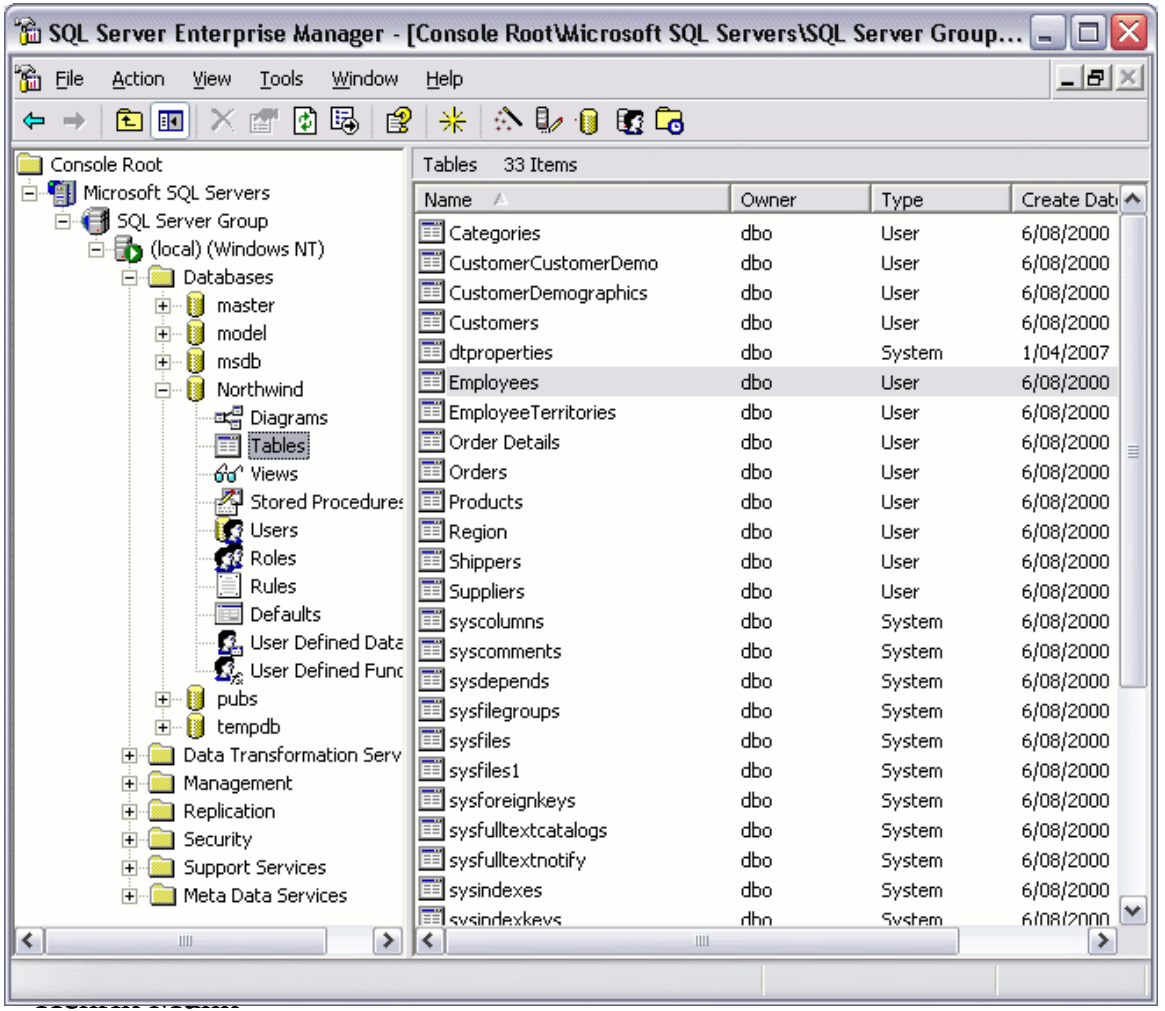
- SQL database
  - Relationel database: En database, der organiserer data i tabeller med rækker og kolonner, hvor relationer mellem tabeller opretholdes gennem nøgler.
- NoSQL databaser
  - Dokumentdatabaser: Disse databaser gemmer og håndterer data i dokumentformat, eks. JSON
  - Key-Value: Disse databaser gemmer og henter data baseret på et nøgle-værdi-paradigme (Key/Value). De er enkle og hurtige og bruges ofte til helt simpel data (eks. caching og session storage).
  - Grafdatabaser: Disse databaser bruger grafstrukturer til at repræsentere og gemme data og understøtter komplekse relationer. De er nyttige til netværksanalyse og sociale netværk.
  - Kolonnefamiliedatabase (Column-family database / wide column database): En database, der organiserer data i kolonnefamilier, hvor hver familie indeholder rækker med kolonner og understøtter stor fleksibilitet i datamodellering
-

Typer af databaser (SQL og NoSQL



Hvad er DBMS?

- En database administreres af et Database Management System (DBMS), der fungerer som en softwareplatform til at administrere og organisere dataene.
- DBMS giver mulighed for at oprette, opdatere, slette og hente data ved hjælp af sprog som SQL (Structured Query Language).



Oracle Database og Microsoft (MS) SQL Database

- MS SQL (database) Server: MS SQL Server er et relationelt databaseadministrationssystem udviklet af Microsoft. Det er et omfattende og robust DBMS, der bruges til at administrere store mængder strukturerede data.
- Oracle Database: Oracle Database er et relationelt databaseadministrationssystem udviklet af Oracle Corporation. Det er kendt for sin høje ydeevne, pålidelighed og skalerbarhed.
- Både MS SQL Server og Oracle Database giver mulighed for at oprette, administrere og få adgang til relationelle databaser ved hjælp af SQL (Structured Query Language) til at udføre forespørgsler og manipulation af data. De har dog forskellige funktionssæt, administrationsegenskaber og licensmodeller

Tabeller i relationelle databaser

- Resultatet af alle forespørgsler til en relationel database er en tabel.
  - Hvis man vil have data fra en relationel database, får man det i en tabel.
  - Et tomt svar fra en relationel database er en tom tabel

Opskriftstabel

Tabel: Opskrift

OpskriftID	Navn	Beskrivelse	Fremgangsmåde	BilledeURL	Sværhedsgrad	Tidsmæssig varighed
1	Pandekager	Lækre pandekager lavet af en simpel dej, perfekte til morgenmad eller dessert.	1. Bland mel, sukker og salt i en skål. \n2. Pisk æggene og mælken sammen i en anden skål. \n3. ...	pandekager.jpg	Let	30 minutter
2	Rabarbertærte	En klassisk tærte med friske rabarber og en sprød smuldredej. Perfekt til sommerdessert eller eftermiddagste.	1. Forvarm ovnen til 180 grader Celsius. \n2. Skær rabarberne i stykker og bland dem med sukkeret. ...	rabarbertaerte.jpg	Medium	1 time

Ingredienstabel

Tabel: Ingrediens

IngrediensID	OpskriftID	Navn	Mængde	Enhed	Kommentar
1	1	Mel	200	g	
2	1	Sukker	2	spsk.	
3	1	Salt	1	knsp.	
4	1	Æg	2	stk.	
5	1	Mælk	500	ml	
6	1	Smør	2	spsk.	Smeltet
7	2	Rabarber	500	g	Skåret i stykker
8	2	Sukker	150	g	
9	2	Hvedemel	250	g	
10	2	Smør	125	g	Koldt og i tern
11	2	Vaniljesukker	1	tsk.	
12	2	Æg	1	stk.	Pisket
13	2	Mel	50	g	Til smuldredejen
14	2	Sukker	50	g	Til smuldredejen

“Queries” og “joins”

- Når man sætter to eller flere tabeller sammen hedder det et “Join”
- Når man sender en besked til en database om at udføre et stykke arbejde, kalder man det en “forespørgsel” (Query).
- Eksempel på forespørgsel: “Giv mig alle ingredienserne til “pandekager” fra “ingrediens” tabellen.
  - …så vi finder ud af, at pandekager har id = 1 fra opskrifttabellen og derfor skal vi have alle de rækker fra ingredienstabellen, hvor opskriftid = 1.

Database med tabeller til opskrifter

Tabel: Opskrift

OpskriftID	Navn	Beskrivelse	Fremgangsmåde	BilledeURL	Sværhedsgrad	Tidsmæssig varighed
1	Pandekager	Lækre pandekager lavet af en	1. Bland mel, sukker og salt i en skål. \n2. Pisk	pandekager.jpg	Let	30 minutter

Join mellem to tabeller via “OpskriftID”

Tabel: Ingrediens

IngrediensID	OpskriftID	Navn	Mængde	Enhed	Kommentar
1	1	Mel	200	g	
2	1	Sukker	2	spsk.	
3	1	Salt	1	knsp.	

SQL (Structured Query Language)

- Sproget til at tale med en relationel database hedder SQL (Structured Query Language).
- Eksemplet fra forespørgslen fra før kunne se ud på følgende måde:

```
SELECT Ingrediens.* FROM Ingrediens
JOIN Opskrift ON Ingrediens.OpskriftID = Opskrift.OpskriftID
WHERE Opskrift.Navn = 'Pandekager';
```

- Eller

```
SELECT * FROM Ingrediens
WHERE OpskriftID IN (
SELECT OpskriftID FROM Opskrift
WHERE Navn = 'Pandekager')
```

Grundlæggende SQL - SELECT

- SELECT: Bruges til at hente data fra tabeller.

```sql
-- Hent alle opskrifter
SELECT * FROM Opskrift;

-- Hent navnet på ingredienserne til pandekager
SELECT Navn FROM Ingrediens WHERE OpskriftID = 1;
```

Grundlæggende SQL - INSERT

- INSERT INTO: Bruges til at indsætte nye rækker i tabeller.

```sql
-- Indsæt en ny opskrift
INSERT INTO Opskrift (Navn, Beskrivelse) VALUES ('Kage', 'En lækker kageopskrift');

-- Indsæt en ny ingrediens til pandekager
INSERT INTO Ingrediens (OpskriftID, Navn, Mængde, Enhed) VALUES (1, 'Mel', 200, 'g');
```

Grundlæggende SQL - UPDATE

- UPDATE: Bruges til at opdatere eksisterende rækker i tabeller.

```sql
-- Opdatér beskrivelsen af en opskrift
UPDATE Opskrift SET Beskrivelse = 'En fantastisk kageopskrift' WHERE OpskriftID = 1;

-- Opdatér mængden af en ingrediens
UPDATE Ingrediens SET Mængde = 250 WHERE IngrediensID = 1;
```

Grundlæggende SQL - DELETE

- DELETE FROM: Bruges til at slette rækker fra tabeller.

```sql
-- Slet en opskrift
DELETE FROM Opskrift WHERE OpskriftID = 1;

-- Slet alle ingredienser til pandekager
DELETE FROM Ingrediens WHERE OpskriftID = 1;
```

Grundlæggende SQL – Create table

- Create table: Bruges til at oprette en tabel.

```sql
-- Opret en ny tabel kaldet Opskrift
CREATE TABLE Opskrift (
    OpskriftID INT PRIMARY KEY,
    Navn VARCHAR(100),
    Beskrivelse VARCHAR(500),
    Fremgangsmåde VARCHAR(1000),
    BilledeURL VARCHAR(200),
    Sværhedsgrad VARCHAR(20),
    TidsmaessigVarighed INT
);
```

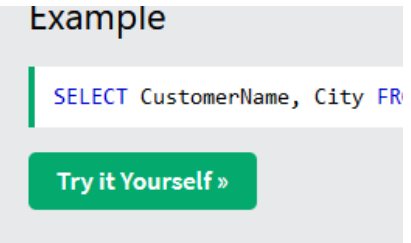
Grundlæggende SQL – Drop table

- Drop table: Bruges til at slette en tabel.

```sql
-- Slet opskriftstabellen
DROP TABLE Opskrift;
```

Opgave – SQL Select

- Afprøv “SQL SELECT” på følgende side
  - https://www.w3schools.com/sql/default.asp
- Klik på “SQL Select” i menuen til venstre og find herefter knappen “Try it Yourself” lidt nede på siden:
- Få vist alle kunderne i tabellen
  - Select * from Customers
- Få vist den kunde, der har CustomerName "Vaffeljernet"
  - Select * from Customers where…..
- Få vist alle de kunder, der bor i “London”
- Få vist alle de danske kunder



Sikring/backup af data

- En transaktionslog i en database bruges til at registrere og opretholde en log over alle ændringer, der foretages på dataene i en database.
- Formålet med transaktionsloggen er at sikre, at databasen forbliver i en konsistent tilstand og kan gendannes til en tidligere tilstand, hvis der opstår fejl eller tab af data.
- Transaktionsloggen giver også mulighed for at identificere problemer eller fejl i databasen.
- Når en transaktion udføres mod databasen, skrives de pågældende ændringer først i transaktionsloggen og derefter anvendes ændringerne på selve databasen.
  - En transaktion er alle forespørgsler til en database – eks. ”select * from opskrift”

Sikring/backup af data

- Når det kommer til backup af en Microsoft SQL Server-database, er transaktionsloggen en vigtig del af backup-processen.
- En fuld databasebackup indeholder normalt både selve databasen og transaktionsloggen.
  - Ved at inkludere transaktionsloggen i backup'en kan man gendanne databasen til et vilkårligt tidspunkt lige før backup'en blev taget.

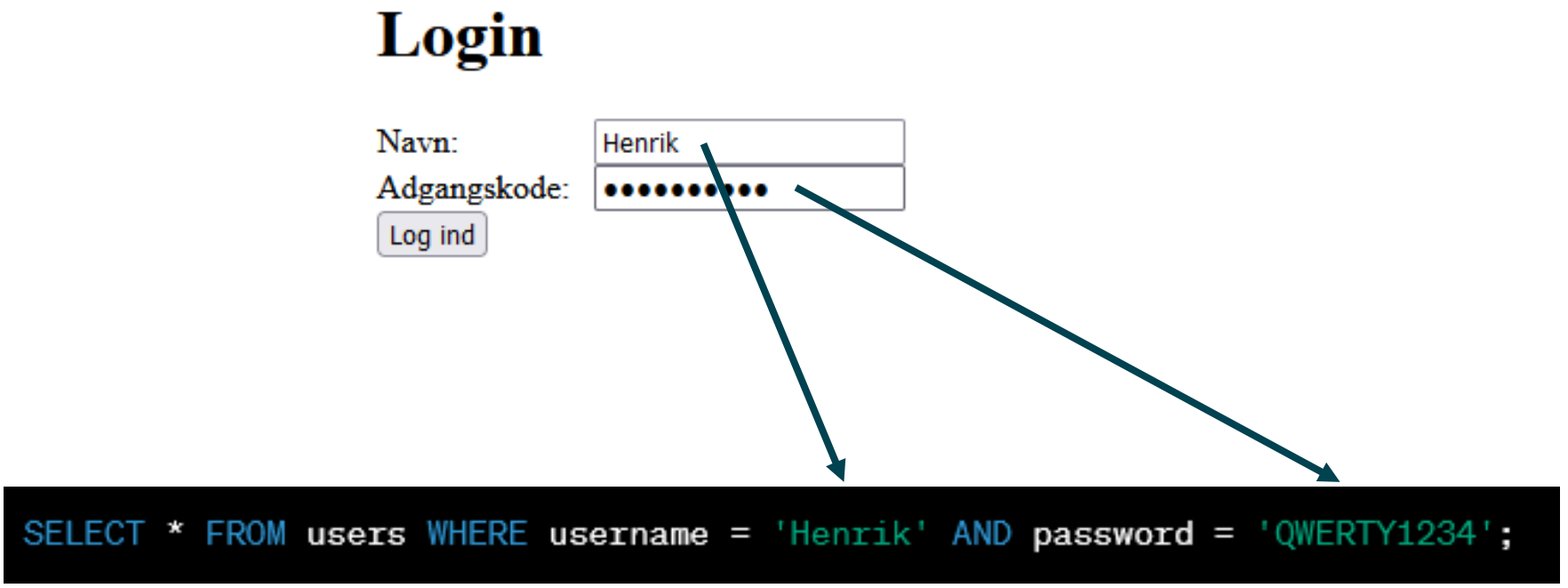
Sikring/backup af data

- Der er forskellige typer af backup, der kan udføres på MS SQL Server:
  - Fuldbackup: En fuldbackup inkluderer hele databasen samt transaktionsloggen. Denne type backup giver den mest komplette genopretning af databasen.
  - Differensbackup: En differensbackup inkluderer kun ændringerne, der er foretaget siden den sidste fuldbackup. Det vil sige ændringer, der er registreret i transaktionsloggen siden den sidste fuldbackup.
  - Transaktionslogbackup: En transaktionslogbackup inkluderer kun ændringerne i transaktionsloggen siden den sidste transaktionslogbackup eller fuld backup.
- Ved at udføre regelmæssige backup'er af både databasen og transaktionsloggen sikres dataintegriteten og muligheden for at gendanne databasen i tilfælde af fejl eller tab af data.
- ***"Hvis du ikke har testet gendannelsen af din backup, har du ingen backup!"***

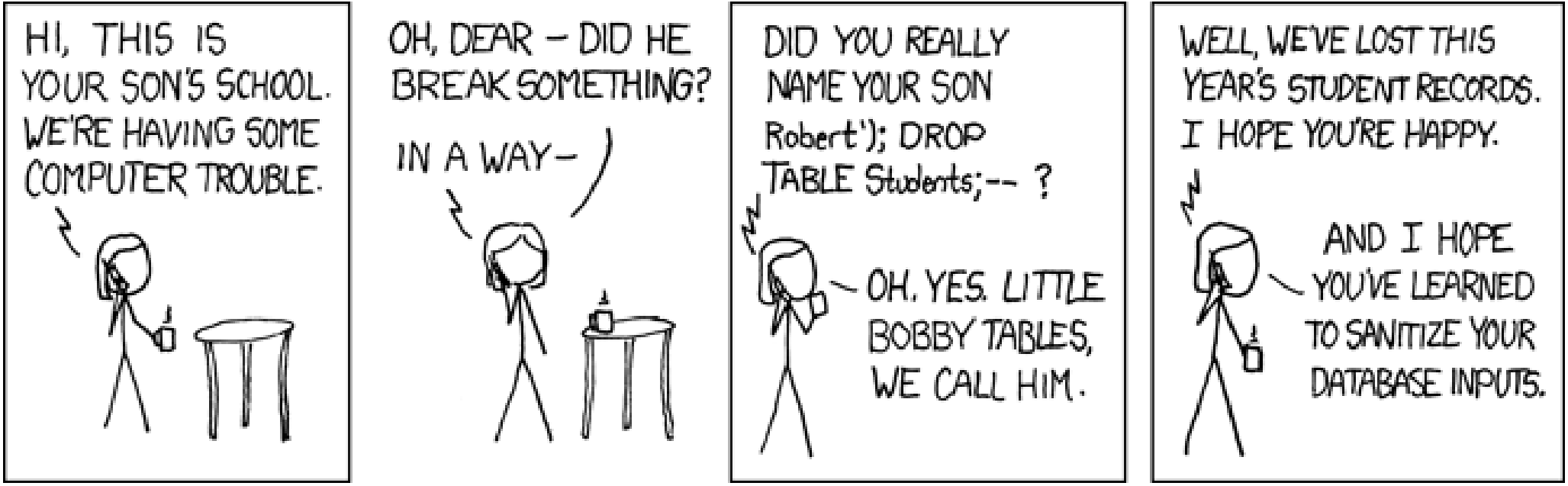
Hackerangreb mod databaser – SQL injection

- Et klassisk angreb mod databaser er et “SQL Injection” angreb.
- Det går ud på, at en angriber forsøger at indsætte (skadelig) SQL-kode i inputfelter, der er beregnet til at interagere med en database. Eksempelvis ved at indsætte SQL-kode i en felt, der er beregnet til et brugernavn. Eller eks. i et søgefelt, hvor man vi finde en bestemt opskrift.
- Det er et problem, når en applikation ikke korrekt validerer og behandler brugerindtastede data.
- Hvis applikationen ikke beskytter sig mod dette, vil den skadelige SQL-kode blive udført direkte på databasen.
- Konsekvenserne af en vellykket SQL injection kan være alvorlige. Angriberen kan opnå uautoriseret adgang til databasen, ændre eller slette data, afsløre fortrolige oplysninger eller endda overtage kontrol over applikationen.

SQL injection



Database jokes – Bobby tables



Opgave – SQL injection

- Eksperimenter med manipulation af SQL til login
  - https://ashkiani.github.io/sql-injection-playground/
- Log nu ind på følgende webside vha. et SQL injection angreb
  - https://demo.owasp-juice.shop/#/
  - (Prøv evt. også https://www.hacksplaining.com/lessons/sql-injection)
- Når du er logget ind, hvad har du så rettigheder til at gøre i Juice shoppen?

Opgave hint til Juiceshop

- Den SQL kode som bliver sendt afsted til databasen ser ud på følgende måde:
  - SELECT * FROM Users WHERE email = ‘Henrik’ AND password = ‘098f6bcd4621d373cade4e832627b4f6’ AND deletedAt IS NULL
- Koden forespørger på både brugernavn og password. Kan du indsætte noget kode i feltet til brugernavnet, som gør udtrykket korrekt og udkommenterer resten af koden?
  - Man udkommenterer ved at anvende to streger ”--”
  - Man viser en tekst ved at anvende ‘ (Eksempelvis ‘Henrik’)
  - Det står i SQL’en, at brugernavn ”AND” password skal findes. Kan man bruge koden ”OR” i stedet for ”AND”?
  - Udtrykket 1 = 1 er altid korrekt :)

Opgaveløsning til Juiceshop

- Anvendt følgende kode i feltet til brugernavn (password feltet må ikke være tomt):
  - ‘ or 1 = 1 --
- Dvs. den SQL kode, der bliver sendt afsted til databasen kommer til at se ud på følgende måde:
  - SELECT * FROM Users WHERE email = ‘’ or 1 = 1 -- AND password = ‘098f6bcd4621d373cade4e832627b4f6’ AND deletedAt IS NULL
  - Efter ”1 = 1” udkommenterer vi resten af SQL’en vha.”--”, så denne del bliver der slet ikke kigget på.
  - Den kan sandsynligvis ikke finde en bruger med ”tomt” brugernavn, men udtrykket 1 = 1 er sandt. Så forespørgslen returnerer den første bruger i tabellen. Det er så til vores held, at det er administratoren.

Referencer

- Center for cybersikkerhed: Logning – en del af et godt cyberforsvar
  - https://www.cfcs.dk/globalassets/cfcs/dokumenter/vejledninger/vejledning-logning-2023.pdf
- Justitsministeriet: nye regler for logning
  - https://www.justitsministeriet.dk/pressemeddelelse/flertal-i-folketinget-vedtager-nye-regler-for-logning/
- Logging Cheat Sheet
  - https://cheatsheetseries.owasp.org/cheatsheets/Logging_Cheat_Sheet.html
- Sundhedsstyrelsen: beskyttelse af sundhedsdata
  - https://sundhedsdatastyrelsen.dk/da/borger/beskyttelse_af_sundhedsdata