SQL Rapport

# Forside



Magnus Rasch

Data og kommunikation

# Indholdsfortegnelse

[Forside 1](#_Toc94853863)

[Indholdsfortegnelse 2](#_Toc94853864)

[Indledning 3](#_Toc94853865)

[Diagram over Skole databasen 3](#_Toc94853866)

[Koden af databasen 4](#_Toc94853867)

[Database tabel kode 4](#_Toc94853868)

[Data til databasen 5](#_Toc94853869)

[SELECT 1 koden 8](#_Toc94853870)

[SELECT 2 koden 9](#_Toc94853871)

[SELECT 3 koden 10](#_Toc94853872)

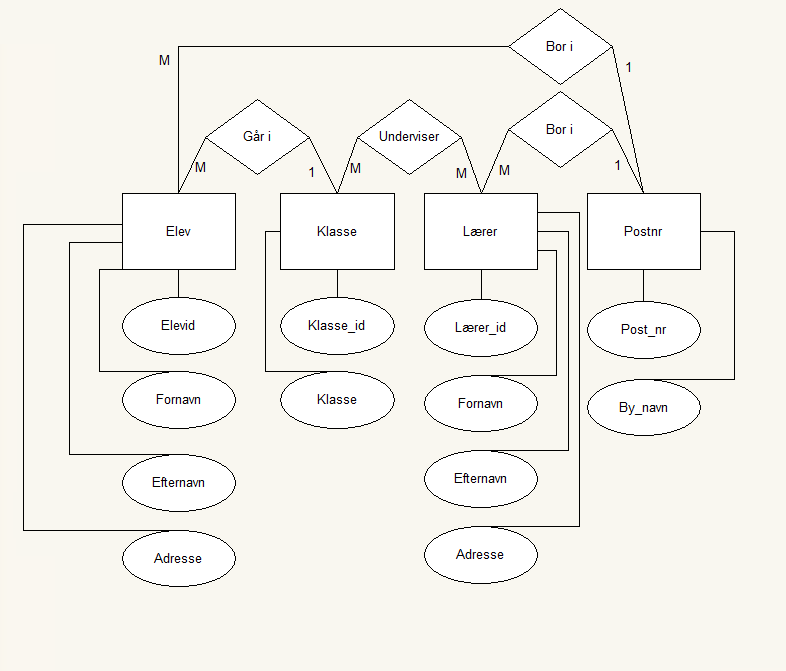
[Konklusion 11](#_Toc94853873)

[Logbog for SQL opgave 11](#_Toc94853874)

# Indledning

Jeg fik til opgave at opsætte en database i en server på en virtuel maskine med nogle forudbestemte redskaber som man skulle bruge for at fuldføre opgaven. HyperV var det redskab som vi skulle bruge til at oprette en virtuel maskine der kan køre serveren. Databasen bliver lavet et redskab der hedder SQL som er et system der er lavet til at oprette tabeller, lagre information og hive information frem til brug.

# Diagram over Skole databasen



Her på skærmbilledet ovenfor kan man se et diagram over databasen for skole. Jeg har markeret hvad der er primary keys og hvad der er foreign keys. Primary Key bliver brugt til at identificere hvert arkiv i en tabel så man nemmere kan finde hvad man søger efter. En Primary Key skal også indeholde en unik værdi, det kan ikke være ”NULL” værdier.

# Koden af databasen

## Database tabel kode

Nedenunder kan man se koden til oprettelsen af de forskellige tabeller som skal have information gemt inde dem til lagring af de forskellige postnumre, elever, lærer og klasser som skolen har. Det er også her at databasen bliver lavet og navngivet på første linje, og så den næste linje er til at bruge den. Selve kodnings sproget er lige til fordi meget af indeks funktionerne kan man nogenlunde forstå ved at læse dem. Det er også her hvor man angiver hvad der er ”PRIMARY KEY” og hvad der er ”FOREIGN KEY” så man normalisere sin database så det er nemmere at gemme og søge efter information.

CREATE DATABASE Elever\_og\_Lærer;

USE Elever\_og\_Lærer;

CREATE TABLE Klasse

(

Klasse\_id int NOT NULL IDENTITY (1,1) PRIMARY KEY,

Klasse varchar(20)

);

CREATE TABLE post\_nr\_by

(

post\_nr int NOT NULL PRIMARY KEY,

by\_navn varchar(30),

);

CREATE TABLE Underviser

(

Underviser\_id varchar(5) NOT NULL PRIMARY KEY,

);

CREATE TABLE Elev

(

elev\_id int IDENTITY (1,1) NOT NULL PRIMARY KEY,

fornavn varchar(30),

efternavn varchar(30),

adresse varchar(40),

post\_nr int NOT NULL,

Klasse\_id int NOT NULL,

FOREIGN KEY (post\_nr) REFERENCES post\_nr\_by(post\_nr),

FOREIGN KEY (Klasse\_id) REFERENCES Klasse(Klasse\_id)

);

CREATE TABLE Lærer

(

Lærer\_id int IDENTITY (1,1) NOT NULL PRIMARY KEY,

fornavn varchar(30),

efternavn varchar(30),

adresse varchar(40),

post\_nr int NOT NULL,

Klasse\_id int NOT NULL

Underviser\_id varchar(5)

FOREIGN KEY (Underviser\_id) REFERENCES Underviser(Underviser\_id),

FOREIGN KEY (post\_nr) REFERENCES post\_nr\_by(post\_nr),

FOREIGN KEY (Klasse\_id) REFERENCES Klasse(Klasse\_id)

);

”varchar(30)” bliver brugt til at bestemme hvor meget informationen skal fylde i størrelsen, så man sætter derfor et maksimum for hvor mange tegn som kan man skrive in de forskellige rækker i tabellerne. ”NOT NULL” betyder at der skal skrives information ind ellers vil der være en fejl. ”REFERENCES” bliver brugt til at refererer hvor henne ”FOREGIN KEY” bliver koblet hen til så tabeller giver mening når gerne vil vide for eksempel hvilken adresse hører til postnummer 2650.

## Data til databasen

USE Elever\_og\_Lærer;

INSERT INTO post\_nr\_by (post\_nr, by\_navn)

VALUES (2650, 'Hvidovre');

INSERT INTO post\_nr\_by (post\_nr, by\_navn)

VALUES (2300, 'København S');

INSERT INTO post\_nr\_by (post\_nr, by\_navn)

VALUES (2500, 'Valby');

INSERT INTO post\_nr\_by (post\_nr, by\_navn)

VALUES (2610, 'Rødovre');

INSERT INTO post\_nr\_by (post\_nr, by\_navn)

VALUES (3650, 'Ølstykke');

INSERT INTO post\_nr\_by (post\_nr, by\_navn)

VALUES (2830, 'Virum');

INSERT INTO post\_nr\_by (post\_nr, by\_navn)

VALUES (2770, 'Kastrup');

INSERT INTO post\_nr\_by (post\_nr, by\_navn)

VALUES (1824, 'Frederiksberg C');

INSERT INTO post\_nr\_by (post\_nr, by\_navn)

VALUES (2740, 'Skovlunde');

INSERT INTO post\_nr\_by (post\_nr, by\_navn)

VALUES (2750, 'Ballerup');

Ovenfor kan man se at jeg har indtastet en masse kode som bliver til information til min database. Man kan se i hver linje hvor der står ”INSERT INTO” at jeg kan vælge hvilken tabel som jeg vil sætte den nye information ind og med ”VALUES” kan jeg skrive ind den information som jeg gerne vil have lagret. Man skriver information i rækkefølge med hvordan tabellen virker så det er meget lige til hvad man skal skrive efter hinanden og hvor der skal stå.

INSERT INTO Klasse (Klasse)

VALUES ('1A');

INSERT INTO Klasse (Klasse)

VALUES ('9B');

INSERT INTO Klasse (Klasse)

VALUES ('4D');

INSERT INTO Elev (fornavn, efternavn, adresse, post\_nr, Klasse\_id)

VALUES ('Bo', 'Andersen', 'Gammel Byvej 12', 2650, 1);

INSERT INTO Elev (fornavn, efternavn, adresse, post\_nr, Klasse\_id)

VALUES ('Frederikke', 'Hansen', 'Amager Boulevard 5', 2300, 2);

INSERT INTO Elev (fornavn, efternavn, adresse, post\_nr, Klasse\_id)

VALUES ('Jens', 'Mikkelsen', 'Lily Brobergs Vej 17', 2500, 3);

INSERT INTO Elev (fornavn, efternavn, adresse, post\_nr, Klasse\_id)

VALUES ('Philip', 'Mortensen', 'Brunevang 90', 2610, 1);

INSERT INTO Elev (fornavn, efternavn, adresse, post\_nr, Klasse\_id)

VALUES ('Kasper', 'Frederiksen', 'Bryggertorvet 32', 3650, 2);

INSERT INTO Elev (fornavn, efternavn, adresse, post\_nr, Klasse\_id)

VALUES ('Milla', 'Jørgensen', 'Virum Torv 25', 2830, 3);

INSERT INTO Elev (fornavn, efternavn, adresse, post\_nr, Klasse\_id)

VALUES ('Fie', 'Kudsen', 'Allen 85', 2770, 1);

INSERT INTO Elev (fornavn, efternavn, adresse, post\_nr, Klasse\_id)

VALUES ('Henrik', 'Madsen', 'Lily Brobergs Vej 53', 2500, 2);

INSERT INTO Lærer (fornavn, efternavn, adresse, post\_nr, Klasse\_id, Underviser\_id)

VALUES ('Tom', 'It', 'Sankt Thomas Alle 3', 1824, 1, 'AB');

INSERT INTO Lærer (fornavn, efternavn, adresse, post\_nr, Klasse\_id, Underviser\_id)

VALUES ('Lars', 'Henriksen', 'Nissedalen 76', 2740, 2, 'AC');

INSERT INTO Lærer (fornavn, efternavn, adresse, post\_nr, Klasse\_id, Underviser\_id)

VALUES ('Mia', 'Hansen', 'Østervej 16', 2750, 3, 'B');

INSERT INTO Underviser (Underviser\_id)

VALUES ('A');

INSERT INTO Underviser (Underviser\_id)

VALUES ('B');

INSERT INTO Underviser (Underviser\_id)

VALUES ('C');

INSERT INTO Underviser (Underviser\_id)

VALUES ('AB');

INSERT INTO Underviser (Underviser\_id)

VALUES ('AC');

INSERT INTO Underviser (Underviser\_id)

VALUES ('BC');

INSERT INTO Underviser (Underviser\_id)

VALUES ('ABC');

Ovenfor kan man se resten af informationen som skal ind i databasen. I den forrige del af koden så indtastede jeg information til postnumre og på denne del af koden indtaster jeg information til elever og lærerne som er en del af den skole som man laver en database over. Man kan se at der skal stå lidt flere informationer i de her tabeller men det er fordi at der er mere information som skal lagres så det er ligesom det med postnumrene, men bare om personer i stedet for. Der er også skrevet adressen og postnumre ind, og postnumrene vil fungere som ”FOREIGN KEYS” til den tabel som har oversigten over postnumre som er ”PRIMARY KEY” så man kan finde ud af hvilken by postnumrene hører til.

”Underviser” tabellen er til for at tildele klasser som lærerne skal under vise. Hvert bogstav står for deres eget klasseværelse. 1A er blevet tildelt bogstavet A, 9B er tildelt B og 4D er tildelt C. Hvis en lærer har mere end et bogstav i deres Underviser\_id, så betyder det at de skal undervise i flere klasser ad gangen.

## SELECT 1 koden

USE Elever\_og\_Lærer

SELECT Elev.fornavn, Elev.efternavn, Elev.adresse, post\_nr\_by.by\_navn, Elev.post\_nr

FROM Elev

INNER JOIN post\_nr\_by ON Elev.post\_nr = post\_nr\_by.post\_nr

ORDER BY post\_nr asc;



Ovenfor kan man se koden til den SELECT som skal finde noget information frem og vise det på skærmen. Der også en ”JOIN” funktion indblandet så den vil tage fra to tabeller og vise den information frem så man valgt i ”SELECT” linjen. Hvis der er information som over lapper så viser den kun informationen en gang hvilket er ideelt. Efter koden er løbet igennem så vil man få vist Fornavn, efternavn, adresse, By navn og Postnummer på skærmen. Rækkefølgen går efter det højeste tal og op til det laveste.

## SELECT 2 koden

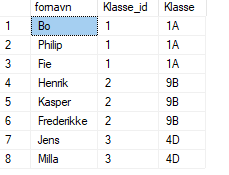
USE Elever\_og\_Lærer

SELECT Elev.fornavn, Elev.Klasse\_id, Klasse.Klasse

FROM Elev

FULL JOIN Klasse ON Elev.Klasse\_id = Klasse.Klasse\_id

ORDER BY Klasse\_id ASC;



Ovenfor kan man se den anden SELECT og den ville vise fornavn, klasse ID og klasse.

## SELECT 3 koden

USE Elever\_og\_Lærer

SELECT Lærer.fornavn, Lærer.efternavn, Lærer.Lærer\_id, Klasse.Klasse

FROM Lærer

RIGHT JOIN Klasse ON Lærer.Klasse\_id = Klasse.Klasse\_id



Ovenfor kan man se at der er en SELECT til at hente informationer om en lærer så fornavn, efternavn og den klasse de hører til vil blive vist på skærmen.

# Konklusion

Det jeg fik ud af opgaven det er læring til hvad SQL er og hvad det kan bruges til, jeg har en bedre ide om hvordan databaser skal se ud og hvordan de er organiseret. Der er selvfølgelig en masse mere jeg kan få lært om SQL og databaser i det hele taget, men nu er jeg i hvert fald blevet mere klog på det end da jeg startede på opgaven.

# Logbog for SQL opgave

17-01-2022: Læste nogle dokumenter angående hvad SQL er og hvordan det fungere.

18-01-2022: Var ikke tilstede

19-01-2022: Har lært en del om hvordan koden ser ud med SQL og hvordan opsætningen virker, har lavet selve skelettet til databasen nu. Jeg mangler bare selve informationen som skal blive sat ind.

20-01-2022: Gik I gang med at skrive kode til SQL databasen, løb ind i nogle problemer som jeg småt er i gang med at løse.

27-01-2022: Fik færdiggjort min database og fik også lavet tre SELECT som kan vise noget frem fra databasen.

31-01-2022: Jeg fik lavet et diagram over hvordan databasen ville se ud.

1-02-2022: Jeg blev færdig med min rapport og er klar til at aflevere min opgave. UPDATE Fik noget feedback så jeg er gået i gang med at rette de fejl som jeg havde i koden og rapporten.

2-02-2022: Lavede en ny tabel til min database og laver derefter et skema til klasse fordelingen mellem lærerne.

4-02-2022: Fik lavede nogle af de sidste rettelser, gør klar til at aflevere opgaven.