# MyAdmin og SQL

I timerne i dag skal vi arbejde med MyAdmin og SQL og bliver mere bekendt med MyAdmin og hvordan man bruger en MySQL database. Den minder en del om SQLite som vi allerede har set på, og sproget er jo det samme (SQL).

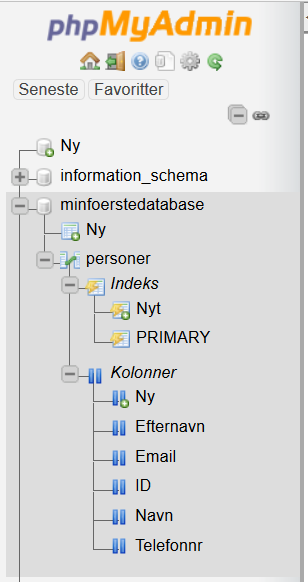
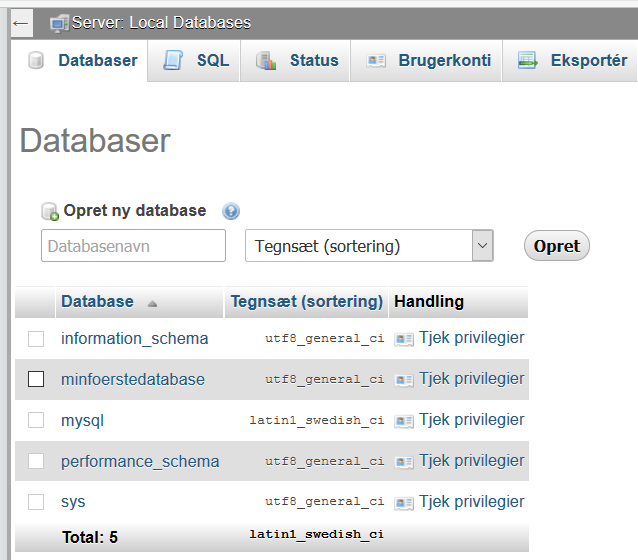
Brug w3schools til at læse og evt. genopfriske din viden om SQL: <https://www.w3schools.com/sql/default.asp>

Nedenfor er en reminder omkring de fire mest anvendte SQL kommandoer:

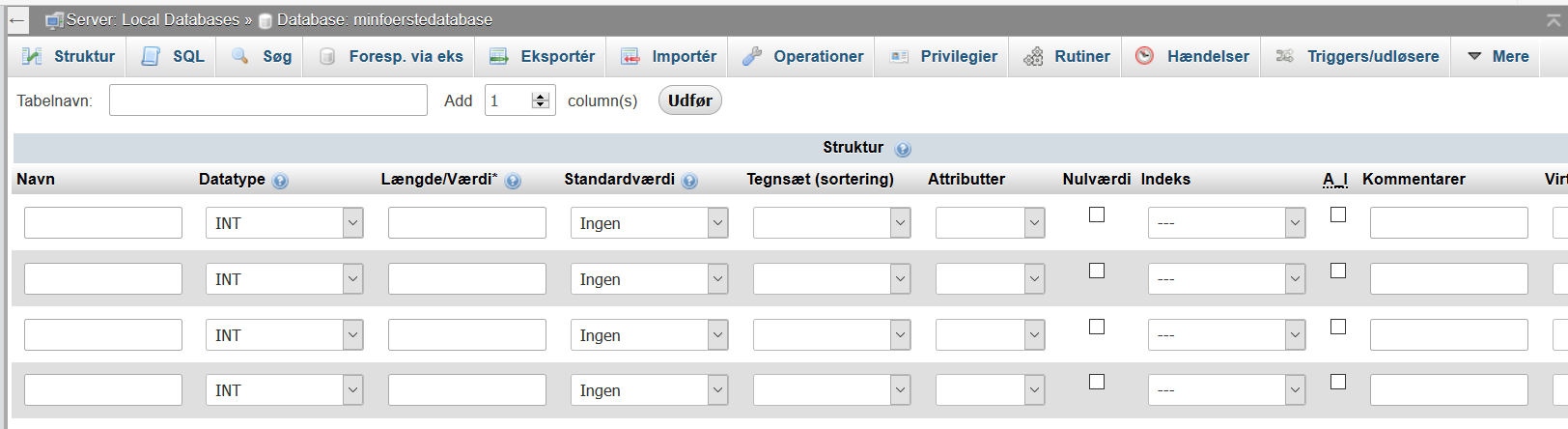
|  |
| --- |
| SELECT \* FROM *tabel\_navn*; Henter alt data fra tabellen med navnet *tabel\_navn*.  INSERT INTO *tabel\_navn* (*kolonne1\_navn*, *kolonne2\_navn*) VALUES (*værdi\_kolonne1*, *værdi\_kolonne1*);  Sætter data ind i tabellen *tabel\_navn.* UPDATE *tabel\_navn* SET *kolonne1\_navn* = *værdi\_kolonne1* WHERE *betingelse*;  Opdaterer kolonne1 værdien hvis betingelsen er opfyldt.  DELETE FROM *tabel\_navn* WHERE *betingelse*;  Sletter data rækker hvis betingelsen er opfyldt. |

**Opgaver:**

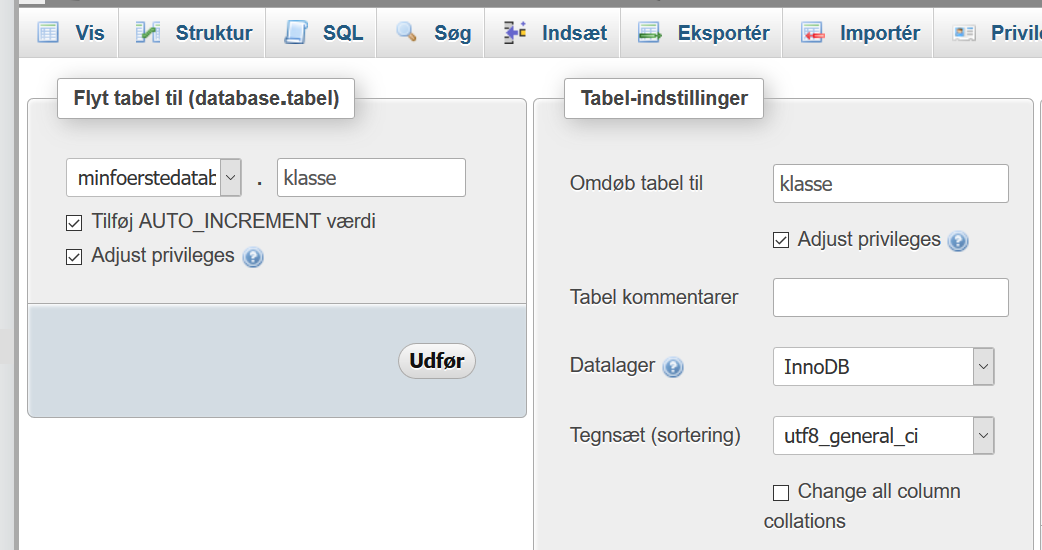
Start phpMyAdmin gennem WAMP serveren. Lav en ny database enten gennem træmenuen til venstre (klik der hvor der står Ny) eller under fanebladet ”database”:



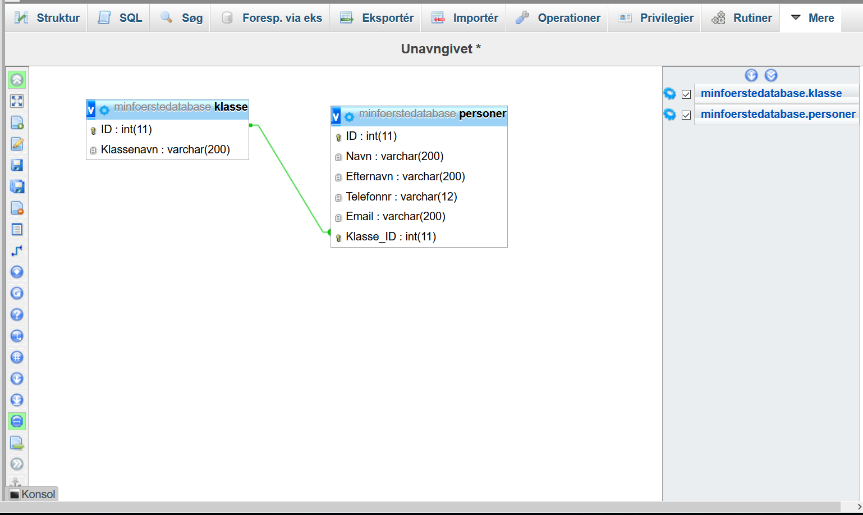
Opret to tabeller. Det gøres gennem træmenuen under den nye database eller ved at klikke på databasen og bruge topmenupunktet ”Struktur”. Nedenfor er vist hvordan interfacet ser ud. Sørg for at den første kolonne indeholder navnet ID, typen int, at checkboksen A.I. er klikket til, sæt indeks på den og at kolonnen er sat til ”primær” således at kolonnen kan bruges som primær nøgle.



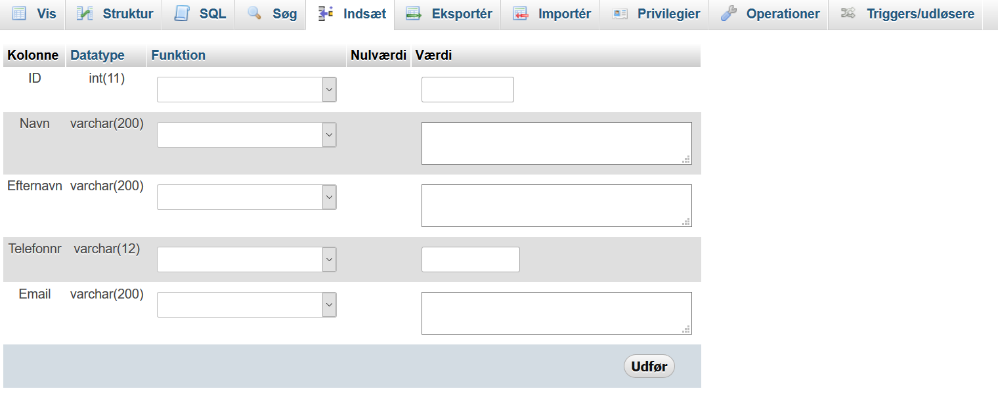
I den ene tabel skal der desuden laves en kolonne (f.eks. ID\_fremmednøgle) der er af typen int som kan bruges som fremmednøgle til den anden tabel som også skal have indeks på.   
  
For at I kan lave relationer mellem tabellerne skal I ind på begge tabeller og ændre ”Datalager” til InnoDB.

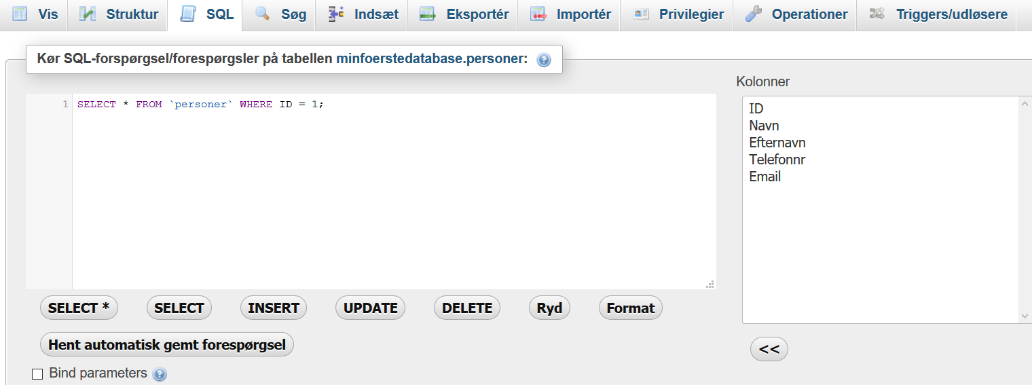


Nu kan I lave relationer og det gøres lettest gennem designeren i MyAdmin.

Klik på databasen i højremenuen og i topmenuen vælges ”Designer” som er helt ude til højre (muligvis under ”mere”). Start designeren, der vises et næsten tomt arbejdsvindue. Klik i den lille menu til venstre på det øverste ikon som hedder ”vis/skjul tabelliste”. Derved skulle de to tabeller vises til højre. Klik dem begge af så de vises på arbejdsbordet og vælg fra den lille menu ”Opret relation”. Det er nu muligt at klikke på ID kolonne i den ene tabel og dernæst på ”ID\_fremmednøglen” i den anden.

På den måde oprettes en relation mellem de to tabeller således at tabellen med ”ID\_fremmednøglen” referere til den anden tabel. Det kunne f.eks. være to tabeller ”personer” og ”klasser” hvor fremmednøglen ligger i ”personer” og peger på ”klasser” tabellen som indeholder alle klasserne på en skole.

I skal nu lægge data i jeres tabel. Start med at læg data i den tabel der har IKKE har fremmednøglen. Klik i højremenuen på ”tabeller” og vælg dernæst ”Indsæt” ud for den rette tabel.

  
Alternativ kan man vælge SQL i topmenuen og selv skrive sine Insert statements (se syntax ovenfor).

Prøv helt begge dele for at blive fortrolig med interfacet samt lære lidt SQL.

Hvis I er i tvivl om data er komme ind i tabellen kan I altid lave en Select gennem topmenupunktet SQL som vist tidligere.

Sørg for at der er data i begge tabeller og at der er referencer fra fremmednøgle kolonnen over til ID kolonnen (primær nøglen) i den anden tabel.

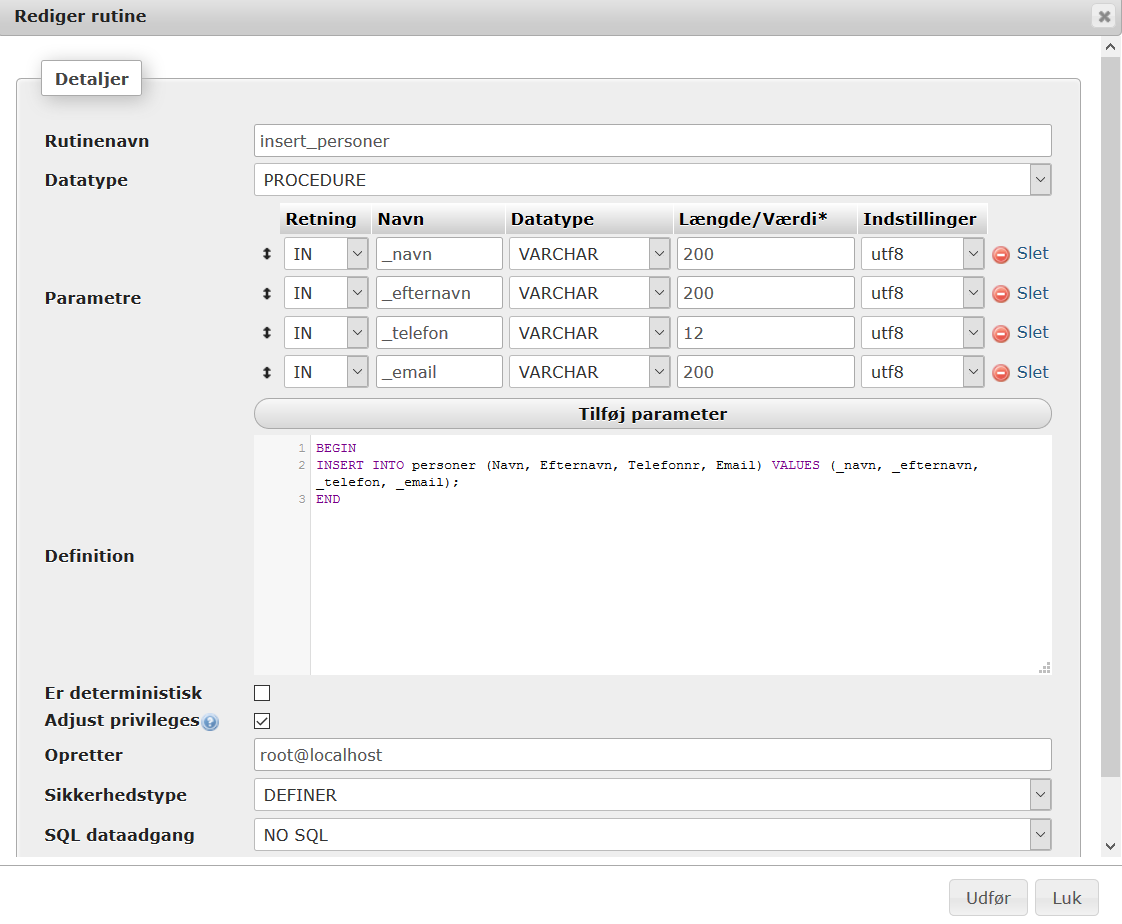
Prøv nu de tre andre almindelige SQL kommandoer af i SQL vinduet. Prøv at Update data, prøv at Select data og prøv at Delete data, husk at I her får brug for WHERE betingelsen. For at se eksempler på hvordan man gør bruge w3school sitet. Gem de SQL koder i skriver så I har dem til senere (evt. i notepad++ el. lign.).

I skal nu vælge data fra mere end en tabel samtidig. Vi har arbejdet med det tidligere så forhåbentlig kan I huske det og ellers kan I få det genopfrisket her. Til at hente data fra flere tabeller benytter man en inner join som vist i denne SQL:

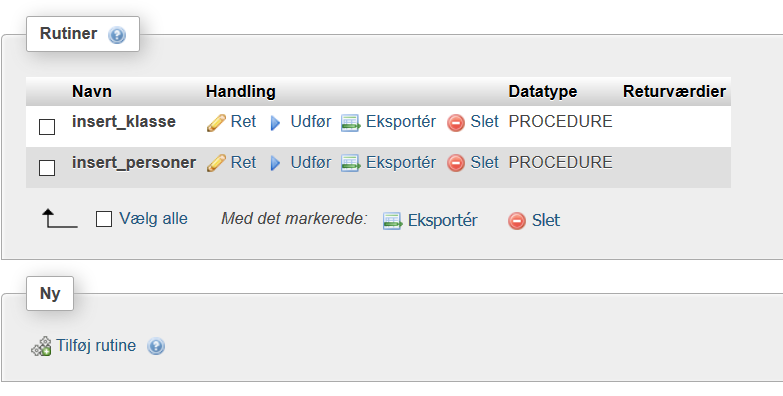
SELECT p.ID, p.Navn, p.Efternavn, P.Email, K.Klassenavn FROM personer p INNER JOIN klasse k ON p.Klasse\_ID = k.ID;

Inner join er en måde at binde to eller flere tabeller sammen således at fremmednøglen i den ene tabel skal svare til primær nøglen i den anden. Læg også mærke til at man kan bruge alias i stedet for tabel navn som ovenfor hvor p er alias for personer (tabel) og k for klasse. På den måde behøves man ikke altid at skrive hele tabelnavnet. *Fordelen ved IKKE at bruge stjerne (SELECT \* FROM…) er at hvis der tilføjes flere kolonner til databasen kan det få fatale konsekvenser hvis man forventer at data kun har et bestemt antal kolonner eller kommer i en bestemt rækkefølge.*

Ofte vil det være bekvemt hvis man ofte skal bruge en bestemt kombination af data fra flere tabeller at lave et ”View”. Det opfører sig mere eller mindre som en ny tabel som blot indeholder data fra rigtige tabeller. Et View bruges f.eks. ofte til at lave større data udtræk fra databasen som bruges i statusoversigter m.m. Desværre har det ikke umiddelbart været muligt at få myAdmin til at acceptere et View med mere end en tabel i men prøv alligevel at lav et View for at prøve det. Linket til oprettelse af et View er lidt hemmeligt i det man skal først klikke på en tabel i træmenuen og dernæst er det muligt at lave et View. Prøv først med den standard SQL som myAdmin selv foreslår, blot for at få det til at virke.

Når man kommunikere med databasen (med SQL) kan man godt sende ens SQL forespørgsler fra PHP som f.eks. SELECT \* FROM personer;. Alternativt er det dog bedre at bruge storeprocedure eller ”Rutiner” som det hedder i myAdmin (i træmenuen hedder det procedure). En procedure er som en funktion der blot indeholder SQL i stedet for et andet sprog. Man kan vælge input parameter til storeproceduren og output parameter og i selve SQL delen kan man henvise til input og output. På den måde kan man skrive en række procedure som kan modtage data som f.eks. skal indsættes i en tabel som vist nedenfor:  
  


Skriv jeres egne procedure med Insert og Select og test evt. først SQL koden v.h.a. SQL menuen (eller brug nogle af de SQL koder I skrev tidligere). Når de er færdige kan I køre dem fra ”Rutiner” menuen:



Klik på Udfør for at køre en rutine og udfyld de felter som vises.

De sidste ting i myAdmin er **Hændelser** og **Triggers/udløsere**. De første er en slags procedure som kan sættes op til at køre en gang eller gentagende gange på et bestemt tidspunkt. Triggers er også en slags procedure men de udløses af bestemte handlinger som udføres på en bestemt tabel. F.eks. kan en Update på tabellen personer udløse en Trigger og denne trigger vil så enten kører før Update SQLen udføres eller lige efter. Det kan være essentielt hvis man har en database hvor data altid skal kunne spores, så kan man lave en backup tabel (f.eks. personer\_audit) der indeholde alle de gamle data.   
Prøv at lav en Trigger på en af jeres tabeller på f.eks. Update, og få den til at lave noget I kan bekræfte er sket bag efter.

I har nu arbejdet med jeres database i lang tid og den er ved at blive mere og mere kompleks. Det er derfor vigtigt at lave backup og det gøres gennem menupunktet ”Eksporter”. Brug som det sidste denne mulighed og prøv først at lav en eksport af databasen som den er sat til som standard. Gem den fil der dannes og åben den med notepad++ el. lignende program. Undersøg hvad der stå i filen og se om I kan genkende de tabeller og den data som I har lagt ind i jeres database. Hvis der er tid kan I gå ind i Eksporter igen og klikke på Brugerdefineret indstillinger og se hvilke ting I selv kan indstille næste gang I skal lave en eksport. Gem disse eksport og brug dem til at genskabe databasen hvis den bliver slettet ved et uheld. Det eneste som ikke er med i Eksporten er selve databasen, men den navn står øverst i filen et sted og kan hurtigt oprettes i modsætning til tabellerne og dataen.