

# INF1050 - Terminaløving uke 10

## Webformularer, SQL og PHP

Gå først gjennom [Appendiks B, Webformularer, PHP og databaser](#) og prøv noen av programmene der!

### Oppgave

Lag et webformular som gir et brukergrensesnitt med omtrent følgende utseende:



Når brukeren klikker på "Utfør spørring", skal SQL-spørringen sendes til Avfallsdatabasen og svaret vises fram i brukergrensesnittet.

Med et slikt formular slipper brukeren å skrive nøkkelordene i SQL-spørringen.

Hint for programmeringen:

Lag først XHTML-koden for webformularet og legg den på filen `sqlform.html`. I `action` kalles `sqlquery.php`. På denne filen legger du innledningsvis et lite php-program som ganske enkelt skriver ut den spørringen du bygger opp. Først når spørringene ser riktige ut, utvider du php-programmet slik at spørringene sendes til databasen og resultatet vises fram. La deg gjerne inspirere av koden på `~inf1050/php/sql.php` !

Hvis du legger alle html- og php-filer (inkludert `autentiser.php`) samt mappen `inc` med innhold i samme mappe, slipper du å tenke så mye på navigasjonsveiene fram til filene.

Merk at hvis det ikke er fylt ut noe i feltet "WHERE", skal det ikke genereres noe WHERE-ledd i spørringen heller.

## Ekstraoppgaver

Dette lille systemet kan utvides og perfektioneres. Her er noen ideer:

- a) Sett webformularet opp i en tabell, slik at brukergrensesnittet blir ryddigere.
- b) Utvid med felter for GROUP BY, HAVING og ORDER BY, og en avkrysningsboks for DISTINCT.
- c) Kall `autentiser.php` automatisk dersom en test viser at brukernavn og passord ikke er gitt.
- d) Istedenfor å la brukeren skrive inn riktig tabellnavn i FROM-feltet, kan vi la ham få velge på en nedtrekksliste mellom de tabellene som fins i databasen. Denne utvidelsen krever en SQL-spørring mot metadatabasen!
- e) På tilsvarende måte kan vi i SELECT-feltet la brukeren velge mellom attributtene i de valgte tabellene. Dialogen må da utføres i to trinn, fordi d) må utføres først.
- f) Integrer `sqlform.html` og `sqlquery.php` på samme fil.
- g) Legg inn feilkontroller.  
Å kontrollere for feil som databasehåndteringssystemet allikevel fanger opp, er kanskje ikke så nyttig. Spøringer som fører til kartesiske produkter oppfattes imidlertid ikke som feilaktige spørringer av databasehåndteringssystemet. Derfor kan det være av interesse å forhindre slike spørringer. Dette kan gjøres på to måter: Enten ved å kontrollere at de nødvendige join-betingelser er til stede, eller ved å lagre metadata om assosiasjoner og generere join-betingelsene automatisk.