

Sist uke

- ☐ Script
- ☐ Mysql

- ☐ Hva er en primærnøkkel?
- ☐ Hva er en fremmednøkkel?
- ☐ Hva betyr det at to tabeller er unionskompatible?
- ☐ Når benyttes GROUP BY gi et eksempel
- ☐ Når benyttes WHERE gi et eksempel

- ☐ Last ned Hobbyhuset.txt fra Fronter og kjør scriptet fra hjemmekatalogen deres på studssh
- ☐ Gå inn i mysql og sjekk at tabellene er opprettet. (show tables;)

Insert, update og delete

- ☐ insert into <tabellnavn> (felt1,felt2,felt3) values (v1,v2,v3);
- ☐ NB: Legger ikke inn verdi i auto_increment felt.
- ☐ insert into <tabellnavn> values (v1,v2,v3);
- ☐ update <tabellnavn> set fornavn='Ole' where etternavn ='Hansen';
- ☐ delete from <tabellnavn> where fornavn='Ole';
- ☐ Farlig: delete from <tabellnavn>;

- ☐ Oppgave: Finn ut hvilke kolonner som er i Vare tabellen.
- ☐ (desc Vare;)
- ☐ Legg inn en post i Vare , sjekk at den er blitt lagt til.
- ☐ Endr Varebetegnelse på varen du la inn og sjekk at det blir endret.
- ☐ Slett varen du la inn og sjekk at den er blitt fjernet.

Lag 5 SQL spørringer til hverandre gitt denne tabellen og utfør oppgavene i mysql.

EMPNO	ENAME	JOB	MGR	HIREDATE	SAL	COMM	DEPTNO
7369	SMITH	CLERK	7902	17-Dec-80	£800.00		20
7499	ALLEN	SALESMAN	7698	20-Feb-81	£1,600.00	£300.00	30
7521	WARD	SALESMAN	7698	22-Feb-81	£1,250.00	£500.00	30
7566	JONES	MANAGER	7839	02-Apr-81	£2,975.00		20
7654	MARTIN	SALESMAN	7698	28-Sep-81	£1,250.00	£1,400.00	30
7698	BLAKE	MANAGER	7893	01-May-81	£2,850.00		30
7782	CLARK	MANAGER	7839	09-Jun-81	£2,450.00		10
7788	SCOTT	ANALYST	7566	09-Nov-81	£3,000.00		20
7839	KING	PRESIDENT		17-Nov-81	£5,000.00		10
7844	TURNER	SALESMAN	7698	08-Sep-81	£1,500.00	£0.00	30
7876	ADAMS	CLERK	7788	23-Sep-81	£1,100.00		20
7900	JAMES	CLERK	7698	03-Dec-81	£950.00		30
7902	FORD	ANALYST	7566	03-Dec-81	£3,000.00		20
7934	MILLER	CLERK	7782	23-Jan-82	£1,300.00		10

Spøringer mot flere tabeller

- ☐ Kartesisk produkt / krysskobling
- ☐ Likekoblinger
- ☐ INNER JOIN syntaks
- ☐ Generelle koblinger
- ☐ Egenkoblinger
- ☐ Ytre koblinger
- ☐ Union, snitt og differanse
- ☐ Mer om gruppering og mengdefunksjoner

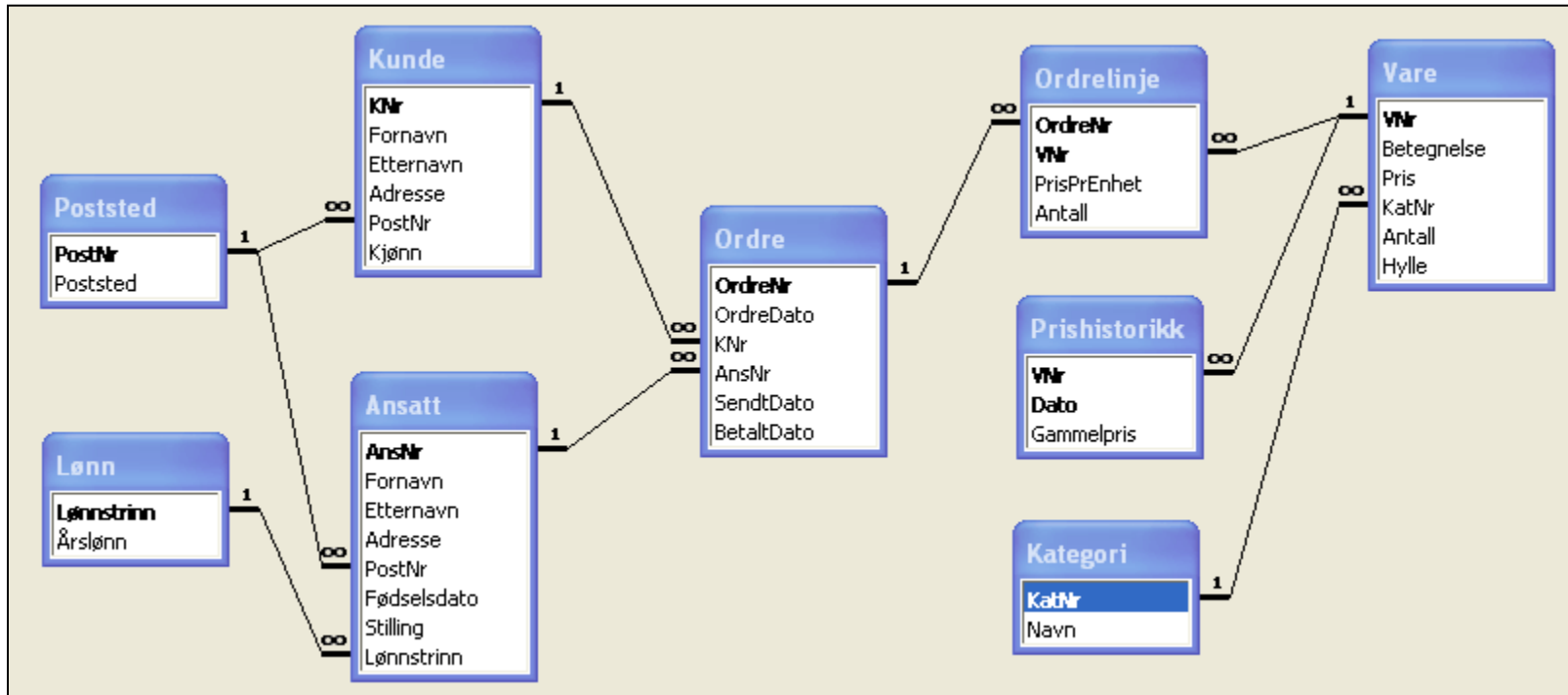
Pensum: Kapittel 4

Motivasjon

- ❑ For å unngå redundans ("dobbeltagring") lagrer vi data i flere tabeller. (Mer om dette i kapittel 8!)
- ❑ Så relasjonsdatabaser består av mange tabeller.
- ❑ Det er logiske sammenhenger mellom data i ulike tabeller.
 - Vi får behov for å skrive spørringer som "kobler" data fra flere tabeller.
 - Koblinger bygger på at samme type verdier finnes i flere tabeller.
 - Koblinger er ofte basert på fremmednøkler (men ikke alltid).

Eksempel: Hobbyhuset

- ❑ Lag en kundeliste sortert på poststed. Hvilke tabeller må brukes?

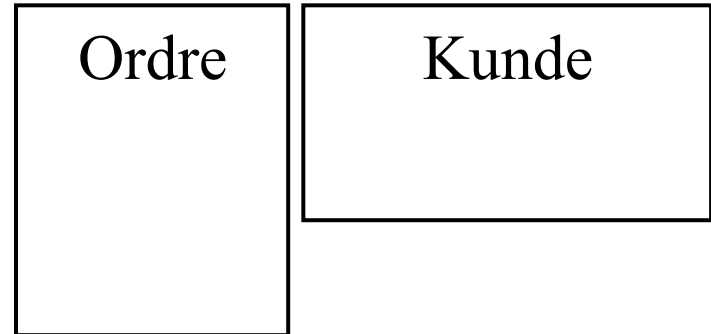


- ❑ Foreslå flere interessante spørringer/rapporter! Hvilke tabeller må brukes i hvert tilfelle?

Flere tabeller i FROM-delen

- ❑ Hva blir resultatet ?

```
SELECT *  
FROM Ordre, Kunde
```



- ❑ Antall rader og kolonner i resultatet ?
 - Får samtlige kolonner fra begge tabeller (*).
 - Enhver rad i Ordre blir koblet med enhver rad i Kunde.
- ❑ Vi kan hente data fra mange tabeller (flere enn 2).
- ❑ Vi må spesifisere hvilke rader som skal kobles.
 - Ordre inneholder KNr ...

Kartesisk produkt (kryssprodukt)

KNr	Navn
1	Per
2	Ola

OrdreNr	KNr	AnsNr
1	1	21
2	2	21
3	2	28

$$2 \times 3 = 6!$$



KNr	Navn	OrdreNr	KNr	AnsNr
1	Per	1	1	21
1	Per	2	2	21
1	Per	3	2	28
2	Ola	1	1	21
2	Ola	2	2	21
2	Ola	3	2	28

Likekobling

KNr	Navn
1	Per
2	Ola

OrdreNr	KNr	AnsNr
1	1	21
2	2	21
3	2	28



KNr	Navn	OrdreNr	KNr	AnsNr
1	Per	1	1	21
2	Ola	2	2	21
2	Ola	3	2	28

❑ Som regel vil vi kun ta med koblingskolonne(r) fra én av tabellene.

➤ De inneholder jo de samme verdiene.

Eksempel på likekobling

- ❑ Vis alle ordrer behandlet av hver ansatt. Sorter utskriften på etternavn og ordredato.

```
SELECT Etternavn, OrdreNr, Ordredato  
FROM Ansatt, Ordre  
WHERE Ansatt.AnsNr = Ordre.AnsNr  
ORDER BY Etternavn, Ordredato
```

- ❑ Likheten i **WHERE** er en koblingsbetingelse.
- ❑ Spørringen kalles en (indre) likekobling (join).
- ❑ AnsNr må prefikses med tabellnavn fordi kolonnenavnet forekommer i begge tabeller.
- ❑ Etternavn, OrdreNr og Ordredato kan prefikses.

Implementasjon av likekobling

- ❑ Hvordan utfører DBHS en likekobling?
 - Eksempel: Ordre og Kunde koblet på KNr
- ❑ Strategi 1: Lag krysskobling og slett ulike (ineffektivt!)
 - Dann alle rad-kombinasjoner (kryssproduktet) og lagre resultatet i en hjelpetabell.
 - Løp gjennom hjelpetabellen og slett kombinasjoner med forskjellige KNr.
- ❑ Med 5 000 ordre og 2 000 kunder vil hjelpetabellen inneholde 10 000 000 (10 millioner) rad-kombinasjoner.
 - Strategi 1 er svært ineffektiv, men er nyttig for å forstå hva resultatet av en likekobling blir.

Mer realistiske implementasjoner

❑ Strategi 2: Sortér og "flett"

- Sortér hver tabell med hensyn på KNr.
- Løp gjennom begge tabeller parallellt ("synkronisert") og skriv ut rad-kombinasjoner som matcher.

❑ Strategi 3: Nøstet løkke

- Løp gjennom Ordre rad for rad.
 - For hver ordre, løp gjennom Kunde og finn "match".
- Hvis Kunde er sortert kan vi bruke binærsøk.
- Hvis vi har en såkalt indeks på Kunde.KNr kan den indre løkka erstattes av et oppslag via indeksen ([mer om dette i kap. 9](#)).

Fremmednøkler og koblinger

- ❑ Vi kobler ofte to tabeller med hensyn på fremmednøkler.

```
SELECT *  
FROM Ordre, Kunde  
WHERE Orde.KNr = Kunde.KNr
```

- ❑ Det er mulig å koble på kolonner som ikke er fremmednøkler eller primærnøkler.
 - Finn kombinasjoner av ansatte og kunder bosatt på samme sted (GIS utfører koblinger basert på geografi ...).
- ❑ Av og til blir det feil å koble med hensyn på samtlige fremmednøkler.
 - Eksempel: Prosjektdeltakelse og prosjektledelse fra leksjon 1.

Syntaks for likekobling

- ❑ Likekoblinger forekommer så hyppig at det er innført en spesiell skrivemåte:

```
SELECT *  
FROM Ordre INNER JOIN Kunde  
      ON Ordre.KNr = Kunde.KNr
```

- ❑ Generelt:

```
T1 INNER JOIN T2 ON T1.kol1 = T2.kol2
```

- ❑ Rekkefølgen av tabellene spiller ingen rolle.

.

Synonymer

- ❑ For å redusere skrivearbeidet kan man innføre synonymer (kortnavn) for tabellene.

```
SELECT O.OrdreNr, K.KNr  
FROM Ordre AS O INNER JOIN Kunde AS K  
ON O.KNr = K.KNr
```

- ❑ Hvis man innfører synonymer må de brukes !
 - Synonymene O og K må også brukes i **SELECT**-delen, selv om de først blir "introdusert" i **FROM**-delen.
- ❑ Oracle bruker ikke **AS**.

Likekobling med ekstrabetingelser

- ❑ Kan ha generelle betingelser i tillegg til koblingsbetingelser.

```
SELECT V.VNr, K.Navn  
FROM Vare AS V, Kategori AS K  
WHERE V.KatNr = K.KatNr  
AND V.Pris > 100  
AND K.Navn = 'Keramikk'
```

- ❑ Hvordan bør DBHS utføre slike spørringer ?
 - Koblingsbetingelser først ?
 - Andre betingelser først ?

Dat typer og mening

- ❑ Dette er meningsløst:

```
SELECT *  
FROM Ansatt AS A INNER JOIN Ordre AS O  
ON A.AnsNr = O.KNr
```

- ❑ Koblingskolonnene må ha samme datatype, og også samme mening.
- ❑ Koblingskolonner trenger imidlertid ikke ha samme navn.
- ❑ Vil DBHS godta spørringen?

Koble flere tabeller enn 2

- ❑ Hvilke kunder har kjøpt vare 1014 ?

```
SELECT K.*  
FROM  
    Kunde AS K INNER JOIN  
        (Ordre AS O INNER JOIN Ordrelinje AS OL  
            ON O.OrdreNr = OL.OrdreNr)  
        ON K.KNr = O.KNr  
WHERE OL.VNr = 1014
```

- ❑ Vi kan tenke oss at selve koblingen blir utført slik:
 1. Ordre og Ordrelinje blir koblet.
 2. Resultatet blir koblet med Kunde.
- ❑ Lettere å skrive/lese med vanlig WHERE-betingelse ?
- ❑ Skriv setningen med WHERE-betingelser.

Likekobling med gruppering

- ❑ Finn antall ordre pr. kunde:

```
SELECT K.KNr, COUNT(*) AS [Antall ordre]  
FROM Kunde AS K, Ordre AS O  
WHERE K.KNr = O.KNr  
GROUP BY K.KNr
```

- ❑ Hva om vi kun vil vise kunder med flere enn 10 ordre ?
- ❑ Totalt varekjøp pr. kunde:
 - Hvilke tabeller må kobles?
 - Hva skal vi gruppere på?

Ytre koblinger (høyre og venstre)

- ❑ Indre likekoblinger tar bare med verdier som finnes i begge tabeller. Det er ikke alltid tilstrekkelig.

- ❑ Vis kunder med tilhørende ordrer. Samtlige kunder skal med.

```
SELECT K.KNr, O.OrdreNr  
FROM Kunde AS K LEFT OUTER JOIN Ordre AS O  
ON K.KNr = O.KNr
```

- ❑ Resultatet består av:
 - Radene i en tilsvarende likekobling, og dessuten
 - én rad for hver kunde uten ordrer. For disse står det et nullmerke i OrdreNr.
- ❑ "Venstre" og "høyre" refererer til rekkefølgen i FROM.

Venstre ytre kobling

KNr	Navn
1	Per
2	Ola
3	Lise

OrdreNr	KNr	AnsNr
1008	1	25
1009	2	25
1010	1	28



KNr	Navn	OrdreNr	KNr	AnsNr
1	Per	1008	1	25
2	Ola	1009	2	25
1	Per	1010	1	28
3	Lise			



❑ Ytre kobling = likekobling + 1 rad for hver uten "match".

Generelle koblinger

- ❑ Det er mulig å koble med andre operatører enn likhet (=).
- ❑ Finn varer som har blitt solgt med rabatt (en eller flere ganger):

```
SELECT DISTINCT VNr  
FROM Vare AS V, Ordrelinje AS OL  
WHERE OL.PrisPrEnhet < V.Pris
```

- ❑ Noen GIS-eksempler med ”geografiske operatører”:
 - Finn byer i Telemark (punkt INNENFOR polygon)
 - Finn veier som krysser Mjøsa (linje KRYSSER linje)
 - Finn eiendommer som blir berørt av en veiutbygging (polygon OVERLAPPER polygon)

Egenkoblinger

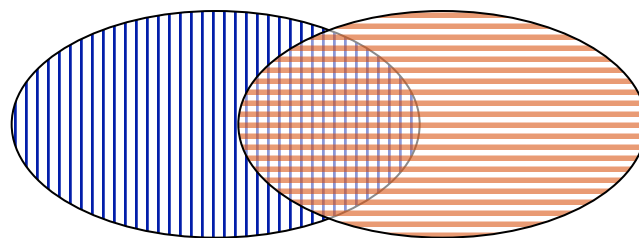
- ❑ Tabeller kan kobles med "seg selv".
- ❑ Finn alle kombinasjoner av varer med samme pris:

```
SELECT V1.VareID, V2.VareID, V1.Pris  
FROM Vare AS V1, Vare AS V2  
WHERE V1.Pris = V2.Pris
```

- ❑ Tenk slik: DBHS "lager" 2 kopier av tabellen Vare, og kobler disse på vanlig måte.
- ❑ Tabeller spiller av og til flere "roller", som også kan medføre behov for å bruke flere "kopier" av samme tabell i en spørring. Eksempel: Avgangsflyplass og ankomstflyplass i en flyavgang, se oppg 4 i kap. 4.

Snitt, union og differanse

- ❑ En tabell består av en mengde rader. Vi har tilgang på standard mengdeoperasjoner:



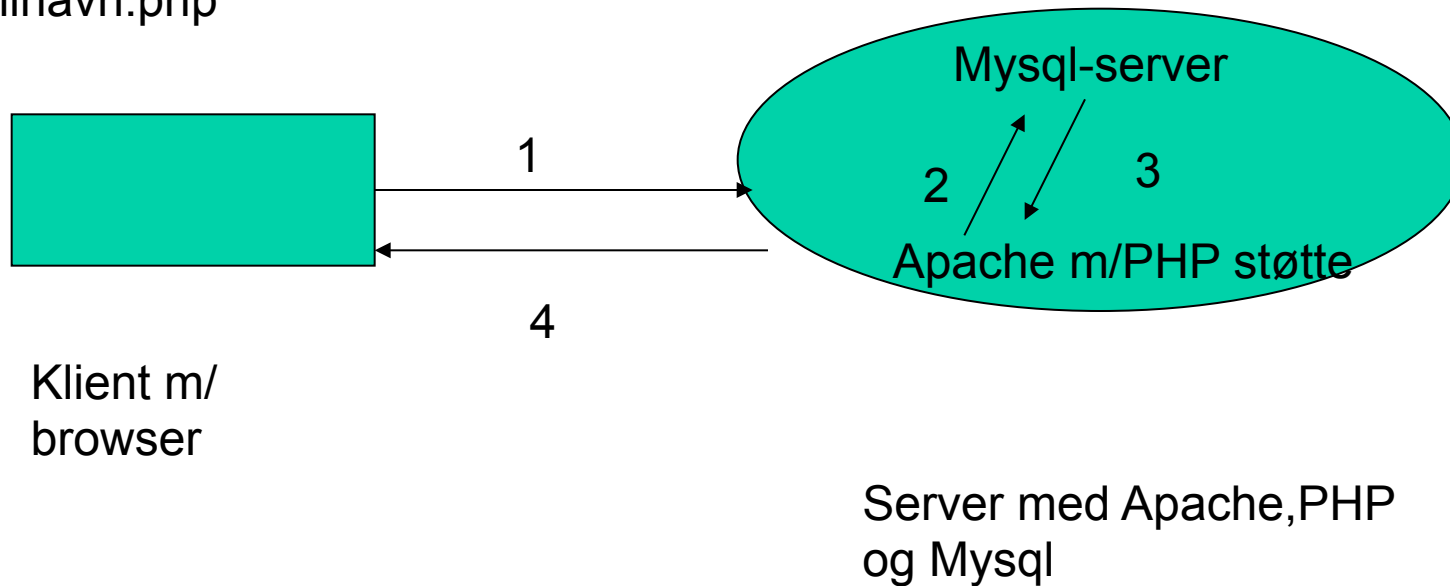
- ❑ Anta historiske ordredata er lagret i tabeller Ordre2001 og Ordre2002:

```
SELECT KNr FROM Ordre2001  
UNION  
SELECT KNr FROM Ordre2002
```

- ❑ UNION svarer i en viss forstand til OR, INTERSECT til AND og MINUS/EXCEPT til AND + NOT.

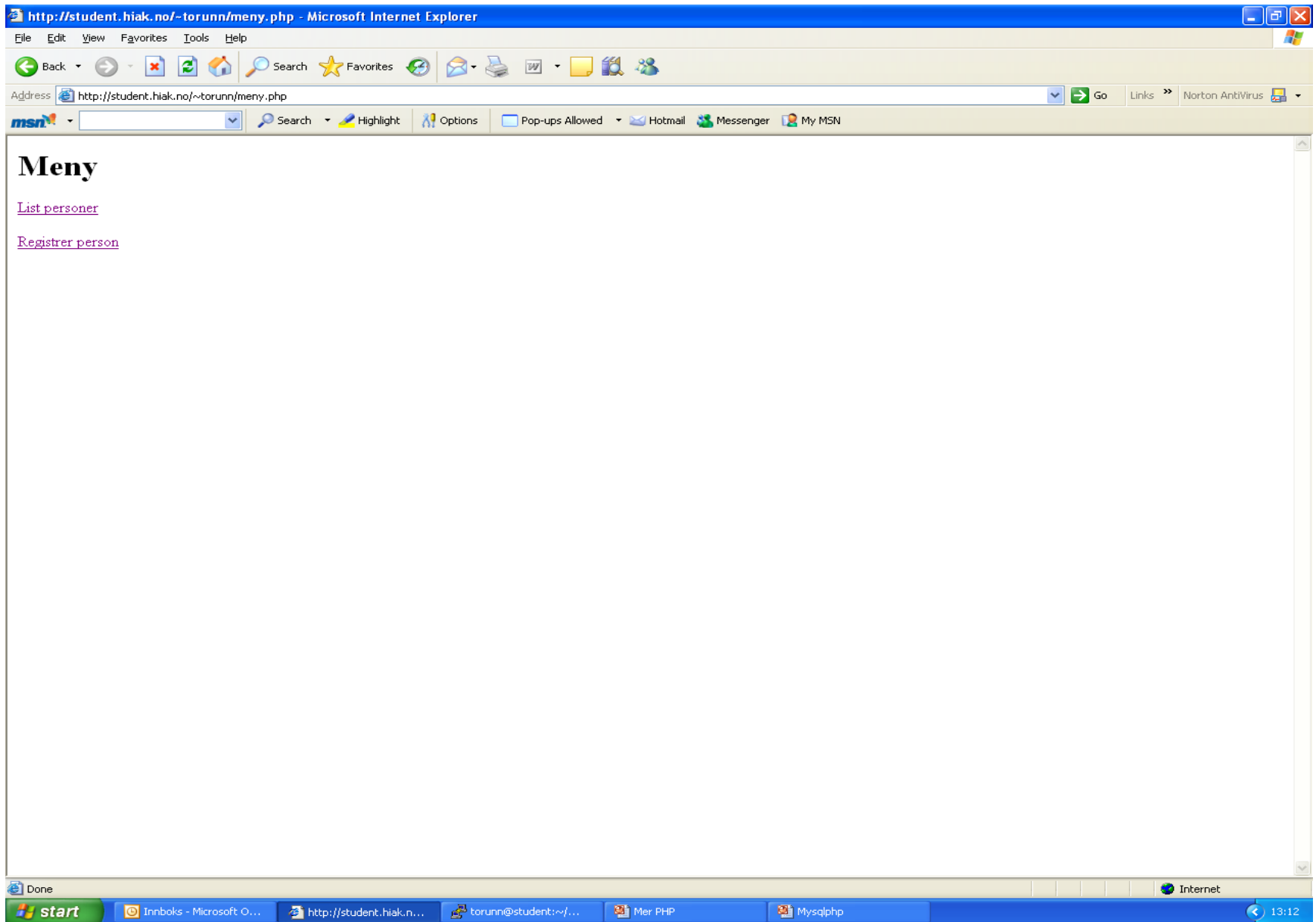
Eksempel på PHP mot database:

<http://studssh.cs..hioa.no/~studentnr/filnavn.php>



Enkel PHP-programmering:

- ☐ Jeg har en database med en tabell som heter person
- ☐ Person har to felt: navn og alder
- ☐ Jeg vil lage en meny som gir meg mulighet til å liste alle personer eller å legge en ny person inn i tabellen
- ☐ PHP filene må ligge på www-katalogen på studssh.cs.hioa.no, de kan overføres med Winscp.
- ☐ Editoren pico på studssh kan benyttes for å gjøre endringer i php-filene.



Koden som ligger i meny.php er

```
<html>
<head></head>
<body>
<h1>Meny</h1>

<a href="hent.php">List personer</a>
<p>
<a href="innskjema.html">Registrer person </a>
</body>
</html>
```

Kode for å liste personer

```
<html>  
<head></head>  
<body>  
<?php  
$con=mysqli_connect("student.cs.hioa.no","torunngj","", "torunngj");
```

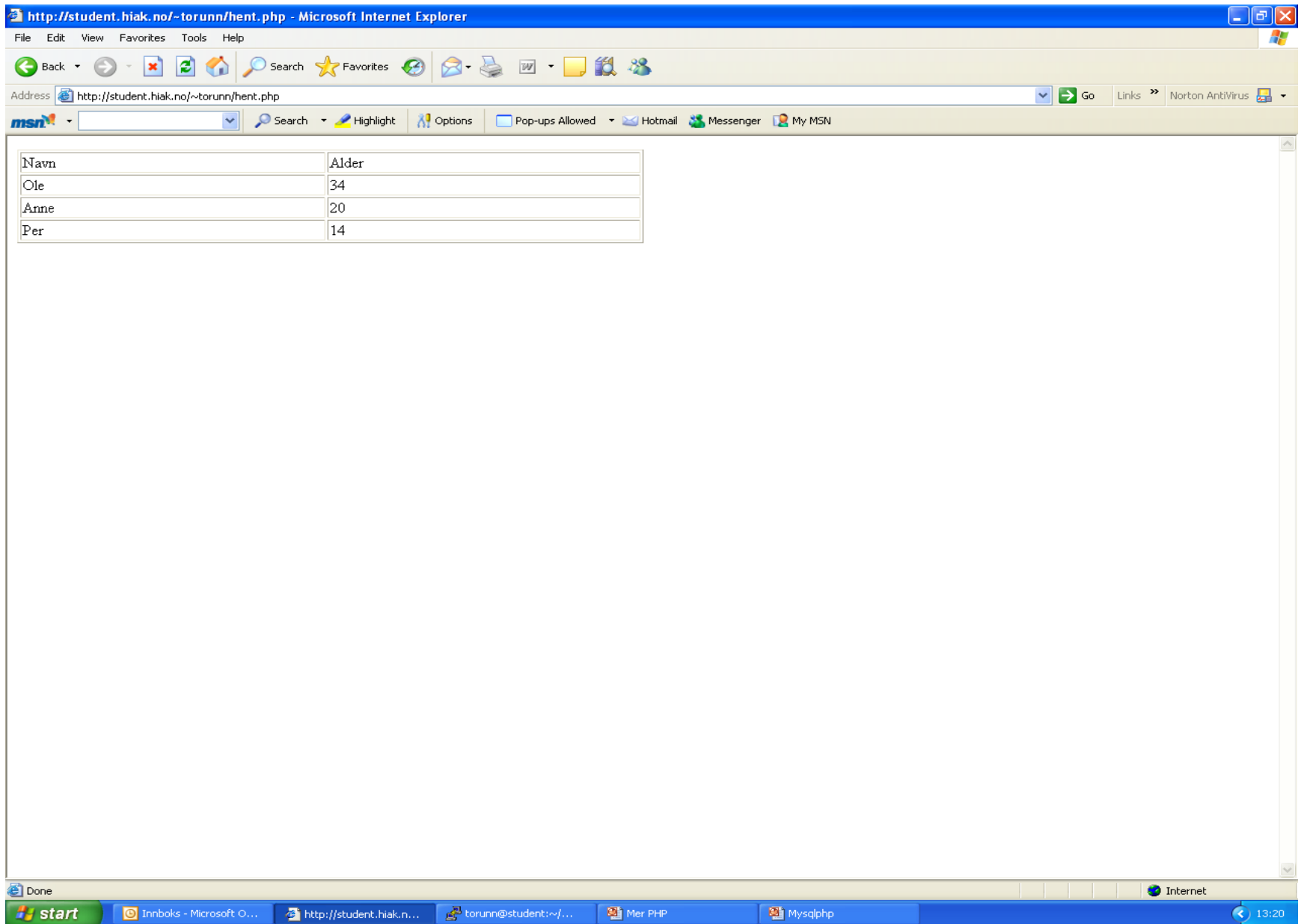
NB! HUSK AT DERE IKKE HAR PASSORD PÅ MYSQL_DB!!

```
$sql=("select * from PERSON");
```

```
if ($result=mysqli_query($db,$query)){  
  
$num_results=mysqli_num_rows($result);  
  
for ($i=0; $i <$num_results; $i++){  
  
    $rad=mysqli_fetch_row($result);  
    echo $rad[0];  
  
    echo $rad[1];  
    }  
    }  
    mysqli_close($db);  
  
> </body> </html>
```

Hva gjorde vi?

- ☐ Kjørte php-rutine for å koble til database
- ☐ Lagde sql-setning
- ☐ Sendte sql-setningen
- ☐ Mottok en tabell som resultat av spørringen
- ☐ Hentet ut verdiene i resultat-tabellen ved hjelp av ferdige php-rutiner



Hva skjer?

- ☐ Dere kaller på en HTMLside med PHP-kode.
 - ☐ Apache serveren kaller på PHP-programmet på serveren som utfører denne koden
 - ☐ Resultatet puttes inn i HTML-siden som sendes i retur til klienten
 - ☐ Prosessering på serveren, presentasjon på klienten → 2-lags arkitektur.
 - ☐ Hvis databaseserveren hadde vært en egen server: Presentasjon på klient, Prosessering på web-server, Databasehåndtering på database-server → 3-lags arkitektur
-
- ☐ Prøv gjerne å lage noe lignende mot EMP og DEPT tabellene.
 - ☐ Husk at copy-paste fra PowerPoint og PDF gir masse feil tegn! Bedre å skrive inn koden selv i f.eks pico på unix-serveren.