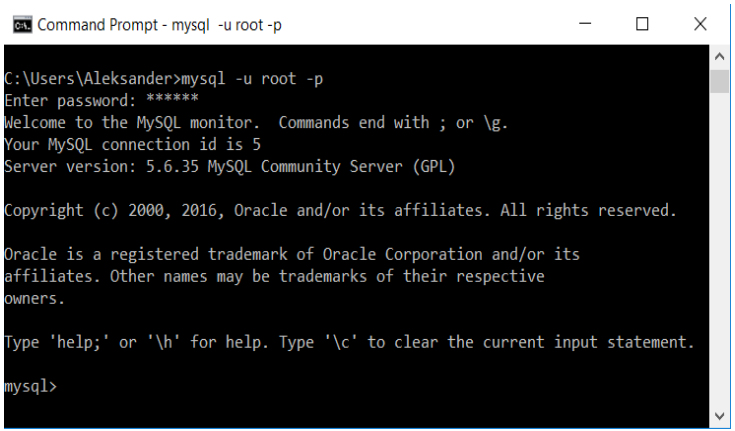
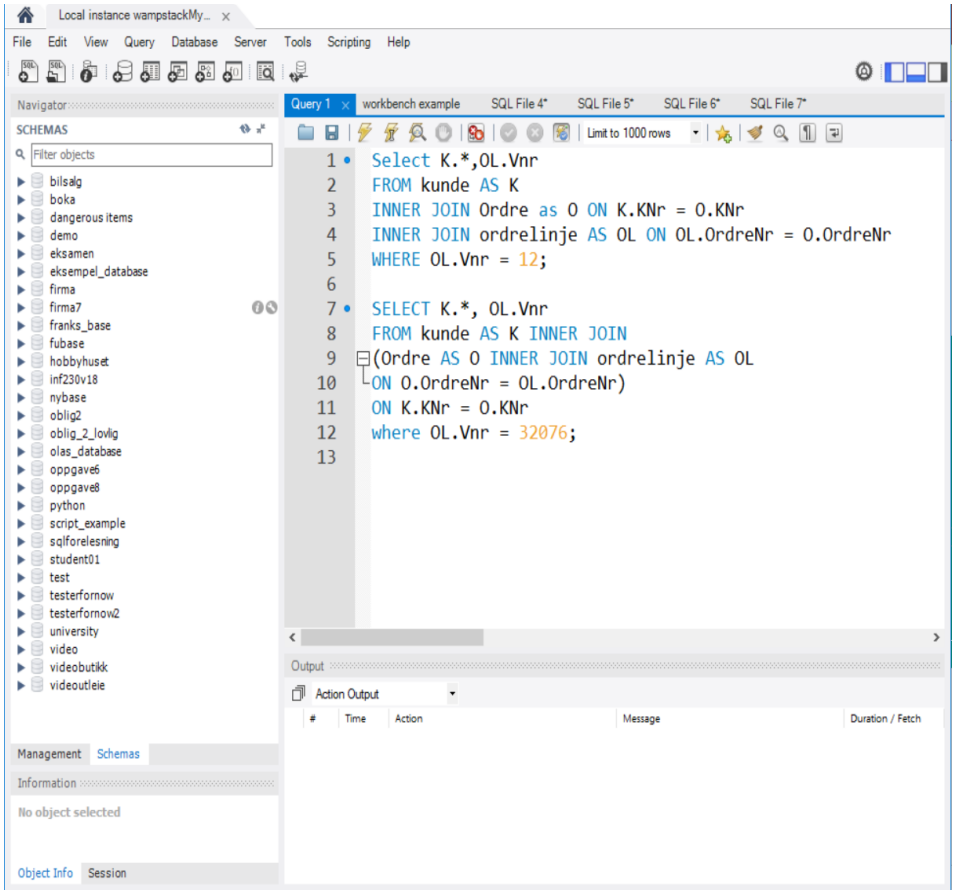
Lecture 28.02.18

Today

- Workbench
- Advanced queries

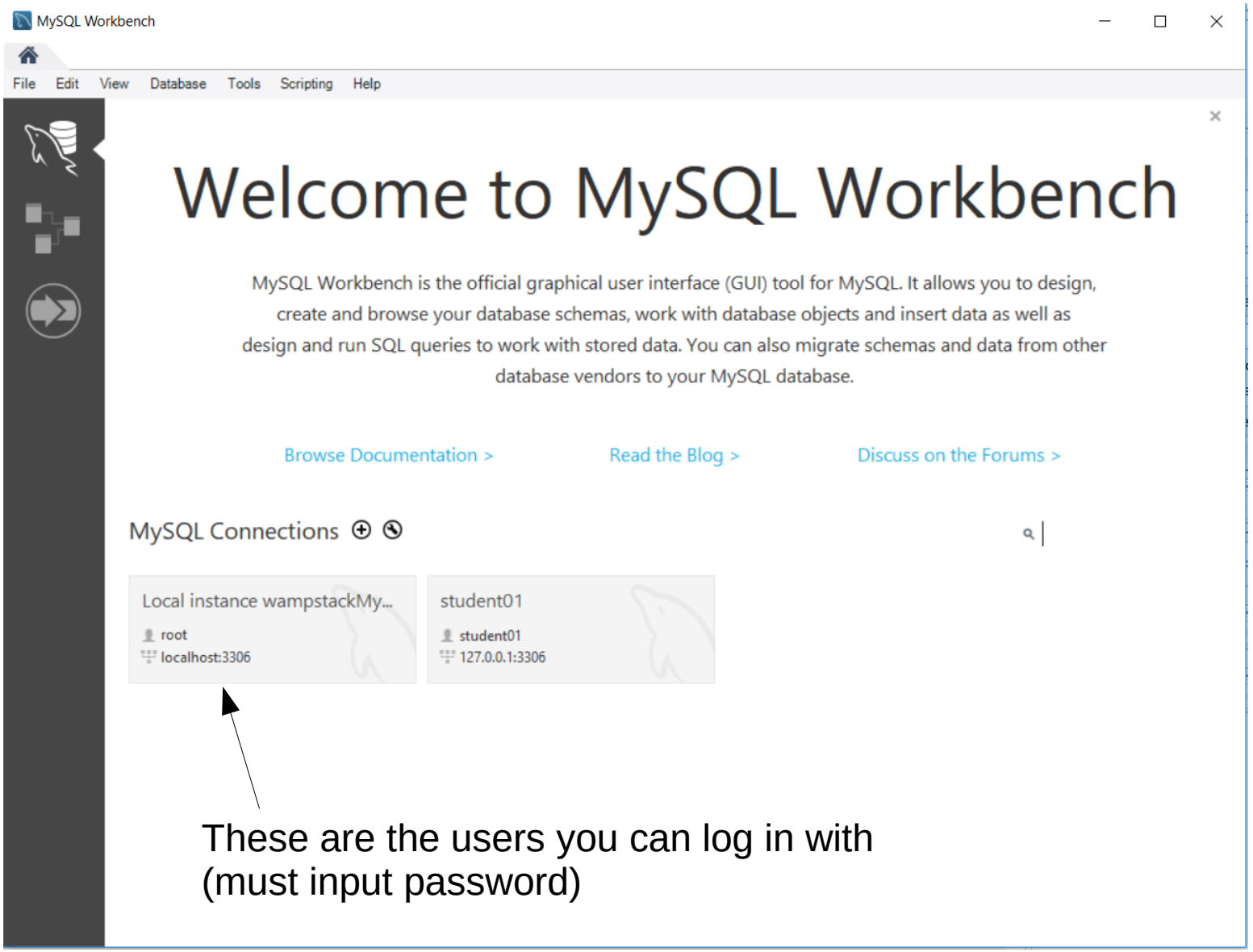
Workbench


Command Prompt - mysql -u root -p
C:\Users\Aleksander>mysql -u root -p
Enter password: ******
Welcome to the MySQL monitor. Commands end with ; or \g.
Your MySQL connection id is 5
Server version: 5.6.35 MySQL Community Server (GPL)
Copyright (c) 2000, 2016, Oracle and/or its affiliates. All rights reserved.
Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.
Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.
mysql>

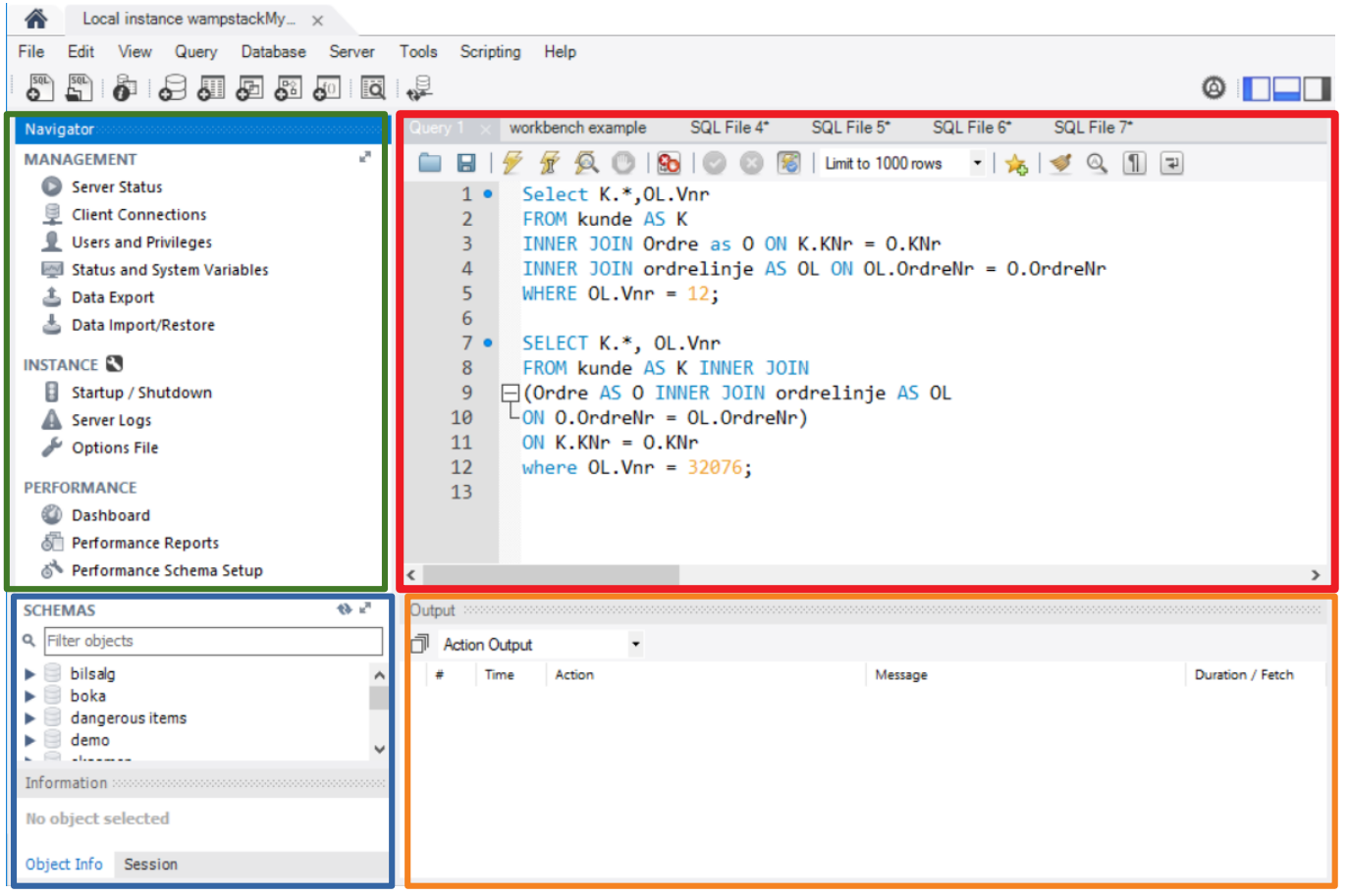

Local instance wampstackMy...
File Edit View Query Database Server Tools Scripting Help
Navigator
SCHEMAS
Filter objects
bilsalg
boka
dangerous items
demo
eksamen
eksempel_database
firma
firma7
franks_base
fubase
hobbyhuset
inf230v18
nybase
oblig2
oblig_2_lovlig
olas_database
oppgave6
oppgave8
python
script_example
sqlforelesning
student01
test
testerfornow
testerfornow2
university
video
videobutikk
videoutleie
Management Schemas
Information
No object selected
Object Info Session
Query 1 workbench example SQL File 4* SQL File 5* SQL File 6* SQL File 7*
Limit to 1000 rows
1 Select K.*,OL.Vnr
2 FROM kunde AS K
3 INNER JOIN Ordre as O ON K.KNr = O.KNr
4 INNER JOIN ordrelinje AS OL ON OL.OrdreNr = O.OrdreNr
5 WHERE OL.Vnr = 12;
6
7 SELECT K.*, OL.Vnr
8 FROM kunde AS K INNER JOIN
9 (Ordre AS O INNER JOIN ordrelinje AS OL
10 ON O.OrdreNr = OL.OrdreNr)
11 ON K.KNr = O.KNr
12 where OL.Vnr = 32076;
13
Output
Action Output
Time Action Message Duration / Fetch

Installing workbench

- https://www.mysql.com/products/workbench/
- Download mysql workbench installer
  - No need to create an account
- Open installer and follow instructions


MySQL Workbench
File Edit View Database Tools Scripting Help
Welcome to MySQL Workbench
MySQL Workbench is the official graphical user interface (GUI) tool for MySQL. It allows you to design, create and browse your database schemas, work with database objects and insert data as well as design and run SQL queries to work with stored data. You can also migrate schemas and data from other database vendors to your MySQL database.
Browse Documentation >
Read the Blog >
Discuss on the Forums >
MySQL Connections
Local instance wampstackMy...
root
localhost:3306
student01
student01
127.0.0.1:3306
These are the users you can log in with
(must input password)

Manager
Server related tools



Editor
Write and run sql code

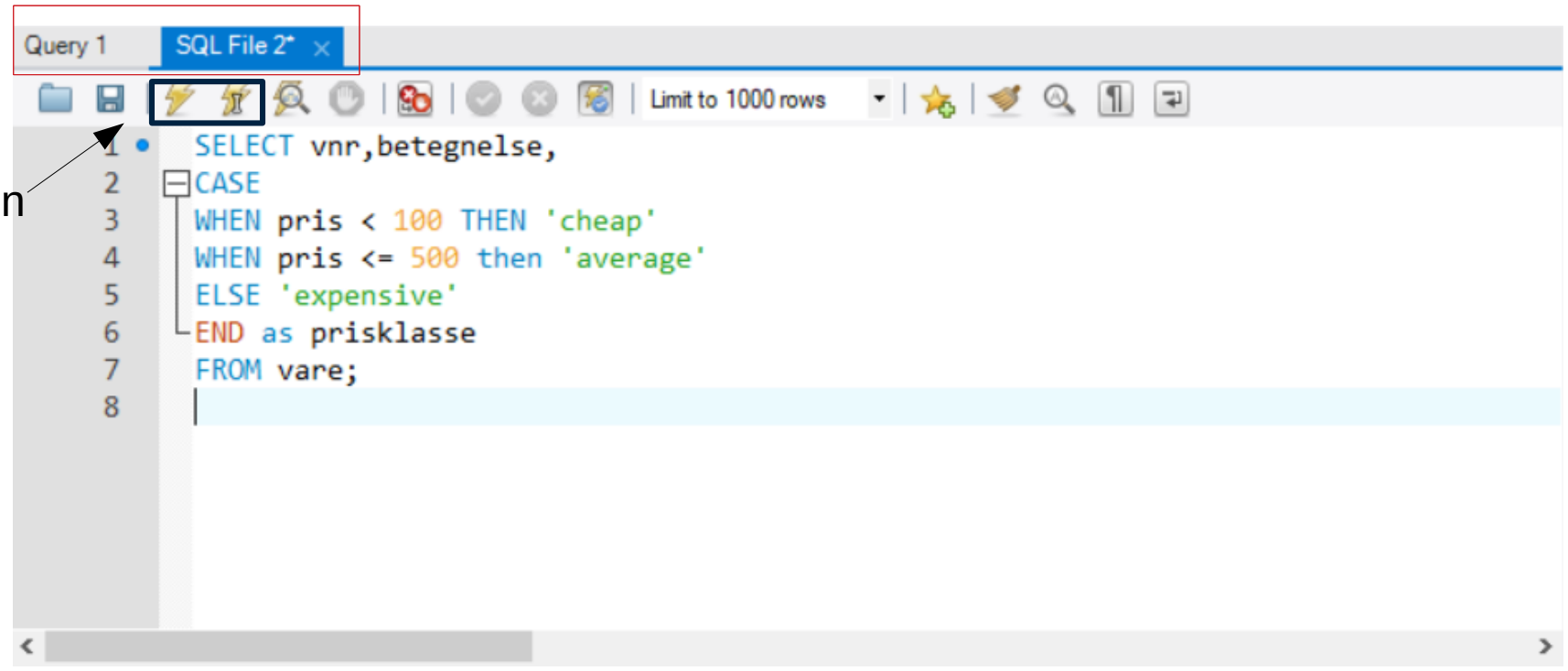
Databases
The contents of your DBMS

Output
Results of queries show up here

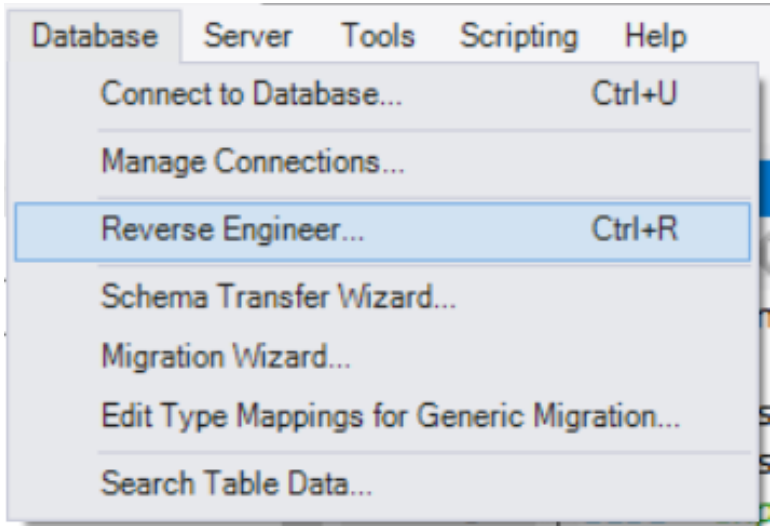
Editor

Tabs

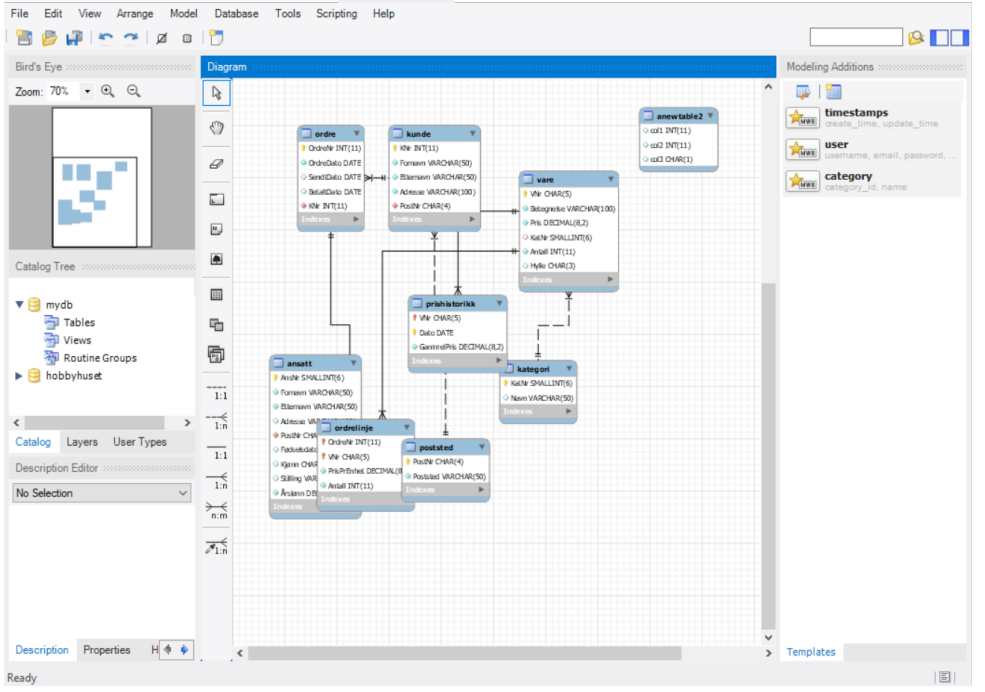
Run sql



Reverse engineer



Choose database



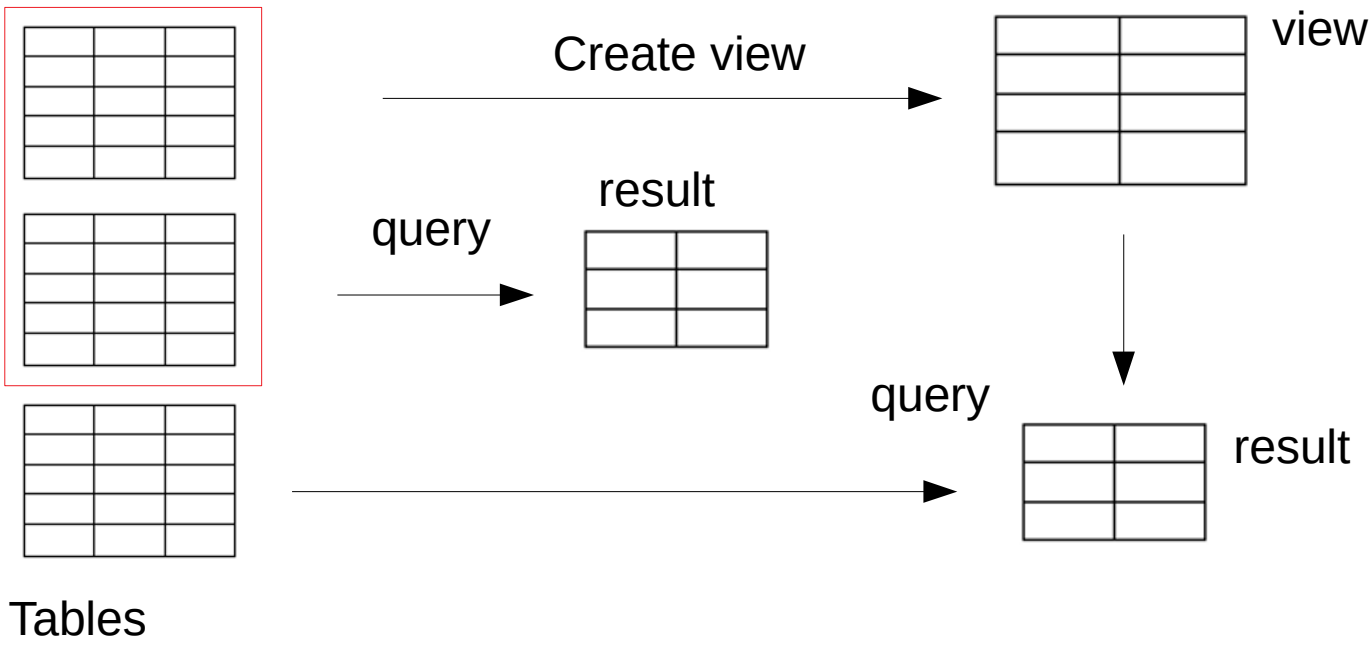
Advanced sql

Views

- Store queries as virtual tables
- Why?
  - Simplify queries
  - Limit access to databases for certain users
  - Database robustness

Views

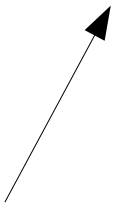
- Queries can be stored as views



View syntax

```
CREATE VIEW DyreVarer AS
  SELECT *
  FROM Vare
  WHERE Pris > 500
```

```
SELECT *
FROM DyreVarer
```



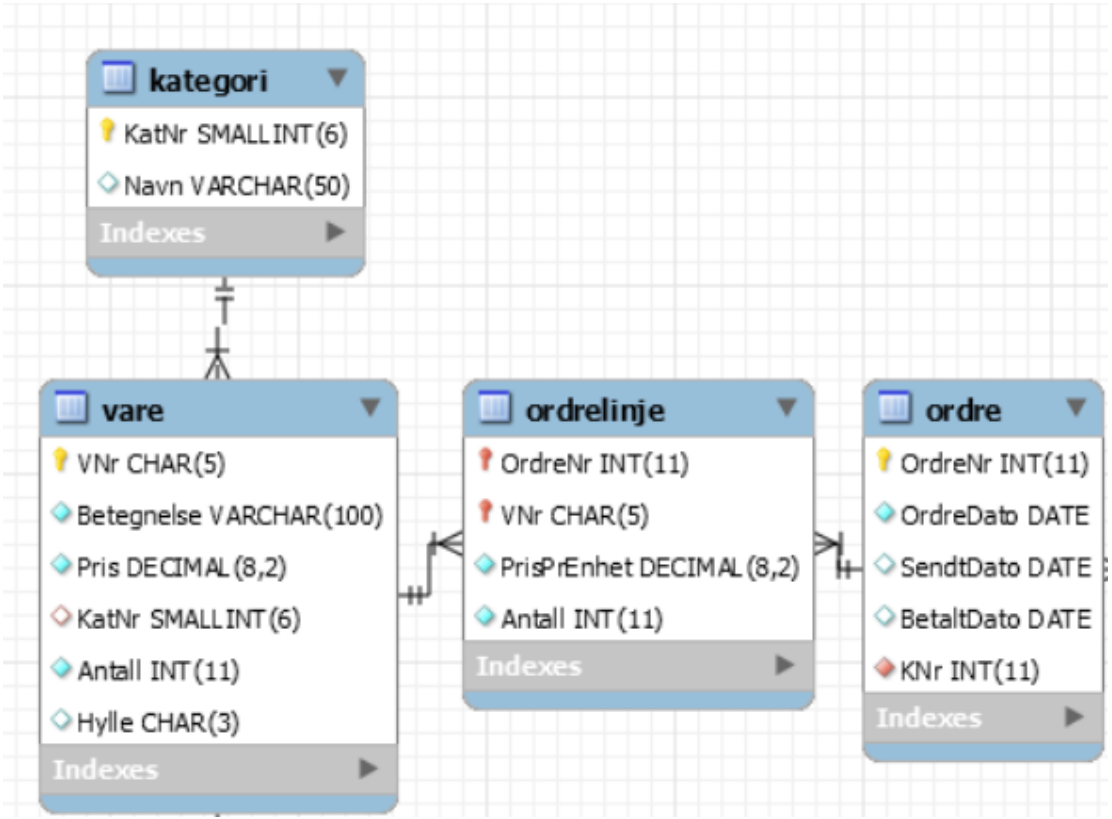
We can query the view

Create view naming

- We can change the names of the columns of the view when we create them

```
CREATE VIEW Keramikk( Kode, Navn ) AS
SELECT VNr, Betegnelse
FROM Vare
WHERE KatNr = 3
```

View example




```
create view salg AS
select OL.*
from ordre AS O
INNER JOIN ordrelinje AS OL
ON OL.ordreNr=O.ordreNr
INNER JOIN vare AS V
ON OL.vnr=V.vnr
INNER JOIN Kategori AS K
ON V.katnr=K.katnr;
```

View example query

- We can query against the view
  - Get total amount sold for a single month

```
SELECT vnr, SUM(antall*prisprenhet) AS totalt, ordredato
from salg
WHERE YEAR(ordredato) = 2015
AND MONTH(ordredato) = 1
GROUP BY vnr;
```

Non-view example query

- It is possible to do the same without a view
  - A lot longer query

```
SELECT OL.vnr, SUM(OL.antall*OL.prisprenhet) AS totalt
from ordre AS O
INNER JOIN ordrelinje AS OL
ON OL.ordreNr=O.ordreNr
INNER JOIN vare AS V
ON OL.vnr=V.vnr
INNER JOIN Kategori AS K
ON V.katnr=K.katnr
WHERE YEAR(O.ordredato) = 2015
AND MONTH(O.ordredato) = 1
GROUP BY OL.vnr;
```

The "old" view

Create view check option

- Can prevent updates from hiding rows that conflict with a WHERE statement in a view

```
CREATE VIEW DyreVarer AS
SELECT *
FROM Vare
WHERE Pris > 1000
WITH CHECK OPTION
```

This will fail

```
UPDATE DyreVarer
SET Pris = 999
WHERE Pris BETWEEN 1000 and 1050
```

View updates

- What will happen to the original table vare if we change the number of categories in the view to 2 from 5?

```
CREATE VIEW AntallVarerPrKategori AS
SELECT KatNr, COUNT(*) AS AntallVarer
FROM Vare
GROUP BY KatN
```

- General rules
  - Can't update grouped columns
  - Primary key has to follow

Robust database

- Sometimes we wish to restructure our database
  - Old applications rely on the "old" structure

- Create views with the same structure and names as the "old" database

Subqueries

ansnr	etternavn	stilling	lønn
1	Hansen	Selger	500.000
2	Mo	Programmerer	600.000
3	Jensen	Selger	500.000
4	Karlsen	Sekretær	400.000

snitt
560.000

Find employees that have above average salary

Query inside a query

```
SELECT * FROM ansatt
WHERE årslønn>(SELECT AVG(årslønn) FROM ansatt);
```

Subqueries

```
SELECT * FROM ansatt
WHERE årslønn>(SELECT AVG(årslønn) FROM ansatt);
```

Requirements

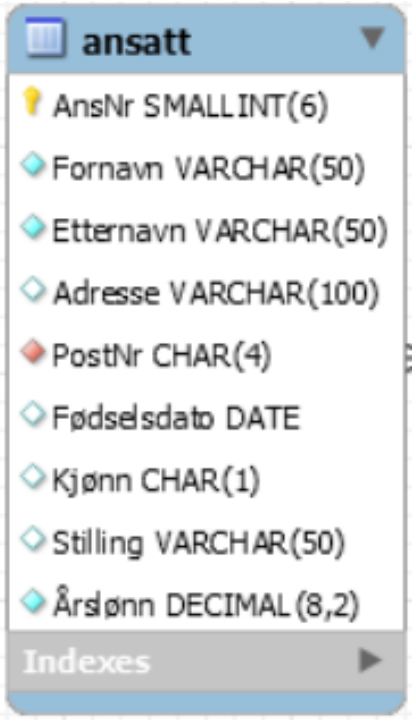
- The subquery has to return one row and one column to be used with the ">" operator

Correlated subqueries

- Which itemS are the least expensive in their category?

```
SELECT V1.Vnr,V1.pris
FROM vare AS V1
WHERE V1.pris =
(SELECT MIN(V2.pris) FROM vare AS V2
WHERE V1.KatNr = V2.katnr);
```

Queries in the from statement



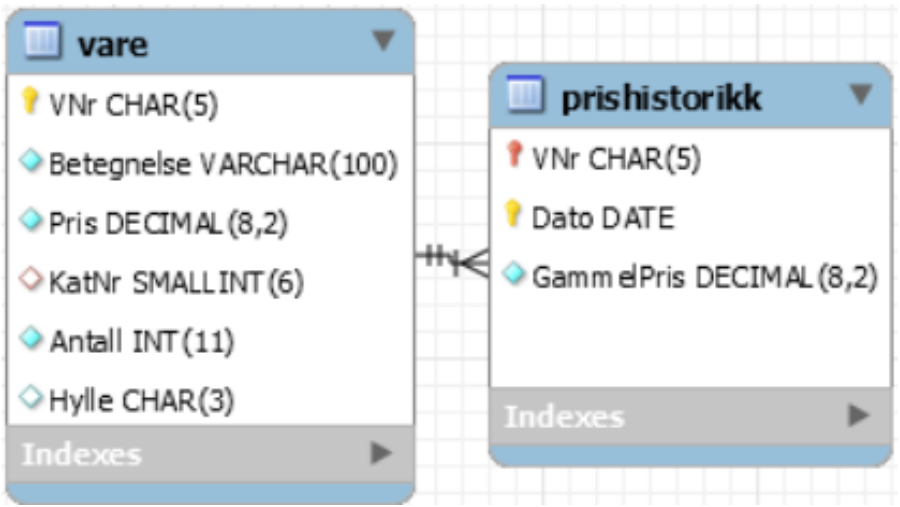
- Find the number of unique positions

```
SELECT COUNT(*) AS
antallstillinger
FROM
(SELECT DISTINCT stilling
FROM ansatt) AS positions;
```

Every derived table needs its own name!!!

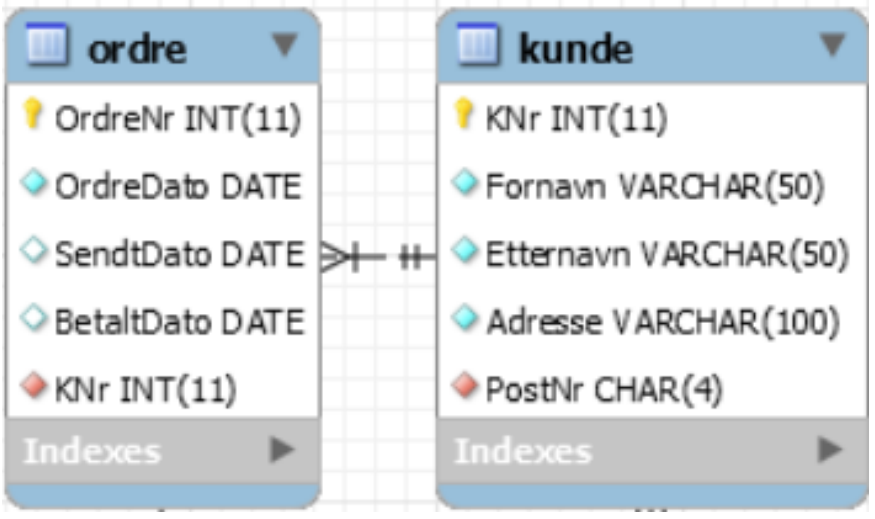
IN, SOME, ALL, ANY, EXISTS

- Sometimes subqueries return more than one row
  - Use IN, SOME, ALL, ANY and EXISTS




```
SELECT *
FROM vare AS V
WHERE pris >= ALL
(SELECT gammelpris FROM prishistorikk AS H
WHERE V.vnr = H.vnr);
```

IN, SOME, ALL, ANY, EXISTS

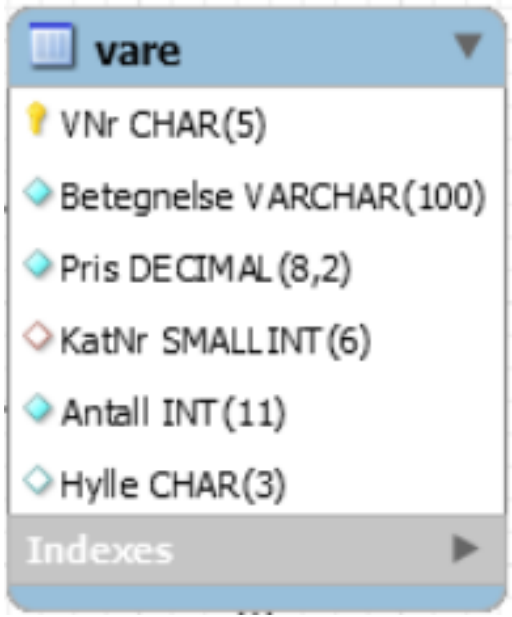


Find the customers that don't have orders

```
SELECT *
FROM kunde as K
WHERE NOT EXISTS
(SELECT O.knr
FROM ordre AS O
WHERE O.knr=K.knr);
```

Case

- Display text instead of values




```
SELECT vnr,betegnelse,
CASE
WHEN pris < 100 THEN 'cheap'
WHEN pris <= 500 then 'average'
ELSE 'expensive'
END as prisklasse
FROM vare;
```