Plenary 26.02.18

Today

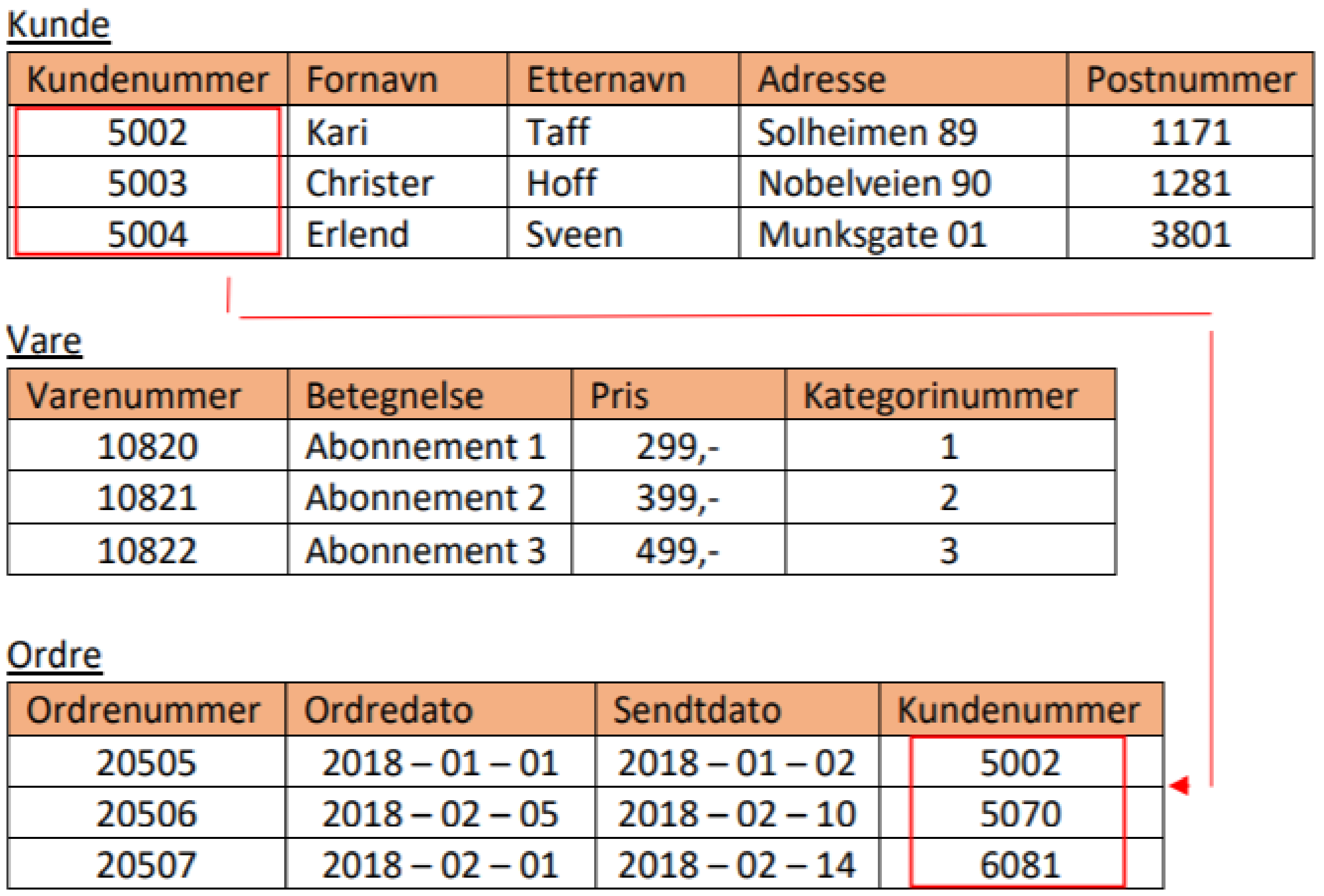
- Mandatory exercise
- Scripting
- Joins recap

Mandatory exercise

- Some info that telenor would like to store
  - Customer info, subscriptions, sales, employees.

Mandatory exercise

Connections and unique identifiers



Kunde

Kundenummer	Fornavn	Etternavn	Adresse	Postnummer
5002	Kari	Taff	Solheimen 89	1171
5003	Christer	Hoff	Nobelveien 90	1281
5004	Erlend	Sveen	Munksgate 01	3801

Vare

Varenummer	Betegnelse	Pris	Kategorinummer
10820	Abonnement 1	299,-	1
10821	Abonnement 2	399,-	2
10822	Abonnement 3	499,-	3

Ordre

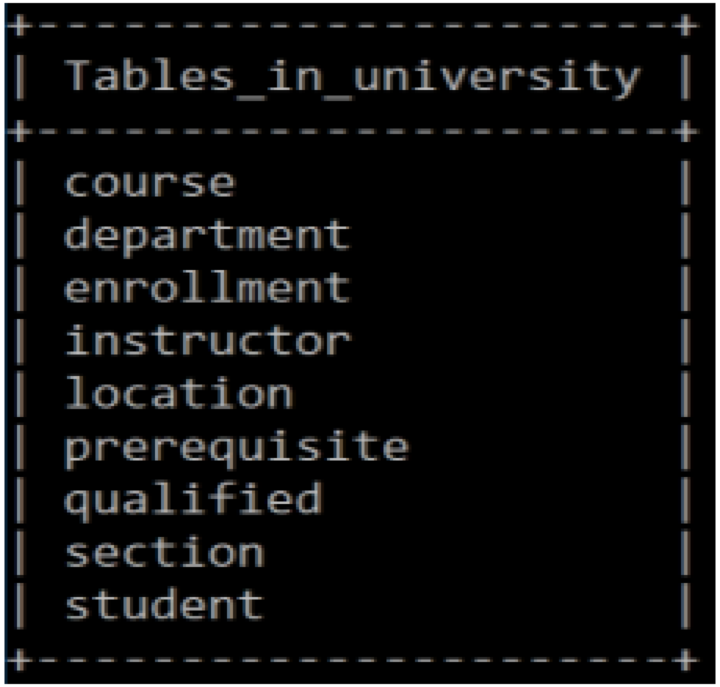
Ordrenummer	Ordredato	Sendtdato	Kundenummer
20505	2018 – 01 – 01	2018 – 01 – 02	5002
20506	2018 – 02 – 05	2018 – 02 – 10	5070
20507	2018 – 02 – 01	2018 – 02 – 14	6081

Mandatory exercise

- Unique identifiers
  - Kundenummer, varenummer, ordrenummer

Mandatory exercise part 2

```
CREATE DATABASE University;
USE university;
SOURCE university.sql;

SHOW tables;
```



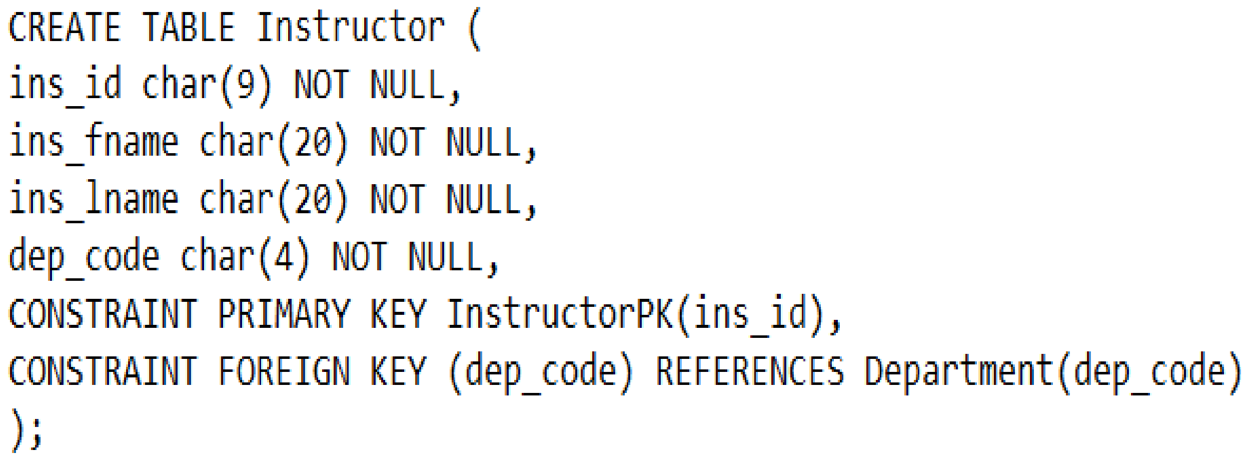
```
+-----------------------+
| Tables_in_university  |
+-----------------------+
| course                |
| department            |
| enrollment            |
| instructor            |
| location              |
| prerequisite          |
| qualified             |
| section               |
| student               |
+-----------------------+
```

Mandatory exercise part 2

- Primary and foreign keys
  - Primary keys are one or more columns that act as a unique identifier for each row
  - Foreign keys are columns that are connected to columns in other tables. You cannot insert values into the child table that does not exist in the parent.

Mandatory exercise part 2

- You can use describe to find the primary keys
- Better to look at the source file (university.sql)



```
CREATE TABLE Instructor (
ins_id char(9) NOT NULL,
ins_fname char(20) NOT NULL,
ins_lname char(20) NOT NULL,
dep_code char(4) NOT NULL,
CONSTRAINT PRIMARY KEY InstructorPK(ins_id),
CONSTRAINT FOREIGN KEY (dep_code) REFERENCES Department(dep_code)
);
```

Mandatory exercise part 2

- Register data
  - Insert into

```
INSERT INTO student
(stu_id,stu_fname,stu_lname) VALUES
(1,'Aleksander','Hykkerud');
```

Mandatory exercise part 2

```
INSERT INTO student
(stu_id,stu_fname,stu_lname) VALUES
(1,'Aleksander','Hykkerud');


INSERT INTO location (loc_code, loc_name,
loc_country) VALUES ('1', 'MOSS', 'NO');
```

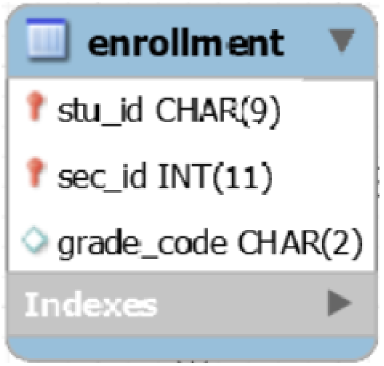
Mandatory exercise part 2

```
INSERT INTO department (dep_code, dep_name) values('IMT', 'Institutt for matematikk og teknologi');

INSERT INTO course(crs_code, crs_title, crs_credits, dep_code, crs_description) VALUES ('INF230', 'Datahåndtering og analyse', '10', 'IMT', 'Et kurs i datahåndtering og

analyse');
```

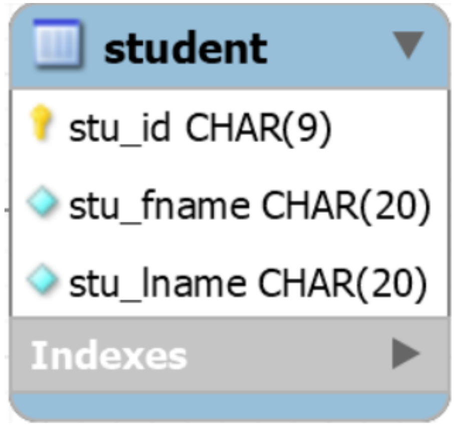
Mandatory exercise part 2

```
INSERT INTO instructor (ins_id, ins_fname,
ins_lname, dep_code) VALUES ('1', 'Ingunn',
'Burud','IMT');
```

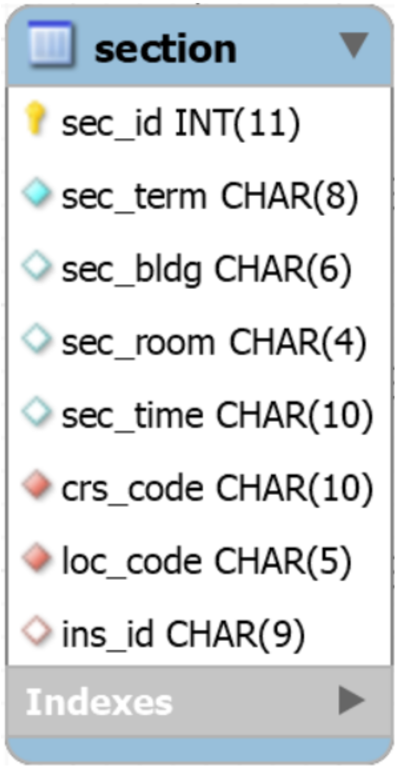
Mandatory exercise 6.a



enrollment
stu_id CHAR(9)
sec_id INT(11)
grade_code CHAR(2)
Indexes

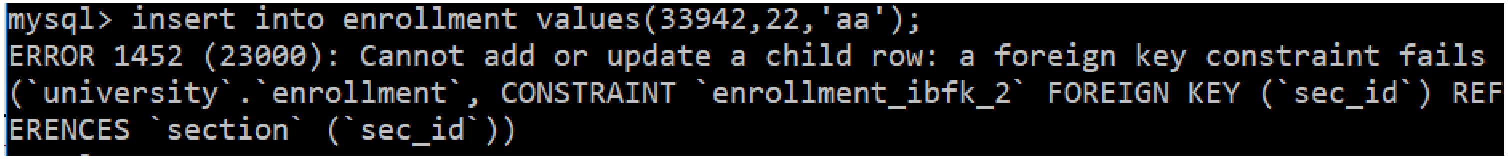


student
stu_id CHAR(9)
stu_fname CHAR(20)
stu_lname CHAR(20)
Indexes



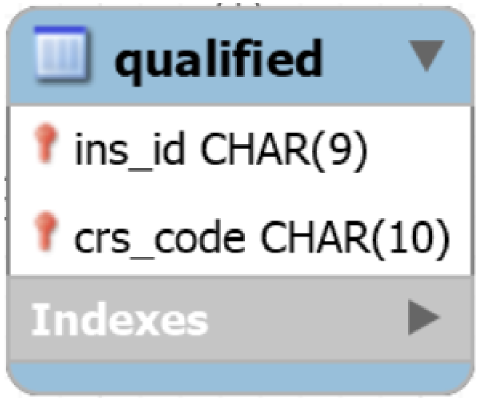
section
sec_id INT(11)
sec_term CHAR(8)
sec_bldg CHAR(6)
sec_room CHAR(4)
sec_time CHAR(10)
crs_code CHAR(10)
loc_code CHAR(5)
ins_id CHAR(9)
Indexes

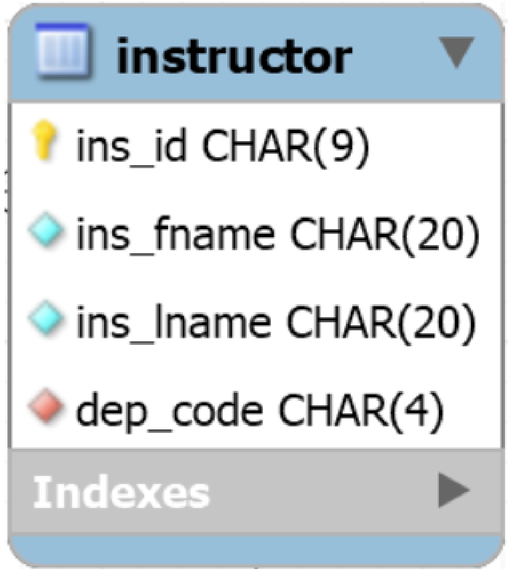
We have no sec_id in section. Foreign key will prevent us from entering a row here

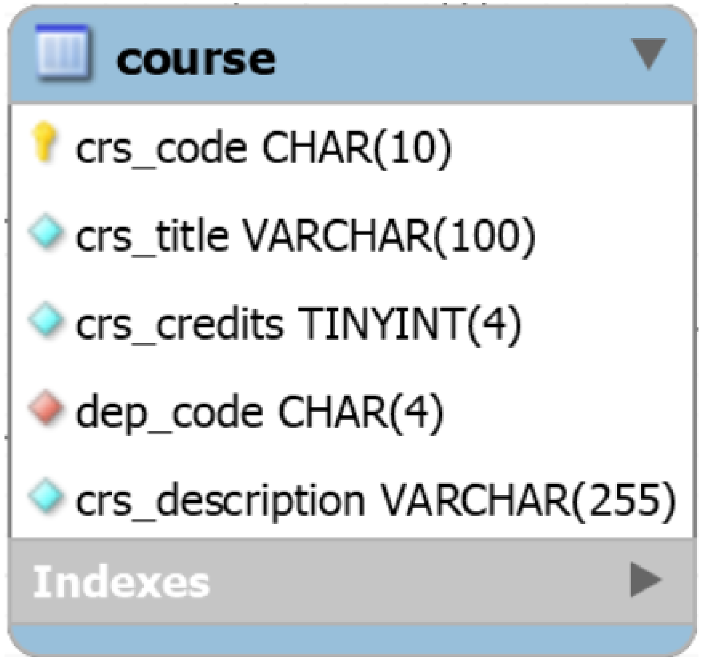


```
mysql> insert into enrollment values(33942,22,'aa');
ERROR 1452 (23000): Cannot add or update a child row: a foreign key constraint fails
(`university`.`enrollment`, CONSTRAINT `enrollment_ibfk_2` FOREIGN KEY (`sec_id`) REF
ERENCES `section` (`sec_id`))
```

Mandatory exercise 6.b

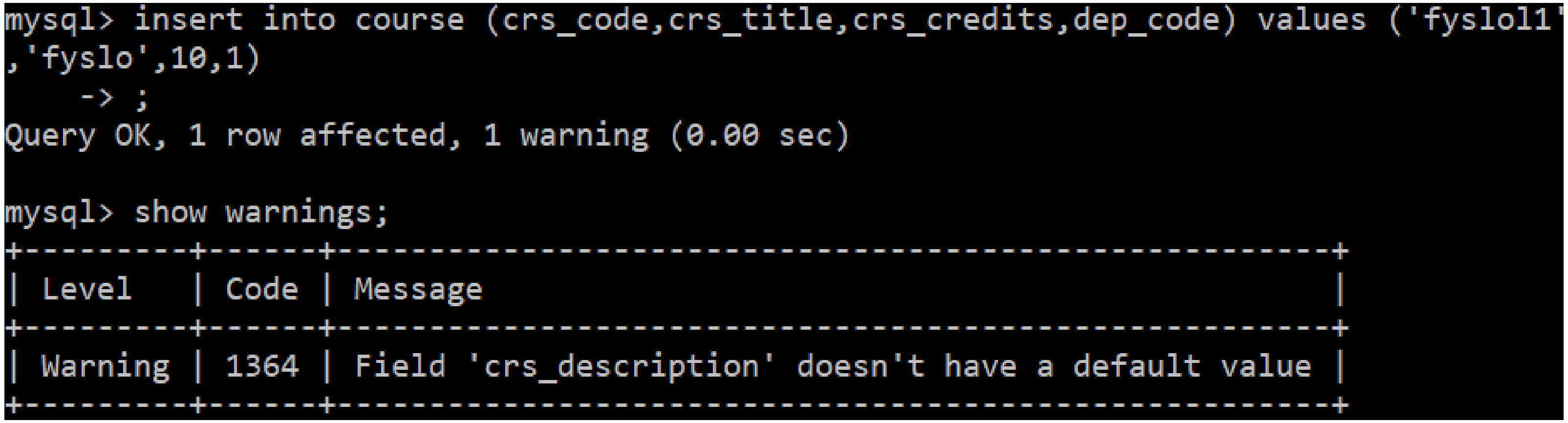






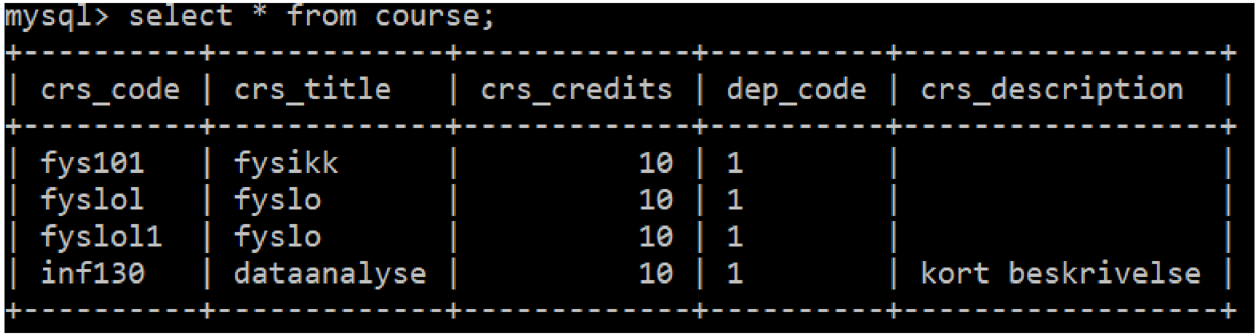
We have registered an instructor and a course, so it is possible to add a qualified instructor

Mandatory exercise 6.d



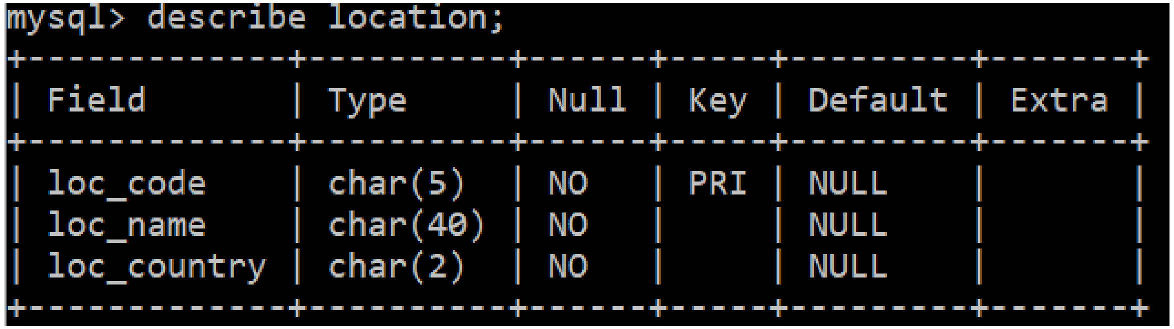
```
mysql> insert into course (crs_code,crs_title,crs_credits,dep_code) values ('fyslol1'
,'fyslo',10,1)
    -> ;
Query OK, 1 row affected, 1 warning (0.00 sec)

mysql> show warnings;
+---------+------+-------------------------------------------------+
| Level   | Code | Message                                         |
+---------+------+-------------------------------------------------+
| Warning | 1364 | Field 'crs_description' doesn't have a default value |
+---------+------+-------------------------------------------------+
```

Mysql adds an empty string for you. Might not give the same result for other database programs



```
mysql> select * from course;
+----------+-------------+-------------+----------+-----------------+
| crs_code | crs_title   | crs_credits | dep_code | crs_description |
+----------+-------------+-------------+----------+-----------------+
| fys101   | fysikk      |          10 | 1        |                 |
| fyslol   | fyslo       |          10 | 1        |                 |
| fyslol1  | fyslo       |          10 | 1        |                 |
| inf130   | dataanalyse |          10 | 1        | kort beskrivelse |
+----------+-------------+-------------+----------+-----------------+
```

Mandatory exercise part 2



```
mysql> describe location;
+-------------+----------+------+-----+---------+-------+
| Field       | Type     | Null | Key | Default | Extra |
+-------------+----------+------+-----+---------+-------+
| loc_code    | char(5)  | NO   | PRI | NULL    |       |
| loc_name    | char(40) | NO   |     | NULL    |       |
| loc_country | char(2)  | NO   |     | NULL    |       |
+-------------+----------+------+-----+---------+-------+
```

Loc_country can only have 2 letters and cannot be NULL.

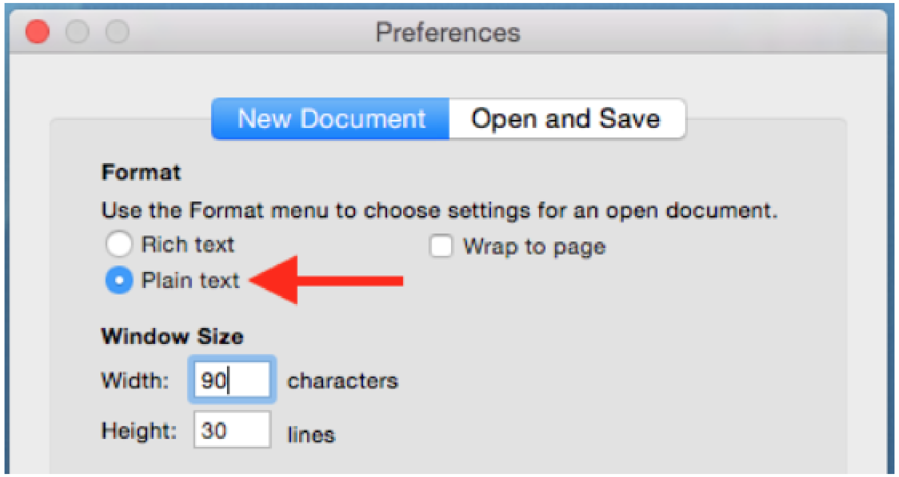
Scripting

- Typing sql in cmd can be a pain
  - Results are not saved
- We can write our code in a plain text editor
  - Run the file with the SOURCE command
- Open university.sql

Scripting

- Open notepad/textedit

TextEdit > Preferences


Preferences

New Document | Open and Save

Format

Use the Format menu to choose settings for an open document.

( ) Rich text
[✓ Plain text]
[ ] Wrap to page

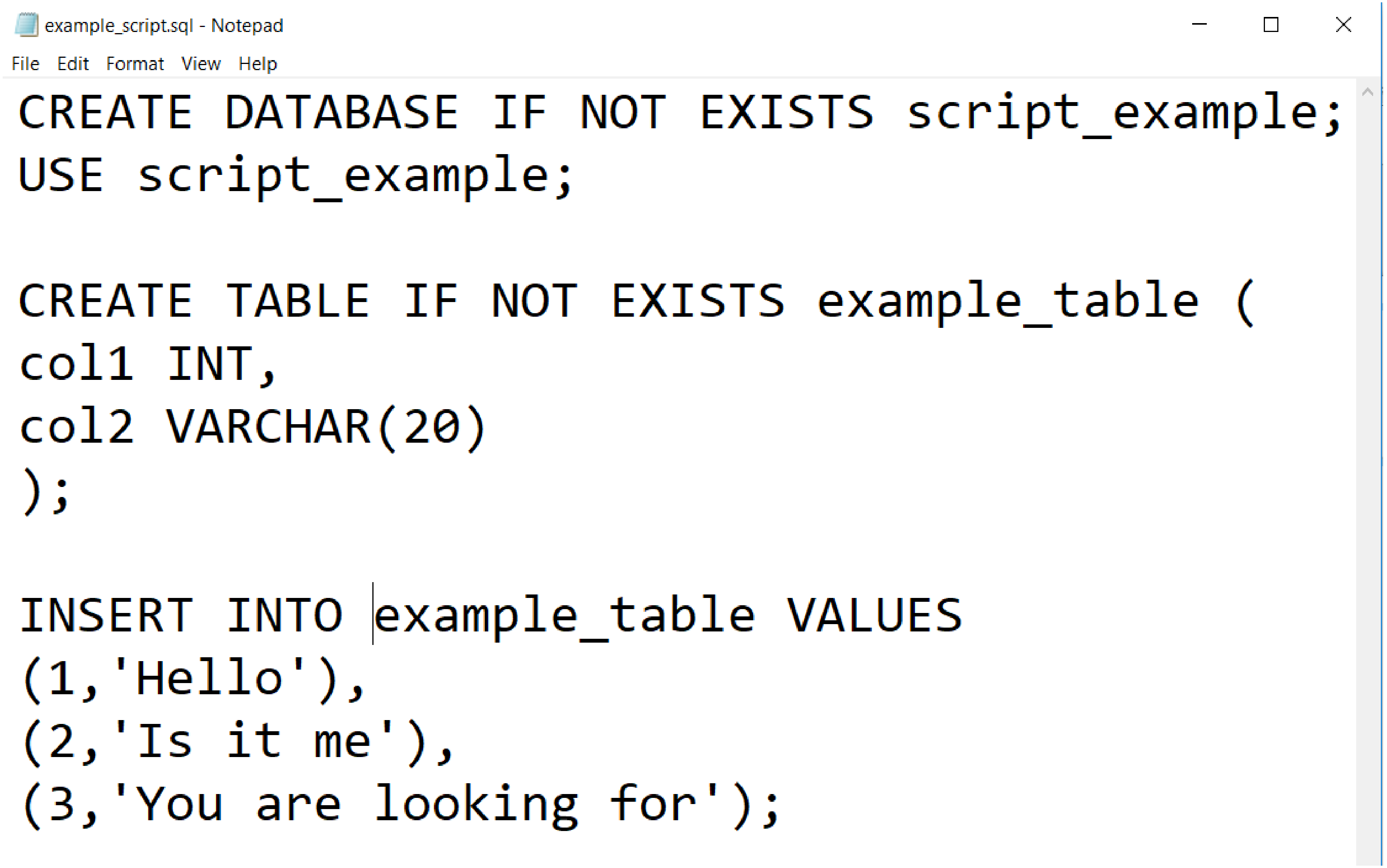
Window Size

Width: 90 characters

Height: 30 lines

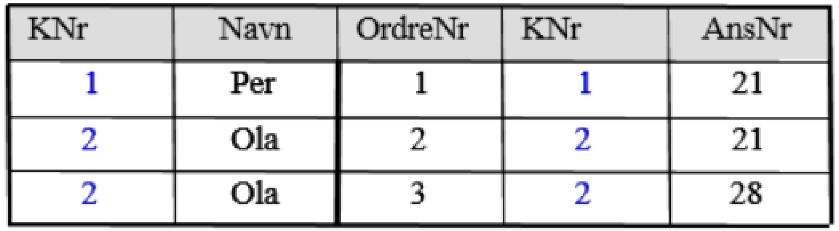
Scripting

- Write your sql code in the text editor
- Save the file
- Navigate to the folder where the file is in cmd/terminal
- Start mysql and run the file with SOURCE filename.sql



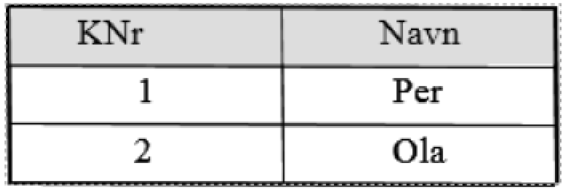
```
CREATE DATABASE IF NOT EXISTS script_example;
USE script_example;

CREATE TABLE IF NOT EXISTS example_table (
col1 INT,
col2 VARCHAR(20)
);

INSERT INTO example_table VALUES
(1,'Hello'),
(2,'Is it me'),
(3,'You are looking for');
```

Joins recap

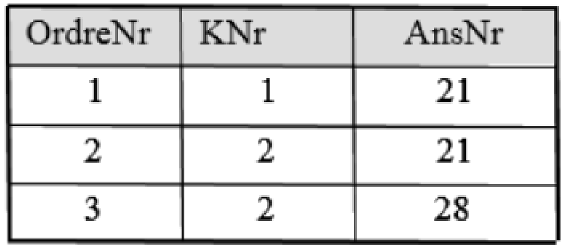


KNr	Navn	OrdreNr	KNr	AnsNr
1	Per	1	1	21
2	Ola	2	2	21
2	Ola	3	2	28

We have split tables to save space (redundancy)

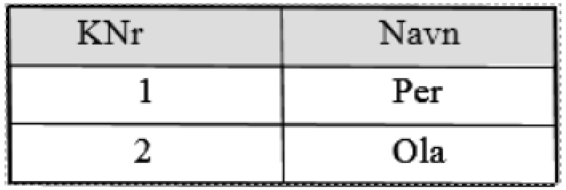


KNr	Navn
1	Per
2	Ola

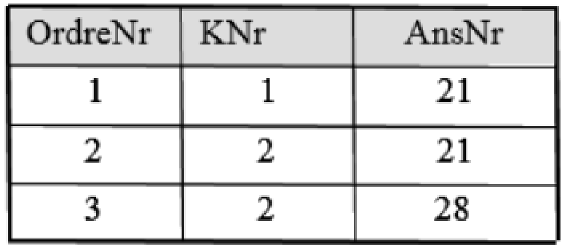


OrdreNr	KNr	AnsNr
1	1	21
2	2	21
3	2	28

Joins recap

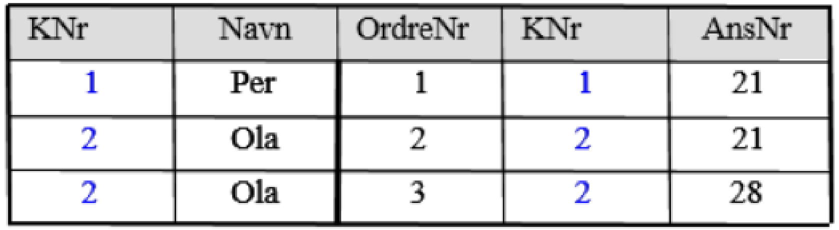


KNr	Navn
1	Per
2	Ola



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

We want the original table back with a query



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

Joins recap

SELECT *

FROM customer, orders

KNr	Navn
1	Per
2	Ola

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

2 × 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

DANGER!!!

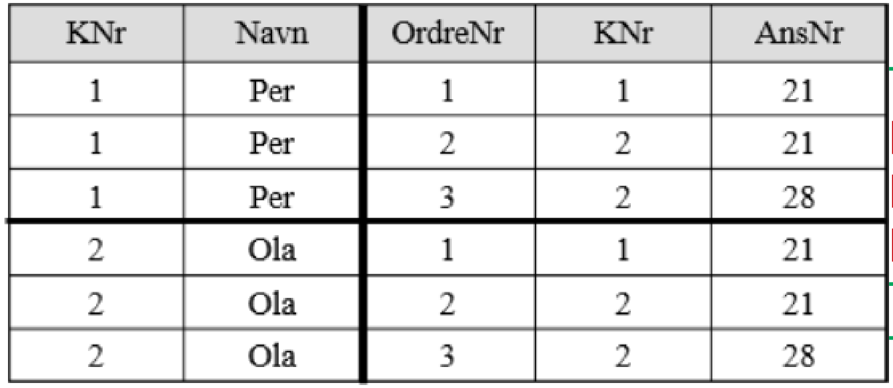
This will combine every row of customer with every row of orders

Joins recap

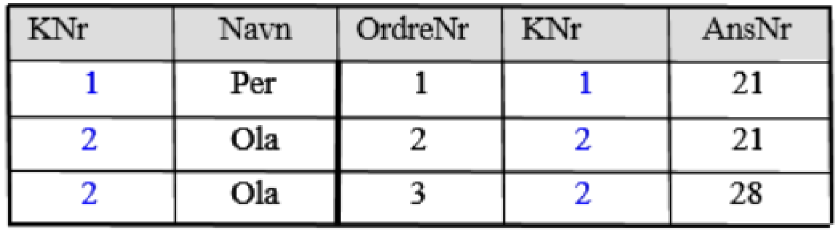
SELECT *

FROM customer, orders

WHERE customer.knr = orders.knr



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



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

Joins recap

Safer way of joining

```
SELECT *
FROM customer INNER JOIN orders
ON customer.knr = orders.knr
```

Joins recap

Other joins

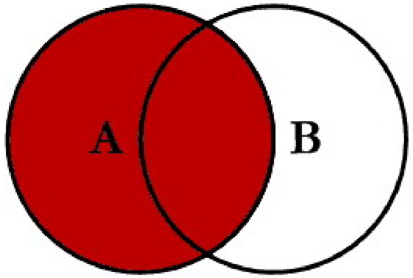
INNER JOIN

FULL OUTER JOIN

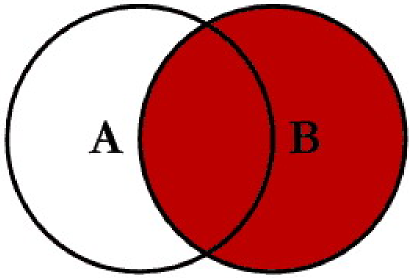
LEFT JOIN

RIGHT JOIN

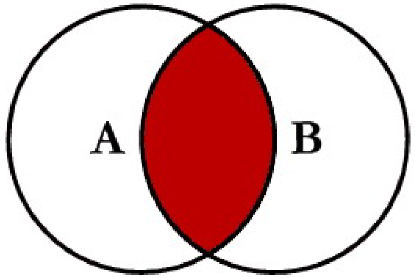
SQL JOINS



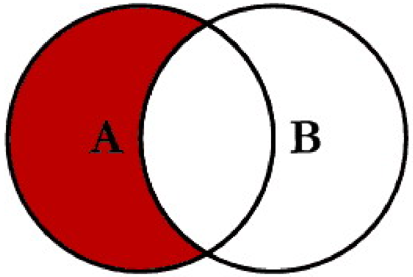
```
SELECT <select_list>
FROM TableA A
LEFT JOIN TableB B
ON A.Key = B.Key
```



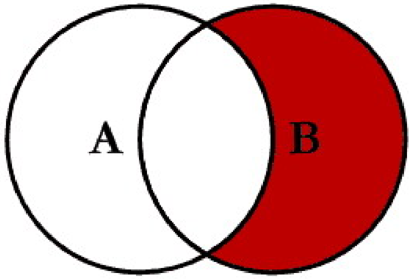
```
SELECT <select_list>
FROM TableA A
RIGHT JOIN TableB B
ON A.Key = B.Key
```



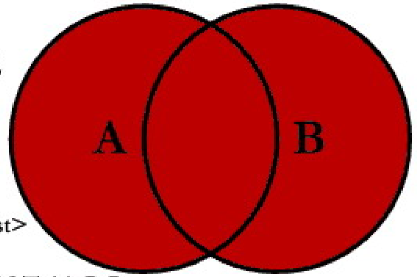
```
SELECT <select_list>
FROM TableA A
INNER JOIN TableB B
ON A.Key = B.Key
```



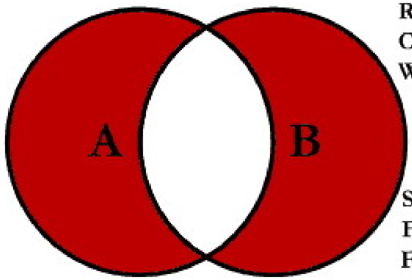
```
SELECT <select_list>
FROM TableA A
LEFT JOIN TableB B
ON A.Key = B.Key
WHERE B.Key IS NULL
```



```
SELECT <select_list>
FROM TableA A
RIGHT JOIN TableB B
ON A.Key = B.Key
WHERE A.Key IS NULL
```



```
SELECT <select_list>
FROM TableA A
FULL OUTER JOIN TableB B
ON A.Key = B.Key
```



```
SELECT <select_list>
FROM TableA A
FULL OUTER JOIN TableB B
ON A.Key = B.Key
WHERE A.Key IS NULL
OR B.Key IS NULL
```

Joins recap

```
SELECT kunde.knr,ordre.ordrenr
FROM kunde INNER JOIN ordre
ON kunde.knr = ordre.knr;
```

Joins recap

```
SELECT kunde.knr,ordre.ordrenr
FROM kunde LEFT JOIN ordre
ON kunde.knr = ordre.knr;
```

Joins recap

```
SELECT kunde.knr,ordre.ordrenr
FROM kunde LEFT JOIN ordre
ON kunde.knr = ordre.knr
WHERE ordrenr is NULL;
```