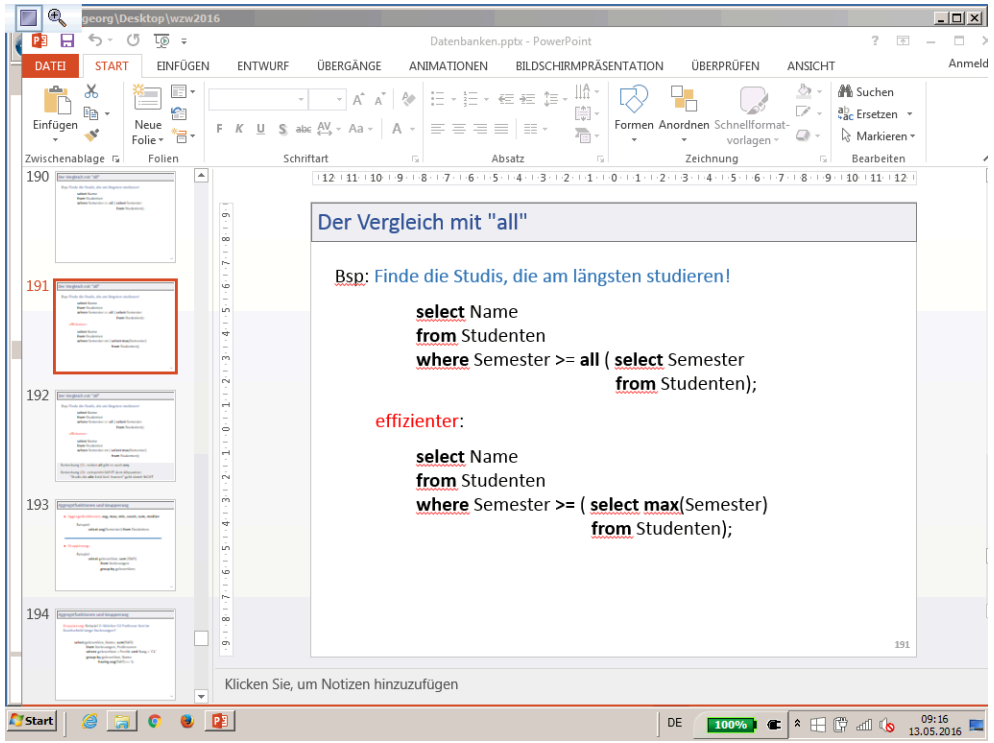
Script **generated by TTT**

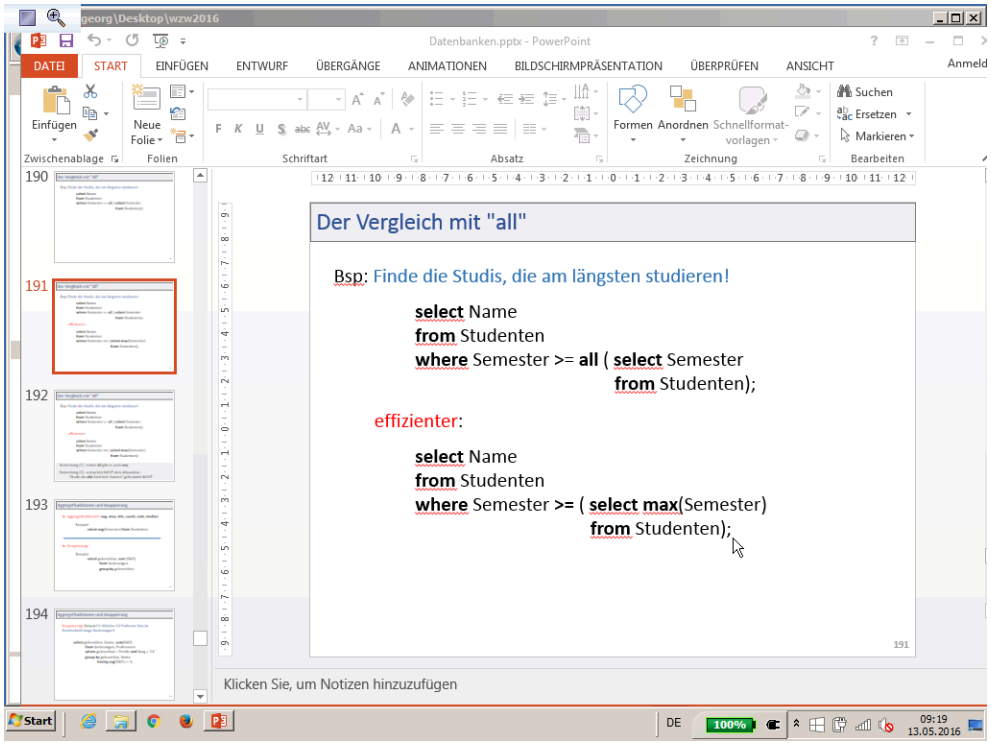
Title: groh: profile1 (13.05.2016)

Date: Fri May 13 09:16:18 CEST 2016

Duration: 87:56 min

Pages: 47





Aggregatfunktionen und Gruppierung

Gruppierung: Beispiel 2: Welcher C4 Professor liest im Durchschnitt lange Vorlesungen?

```
select gelesenVon, Name, sum(SWS)
        from Vorlesungen, Professoren
        where gelesenVon = PersNr and Rang = 'C4'
        group by gelesenVon, Name
                having avg(SWS) >= 3;
```

Aggregatfunktionen und Gruppierung

Gruppierung: Beispiel 2: Welcher C4 Professor liest im Durchschnitt lange Vorlesungen?

Aggregatfkt.: wird für jede Gruppe getrennt berechnet

```
select gelesenVon, Name, sum(SWS)
    from Vorlesungen, Professoren
    where gelesenVon = PersNr and Rang = 'C4'
    group by gelesenVon, Name
        having avg(SWS) >= 3;
```

wie gehabt: sortiert Tupel aus Kreuzprodukt aus (Theta Join bzw. Selektion)

sortiert bstimmte Gruppen aus

$\sigma_{avg(SWS)>3}(\gamma_{gelesenVon, Name; avg(SWS)}(\sigma_{gelesenVon = PersNr \wedge Rang = 'C4'}(Vorlesung \times Professoren)))$

VorlNr	Titel	SWS	gelesen Von	PersNr	Name	Rang	Raum	avg(SWS)
5041	Ethik	4	2125	2125	Sokrates	C4	226	3.333
5049	Mäeutik	2	2125	2125	Sokrates	C4	226	
4052	Logik	4	2125	2125	Sokrates	C4	226	
4630	Die 3 Kritiken	4	2137	2137	Kant	C4	7	4
5001	Grundzüge	4	2137	2137	Kant	C4	7	

```
select gelesenVon, Name, sum(SWS)
    from Vorlesungen, Professoren
    where gelesenVon = PersNr and Rang
    group by gelesenVon, Name
        having avg(SWS) >= 3;
```

Berechnung von Aggregatfunktion sum

$\sigma_{avg(SWS)>3}(\gamma_{gelesenVon, Name; avg(SWS),sum(SWS)}(\sigma_{gelesenVon = PersNr \wedge Rang = 'C4'}(Vorlesung \times Professoren)))$

VorlNr	Titel	SWS	gelesen Von	PersNr	Name	Rang	Raum	avg(SWS)	sum(SWS)
5041	Ethik	4	2125	2125	Sokrates	C4	226	3.333	10
5049	Mäeutik	2	2125	2125	Sokrates	C4	226		
4052	Logik	4	2125	2125	Sokrates	C4	226		
4630	Die 3 Kritiken	4	2137	2137	Kant	C4	7	4	8
5001	Grundzüge	4	2137	2137	Kant	C4	7		

Projektion

$\Pi_{gelesenVon,sum(SWS)}(\sigma_{avg(SWS)>3}(\gamma_{gelesenVon, Name; avg(SWS),sum(SWS)}(\sigma_{gelesenVon = PersNr \wedge Rang = 'C4'}(Vorlesung \times Professoren))))$

gelesen Von	Name	sum(SWS)
2125	Sokrates	10
2137	Kant	8

$\sigma_{avg(SWS)>3}(\gamma_{gelesenVon, Name; avg(SWS)}(\sigma_{gelesenVon = PersNr \wedge Rang = 'C4'}(Vorlesung \times Professoren)))$

VorlNr	Titel	SWS	gelesen Von	PersNr	Name	Rang	Raum	avg(SWS)
5041	Ethik	4	2125	2125	Sokrates	C4	226	3.333
5049	Mäeutik	2	2125	2125	Sokrates	C4	226	
4052	Logik	4	2125	2125	Sokrates	C4	226	
4630	Die 3 Kritiken	4	2137	2137	Kant	C4	7	4
5001	Grundzüge	4	2137	2137	Kant	C4	7	

```
select gelesenVon, Name, sum(SWS)
    from Vorlesungen, Professoren
    where gelesenVon = PersNr and Rang
    group by gelesenVon, Name
        having avg(SWS) >= 3;
```

Berechnung von Aggregatfunktion sum

$\sigma_{avg(SWS)>3}(\gamma_{gelesenVon, Name; avg(SWS),sum(SWS)}(\sigma_{gelesenVon = PersNr \wedge Rang = 'C4'}(Vorlesung \times Professoren)))$

VorlNr	Titel	SWS	gelesen Von	PersNr	Name	Rang	Raum	avg(SWS)	sum(SWS)
5041	Ethik	4	2125	2125	Sokrates	C4	226	3.333	10
5049	Mäeutik	2	2125	2125	Sokrates	C4	226		
4052	Logik	4	2125	2125	Sokrates	C4	226		
4630	Die 3 Kritiken	4	2137	2137	Kant	C4	7	4	8
5001	Grundzüge	4	2137	2137	Kant	C4	7		

Projektion

$\Pi_{gelesenVon,sum(SWS)}(\sigma_{avg(SWS)>3}(\gamma_{gelesenVon, Name; avg(SWS),sum(SWS)}(\sigma_{gelesenVon = PersNr \wedge Rang = 'C4'}(Vorlesung \times Professoren))))$

gelesen Von	Name	sum(SWS)
2125	Sokrates	10
2137	Kant	8

Geschachtelte Anfragen

3. Unteranfrage in der **from**-Klausel:

--> Verwertung der Ergebnismenge einer Unteranfrage:

Beispiel: Wer hört mehr als 2 Vorlesungen?

```
select tmp.MatrNr, tmp.Name, tmp.VorlAnzahl
from ( select s.MatrNr, s.Name, count(*) as VorlAnzahl
       from Studenten s, hören h
       where s.MatrNr=h.MatrNr
       group by s.MatrNr, s.Name ) tmp
where tmp.VorlAnzahl > 2;
```


Allquantifizierung durch count-Aggregation

- Allquantifizierung kann immer auch durch eine **count**-Aggregation ausgedrückt werden.
- Bsp: MatrNr der Studenten, die *alle* Vorlesungen hören:

```
select h.MatrNr
  from hören h
  group by h.MatrNr
       having count (*) = (select count (*) from Vorlesungen);
```

(count(*) nach having zählt alle Tupel einer Gruppe)

Allquantifizierung durch count-Aggregation

Bsp: MatrNr der Studenten die alle vierstündigen Vorlesungen hören:

```
select s.MatrNr
from Studenten s
where not exists
      ( select v.*
        from Vorlesungen v
        where v.SWS=4 and not exists
              ( select h.*
                from hören h
                where h.VorlNr = v.VorlNr and h.MatrNr = s.MatrNr ) );
```

oder:

```
select h.MatrNr
from hören h, Vorlesungen v
where h.VorlNr = v.VorlNr and h.MatrNr = s.MatrNr and v.SWS = 4
group by h.MatrNr
      having count (*) = ( select count (*)
                           from Vorlesungen v
                           where v.SWS=4 );
```


Allquantifizierung durch count-Aggregation

Bsp: MatrNr der Studenten die alle vierstündigen Vorlesungen hören:

```
select s.MatrNr
from Studenten s
where not exists
      ( select v.*
        from Vorlesungen v
        where v.SWS=4 and not exists
              ( select h.*
                from hören h
                where h.VorlNr = v.VorlNr and h.MatrNr = s.MatrNr ) );
```

folgt sinngemäß der logischen Identität $\forall x\, Bed(x) \equiv \neg\exists x\, \neg Bed(x)$
(wobei x eine Variable und $Bed(x)$ ein prädikatenlogischer Ausdruck ist)

Allquantifizierung durch count-Aggregation

Bsp: MatrNr der Studenten die alle vierstündigen Vorlesungen hören:

```
select s.MatrNr
from Studenten s
where not exists
      ( select v.*
        from Vorlesungen v
        where v.SWS=4 and not exists
              ( select h.*
                from hören h
                where h.VorlNr = v.VorlNr and h.MatrNr = s.MatrNr ) );
```

folgt sinngemäß der logischen Identität $\forall x\ Bed(x) \equiv \neg\exists x\ \neg Bed(x)$
(wobei x eine Variable und $Bed(x)$ ein prädikatenlogischer Ausdruck ist)

Allquantifizierung durch count-Aggregation

Bsp: MatrNr der Studenten die alle vierstündigen Vorlesungen hören:

```
select s.MatrNr
from Studenten s
where not exists
      ( select v.*
        from Vorlesungen v
        where v.SWS=4 and not exists
              ( select h.*
                from hören h
                where h.VorlNr = v.VorlNr and h.MatrNr = s.MatrNr ) );
```

folgt sinngemäß der logischen Identität $\forall x\ Bed(x) \equiv \neg\exists x\ \neg Bed(x)$
(wobei x eine Variable und $Bed(x)$ ein prädikatenlogischer Ausdruck ist)

208

Allquantifizierung durch count-Aggregation

Bsp: MatrNr der Studenten die alle vierstündigen Vorlesungen hören:

```
select s.MatrNr
from Studenten s
where not exists
      ( select v.*
        from Vorlesungen v
        where v.SWS=4 and not exists
              ( select h.*
                from hören h
                where h.VorlNr = v.VorlNr and h.MatrNr = s.MatrNr ) );
```

folgt sinngemäß der logischen Identität $\forall x\ Bed(x) \equiv \neg\exists x\ \neg Bed(x)$
(wobei x eine Variable und $Bed(x)$ ein prädikatenlogischer Ausdruck ist)

208

Allquantifizierung durch count-Aggregation

Bsp: MatrNr der Studenten die alle vierstündigen Vorlesungen hören:

```
select s.MatrNr
from Studenten s
where not exists
      ( select v.*
        from Vorlesungen v
        where v.SWS=4 and not exists
              ( select h.*
                from hören h
                where h.VorlNr = v.VorlNr and h.MatrNr = s.MatrNr ) );
```

oder:

```
select h.MatrNr
from hören h, Vorlesungen v
where h.VorlNr = v.VorlNr and h.MatrNr = s.MatrNr and v.SWS = 4
group by h.MatrNr
      having count (*) = ( select count (*)
                           from Vorlesungen v
                           where v.SWS=4 );
```

209

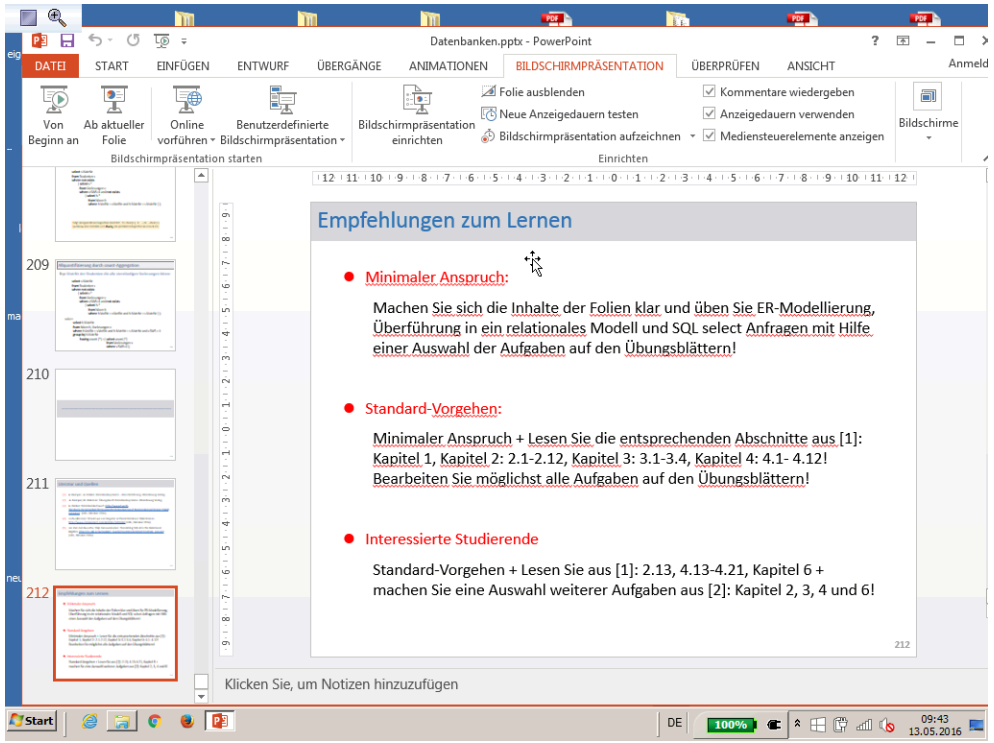
Allquantifizierung durch count-Aggregation

Bsp: MatrNr der Studenten die alle vierstündigen Vorlesungen hören:

```
select s.MatrNr
from Studenten s
where not exists
      ( select v.*
        from Vorlesungen v
        where v.SWS=4 and not exists
              ( select h.*
                from hören h
                where h.VorlNr = v.VorlNr and h.MatrNr = s.MatrNr ) );
```

oder:

```
select h.MatrNr
from hören h, Vorlesungen v
where h.VorlNr = v.VorlNr and h.MatrNr = s.MatrNr and v.SWS = 4
group by h.MatrNr
      having count (*) = ( select count (*)
                           from Vorlesungen v
                           where v.SWS=4 );
```

Technische Universität München

Einführung in die Informatik
für Hörer anderer Fachrichtungen (WZW)
IN8003

PD Dr. Georg Groh

Social Computing
Research Group

Fakultät für Informatik

Aufgaben

Professoren

PersNr	Name	Rang	Raum
2125	Sokrates	C4	226
2126	Russel	C4	232
2127	Kopernikus	C3	310
2133	Popper	C3	52
2134	Augustinus	C3	309
2136	Curie	C4	36
2137	Kant	C4	7

Studenten

MatrNr	Name	Semester
24002	Xenokrates	18
25403	Jonas	12
26120	Fichte	10
26830	Aristoxenos	8
27550	Schopenhauer	6
28106	Carnap	3
29120	Theophrastos	2
29555	Feuerbach	2

Vorlesungen

VorlNr	Titel	SWS	gelesenVon
5001	Grundzüge	4	2137
5041	Ethik	4	2125
5043	Erkenntnistheorie	3	2126
5049	Mäeutik	2	2125
4052	Logik	4	2125
5052	Wissenschaftstheorie	3	2126
5216	Bioethik	2	2126
5259	Der Wiener Kreis	2	2133
5022	Glaube und Wissen	2	2134
4630	Die 3 Kritiken	4	2137

voraussetzen

Vorgänger	Nachfolger
5001	5041
5001	5043
5001	5049
5041	5216
5043	5052
5041	5052
5052	5259

hören

MatrNr	VorlNr
26120	5001
27550	5001
27550	4052
28106	5041
28106	5052
28106	5216
28106	5259
29120	5001
29120	5041
29120	5049
29555	5022
25403	5022
29555	5001

A

PersNr	Name
3002	Platon
3003	Aristoteles
3004	Wittgenstei
3005	Rhetikus
3006	Newton
3007	Spinoza

Keplersche Gesetze	2127
Gott und Natur	2134

prüfen

MatrNr	VorlNr	PersNr	Note
28106	5001	2126	1
25403	5041	2125	2
27550	4630	2137	2

Bestimmen Sie mit einem `select` statement, **welche Professoren welche Assistenten** haben. **Spalten** der Ergebnistabelle: **Name** Professor, **Name** Assistent.

```
select p.Name, a.Name
       from Assistenten a, Professoren p
       where a.Boss = p.PersNr ;
```

Aufgaben

Professoren			
PersNr	Name	Rang	Raum
2125	Sokrates	C4	226

Studenten		
MatrNr	Name	Semester
24002	Xenokrates	18

Formulieren Sie eine **SQL-Anfrage**, aus der die Namen der Studenten hervorgehen, die mindestens eine Vorlesung mit drei Semesterwochenstunden **und** mindestens eine Vorlesung mit vier Semesterwochenstunden gehört haben! Die Ergebnisrelation soll das Attribut *NameStudent* besitzen.

Variante 1 (per Join):

```
select distinct(s.Name) as NameStudent
from Studenten s, hören h, hören hh, Vorlesungen v, Vorlesungen vv
where s.MatrNr = h.MatrNr and
      s.MatrNr = hh.MatrNr and
      h.VorlNr = v.VorlNr and
      v.SWS = 3 and
      hh.VorlNr = vv.VorlNr and
      vv.SWS = 4
```

oder

Variante 2 (per Unteranfragen):

```
select distinct(s.Name) as NameStudent
from Studenten s, hören h, hören hh
where s.MatrNr = h.MatrNr and
      h.VorlNr in  (select v.VorlNr
                    from Vorlesungen v
                    where v.SWS = 3
                   ) and
      hh.VorlNr in (select v.VorlNr
                    from Vorlesungen v
                    where v.SWS = 4
                   )
```


Aufgaben

Professoren			
PersNr	Name	Rang	Raum
2125	Sokrates	C4	226

Studenten		
MatrNr	Name	Semester
24002	Xenokrates	18

Formulieren Sie eine **SQL-Anfrage**, aus der die Namen der Studenten hervorgehen, die mindestens eine Vorlesung mit drei Semesterwochenstunden **und** mindestens eine Vorlesung mit vier Semesterwochenstunden gehört haben! Die Ergebnisrelation soll das Attribut *NameStudent* besitzen.

Variante 1 (per Join):

```
select distinct(s.Name) as NameStudent
from Studenten s, hören h, hören hh, Vorlesungen v, Vorlesungen vv
where s.MatrNr = h.MatrNr and
      s.MatrNr = hh.MatrNr and
      h.VorlNr = v.VorlNr and
      v.SWS = 3 and
      hh.VorlNr = vv.VorlNr and
      vv.SWS = 4
```

oder

Variante 2 (per Unteranfragen):

```
select distinct(s.Name) as NameStudent
from Studenten s, hören h, hören hh
where s.MatrNr = h.MatrNr and
      h.VorlNr in  (select v.VorlNr
                    from Vorlesungen v
                    where v.SWS = 3
                   ) and
      hh.VorlNr in (select v.VorlNr
                    from Vorlesungen v
                    where v.SWS = 4
                   )
```

220

Aufgaben

Professoren			
PersNr	Name	Rang	Raum
2125	Sokrates	C4	226

Studenten		
MatrNr	Name	Semester
24002	Xenokrates	18

Formulieren Sie eine **SQL-Anfrage**, aus der die Namen der Studenten hervorgehen, die mindestens eine Vorlesung mit drei Semesterwochenstunden **und** mindestens eine Vorlesung mit vier Semesterwochenstunden gehört haben! Die Ergebnisrelation soll das Attribut *NameStudent* besitzen.

Variante 1 (per Join):

```
select distinct(s.Name) as NameStudent
from Studenten s, hören h, hören hh, Vorlesungen v, Vorlesungen vv
where s.MatrNr = h.MatrNr and
      s.MatrNr = hh.MatrNr and
      h.VorlNr = v.VorlNr and
      v.SWS = 3 and
      hh.VorlNr = vv.VorlNr and
      vv.SWS = 4
```

oder

Variante 2 (per Unteranfragen):

```
select distinct(s.Name) as NameStudent
from Studenten s, hören h, hören hh
where s.MatrNr = h.MatrNr and
      h.VorlNr in  (select v.VorlNr
                    from Vorlesungen v
                    where v.SWS = 3
                   ) and
      hh.VorlNr in (select v.VorlNr
                    from Vorlesungen v
                    where v.SWS = 4
                   )
```

220

Aufgaben

Professoren			
PersNr	Name	Rang	Raum
2125	Sokrates	C4	226

Studenten		
MatrNr	Name	Semester
24002	Xenokrates	18

Formulieren Sie eine **SQL-Anfrage**, aus der die Namen der Studenten hervorgehen, die mindestens eine Vorlesung mit drei Semesterwochenstunden **und** mindestens eine Vorlesung mit vier Semesterwochenstunden gehört haben! Die Ergebnisrelation soll das Attribut *NameStudent* besitzen.

Variante 1 (per Join):

```
select distinct(s.Name) as NameStudent
from Studenten s, hören h, hören hh, Vorlesungen v, Vorlesungen vv
where s.MatrNr = h.MatrNr and
      s.MatrNr = hh.MatrNr and
      h.VorlNr = v.VorlNr and
      v.SWS = 3 and
      hh.VorlNr = vv.VorlNr and
      vv.SWS = 4
```

oder

Variante 2 (per Unteranfragen):

```
select distinct(s.Name) as NameStudent
from Studenten s, hören h, hören hh
where s.MatrNr = h.MatrNr and
      h.VorlNr in  (select v.VorlNr
                    from Vorlesungen v
                    where v.SWS = 3
                   ) and
      hh.VorlNr in (select v.VorlNr
                    from Vorlesungen v
                    where v.SWS = 4
                   )
```

220

Aufgaben

Professoren			
PersNr	Name	Rang	Raum
2125	Sokrates	C4	226

Studenten		
MatrNr	Name	Semester
24002	Xenokrates	18

Formulieren Sie eine **SQL-Anfrage**, aus der die Namen der Studenten hervorgehen, die mindestens eine Vorlesung mit drei Semesterwochenstunden **und** mindestens eine Vorlesung mit vier Semesterwochenstunden gehört haben! Die Ergebnisrelation soll das Attribut *NameStudent* besitzen.

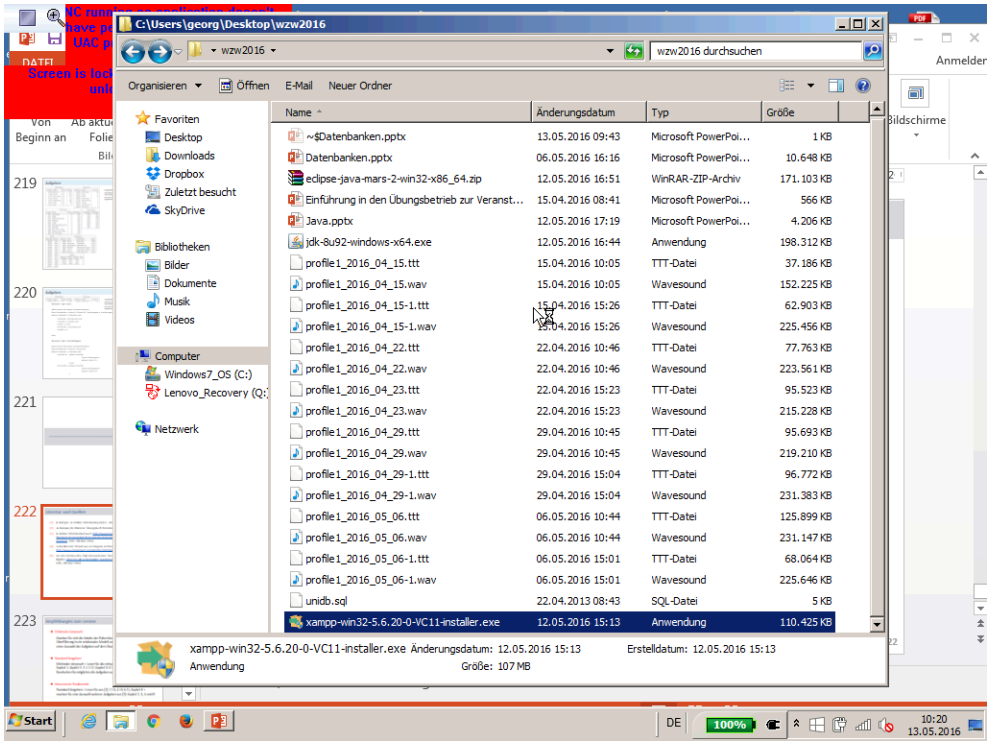
Variante 1 (per Join):

```
select distinct(s.Name) as NameStudent
from Studenten s, hören h, hören hh, Vorlesungen v, Vorlesungen vv
where s.MatrNr = h.MatrNr and
      s.MatrNr = hh.MatrNr and
      h.VorlNr = v.VorlNr and
      v.SWS = 3 and
      hh.VorlNr = vv.VorlNr and
      vv.SWS = 4
```

oder

Variante 2 (per Unteranfragen):

```
select distinct(s.Name) as NameStudent
from Studenten s, hören h, hören hh
where s.MatrNr = h.MatrNr and
      h.VorlNr in   (select v.VorlNr
                     from Vorlesungen v
                     where v.SWS = 3
                ) and
      hh.VorlNr in (select v.VorlNr
                     from Vorlesungen v
                     where v.SWS = 4
                )
```

Aufgaben

Professoren			
PersNr	Name	Rang	Raum
2125	Sokrates	C4	226

Studenten		
MatrNr	Name	Semester
24002	Xenokrates	18

Formulieren Sie eine **SQL-Anfrage**, aus der die Namen der Studenten hervorgehen, die mindestens eine Vorlesung mit drei Semesterwochenstunden **und** mindestens eine Vorlesung mit vier Semesterwochenstunden gehört haben! Die Ergebnisrelation soll das Attribut *NameStudent* besitzen.

Variante 1 (per Join):

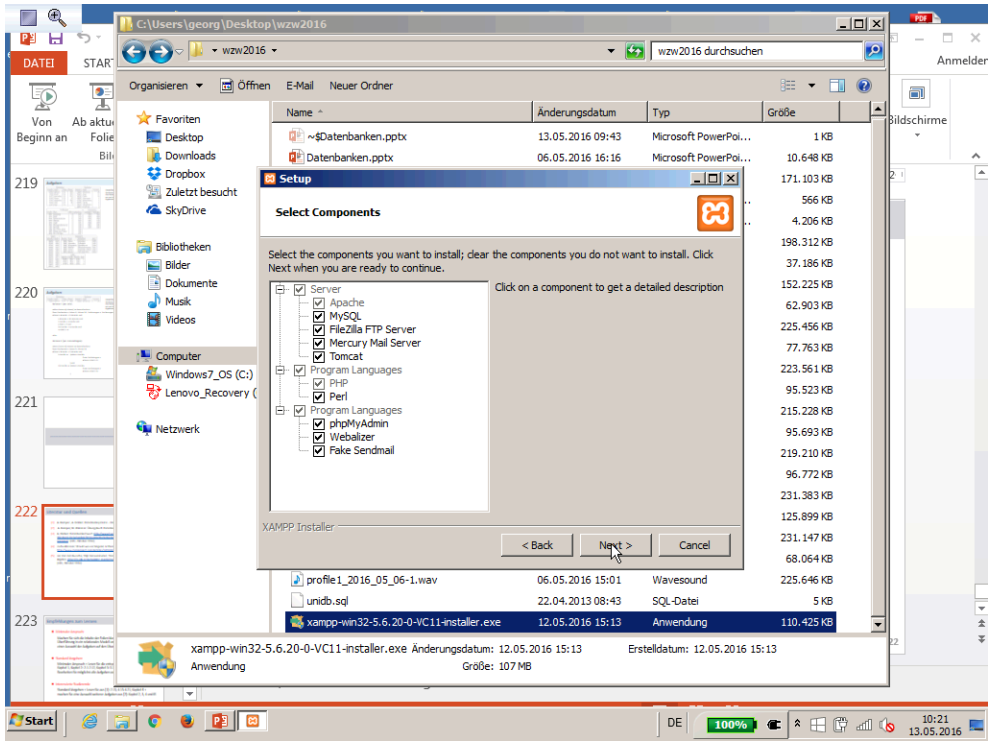
```
select distinct(s.Name) as NameStudent
from Studenten s, hören h, hören hh, Vorlesungen v, Vorlesungen vv
where s.MatrNr = h.MatrNr and
      s.MatrNr = hh.MatrNr and
      h.VorlNr = v.VorlNr and
      v.SWS = 3 and
      hh.VorlNr = vv.VorlNr and
      vv.SWS = 4
```

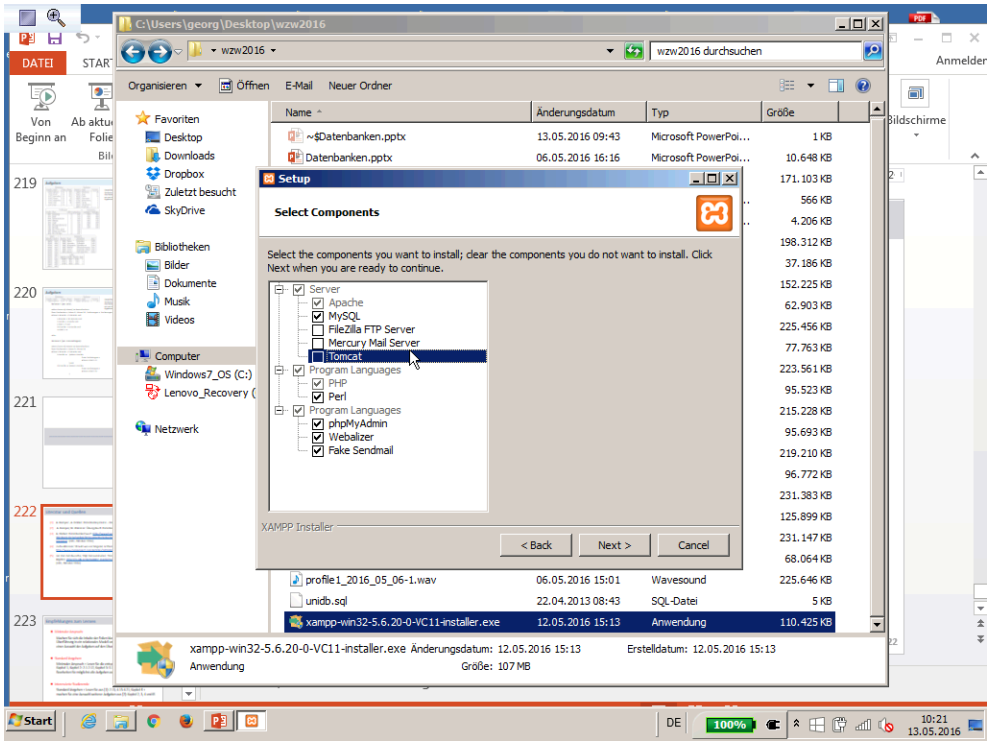
oder

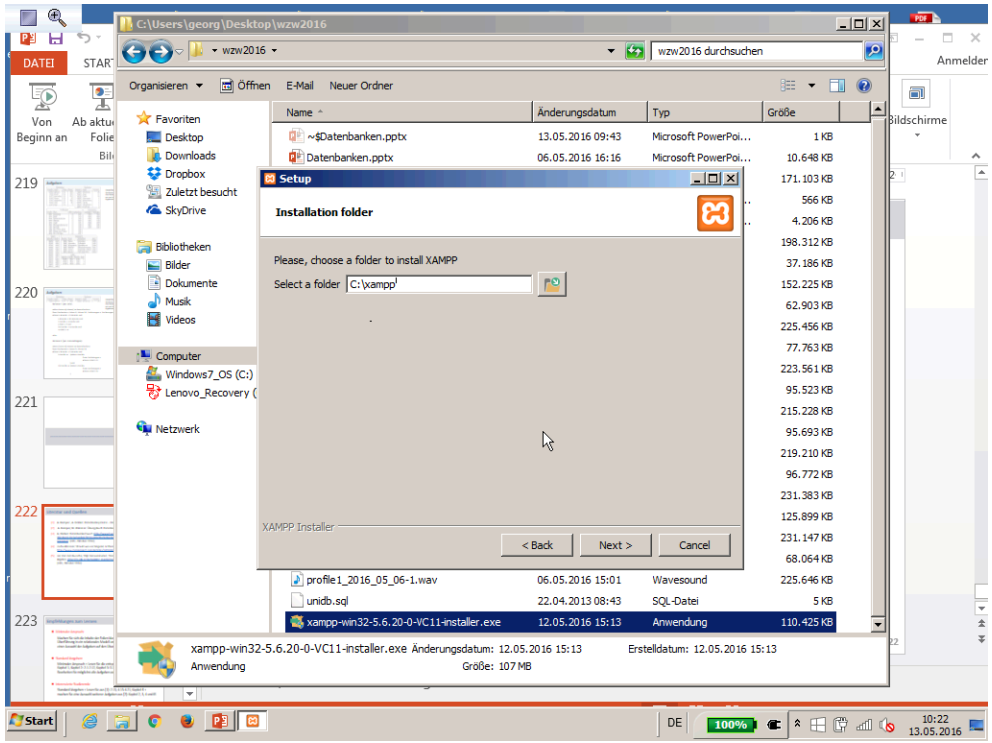
Variante 2 (per Unteranfragen):

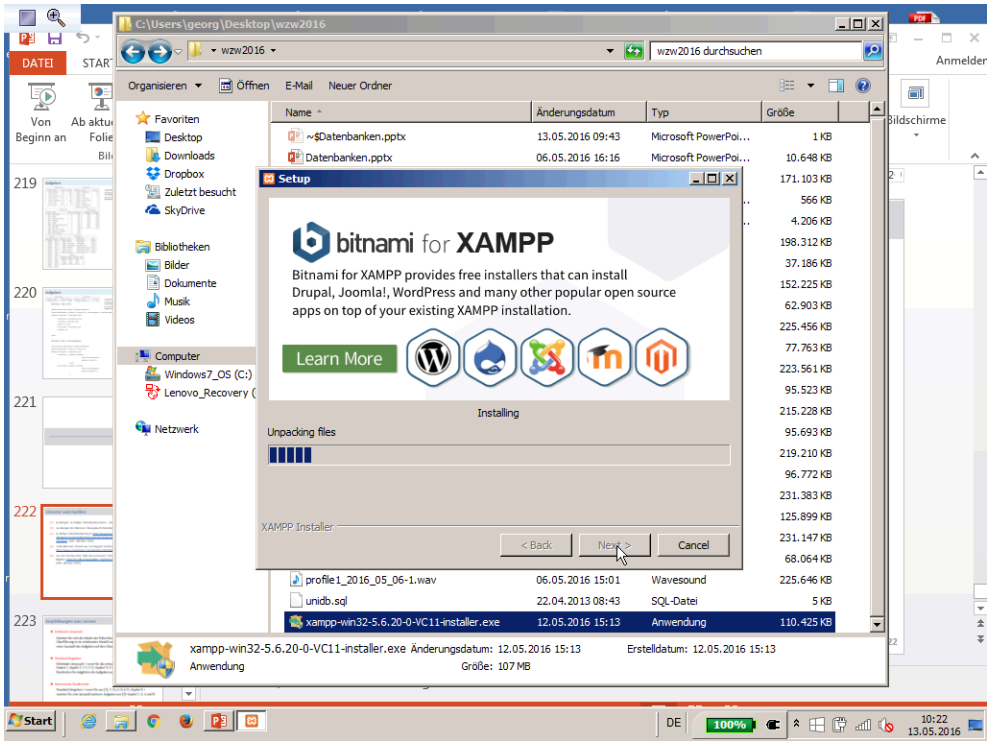
```
select distinct(s.Name) as NameStudent
from Studenten s, hören h, hören hh
where s.MatrNr = h.MatrNr and
      h.VorlNr in   (select v.VorlNr
                     from Vorlesungen v
                     where v.SWS = 3
                ) and
      hh.VorlNr in (select v.VorlNr
                     from Vorlesungen v
                     where v.SWS = 4
                )
```

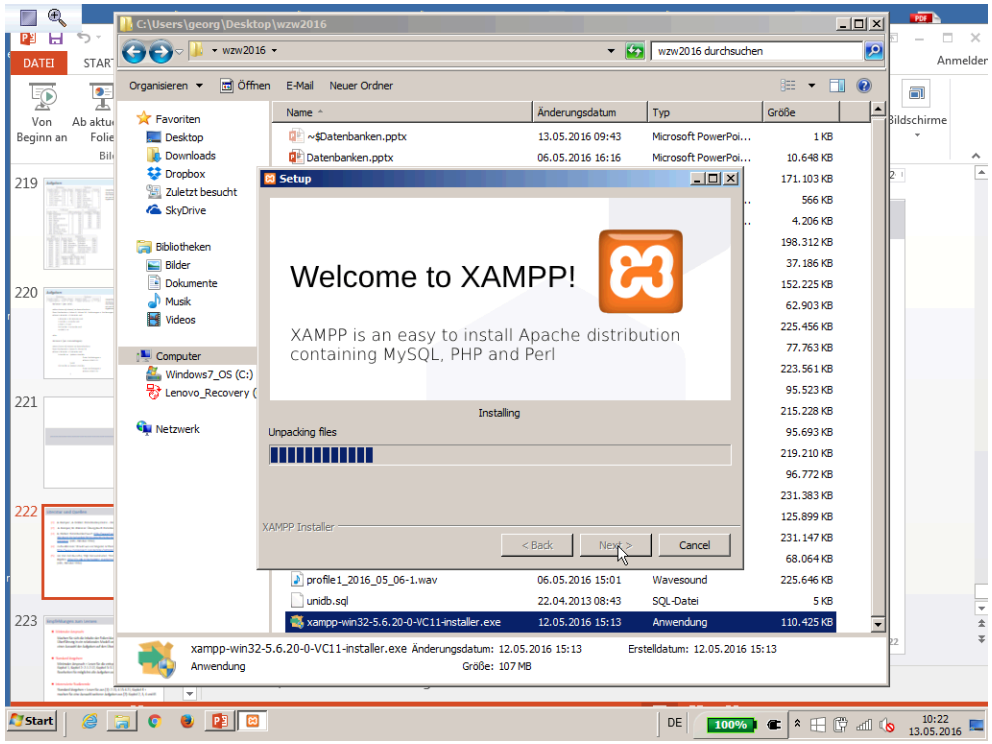
220

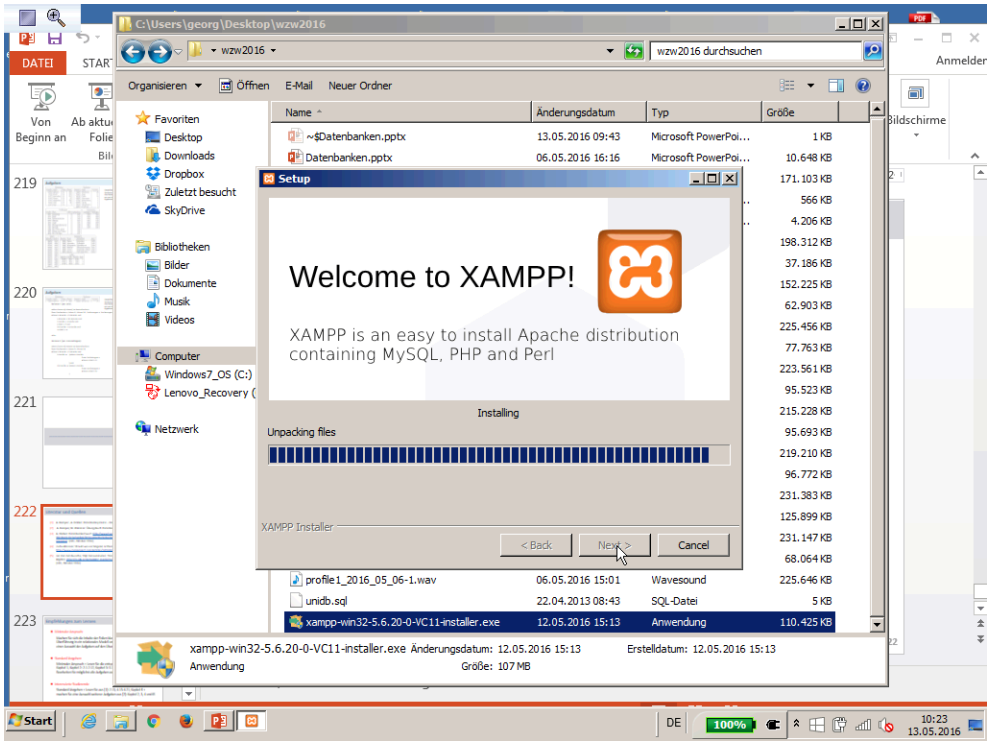


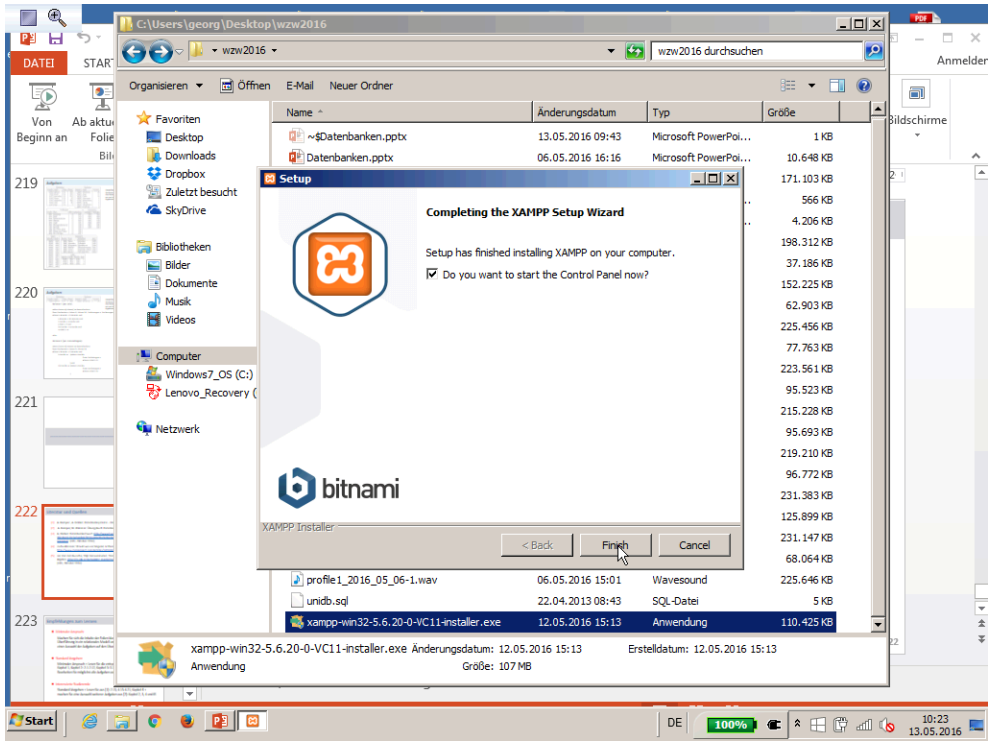

Setup
Select Components
Select the components you want to install; clear the components you do not want to install. Click Next when you are ready to continue.
Server
Apache
MySQL
FileZilla FTP Server
Mercury Mail Server
Tomcat
Program Languages
PHP
Perl
Program Languages
phpMyAdmin
Webalizer
Fake Sendmail
XAMPP Installer
< Back
Next >
Cancel

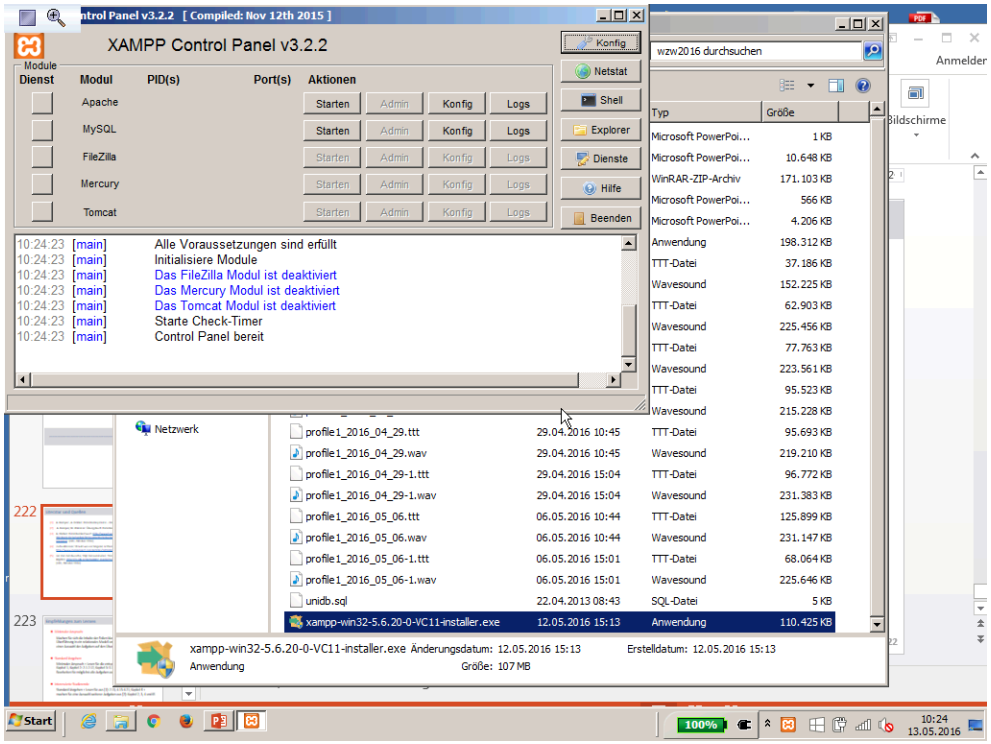

Setup
Installation folder
Please, choose a folder to install XAMPP
Select a folder
C:\xampp
XAMPP Installer
< Back
Next >
Cancel

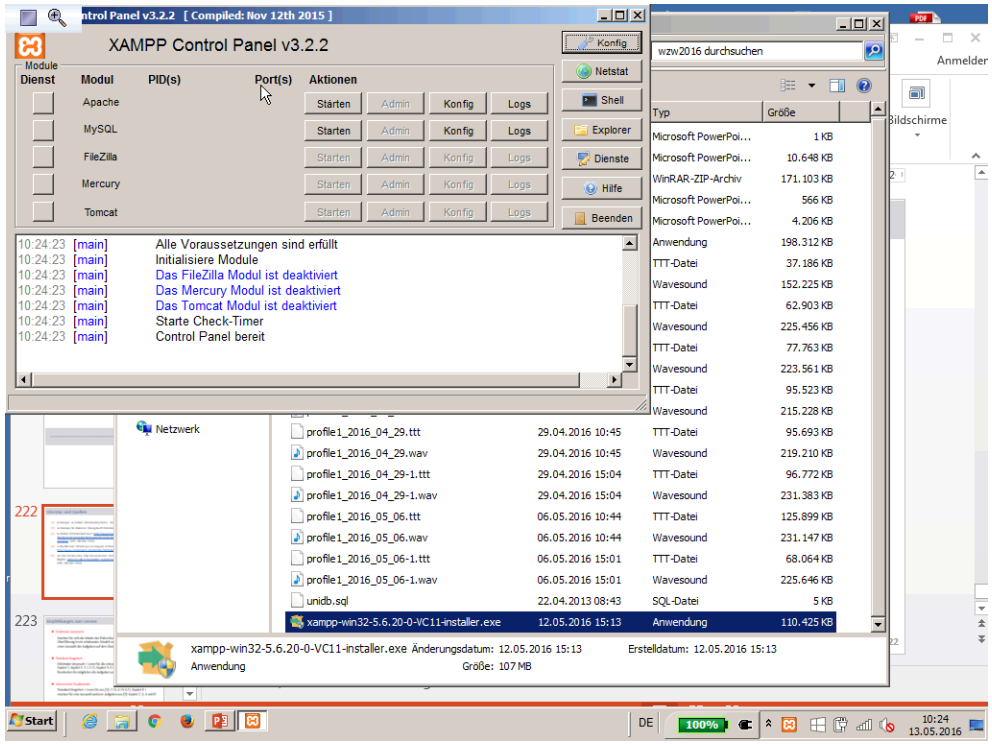

Setup
bitnami for XAMPP
Bitnami for XAMPP provides free installers that can install Drupal, Joomla!, WordPress and many other popular open source apps on top of your existing XAMPP installation.
Learn More
Installing
Unpacking files
XAMPP Installer
< Back
Next >
Cancel

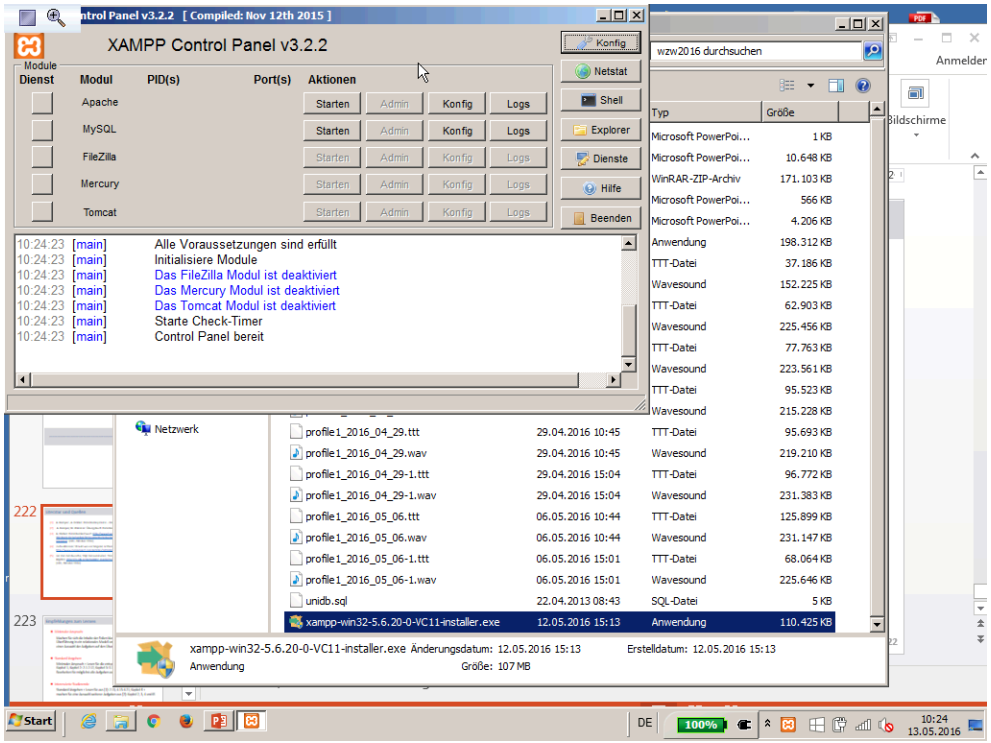

Setup
Welcome to XAMPP!
XAMPP is an easy to install Apache distribution containing MySQL, PHP and Perl
Installing
Unpacking files
XAMPP Installer
< Back
Next >
Cancel

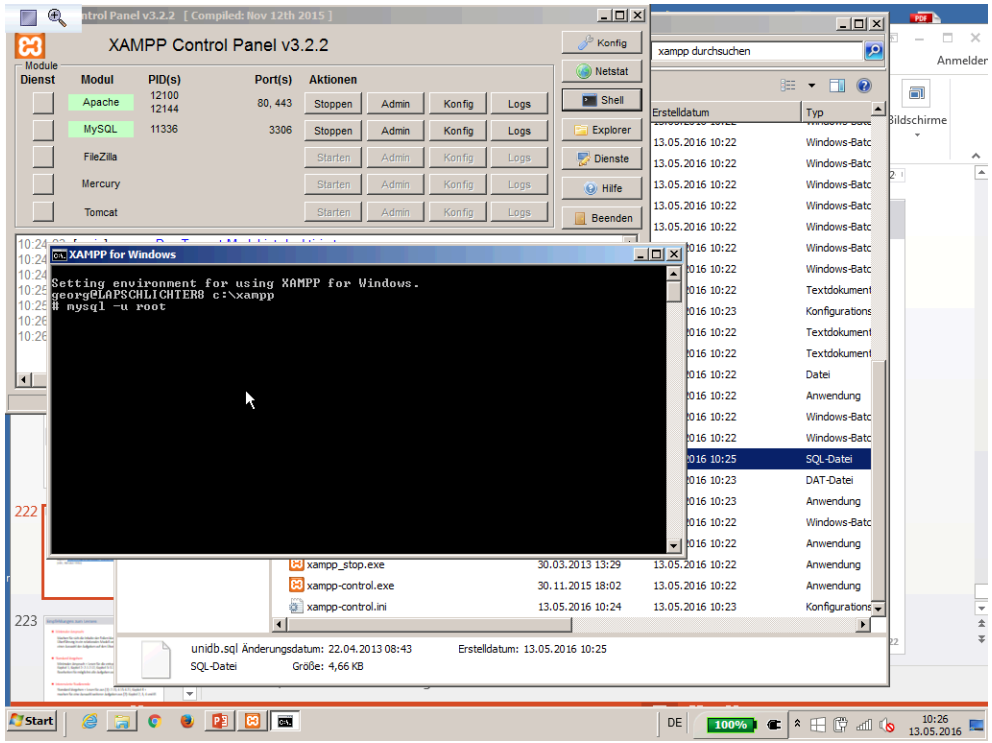

Setup
Welcome to XAMPP!
XAMPP is an easy to install Apache distribution containing MySQL, PHP and Perl
Installing
Unpacking files
XAMPP Installer
< Back
Next >
Cancel

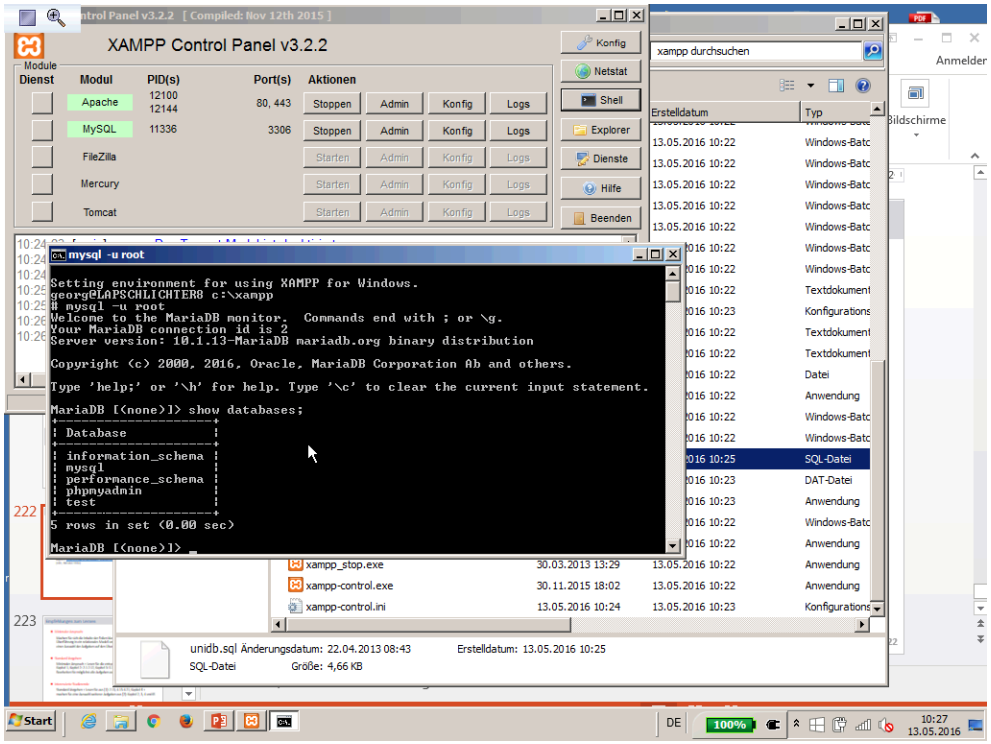

Setup
Completing the XAMPP Setup Wizard
Setup has finished installing XAMPP on your computer.
Do you want to start the Control Panel now?
bitnami
XAMPP Installer
< Back
Finish
Cancel

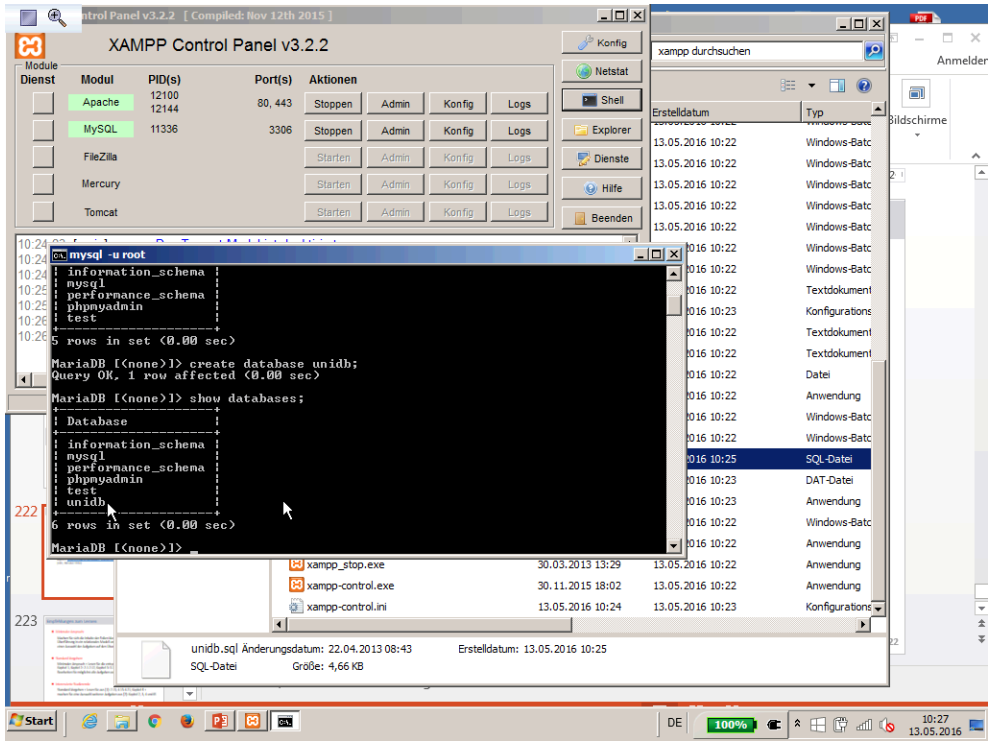

XAMPP Control Panel v3.2.2
Module
Dienst
Modul
PID(s)
Port(s)
Aktionen
Apache
MySQL
FileZilla
Mercury
Tomcat
Starten
Admin
Konfig
Logs
Konfig
Netstat
Shell
Explorer
Dienste
Hilfe
Beenden
10:24:23 [main] Alle Voraussetzungen sind erfüllt
10:24:23 [main] Initialisiere Module
10:24:23 [main] Das FileZilla Modul ist deaktiviert
10:24:23 [main] Das Mercury Modul ist deaktiviert
10:24:23 [main] Das Tomcat Modul ist deaktiviert
10:24:23 [main] Starte Check-Timer
10:24:23 [main] Control Panel bereit

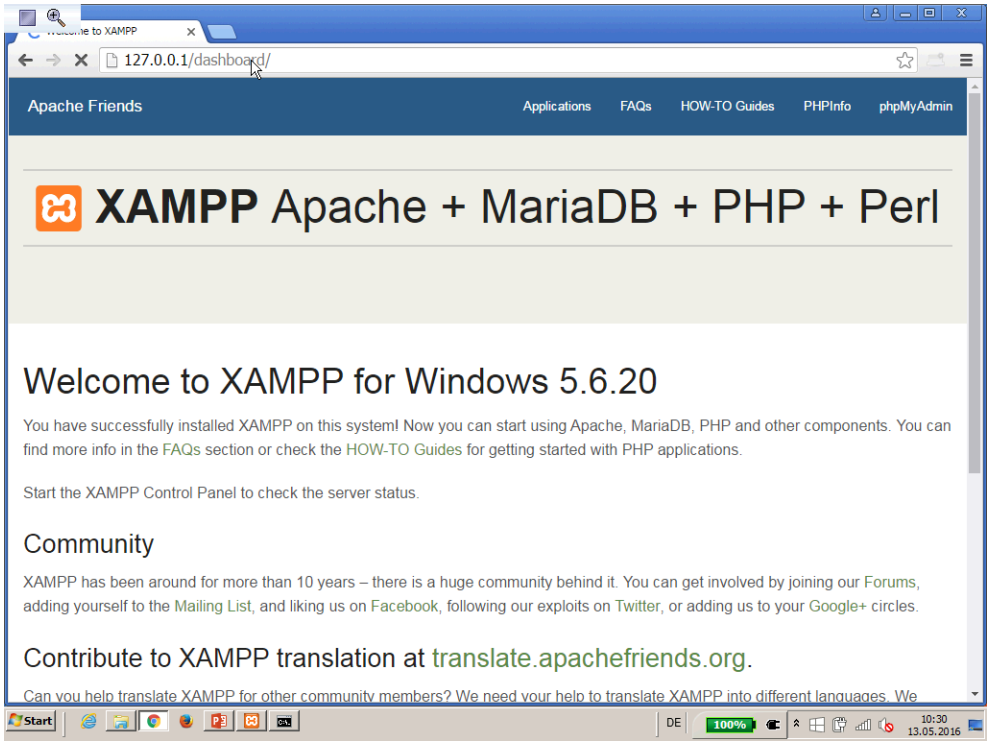

XAMPP Control Panel v3.2.2
Module
Dienst
Modul
PID(s)
Port(s)
Aktionen
Apache
MySQL
FileZilla
Mercury
Tomcat
Starten
Admin
Konfig
Logs
Konfig
Netstat
Shell
Explorer
Dienste
Hilfe
Beenden
10:24:23 [main] Alle Voraussetzungen sind erfüllt
10:24:23 [main] Initialisiere Module
10:24:23 [main] Das FileZilla Modul ist deaktiviert
10:24:23 [main] Das Mercury Modul ist deaktiviert
10:24:23 [main] Das Tomcat Modul ist deaktiviert
10:24:23 [main] Starte Check-Timer
10:24:23 [main] Control Panel bereit

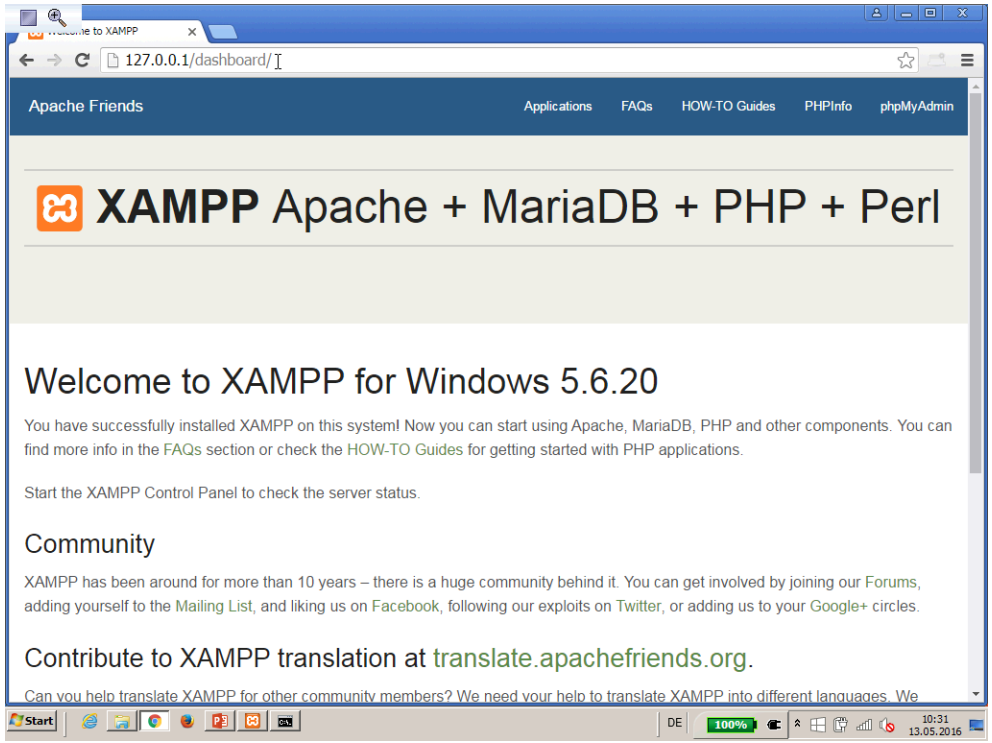

XAMPP Control Panel v3.2.2
Module
Dienst
Modul
PID(s)
Port(s)
Aktionen
Apache
MySQL
FileZilla
Mercury
Tomcat
Starten
Admin
Konfig
Logs
Konfig
Netstat
Shell
Explorer
Dienste
Hilfe
Beenden
10:24:23 [main] Alle Voraussetzungen sind erfüllt
10:24:23 [main] Initialisiere Module
10:24:23 [main] Das FileZilla Modul ist deaktiviert
10:24:23 [main] Das Mercury Modul ist deaktiviert
10:24:23 [main] Das Tomcat Modul ist deaktiviert
10:24:23 [main] Starte Check-Timer
10:24:23 [main] Control Panel bereit
wzw2016 durchsuchen
Typ
Größe
Netzwerk
xampp-win32-5.6.20-0-VC11-installer.exe Änderungsdatum: 12.05.2016 15:13
Erstelldatum: 12.05.2016 15:13
Anwendung
Größe: 107 MB

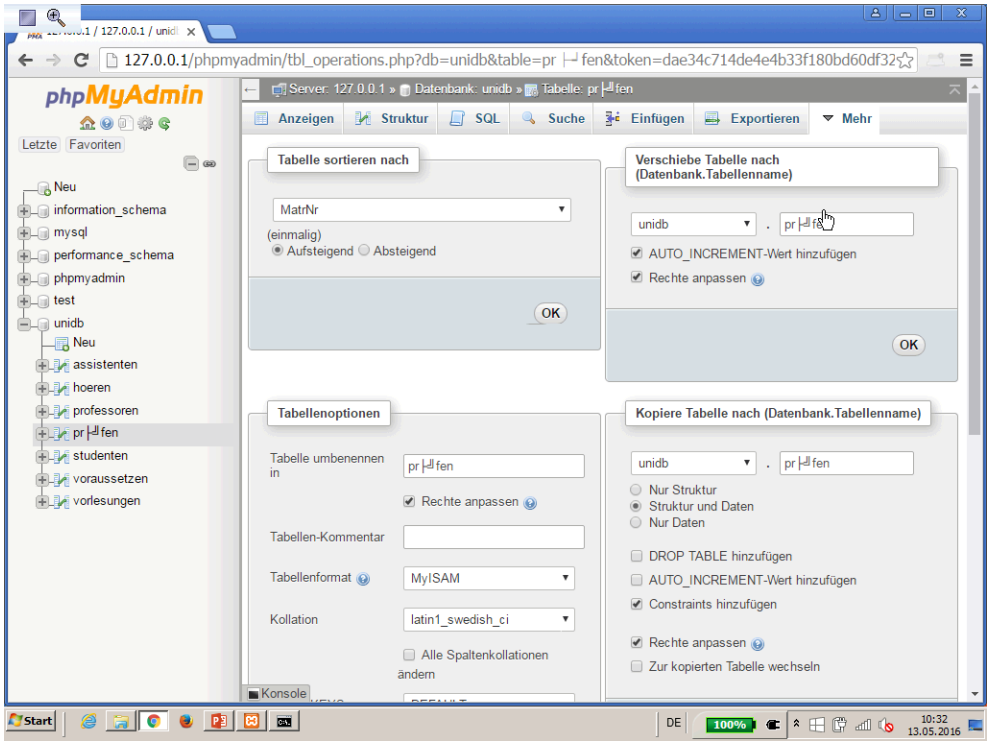

XAMPP Control Panel v3.2.2
XAMPP for Windows
Setting environment for using XAMPP for Windows.
georg@LAPSCHLICHTER8 c:\xampp
mysql -u root
xampp durchsuchen
unidb.sql Änderungsdatum: 22.04.2013 08:43
Erstelldatum: 13.05.2016 10:25
SQL-Datei
Größe: 4,66 KB

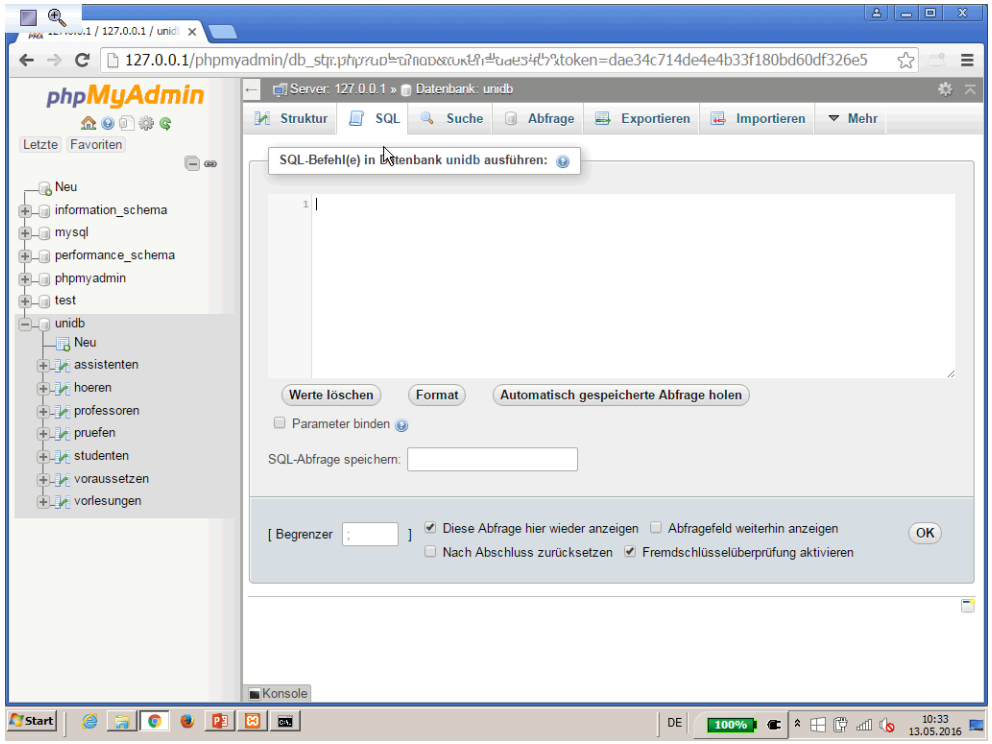

XAMPP Control Panel v3.2.2
mysql -u root
Setting environment for using XAMPP for Windows.
georg@LAPSCHLICHTER8 c:\xampp
mysql -u root
Welcome to the MariaDB monitor. Commands end with ; or \g.
Your MariaDB connection id is 2
Server version: 10.1.13-MariaDB mariadb.org binary distribution
Copyright (c) 2000, 2016, Oracle, MariaDB Corporation Ab and others.
Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.
MariaDB [(none)]> show databases;
+--------------------+
| Database |
+--------------------+
| information_schema |
| mysql |
| performance_schema |
| phpmyadmin |
| test |
+--------------------+
5 rows in set (0.00 sec)
MariaDB [(none)]>
unidb.sql Änderungsdatum: 22.04.2013 08:43
Erstelldatum: 13.05.2016 10:25
SQL-Datei
Größe: 4,66 KB

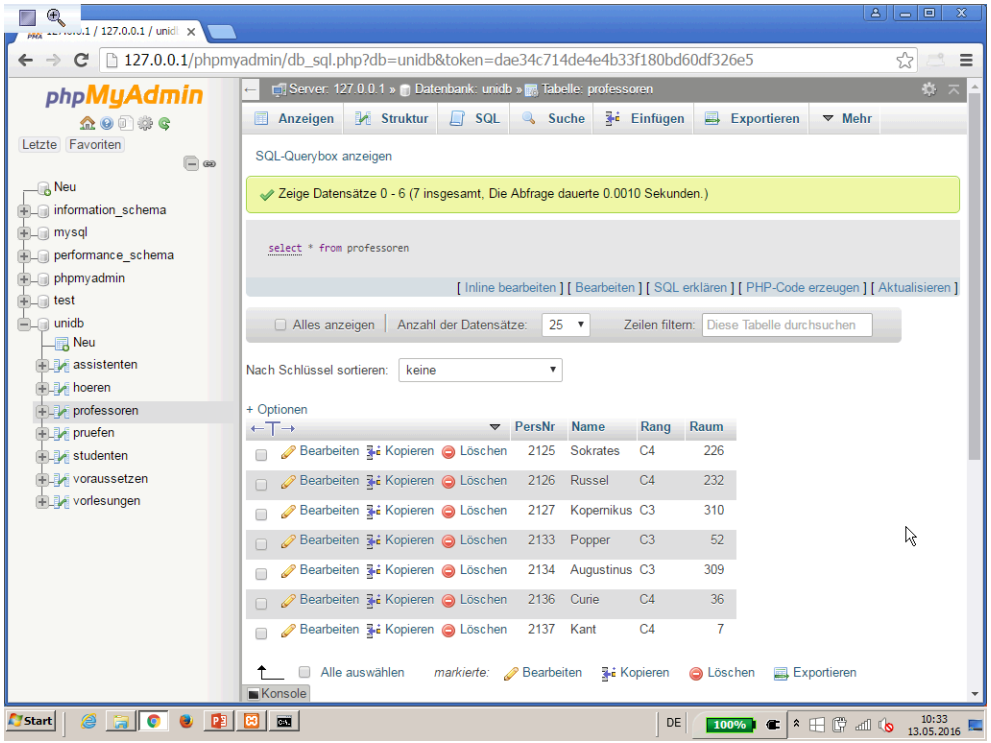

XAMPP Control Panel v3.2.2
mysql -u root
| information_schema |
| mysql |
| performance_schema |
| phpmyadmin |
| test |
5 rows in set (0.00 sec)
MariaDB [(none)]> create database unidb;
Query OK, 1 row affected (0.00 sec)
MariaDB [(none)]> show databases;
| information_schema |
| mysql |
| performance_schema |
| phpmyadmin |
| test |
| unidb |
6 rows in set (0.00 sec)
MariaDB [(none)]>
unidb.sql Änderungsdatum: 22.04.2013 08:43
Erstelldatum: 13.05.2016 10:25
SQL-Datei
Größe: 4,66 KB

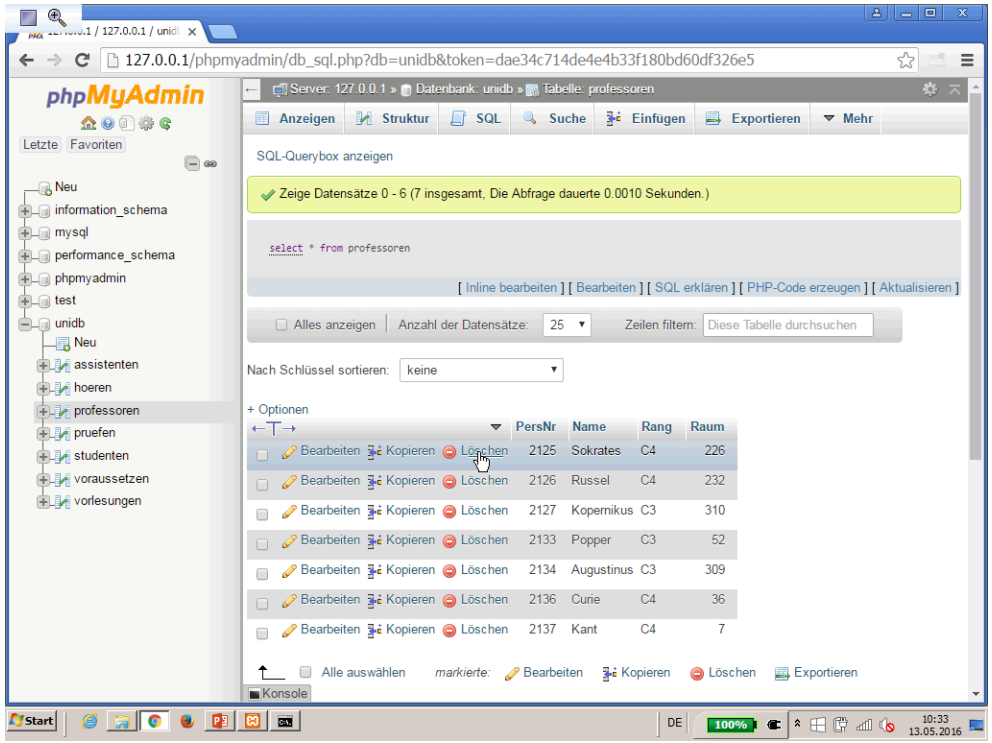

127.0.0.1/dashboard/
Apache Friends
Applications
FAQs
HOW-TO Guides
PHPInfo
phpMyAdmin
XAMPP Apache + MariaDB + PHP + Perl
Welcome to XAMPP for Windows 5.6.20
You have successfully installed XAMPP on this system! Now you can start using Apache, MariaDB, PHP and other components. You can find more info in the FAQs section or check the HOW-TO Guides for getting started with PHP applications.
Start the XAMPP Control Panel to check the server status.
Community
XAMPP has been around for more than 10 years – there is a huge community behind it. You can get involved by joining our Forums, adding yourself to the Mailing List, and liking us on Facebook, following our exploits on Twitter, or adding us to your Google+ circles.
Contribute to XAMPP translation at translate.apachefriends.org.
Can you help translate XAMPP for other community members? We need your help to translate XAMPP into different languages. We


127.0.0.1/dashboard/
Apache Friends
Applications
FAQs
HOW-TO Guides
PHPInfo
phpMyAdmin
XAMPP Apache + MariaDB + PHP + Perl
Welcome to XAMPP for Windows 5.6.20
You have successfully installed XAMPP on this system! Now you can start using Apache, MariaDB, PHP and other components. You can find more info in the FAQs section or check the HOW-TO Guides for getting started with PHP applications.
Start the XAMPP Control Panel to check the server status.
Community
XAMPP has been around for more than 10 years – there is a huge community behind it. You can get involved by joining our Forums, adding yourself to the Mailing List, and liking us on Facebook, following our exploits on Twitter, or adding us to your Google+ circles.
Contribute to XAMPP translation at translate.apachefriends.org.
Can you help translate XAMPP for other community members? We need your help to translate XAMPP into different languages. We


127.0.0.1/phpmyadmin/tbl_operations.php?db=unidb&table=pr fen&token=dae34c714de4e4b33f180bd60df32
phpMyAdmin
Server: 127.0.0.1 » Datenbank: unidb » Tabelle: pr fen
Anzeigen
Struktur
SQL
Suche
Einfügen
Exportieren
Mehr
Letzte
Favoriten
Neu
information_schema
mysql
performance_schema
phpmyadmin
test
unidb
assistenten
hoeren
professoren
pr fen
studenten
voraussetzen
vorlesungen
Tabelle sortieren nach
MatrNr
(einmalig)
Aufsteigend
Absteigend
OK
Verschiebe Tabelle nach (Datenbank.Tabellenname)
unidb
AUTO_INCREMENT-Wert hinzufügen
Rechte anpassen
Tabellenoptionen
Tabelle umbenennen in
Tabellen-Kommentar
Tabellenformat
MyISAM
Kollation
latin1_swedish_ci
Alle Spaltenkollationen ändern
Kopiere Tabelle nach (Datenbank.Tabellenname)
Nur Struktur
Struktur und Daten
Nur Daten
DROP TABLE hinzufügen
Constraints hinzufügen
Zur kopierten Tabelle wechseln
Konsole


127.0.0.1/phpmyadmin/db_str
token=dae34c714de4e4b33f180bd60df326e5
phpMyAdmin
Server: 127.0.0.1 » Datenbank: unidb
Struktur
SQL
Suche
Abfrage
Exportieren
Importieren
Mehr
Letzte
Favoriten
Neu
information_schema
mysql
performance_schema
phpmyadmin
test
unidb
assistenten
hoeren
professoren
pruefen
studenten
voraussetzen
vorlesungen
SQL-Befehl(e) in Datenbank unidb ausführen:
Werte löschen
Format
Automatisch gespeicherte Abfrage holen
Parameter binden
SQL-Abfrage speichern:
[Begrenzer ;]
Diese Abfrage hier wieder anzeigen
Abfragefeld weiterhin anzeigen
Nach Abschluss zurücksetzen
Fremdschlüsselüberprüfung aktivieren
OK
Konsole





Teil II:
Programmierung, Java
Algorithmen, Datenstrukturen

Programme

- *Erinnerung:* Informatik: Gegenstand: Informationen + deren Verarbeitung (Repräsentation, Speicherung, Abbildung, Übertragung etc.) mit Computersystemen.
- Bisher: DBMS: realisieren eine Art, Daten systematisch zu speichern und ermöglichen Anfragen auf Datenbasis (Eingabe: bspw. SQL select, Ausgabe: Antworttabelle).
- Allgemeinere Art der Verarbeitung von Informationen? --> mit Programmen (in formalen Programmiersprachen) auf (programmierbaren) Computersystemen.

Achtung: Informatik $\neq$ Programmieren !!!!!!!!!