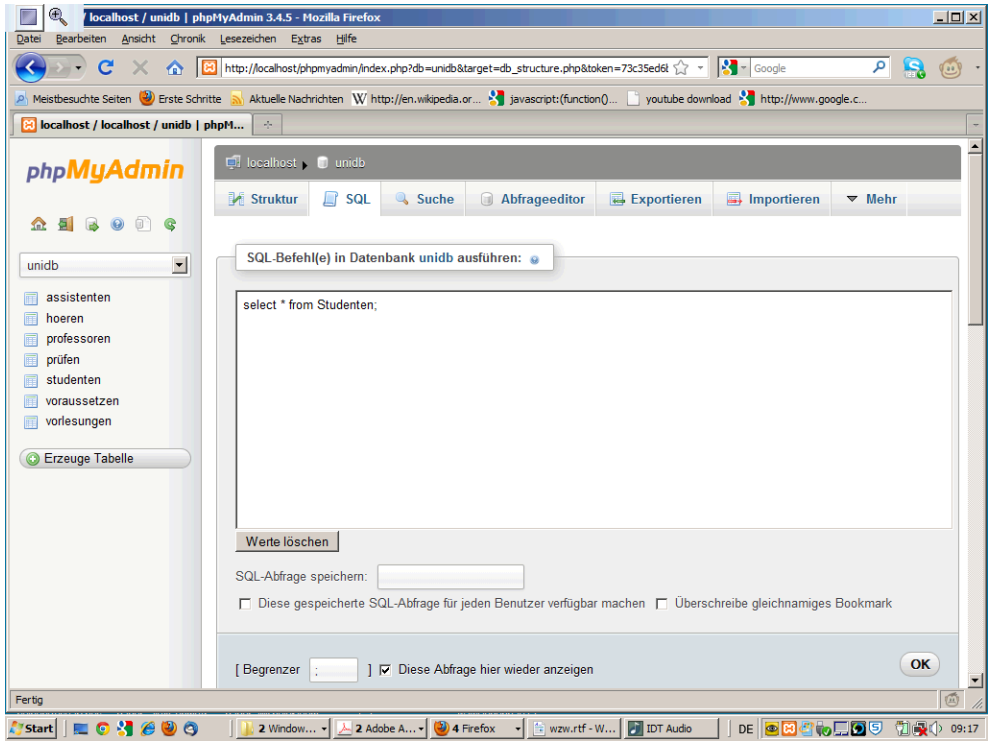
Script **generated by TTT**

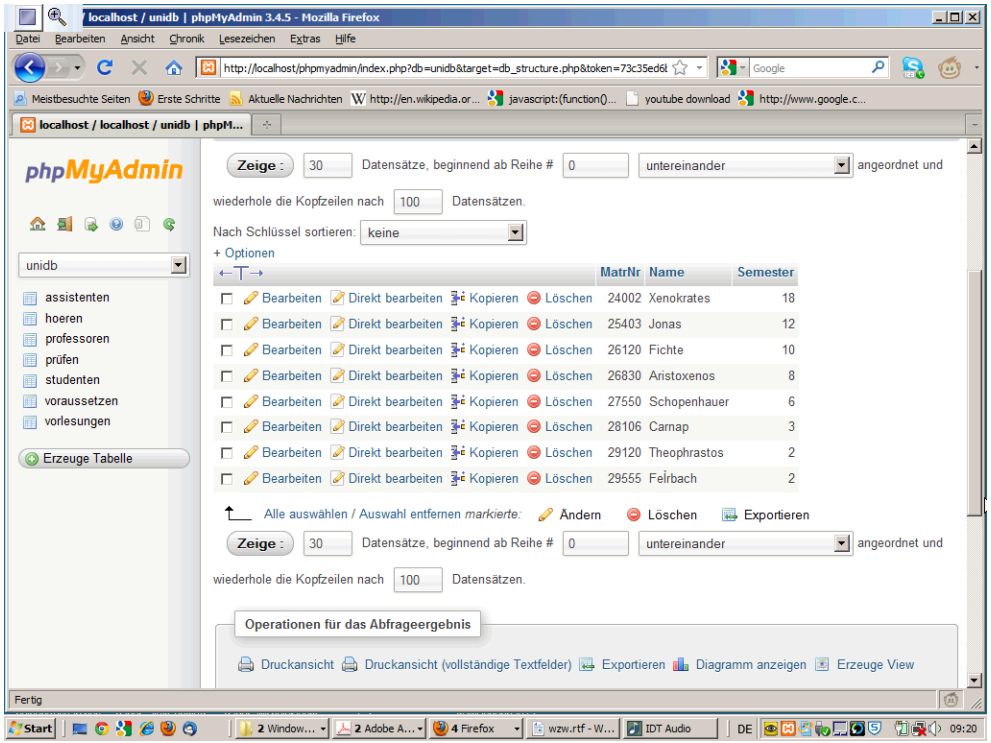
Title: Groh: wzw_2012 (01.06.2012)

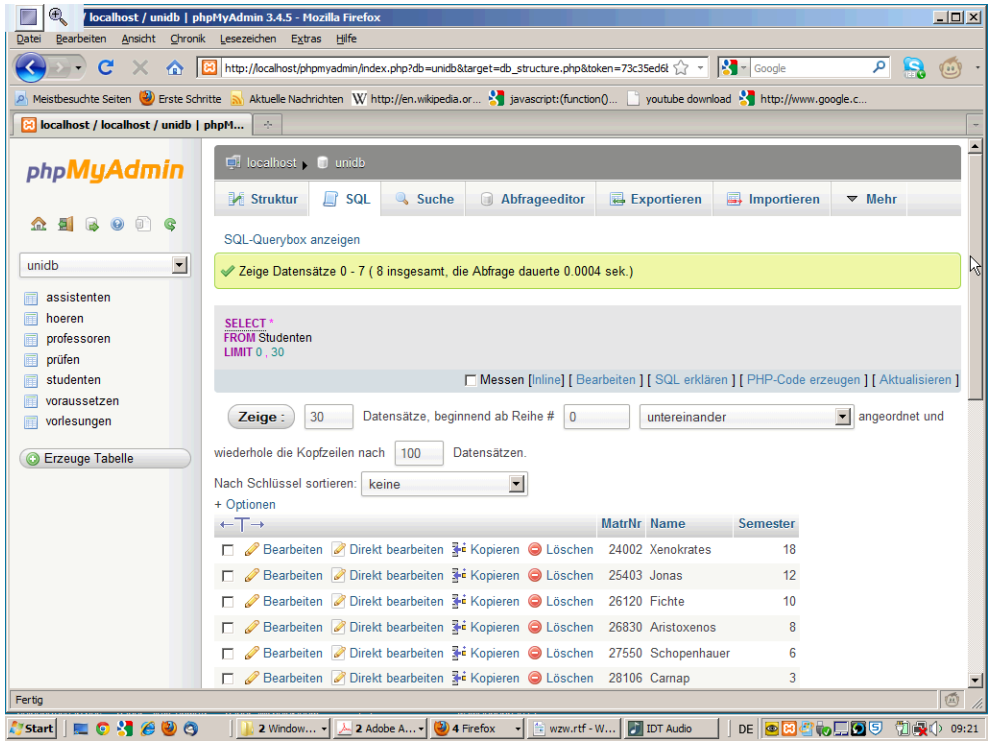
Date: Fri Jun 01 09:17:47 CEST 2012

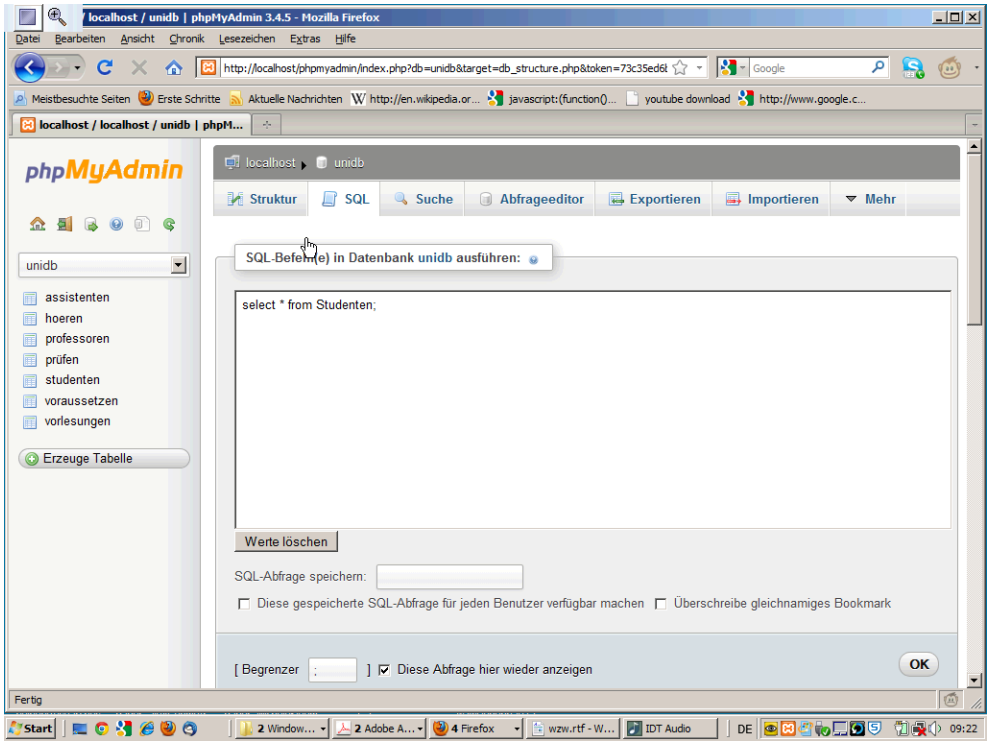
Duration: 89:58 min

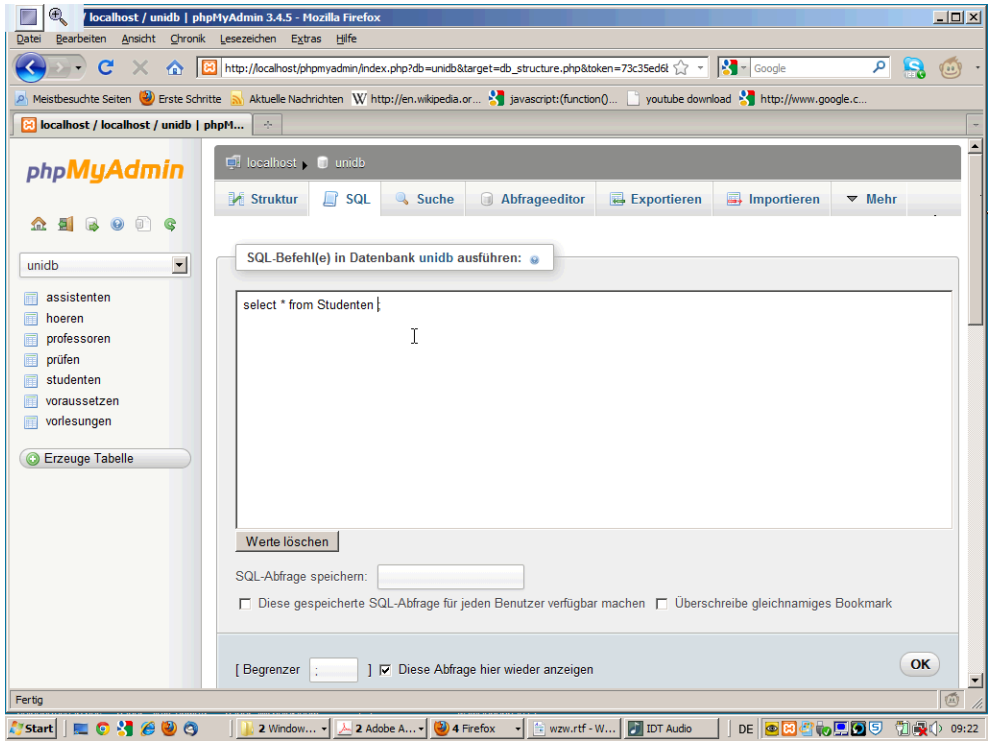
Pages: 35

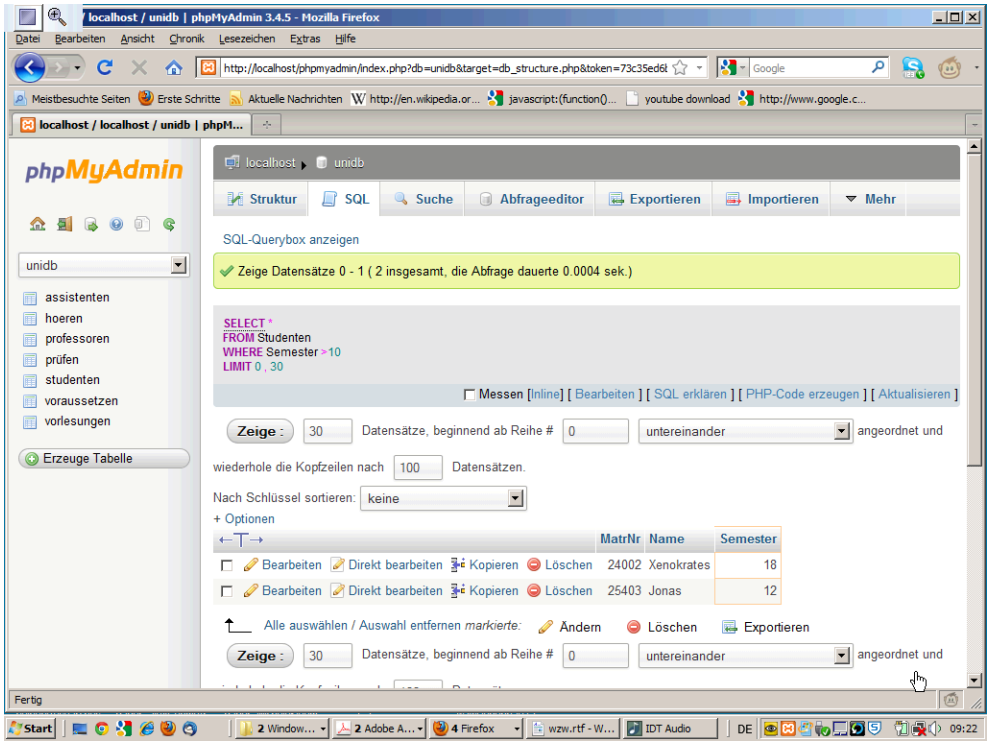


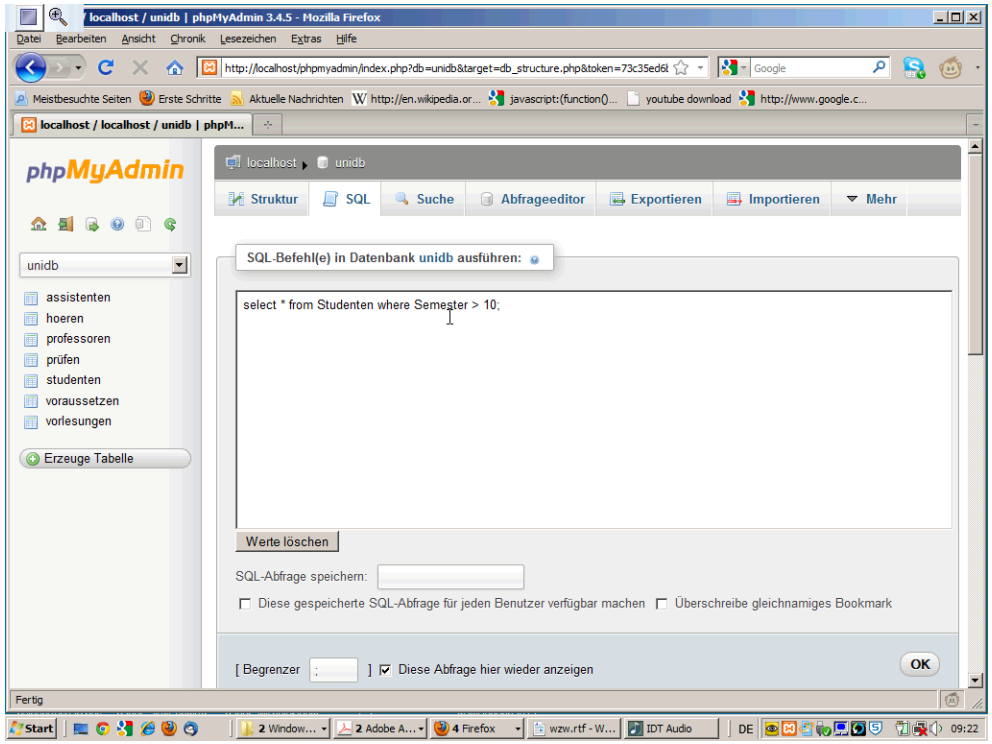


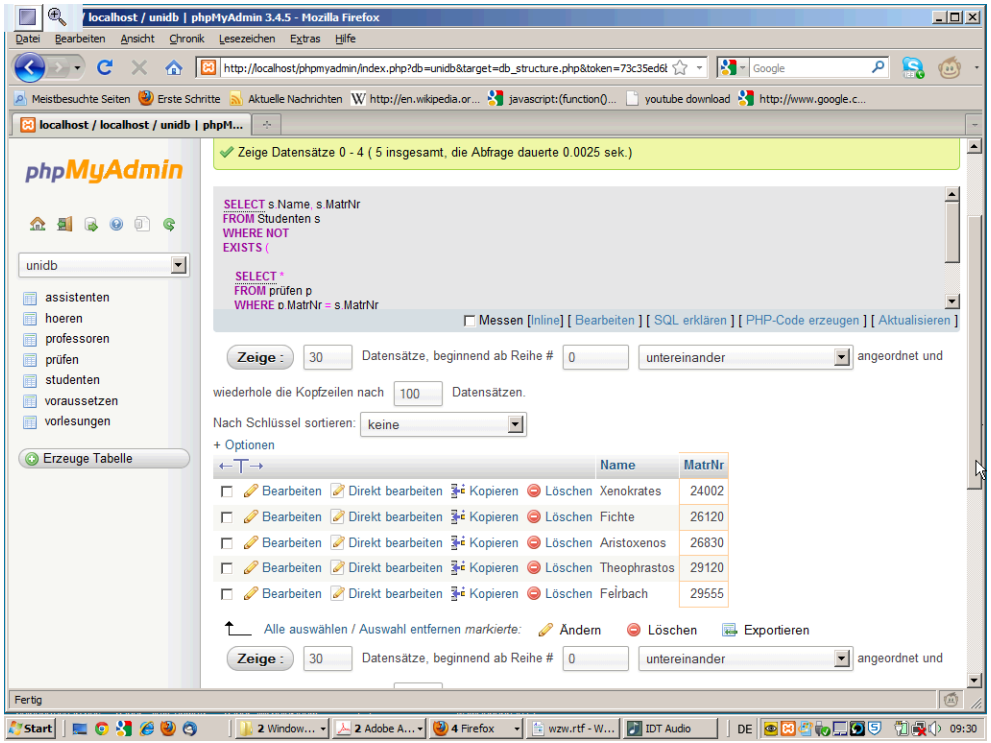


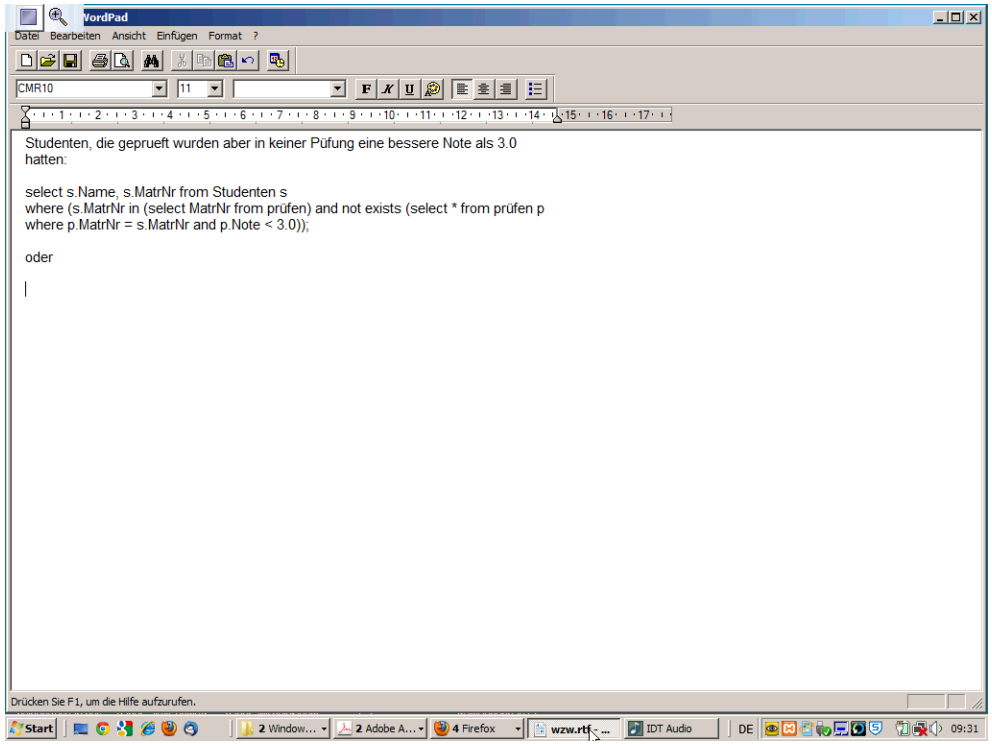

localhost / unidb | phpMyAdmin 3.4.5 - Mozilla Firefox
phpMyAdmin
localhost
unidb
Struktur
SQL
Suche
Abfrageeditor
Exportieren
Importieren
Mehr
unidb
assistenten
hoeren
professoren
prüfen
studenten
voraussetzen
vorlesungen
Erzeuge Tabelle
SQL-Befehl(e) in Datenbank unidb ausführen:
select * from Studenten;
Werte löschen
SQL-Abfrage speichern:
Diese gespeicherte SQL-Abfrage für jeden Benutzer verfügbar machen
Überschreibe gleichnamiges Bookmark
[Begrenzer ;]
Diese Abfrage hier wieder anzeigen
OK
Fertig

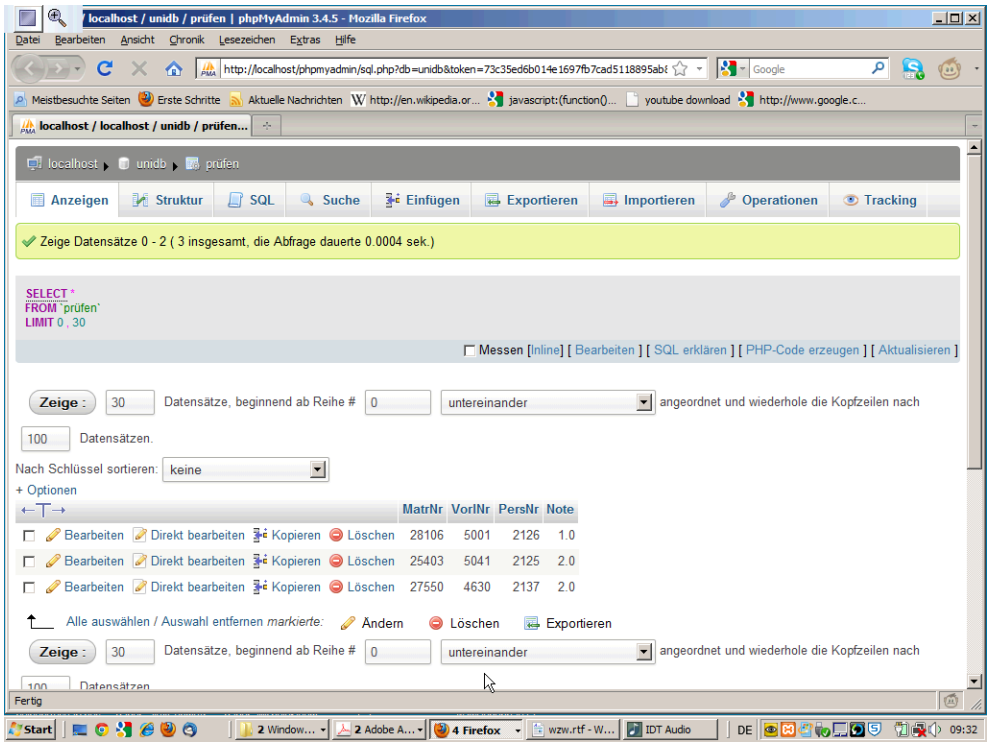

localhost / unidb | phpMyAdmin 3.4.5 - Mozilla Firefox
phpMyAdmin
localhost
unidb
Struktur
SQL
Suche
Abfrageeditor
Exportieren
Importieren
Mehr
unidb
assistenten
hoeren
professoren
prüfen
studenten
voraussetzen
vorlesungen
Erzeuge Tabelle
SQL-Befehl(e) in Datenbank unidb ausführen:
select * from Studenten
Werte löschen
SQL-Abfrage speichern:
Diese gespeicherte SQL-Abfrage für jeden Benutzer verfügbar machen
Überschreibe gleichnamiges Bookmark
[Begrenzer ;]
Diese Abfrage hier wieder anzeigen
OK
Fertig

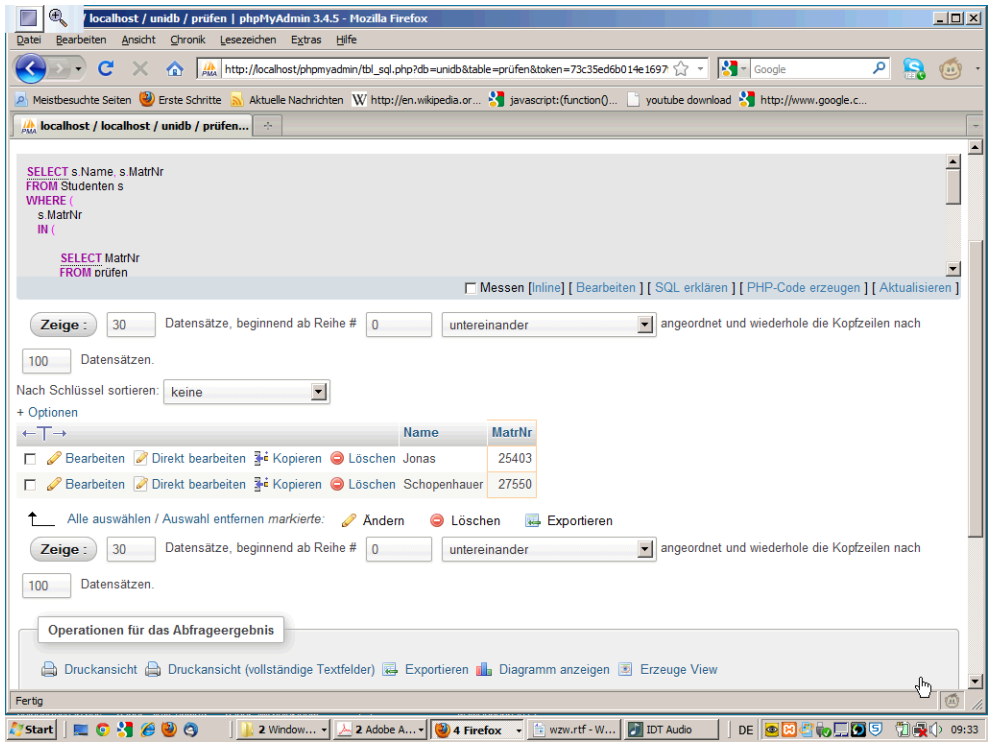

localhost / unidb | phpMyAdmin 3.4.5 - Mozilla Firefox
phpMyAdmin
localhost
unidb
Struktur
SQL
Suche
Abfrageeditor
Exportieren
Importieren
Mehr
unidb
assistenten
hoeren
professoren
prüfen
studenten
voraussetzen
vorlesungen
Erzeuge Tabelle
SQL-Querybox anzeigen
Zeige Datensätze 0 - 1 (2 insgesamt, die Abfrage dauerte 0.0004 sek.)
SELECT *
FROM Studenten
WHERE Semester >10
LIMIT 0 , 30
Messen [Inline] [Bearbeiten] [SQL erklären] [PHP-Code erzeugen] [Aktualisieren]
Zeige : 30 Datensätze, beginnend ab Reihe # 0 untereinander angeordnet und
wiederhole die Kopfzeilen nach 100 Datensätzen.
Nach Schlüssel sortieren: keine
+ Optionen
MatrNr Name Semester
Bearbeiten Direkt bearbeiten Kopieren Löschen 24002 Xenokrates 18
Bearbeiten Direkt bearbeiten Kopieren Löschen 25403 Jonas 12
Alle auswählen / Auswahl entfernen markierte: Ändern Löschen Exportieren
Zeige : 30 Datensätze, beginnend ab Reihe # 0 untereinander angeordnet und
Fertig

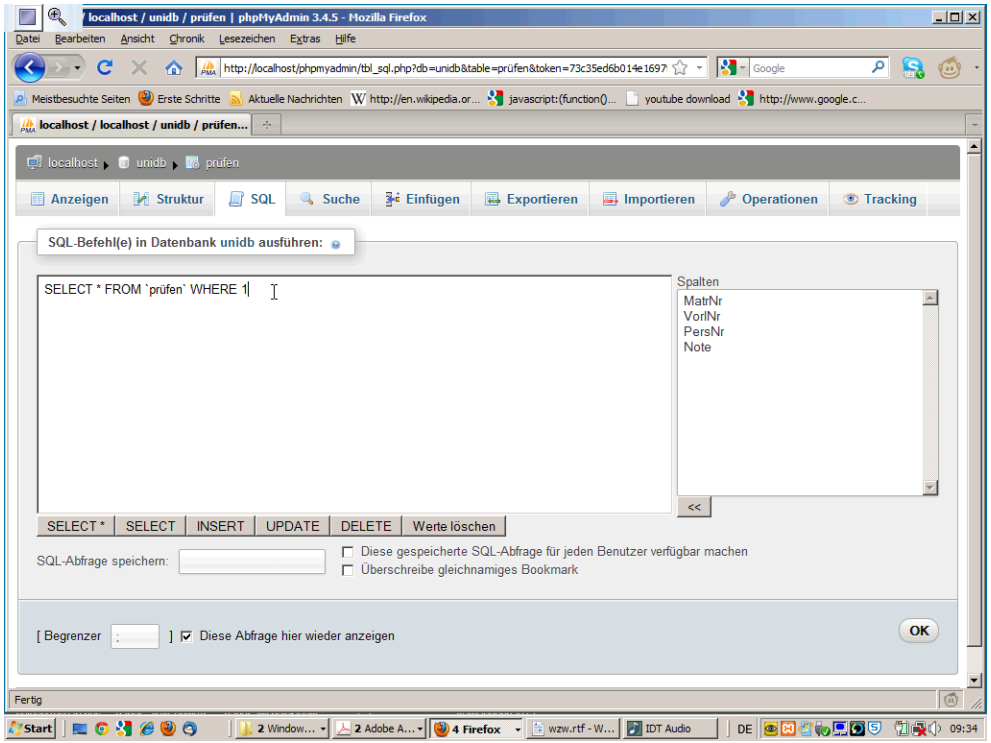

localhost / unidb | phpMyAdmin 3.4.5 - Mozilla Firefox
phpMyAdmin
localhost
unidb
Struktur
SQL
Suche
Abfrageeditor
Exportieren
Importieren
Mehr
unidb
assistenten
hoeren
professoren
prüfen
studenten
voraussetzen
vorlesungen
Erzeuge Tabelle
SQL-Befehl(e) in Datenbank unidb ausführen:
select * from Studenten where Semester > 10;
Werte löschen
SQL-Abfrage speichern:
Diese gespeicherte SQL-Abfrage für jeden Benutzer verfügbar machen
Überschreibe gleichnamiges Bookmark
[Begrenzer ;]
Diese Abfrage hier wieder anzeigen
OK
Fertig

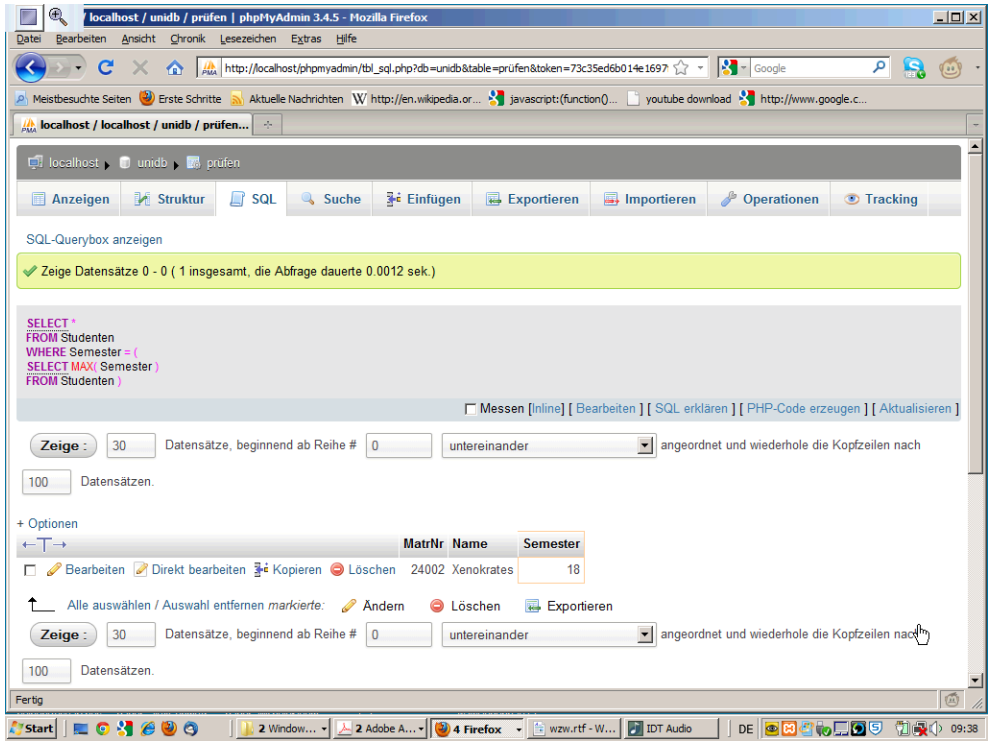

localhost / unidb | phpMyAdmin 3.4.5 - Mozilla Firefox
phpMyAdmin
unidb
assistenten
hoeren
professoren
prüfen
studenten
voraussetzen
vorlesungen
Erzeuge Tabelle
Zeige Datensätze 0 - 4 (5 insgesamt, die Abfrage dauerte 0.0025 sek.)
SELECT s.Name, s.MatrNr
FROM Studenten s
WHERE NOT
EXISTS (
SELECT *
FROM prüfen p
WHERE p.MatrNr = s.MatrNr
Name
MatrNr
Xenokrates 24002
Fichte 26120
Aristoxenos 26830
Theophrastos 29120
Feirbach 29555
Bearbeiten Direkt bearbeiten Kopieren Löschen
Alle auswählen / Auswahl entfernen markierte: Ändern Löschen Exportieren

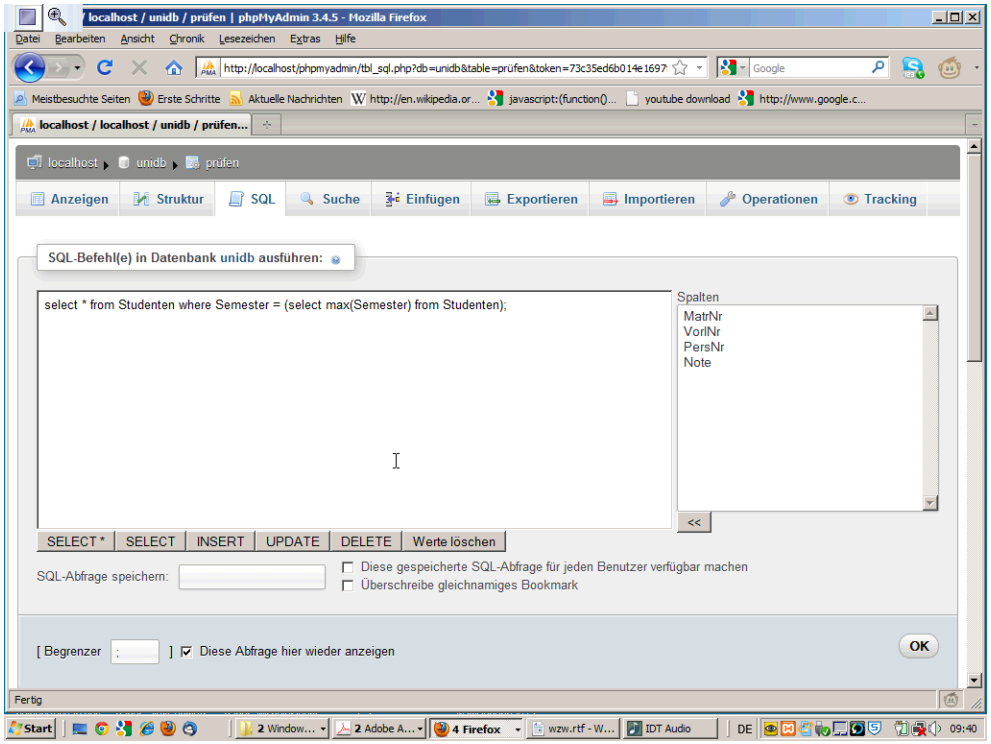

WordPad
Studenten, die geprueft wurden aber in keiner Püfung eine bessere Note als 3.0
hatten:
select s.Name, s.MatrNr from Studenten s
where (s.MatrNr in (select MatrNr from prüfen) and not exists (select * from prüfen p
where p.MatrNr = s.MatrNr and p.Note < 3.0));
oder

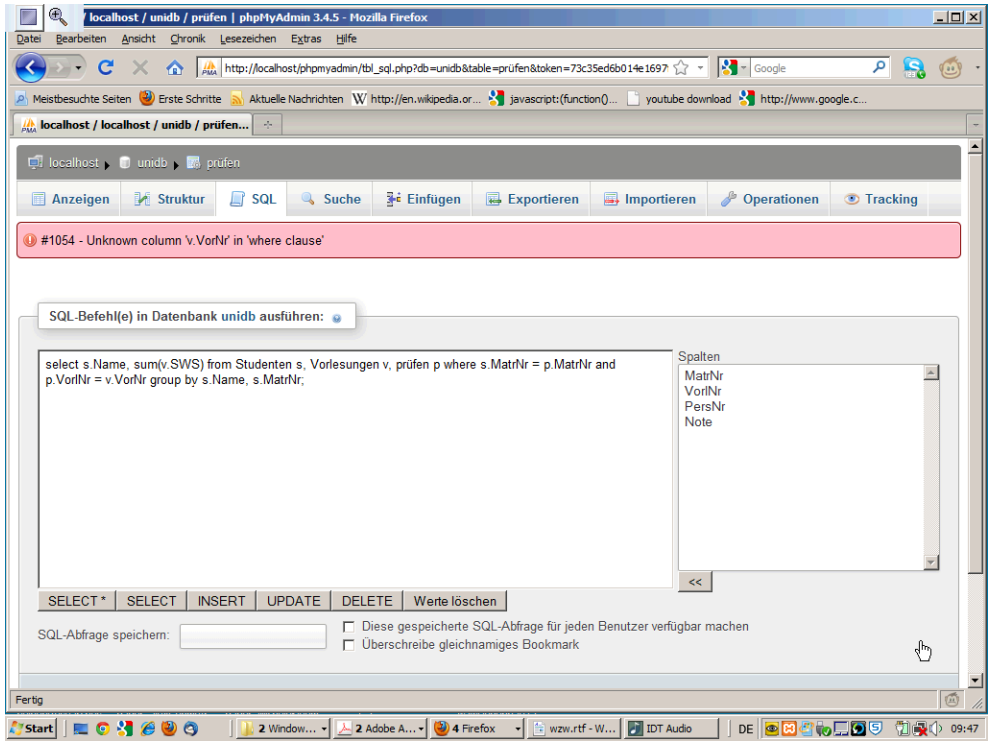

localhost / unidb / prüfen | phpMyAdmin 3.4.5 - Mozilla Firefox
Anzeigen Struktur SQL Suche Einfügen Exportieren Importieren Operationen Tracking
Zeige Datensätze 0 - 2 (3 insgesamt, die Abfrage dauerte 0.0004 sek.)
SELECT *
FROM `prüfen`
LIMIT 0 , 30
MatrNr VorlNr PersNr Note
28106 5001 2126 1.0
25403 5041 2125 2.0
27550 4630 2137 2.0
Bearbeiten Direkt bearbeiten Kopieren Löschen
Alle auswählen / Auswahl entfernen markierte: Ändern Löschen Exportieren

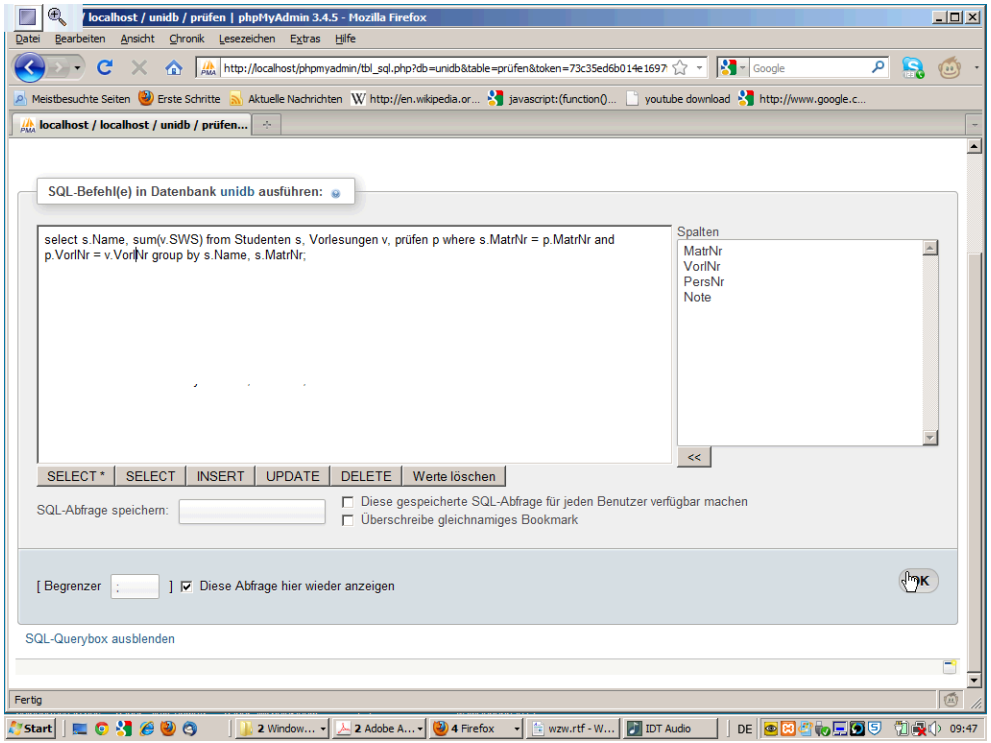

localhost / unidb / prüfen | phpMyAdmin 3.4.5 - Mozilla Firefox
SELECT s.Name, s.MatrNr
FROM Studenten s
WHERE (
s.MatrNr
IN (
SELECT MatrNr
FROM prüfen
Name
MatrNr
Jonas 25403
Schopenhauer 27550
Bearbeiten Direkt bearbeiten Kopieren Löschen
Alle auswählen / Auswahl entfernen markierte: Ändern Löschen Exportieren
Operationen für das Abfrageergebnis
Druckansicht Druckansicht (vollständige Textfelder) Exportieren Diagramm anzeigen Erzeuge View

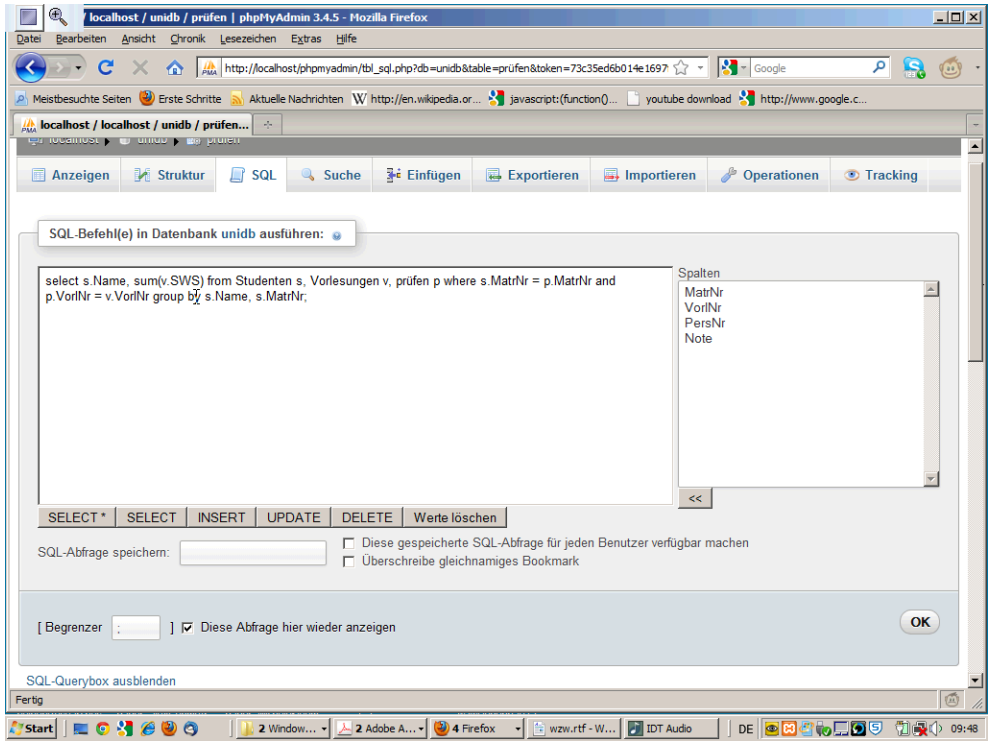

localhost / unidb / prüfen | phpMyAdmin 3.4.5 - Mozilla Firefox
SQL-Befehl(e) in Datenbank unidb ausführen:
SELECT * FROM `prüfen` WHERE 1
Spalten
MatrNr
VorlNr
PersNr
Note
SELECT * | SELECT | INSERT | UPDATE | DELETE | Werte löschen
SQL-Abfrage speichern:
Diese gespeicherte SQL-Abfrage für jeden Benutzer verfügbar machen
Überschreibe gleichnamiges Bookmark
[Begrenzer ;] Diese Abfrage hier wieder anzeigen
OK

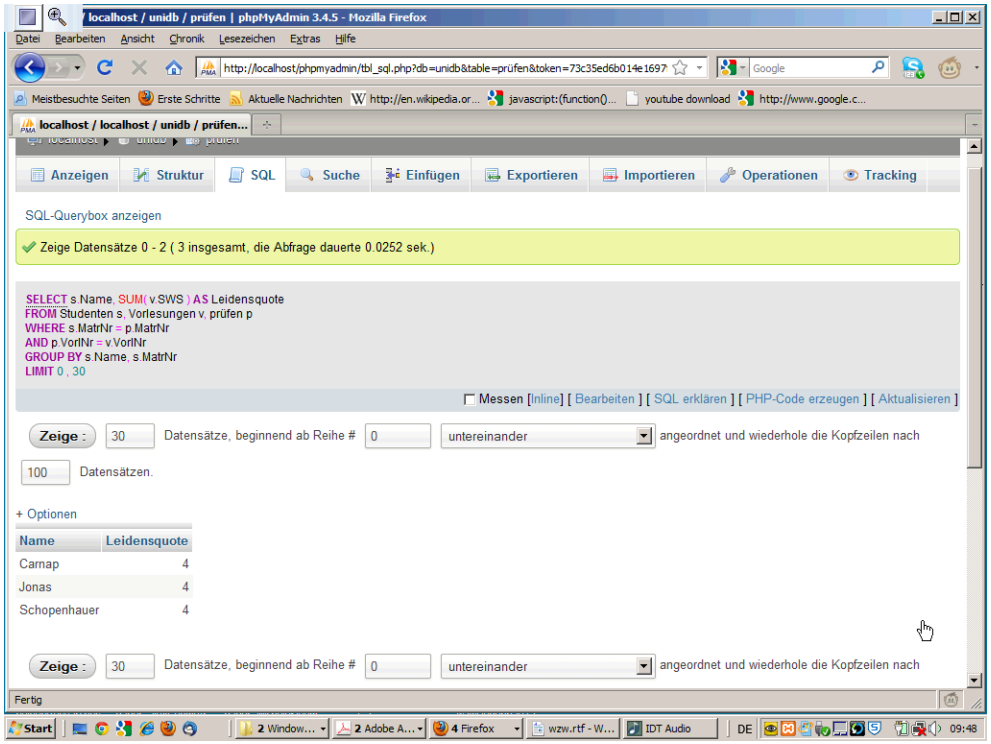

localhost / unidb / prüfen | phpMyAdmin 3.4.5 - Mozilla Firefox
SQL-Querybox anzeigen
Zeige Datensätze 0 - 0 (1 insgesamt, die Abfrage dauerte 0.0012 sek.)
SELECT *
FROM Studenten
WHERE Semester = (
SELECT MAX(Semester)
FROM Studenten)
Messen [Inline] [Bearbeiten] [SQL erklären] [PHP-Code erzeugen] [Aktualisieren]
Zeige : 30 Datensätze, beginnend ab Reihe # 0 untereinander angeordnet und wiederhole die Kopfzeilen nach 100 Datensätzen.
+ Optionen
MatrNr Name Semester
Bearbeiten Direkt bearbeiten Kopieren Löschen 24002 Xenokrates 18
Alle auswählen / Auswahl entfernen markierte: Ändern Löschen Exportieren
Zeige : 30 Datensätze, beginnend ab Reihe # 0 untereinander angeordnet und wiederhole die Kopfzeilen nach 100 Datensätzen.

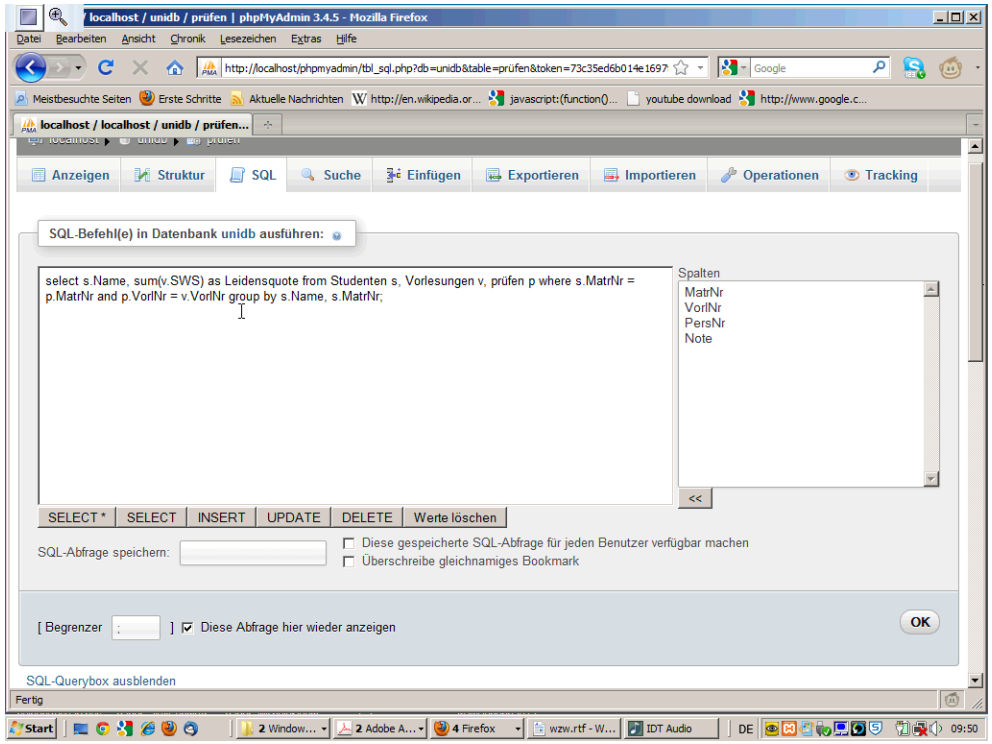

localhost / unidb / prüfen | phpMyAdmin 3.4.5 - Mozilla Firefox
SQL-Befehl(e) in Datenbank unidb ausführen:
select * from Studenten where Semester = (select max(Semester) from Studenten);
Spalten
MatrNr
VorlNr
PersNr
Note
SELECT * | SELECT | INSERT | UPDATE | DELETE | Werte löschen
SQL-Abfrage speichern:
Diese gespeicherte SQL-Abfrage für jeden Benutzer verfügbar machen
Überschreibe gleichnamiges Bookmark
[Begrenzer ;] Diese Abfrage hier wieder anzeigen
OK


localhost / unidb / prüfen | phpMyAdmin 3.4.5 - Mozilla Firefox
#1054 - Unknown column 'v.VorNr' in 'where clause'
SQL-Befehl(e) in Datenbank unidb ausführen:
select s.Name, sum(v.SWS) from Studenten s, Vorlesungen v, prüfen p where s.MatrNr = p.MatrNr and p.VorlNr = v.VorNr group by s.Name, s.MatrNr;
Spalten
MatrNr
VorlNr
PersNr
Note
SELECT * | SELECT | INSERT | UPDATE | DELETE | Werte löschen
SQL-Abfrage speichern:
Diese gespeicherte SQL-Abfrage für jeden Benutzer verfügbar machen
Überschreibe gleichnamiges Bookmark


localhost / unidb / prüfen | phpMyAdmin 3.4.5 - Mozilla Firefox
SQL-Befehl(e) in Datenbank unidb ausführen:
select s.Name, sum(v.SWS) from Studenten s, Vorlesungen v, prüfen p where s.MatrNr = p.MatrNr and p.VorlNr = v.VorlNr group by s.Name, s.MatrNr;
Spalten
MatrNr
VorlNr
PersNr
Note
SELECT * | SELECT | INSERT | UPDATE | DELETE | Werte löschen
SQL-Abfrage speichern:
Diese gespeicherte SQL-Abfrage für jeden Benutzer verfügbar machen
Überschreibe gleichnamiges Bookmark
[Begrenzer ;] Diese Abfrage hier wieder anzeigen
OK
SQL-Querybox ausblenden
Fertig

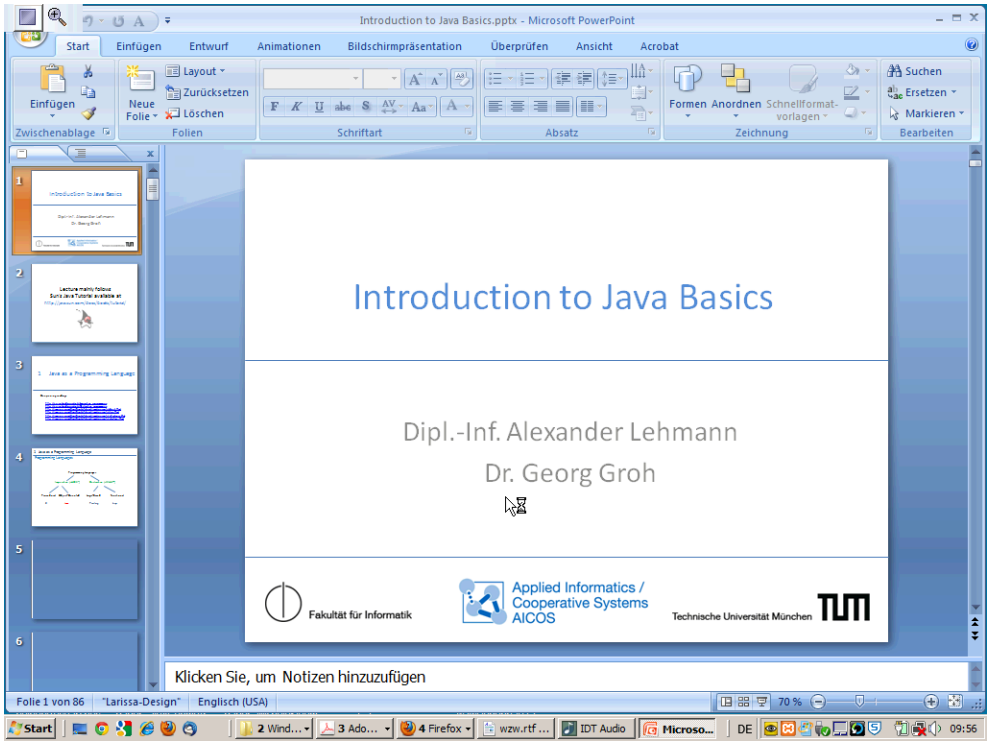

localhost / unidb / prüfen | phpMyAdmin 3.4.5 - Mozilla Firefox
Anzeigen | Struktur | SQL | Suche | Einfügen | Exportieren | Importieren | Operationen | Tracking
SQL-Befehl(e) in Datenbank unidb ausführen:
select s.Name, sum(v.SWS) from Studenten s, Vorlesungen v, prüfen p where s.MatrNr = p.MatrNr and p.VorlNr = v.VorlNr group by s.Name, s.MatrNr;
Spalten
MatrNr
VorlNr
PersNr
Note
SELECT * | SELECT | INSERT | UPDATE | DELETE | Werte löschen
SQL-Abfrage speichern:
Diese gespeicherte SQL-Abfrage für jeden Benutzer verfügbar machen
Überschreibe gleichnamiges Bookmark
[Begrenzer ;] Diese Abfrage hier wieder anzeigen
OK
SQL-Querybox ausblenden
Fertig

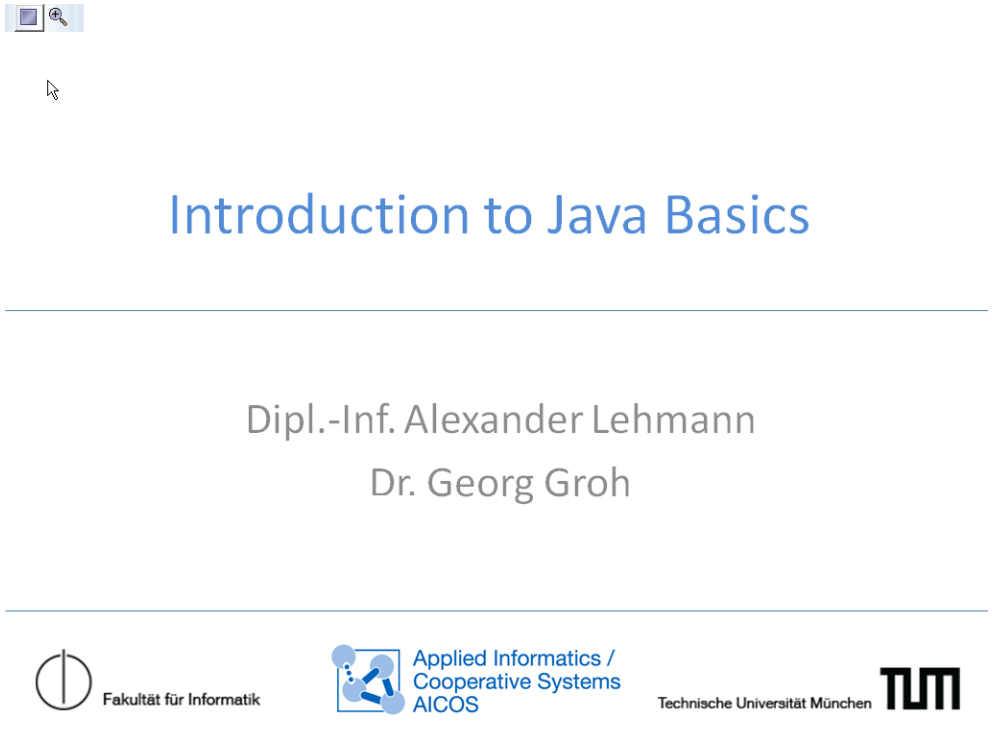

localhost / unidb / prüfen | phpMyAdmin 3.4.5 - Mozilla Firefox
Anzeigen | Struktur | SQL | Suche | Einfügen | Exportieren | Importieren | Operationen | Tracking
SQL-Querybox anzeigen
Zeige Datensätze 0 - 2 (3 insgesamt, die Abfrage dauerte 0.0252 sek.)
SELECT s.Name, SUM(v.SWS) AS Leidensquote
FROM Studenten s, Vorlesungen v, prüfen p
WHERE s.MatrNr = p.MatrNr
AND p.VorlNr = v.VorlNr
GROUP BY s.Name, s.MatrNr
LIMIT 0 , 30
Messen [Inline] [Bearbeiten] [SQL erklären] [PHP-Code erzeugen] [Aktualisieren]
Zeige : 30 Datensätze, beginnend ab Reihe # 0 untereinander angeordnet und wiederhole die Kopfzeilen nach 100 Datensätzen.
+ Optionen
Name | Leidensquote
Carnap 4
Jonas 4
Schopenhauer 4
Zeige : 30 Datensätze, beginnend ab Reihe # 0 untereinander angeordnet und wiederhole die Kopfzeilen nach
Fertig


localhost / unidb / prüfen | phpMyAdmin 3.4.5 - Mozilla Firefox
Anzeigen | Struktur | SQL | Suche | Einfügen | Exportieren | Importieren | Operationen | Tracking
SQL-Befehl(e) in Datenbank unidb ausführen:
select s.Name, sum(v.SWS) as Leidensquote from Studenten s, Vorlesungen v, prüfen p where s.MatrNr = p.MatrNr and p.VorlNr = v.VorlNr group by s.Name, s.MatrNr;
Spalten
MatrNr
VorlNr
PersNr
Note
SELECT * | SELECT | INSERT | UPDATE | DELETE | Werte löschen
SQL-Abfrage speichern:
Diese gespeicherte SQL-Abfrage für jeden Benutzer verfügbar machen
Überschreibe gleichnamiges Bookmark
[Begrenzer ;] Diese Abfrage hier wieder anzeigen
OK
SQL-Querybox ausblenden
Fertig


tsexperiment im Rahmen einer Masterarbeit.pdf - Adobe Acrobat Professional
Kreativitätsexperiment im Rahmen einer Masterarbeit
• Durchführung eines Experiments zur Ideengenerierung unterstützt durch die IdeaStream Plattform (www.ideastream.de)
• Dauer ca. 1 Stunde inkl. Einweisung
• Gruppen von 4 Studenten
• Ort: MI Gebäude, Diplomandenraum Lehrstuhl Prof. Schlichter
• Terminvereinbarung per Doodle-Umfrage:
– Termine ab 29.05
– Link http://www.doodle.com/5pfhakn8n87sc6za
• Verlosung eines iPod-Shuffle (gesponsert durch www.InterFace-AG.de und Amazongutscheine unter den Teilnehmern
• Kontakt: Alexandru Zerva, ma@alexzerva.com


tsexperiment im Rahmen einer Masterarbeit.pdf - Adobe Acrobat Professional
Kreativitätsexperiment im Rahmen einer Masterarbeit
• Durchführung eines Experiments zur Ideengenerierung unterstützt durch die IdeaStream Plattform (www.ideastream.de)
• Dauer ca. 1 Stunde inkl. Einweisung
• Gruppen von 4 Studenten
• Ort: MI Gebäude, Diplomandenraum Lehrstuhl Prof. Schlichter
• Terminvereinbarung per Doodle-Umfrage:
– Termine ab 29.05
– Link http://www.doodle.com/5pfhakn8n87sc6za
• Verlosung eines iPod-Shuffle (gesponsert durch www.InterFace-AG.de und Amazongutscheine unter den Teilnehmern
• Kontakt: Alexandru Zerva, ma@alexzerva.com


Introduction to Java Basics.pptx - Microsoft PowerPoint
Introduction to Java Basics
Dipl.-Inf. Alexander Lehmann
Dr. Georg Groh
Fakultät für Informatik
Applied Informatics / Cooperative Systems AICOS
Technische Universität München
Klicken Sie, um Notizen hinzuzufügen


Introduction to Java Basics
Dipl.-Inf. Alexander Lehmann
Dr. Georg Groh
Fakultät für Informatik
Applied Informatics / Cooperative Systems AICOS
Technische Universität München

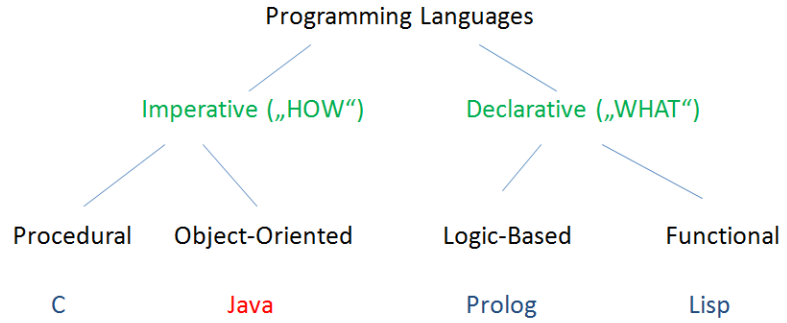
Lecture mainly follows
Sun's Java Tutorial available at
http://java.sun.com/docs/books/tutorial/

1 Java as a Programming Language

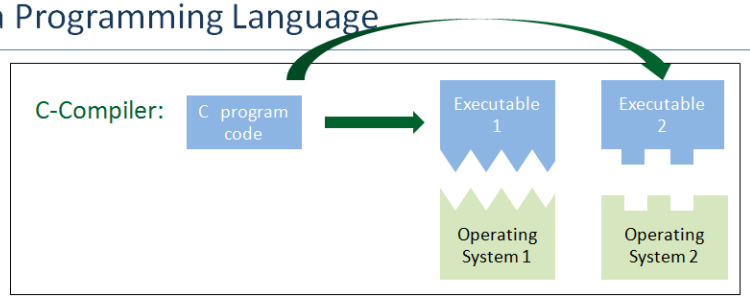
Deepening readings:

http://en.wikipedia.org/wiki/Imperative_programming
http://en.wikipedia.org/wiki/Declarative_programming
http://java.sun.com/docs/books/tutorial/java/concepts/object.html
http://java.sun.com/docs/books/tutorial/java/concepts/class.html
http://java.sun.com/docs/books/tutorial/java/concepts/inheritance.html
http://java.sun.com/docs/books/tutorial/java/concepts/interface.html

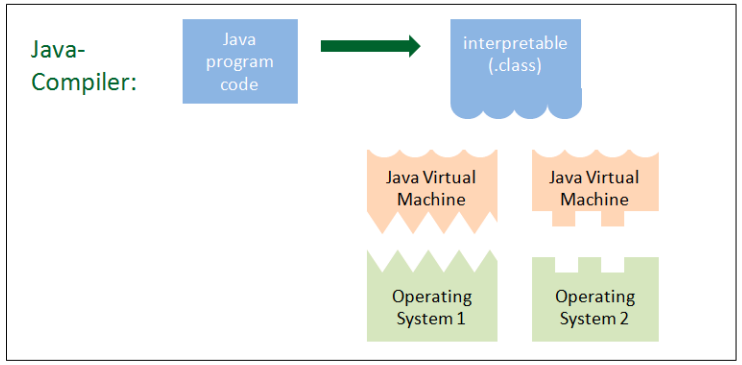
Programming Languages



Compiled Languages: (e.g. C)



Virtual Machine Languages: (e.g. Java)



Java as a Programming Language

Imperative Programming

- Imperative program:
  Sequence of statements
- Instructions change state (especially memory) of computer system

control flow

```
boolean plato;
int horst;
int heiner;
int fritz;
plato = false;
horst = 101;
heiner = 2;
fritz = horst + heiner;
horst = 2000;
```

memory (simplified model)

cell nr	cell name	cell content	
1123	plato	false	data
1124			
1125	horst	101	
1126	heiner	0	
1127			
1128	fritz	0	
4027		boolean plato;	instructions
4028		int horst;	
4029		int heiner;	
4030		int fritz;	
4029		plato = false;	
4030		horst = 101;	

Java as a Programming Language

Procedural Programming

- Group sequences of instructions into named „procedures“ („functions“, „methods“, „sub-routines“ etc.)

$f(x) = (x + x)^2$

```
int doSelfSumSquare(int someNumber){
        int a;
        a = someNumber + someNumber;
        a = a * a;
        return a;
}
```

- Advantages
  - no copying of instruction sequences
  - better testing
  - modularity
  - code re-use
  - etc.

Java as a Programming Language

```
int horst;
int heiner;
horst = 101;
heiner = 2;
heiner = doSelfSumSquare(horst);
heiner = doSelfSumSquare(117);
horst = horst + 2;
```

⋮

```
int doSelfSumSquare(int someNumber){
        int a;
        a = someNumber + someNumber;
        a = a * a;
        return a;
}
```

- In the example: Control flow is transferred to procedure, back to main program, back to procedure and back to main program

Java as a Programming Language

```
int horst;
int heiner;
horst = 101;
heiner = 2;
heiner = doSelfSumSquare(horst);
heiner = doSelfSumSquare(117);
horst = horst + 2;
```

⋮

```
int doSelfSumSquareHeiner(int someNumber){

        int a;
        a = someNumber + someNumber;
        a = a * a;
        a = a * heiner;
        return a;
}
```

- Procedures often access global variables (bad style!)
- Goal: „Keep things local“ (better testing, code re-use etc.)

Java as a Programming Language

Object-oriented Programming

- Object-oriented programming:

  Group data and procedures into objects ←→
  Models of state and behaviour of **real world** objects
  state „fields“ ; behaviour „methods“

- Methods should mainly act on an object's fields
- Classes: Blueprints for objects → Objects: Instances of classes
- Advantages
  - Intuitive models
  - Information hiding
  - Increased modularity, locality etc.
  - Increased code re-use
  - etc.

Java as a Programming Language

```
class Bicycle {
     int cadence = 0;
     int speed = 0;
     int gear = 1;

     void changeCadence(int newValue) {
          cadence = newValue;
     }

     void changeGear(int newValue) {
          gear = newValue;
     }

     void speedUp(int increment) {
          speed = speed + increment;
     }

     void applyBrakes(int decrement) {
          speed = speed - decrement;
     }
}
```

fields (state)

methods (behaviour)

class

Source: [JTutorial]

Java as a Programming Language

```
class BicycleDemo {
     public static void main(String[] args) {
          // Create two different Bicycle objects
          Bicycle bike1 = new Bicycle();
          Bicycle bike2 = new Bicycle();

          // Invoke methods on these objects
          bike1.changeCadence(50);
          bike1.speedUp(10);
          bike1.changeGear(2);

          bike2.changeCadence(50);
          bike2.speedUp(10);
          bike2.changeGear(2);
          bike2.changeCadence(40);
          bike2.speedUp(10);
          bike2.changeGear(3);
     }
}
```

```
class Bicycle {

     int cadence = 0;
     int speed = 0;
     int gear = 1;

     void changeCadence(int newValue) {
          cadence = newValue;
     }

     void changeGear(int newValue) {
          gear = newValue;
     }

     void speedUp(int increment) {
          speed = speed + increment;
     }

     void applyBrakes(int decrement) {
          speed = speed - decrement;
     }
}
```

Source: [JTutorial]