

Datenbanksysteme 2011

Anfang von Kapitel 7:
SQL

Vorlesung vom 16.05.2011

Oliver Vornberger

Institut für Informatik
Universität Osnabrück

SQL

- 1970 Edgar Codd:
A relational model for large shared data banks
- 1975 SEQUEL für System R von IBM
- 1977 Oracle gegründet
- 1979 SQL
- 1992 SQL-2, SQL-92, SQL:1992
- 1999 SQL-3, SQL-99, SQL:1999 (objektorientiert)
- 2003 SQL:2003 (XML)
- 2006 SQL:2006 (XQuery)
- 2008 SQL:2008 (Merge, instead of triggers, ...)

Relationale Datenbanksysteme

- DB2 IBM
- Informix IBM
- Database 11g Oracle
- Access Microsoft
- SQL Server Microsoft
- Ingres Open Source
- Postgres Open Source
- MySQL Open Source

MySQL

Populärstes OpenSource Datenbanksystem

Verfügbar für Linux, Windows, Mac OS X

70.000 Downloads am Tag, > 10.000.000 Installationen

1994 entstanden als Version 3.21 aus mSQL
von Michael Widenius, MySQL AB

2005 Version 5 (mit View, Trigger, Stored Procedures)

2008 Sun Microsystems kauft MySQL (1 Milliarde US-\$)

2010 Oracle kauft Sun Microsystems (7 Milliarden US-\$)

aktuelle Version: 5.5.11

LAMP

- **L**inux
- **A**pache
- **M**ySQL
- **P**HP

phpmyadmin

db.informatik.uos.de / localhost / UniWeb / Studenten | phpMyAdmin 2.11.8.1deb5 - Mozilla Firefox

Datei Bearbeiten Ansicht Chronik Lesezeichen Extras Hilfe

http://dbs.informatik.uos.de/phpmyadmin/index.php?db=UniWeb&lang=de-utf-8&c

Soll Firefox dieses Passwort speichern?

phpMyAdmin

Server: localhost ▶ Datenbank: UniWeb ▶ Tabelle: Studenten

Anzeigen Struktur SQL Suche Einfügen Exportieren Importieren

Operationen Leeren Löschen

Datenbank: UniWeb (11)

UniWeb (11)

- abgelehnt
- Assistenten
- gebucht
- hoeren
- konto
- Personen
- Professoren
- pruefen
- Studenten**
- voraussetzen
- Vorlesungen

Zeige Datensätze 0 - 13 (14 insgesamt, die Abfrage dauerte 0.0001 sek.)

SQL-Befehl:

```
SELECT Name, Titel
FROM Studenten, hoeren, Vorlesungen
WHERE Studenten.MatrNr = hoeren.MatrNr
AND hoeren.VorlNr = Vorlesungen.VorlNr
LIMIT 0, 30
```

☐ Messen [Bearbeiten] [SQL erklären] [PHP-Code erzeugen] [Aktualisieren]

Zeige: 30 Datensätze, beginnend ab 0

untereinander angeordnet und wiederhole die Kopfzeilen nach 100 Datensätzen.

Name	Titel
Jonas	Glaube und Wissen
Fichte	Grundzuege
Schopenhauer	Logik
Schopenhauer	Die 3 Kritiken

Fertig

<http://dbs.informatik.uni-osnabrueck.de/phpmyadmin>

MySQL Workbench: manage connections

The screenshot shows the 'Manage DB Connections' dialog box in MySQL Workbench. On the left, a 'Stored Connections' list contains one entry, 'dbs'. The main area is for configuring the selected connection. The 'Connection Name' is 'dbs'. The 'Connection Method' is 'Standard (TCP/IP)'. The 'Parameters' tab is active, showing fields for 'Hostname' (dbs.informatik.uni-osnabrueck.d), 'Port' (3306), 'Username' (oliver), 'Password' (with 'Store in Vault ...' and 'Clear' buttons), and 'Default Schema' (Uni). At the bottom, there are buttons for 'New', 'Delete', 'Duplicate', 'Move Up', 'Move Down', 'Test Connection', and 'Close'.

Manage DB Connections

Stored Connections

- dbs

Connection Name: Type a name for the connection

Connection Method: Method to use to connect to the RDBMS

Parameters **Advanced**

Hostname: Port: Name or IP address of the server host - TCP/IP port

Username: Name of the user to connect with.

Password: The user's password.

Default Schema: The schema that will be used as default schema

MySQL WorkBench: SQL

The screenshot displays the MySQL Workbench application window. The title bar reads "MySQL Workbench". The menu bar includes "File", "Edit", "View", "Query", "Database", "Plugins", "Scripting", "Community", and "Help". The toolbar contains various icons for file operations, execution, and navigation.

On the left side, the "Object Browser" panel shows a tree structure under "SCHEMAS" with "Uni", "UniTest", and "UniWeb". Below it, the "Object Information" panel is empty.

The main workspace is divided into two panes. The top pane, titled "Query 1", contains the following SQL query:

```
1 select titel, sws
2 from Professoren, Vorlesungen
3 where persnr=gelesenvon
4 and name='Sokrates'
```

The bottom pane, titled "Query 1 Result", shows the results of the query in a table format. The status bar indicates "Fetches 3 records. Duration: 0.000 sec, fetched in: 0.000 sec".

	titel	sws
▶	Logik	4
	Ethik	4
	Mäeutik	2

The status bar at the bottom of the window displays the message "Active schema changed to Uni".

MySQL Referenzhandbuch

The screenshot shows a web browser window titled "MySQL :: MySQL 5.1 Referenzhandbuch - Mozilla Firefox". The address bar shows the URL <http://dev.mysql.com/doc/refman/5.1/de/>. The page features the MySQL logo and Sun Microsystems logo at the top. Below the logo, it says "The world's most popular open source database". The navigation bar includes links for "MySQL.com", "Developer Zone", "Partners & Solutions", and "Customer Login". The main content area is titled "MySQL 5.1 Referenzhandbuch" and contains a "Section Navigation" list with 15 items, including "Vorwort", "1 Allgemeine Informationen über MySQL", "2 Installation von MySQL", "3 Einführung in MySQL: ein MySQL-Tutorial", "4 Benutzung von MySQL-Programmen", "5 Datenbankverwaltung", "6 Replikation bei MySQL", "7 Optimierung", "8 Client- und Hilfsprogramme", "9 Sprachstruktur", "10 Zeichensatz-Unterstützung", "11 Datentypen", "12 Funktionen für die Benutzung in SELECT- und WHERE-Klauseln", "13 SQL-Anweisungssyntax", "14 Speicher-Engines und Tabellentypen", and "15 Erstellung einer eigenen". The left sidebar contains a "Documentation Library" section with a "Table of Contents" and a "Search manual:" field. The bottom of the page shows a "Fertig" button.

MySQL 5.1 Referenzhandbuch

Dies ist eine Übersetzung des MySQL-Referenzhandbuchs, das sich auf dev.mysql.com befindet. Das ursprüngliche Referenzhandbuch ist auf Englisch, und diese Übersetzung ist nicht notwendigerweise so aktuell wie die englische Ausgabe. Das vorliegende deutschsprachige Handbuch behandelt MySQL bis zur Version 5.1.

Copyright 1997-2008 MySQL AB, 2008 Sun Microsystems, Inc.

Diese Dokumentation wird NICHT unter der GPL-Lizenz vertrieben. Die Benutzung dieser Dokumentation unterliegt folgenden Bedingungen: Sie dürfen diese Dokumentation ausschließlich für private Zwecke ausdrucken. Eine Umwandlung in andere Formate ist gestattet, solange die ursprünglichen Inhalte nicht verändert oder in sonstiger Weise editiert werden. Sie dürfen diese Dokumentation weder veröffentlichen noch in sonstiger Form auf irgendeinem Medium vertreiben, außer wenn Sie die Dokumentation in ähnlicher Form wie Sun selbst verteilen (das heißt elektronisch zum Download auf einer Website zusammen mit der Software) oder auf einer CD-ROM oder einem ähnlichen Medium, vorausgesetzt, dass die Dokumentation zusammen mit der Software auf diesem Medium verteilt wird. Jeder sonstige Gebrauch, beispielsweise die

Section Navigation [Toggle]

- MySQL 5.1 Referenzhandbuch
 - Vorwort
 - 1 Allgemeine Informationen über MySQL
 - 2 Installation von MySQL
 - 3 Einführung in MySQL: ein MySQL-Tutorial
 - 4 Benutzung von MySQL-Programmen
 - 5 Datenbankverwaltung
 - 6 Replikation bei MySQL
 - 7 Optimierung
 - 8 Client- und Hilfsprogramme
 - 9 Sprachstruktur
 - 10 Zeichensatz-Unterstützung
 - 11 Datentypen
 - 12 Funktionen für die Benutzung in SELECT- und WHERE-Klauseln
 - 13 SQL-Anweisungssyntax
 - 14 Speicher-Engines und Tabellentypen
 - 15 Erstellung einer eigenen

<http://dev.mysql.com/doc/refman/5.1/de/>

SQL: numerische Datentypen

8	bigint	ganze Zahlen von -2^{63} bis $+2^{63}$
4	int	ganze Zahlen von -2^{31} bis $+2^{31}$
3	mediumint	ganze Zahlen von -2^{23} bis $+2^{23}$
2	smallint	ganze Zahlen von -2^{15} bis $+2^{15}$
1	tinyint	ganze Zahlen von -128 bis +127
1	bit	ganze Zahlen von 0 bis 1
1	boolean	alias für bit
d	decimal(n,k)	feste Genauigkeit, n Stellen, davon k nach Komma 9 digits in 4 Bytes, d.h. $d \approx \lceil n/9 \rceil * 4$
d	numeric(n,k)	alias für decimal
4	float	Gleitkommazahlen von -10^{38} bis $+10^{38}$
8	double, real	Gleitkommazahlen von -10^{308} bis $+10^{308}$

Microsoft SQL Server:

8	money	Währungswerte mit 4 Nachkommastellen (MS SQL)
---	--------------	---

SQL Datentypen für Zeitangaben

1	year	von 1901 to 2155
3	date	von 01.01.0001 bis 31.12.9999 kodiert als $DD + 32*MM + 32*16*YYYY$
8	datetime	von 00.00.0000 00:00:00 bis 31.12.9999 23:59:59 kodiert als $YYYY*10000 + MM*100 + DD$ $HH*10000 + MM*100 + SS$ (nicht als Millisekunden seit 1.1.1970)
4	timestamp	von 01.01.1970 bis 31.12.2037 (beim Einfügen, inkl. Uhrzeit) kodiert als Sekunden nach 1.1.1970
3	time	von -838:59:59 bis 838:59:59 kodiert als $HH*3600 + MM*60 + SS$

SQL: Datentypen für Zeichenketten

n **char**(n) Zeichenkette fester Länge
mit $n \leq 255$ Zeichen

$n+d$ **varchar**(n) Zeichenkette variabler Länge
mit $n \leq 65535$ Zeichen
[zusätzlich d Bytes für Längenangabe]

$n+d$ **text** Zeichenkette variabler Länge
[zusätzlich d Bytes für Längenangabe]

die ersten 256 Zeichen in Originaltabelle
die nächsten Zeichen in 2000-Bytes-Blöcken
in verborgenen Tabellen

SQL: Datentypen für Binärdaten

n binary(n) Binärdaten fester Länge
mit $n \leq 255$ Bytes

n+d varbinary(n) Binärdaten variabler Länge
mit $n \leq 65535$ Bytes
[zusätzlich d Bytes für Längenangabe]

n+x blob Binärkette variabler Länge
[zusätzlich x Bytes für Verwaltung]

die ersten 256 Bytes in Originaltabelle
die nächsten Zeichen in 2000-Bytes-Blöcken
in verborgenen Tabellen

SQL Mengen und Aufzählungen

- 8 `set` Menge von bis zu 64 Elementen
- 2 `enum` Liste von bis zu 65.535 Elementen

SQL: create

```
Create table Personen (  
    persnr          int primary key auto_increment,  
    name            char(30) not null,  
    geschlecht      boolean default 0,  
    note            decimal (3,2),  
    groesse         float,  
    gewicht         double,  
    gebDatum        date,  
    einschulung     year,  
    marathon        time,  
    bemerkung       text,  
    photo           blob,  
    zugriff         timestamp,  
    kombination     set ('rot ', 'gruen', 'blau')  
    ) auto_increment = 100000;
```

SQL: alter, modify, drop

Tabelle um eine Spalte erweitern:

```
alter table Personen  
add    Vorname varchar(15)
```

Tabellenspalte ändern:

```
alter table Personen  
modify Vorname varchar(20)
```

Tabelle um eine Spalte verkürzen:

```
alter table Personen  
drop column Vorname
```

Tabelle entfernen:

```
drop table Personen
```


SQL: Schlüsselworte

select	distinct	in
from	count	not
where	sum	null
order by	avg	exists
asc	max	all
desc	min	some
as	group by	
like	having	
upper		
lower		

[zum Quiz](#)

SQL: select, from, where

1.) Liste alle Studenten:

```
select * from Studenten
```

2.) Liste Personalnummer und Name der C4-Professoren:

```
select PersNr, Name  
from Professoren  
where Rang= 'C4'
```

SQL: count, as, is not, null

3.) Zähle alle Studenten

```
select count(*) from Studenten
```

4.) Liste Name und Studiendauer in Jahren von allen Studenten:

```
select Name, Semester/2 as Studienjahr  
from Studenten  
where Semester is not null
```

SQL: between, in

5.) Liste alle Studenten mit Semesterzahlen zwischen 1 und 4:

```
select *  
from Studenten  
where Semester >= 1 and Semester <= 4
```

alternativ

```
select *  
from Studenten  
where Semester between 1 and 4
```

alternativ

```
select *  
from Studenten  
where Semester in (1,2,3,4)
```

SQL: like, order, distinct

6.) Liste alle Vorlesungen mit `Ethik` im Titel:

```
select * from Vorlesungen
where Titel like '%ETHIK'
```

7.) Liste Personalnummer, Name und Rang aller Professoren, absteigend sortiert nach Rang, innerhalb des Rangs aufsteigend sortiert nach Name:

```
select PersNr, Name, Rang
from Professoren
order by Rang desc, Name asc
```

8.) Liste alle verschiedenen Ränge der Relation Professoren:

```
select distinct Rang
from Professoren
```

SQL: Datum

9.) Liste alle Geburtstage mit ausgeschriebenem Monatsnamen:

```
select  name,  
        Day    (Gebdatum) as Tag,  
        Month  (GebDatum) as Monat,  
        Year   (GebDatum) as Jahr  
from    Studenten
```

10.) Liste das Alter der Studenten in Jahren:

```
select name, year(Now())-year(gebdatum) as Jahre  
from Studenten
```

SQL Datumsfunktionen

11.) Liste die Wochentage der Geburtsdaten der Studenten:

```
select name,  
dayname(GebDatum) as Wochentag  
from Studenten
```

12.) Liste die Kalenderwochen der Geburtsdaten der Studenten:

```
select name,  
week(GebDatum) as Kalenderwoche  
from Studenten
```

SQL: Verbund

13.) Liste den Dozenten der Vorlesung Logik:

```
select  Name, Titel
from    Professoren, Vorlesungen
where   PersNr = gelesenVon and Titel = 'Logik'
```

14.) Liste die Namen der Studenten mit ihren Vorlesungstiteln:

```
select  Name, Titel
from    Studenten, hoeren, Vorlesungen
where   Studenten.MatrNr = hoeren.MatrNr
and     hoeren.VorlNr    = Vorlesungen.VorlNr
```

alternativ:

```
select  s.Name, v.Titel
from    Studenten s, hoeren h, Vorlesungen v
where   s.MatrNr = h.MatrNr
and     h.VorlNr = v.VorlNr
```


SQL: Self Join

- 15.) Liste die Namen der Assistenten, die für denselben Professor arbeiten, für den Aristoteles arbeitet:

```
select  a2.Name
from    Assistenten a1, Assistenten a2
where   a2.boss  =  a1.boss
and     a1.name  =  'Aristoteles'
and     a2.name  != 'Aristoteles'
```