

ACLS















Databases



SS 2020 – Week 3

March 2

Schedule

Calendar Week	8	9	10	11	12	13	14	15	16	17	18	19	20	21
Module Week	1	2	3	4	5	6	7	8	9	10	11	12	13	14
Date	17.02.	24.02.	02.03.	09.03.	16.03.	23.03.	30.03.	06.04.	13.04.	20.04.	27.04.	04.05.	11.05.	18.05.
Topic														
	February		March					April			May			



Introduction & RDB



Structured Query Language



Database & Python



Database & R



Data Warehouse



Not only SQL



Graph Database



Special

Content

• Review + Scripts		10'
• JOINS	🎓	35'
• Nested Queries	🎓	25'
• Views - Procedures	🎓	20'
• SQLite	🎓	20'
• Exercises	🔧	25'
• Exercises	🔧	90'



Content

• Review + Scripts		10'
• JOINS	🎓	35'
• Nested Queries	🎓	25'
• Views - Procedures	🎓	20'
• SQLite	🎓	20'
• Exercises	🔧	25'
• Exercises	🔧	90'



Review (1/3)

SQL



Many Systems* – one Language

*DataBase-Management-Systems (DBMS)

There aren't any alternatives to SQL for speaking to relational databases, but there are many alternatives to writing SQL in your applications.

- SchemeQL and CLSQL
- LINQ (in .Net)
- ScalaQL and ScalaQuery (in Scala)
- SqlStatement
- ActiveRecord
- HaskellIDB
- Hibernate
- ...

⚠ obviously not exhaustive

```
//Query syntax:
IEnumerable<int> numQuery1 =
    from num in numbers
    where num % 2 == 0
    orderby num
    select num;

//Method syntax:
IEnumerable<int> numQuery2 = numbers.Where(num => num % 2 == 0).OrderBy(n => n);
```

MySQL



- Open-source relational database management system (RDBMS)
- Server software itself and the client libraries use dual-licensing distribution

Review (2/3)

CREATE – MODIFY – DELETE



Databases

CREATE DATABASE *databasename*;
SHOW DATABASES;
DROP DATABASE *databasename*;



Tables

CREATE TABLE *tablename*(...); **SHOW TABLES**;
ALTER TABLE *tablename* **ADD** | **MODIFY** | **DROP** ...;
DROP DATABASE *tablename*;



Records

INSERT INTO *tablename*(...) **VALUES** (...);
UPDATE *tablename* **SET** ... **WHERE** ...;
DELETE FROM *tablename* **WHERE** ...;

Review (3/3)

RETRIEVE



SELECT ... FROM *tablename* **WHERE ...;**

Scripts (1/3)

SQL statements:

- SHOW DATABASES;
- USE myfirstdb;
- SELECT * FROM clients;

Command prompt

```
Eingabeaufforderung - mysql -u root -p
Microsoft Windows [Version 10.0.15063]
(c) 2017 Microsoft Corporation. Alle Rechte vorbehalten.

C:\Users\voru>mysql -u root -p
Enter password: *****
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 57
Server version: 5.7.20-log MySQL Community Server (GPL)

Copyright (c) 2000, 2017, 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> SHOW DATABASES;
+-----+
| Database |
+-----+
| information_schema |
| employees |
| myfirstdb |
| mysql |
| performance_schema |
| projectscmd |
| sys |
+-----+
7 rows in set (0.01 sec)

mysql> use myfirstdb;
Database changed
mysql> SELECT * FROM clients;
+-----+
| ID | FirstName | LastName | Address | AboTypeID |
+-----+
| 1 | John | Smith | Hauptstrasse 15 | 1 |
| 2 | Paul | Jones | Main Street 5 | 1 |
| 3 | Mary | Bond | 5th Avenue | 1 |
+-----+
3 rows in set (0.03 sec)

mysql>
```

Script file

```
C:\Users\voru\ZHAW\2018_FS_Databases\09_Sandbox\MySQL\script_example.sql - Notepad++
Datei Bearbeiten Suchen Ansicht Kodierung Sprachen Einstellungen Werkzeuge Makro Ausführen
Erweiterungen Fenster ?
script_example.sql
1  -- List all databases
2  SHOW DATABASES;
3
4  -- use specific database
5  USE myfirstdb;
6
7  -- show all records
8  SELECT * FROM clients;
```

How to execute?

Scripts (2/3)

Option 1

```
> mysql -u root -p < path-to-file/my_script.sql
```

```
> mysql -u root -p -t < path-to-file/my_script.sql
```

Option 2

```
> mysql -u root -p
```

```
Enter password: *****
```

```
mysql> SOURCE path-tofile/my_script.sql;
```

```
C:\Users\voru>mysql -u root -p < C:/script_example.sql
Enter password: *****
Database
information_schema
employees
myfirstdb
mysql
performance_schema
projectscmd
sys
ID      FirstName  LastName  Address  AboTypesID
1       John      Smith     Hauptstrasse 15  1
2       Paul      Jones     Main Street 5    1
3       Mary      Bond      5th Avenue      1

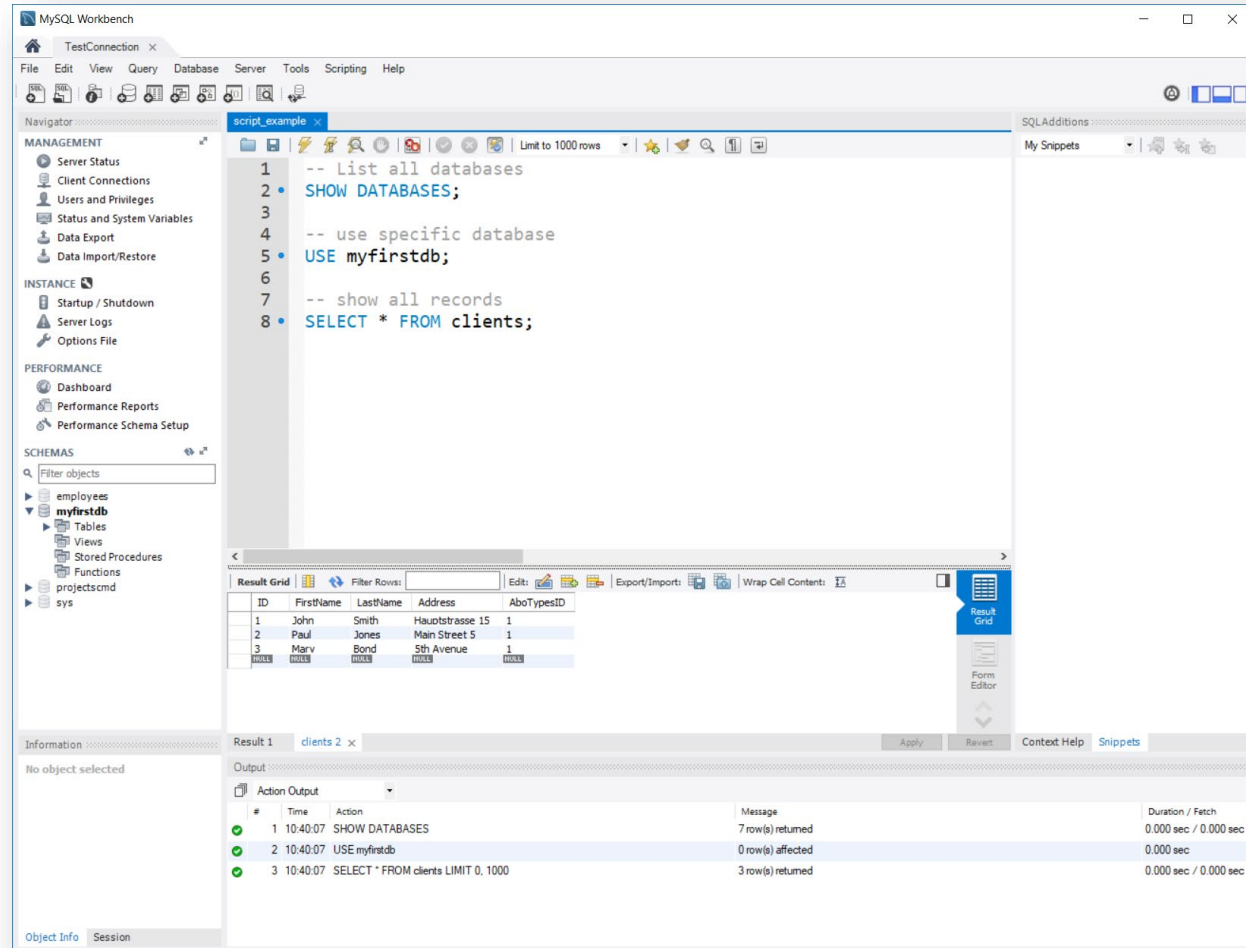
C:\Users\voru>
```

```
C:\Users\voru>mysql -u root -p
Enter password: *****
mysql> SOURCE C:/script_example.sql;
+-----+
| Database |
+-----+
| information_schema |
| employees |
| myfirstdb |
| mysql |
| performance_schema |
| projectscmd |
| sys |
+-----+
7 rows in set (0.00 sec)

Database changed
+-----+
| ID | FirstName | LastName | Address | AboTypesID |
+-----+
| 1 | John | Smith | Hauptstrasse 15 | 1 |
| 2 | Paul | Jones | Main Street 5 | 1 |
| 3 | Mary | Bond | 5th Avenue | 1 |
+-----+
3 rows in set (0.00 sec)

mysql>
```

Scripts (3/3)



The screenshot shows the MySQL Workbench interface. The left sidebar contains the Navigator pane with sections for MANAGEMENT, INSTANCE, PERFORMANCE, and SCHEMAS. The SCHEMAS section is expanded, showing a tree view with 'employees', 'myfirstdb' (selected), 'Tables', 'Views', 'Stored Procedures', 'Functions', 'projects', and 'sys'. The main editor window displays a SQL script named 'script_example' with the following content:

```

1  -- List all databases
2  SHOW DATABASES;
3
4  -- use specific database
5  USE myfirstdb;
6
7  -- show all records
8  SELECT * FROM clients;

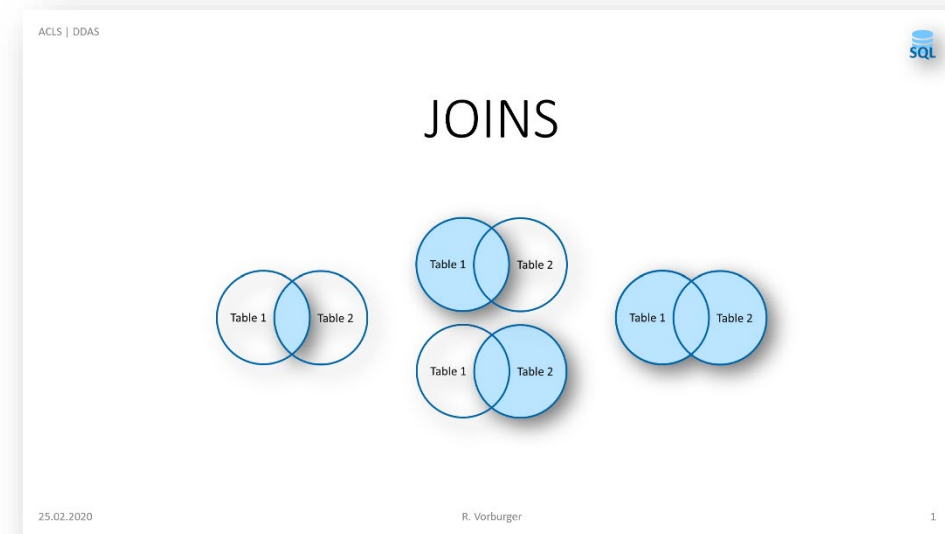
```

Below the script editor, the 'Result Grid' is visible, showing the results of the executed queries. The first query, 'SHOW DATABASES;', returned 7 rows. The second query, 'USE myfirstdb;', affected 0 rows. The third query, 'SELECT * FROM clients LIMIT 0, 1000', returned 3 rows. The 'Output' pane at the bottom shows the execution log with the following details:

#	Time	Action	Message	Duration / Fetch
1	10:40:07	SHOW DATABASES	7 row(s) returned	0.000 sec / 0.000 sec
2	10:40:07	USE myfirstdb	0 row(s) affected	0.000 sec
3	10:40:07	SELECT * FROM clients LIMIT 0, 1000	3 row(s) returned	0.000 sec / 0.000 sec

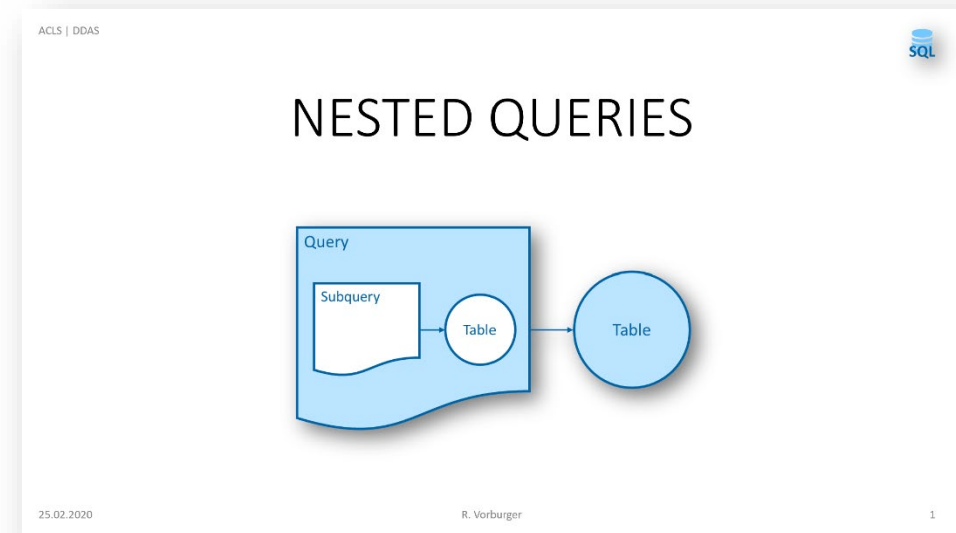
Content

• Review + Scripts		10'
• JOINS	🎓	35'
• Nested Queries	🎓	25'
• Views - Procedures	🎓	20'
• SQLite	🎓	20'
• Exercises	🔧	25'
• Exercises	🔧	90'



Content

• Review + Scripts		10'
• JOINS	🎓	35'
• Nested Queries	🎓	25'
• Views - Procedures	🎓	20'
• SQLite	🎓	20'
• Exercises	🔧	25'
• Exercises	🔧	90'



Content

• Review + Scripts		10'
• JOINS	🎓	35'
• Nested Queries	🎓	25'
• Views - Procedures	🎓	20'
• SQLite	🎓	20'
• Exercises	🔧	25'
• Exercises	🔧	90'

ACLS | DDAS

SQL







Views - Procedures

27.02.2020

R. Vorburger







1

Content

• Review + Scripts		10'
• JOINS		35'
• Nested Queries		25'
• Views - Procedures		20'
• SQLite		20'
• Exercises		25'
• Exercises		90'



Content

• Review + Scripts		10'
• JOINS		35'
• Nested Queries		25'
• Views - Procedures		20'
• SQLite		20'
• Exercises		25'
• Exercises		90'

MSc ACLS Databases and Data

[Dashboard](#) / [My courses](#) / [ACLS DDAS SS20](#) / [Exercises](#) / [Week 03](#)

Week 03

solutions

joins.pdf

nested.pdf

view_procedures.pdf

Download folder