















ACLS

Databases



SS 2020 – Week 4
March 9

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		5'
• Project – Task 1	🎓	20'
• Python and MySQL	🎓	65'
• Python and SQLite	🎓	45'
• Exercises	🔧	90'



Content

• Review		5'
• Project – Task 1	🎓	20'
• Python and MySQL	🎓	65'
• Python and SQLite	🎓	45'
• Exercises	🔧	90'



Review (1/4)

JOINS

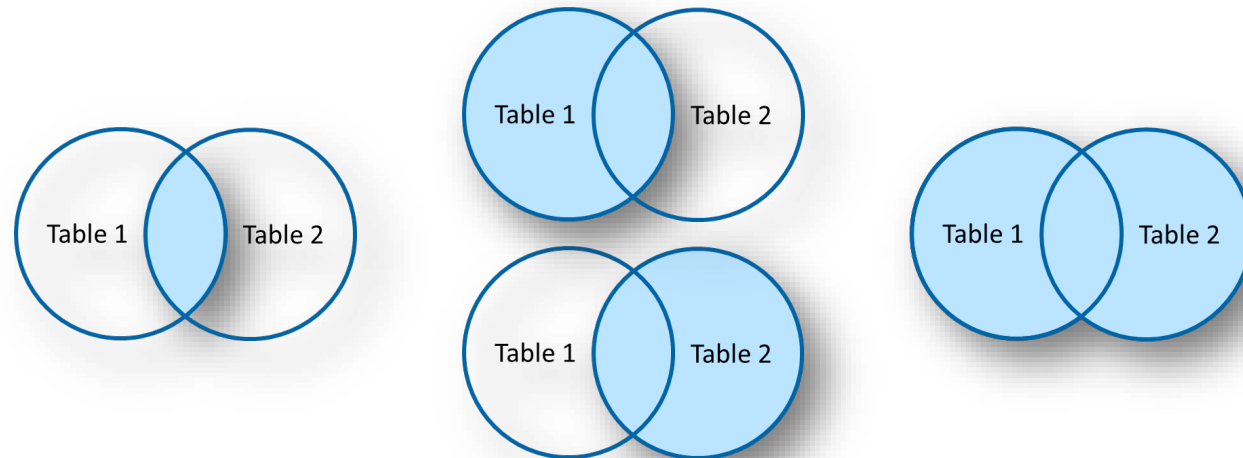
Syntax

SELECT *table1.attribute1, table1.attribute2, ..., table2.attribute1, table2.attribute2, ...*

FROM *table1*

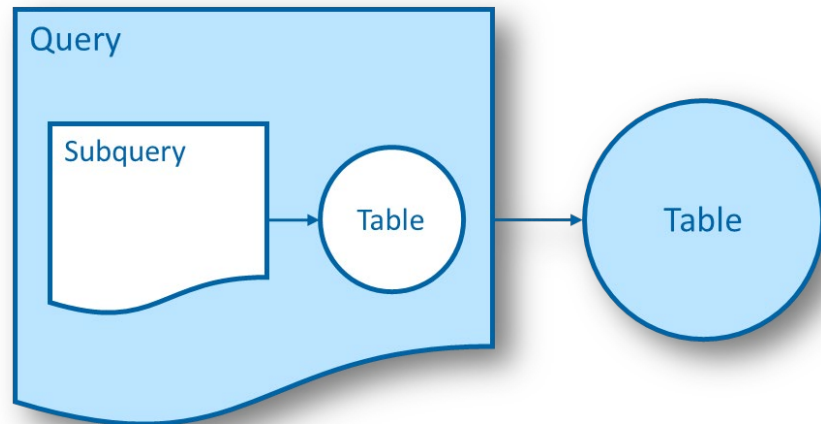
INNER JOIN *table2 ON table1.attributeN = table2.attributeM1*

WHERE *condition;*



Review (2/4)

NESTED QUERIES



Example

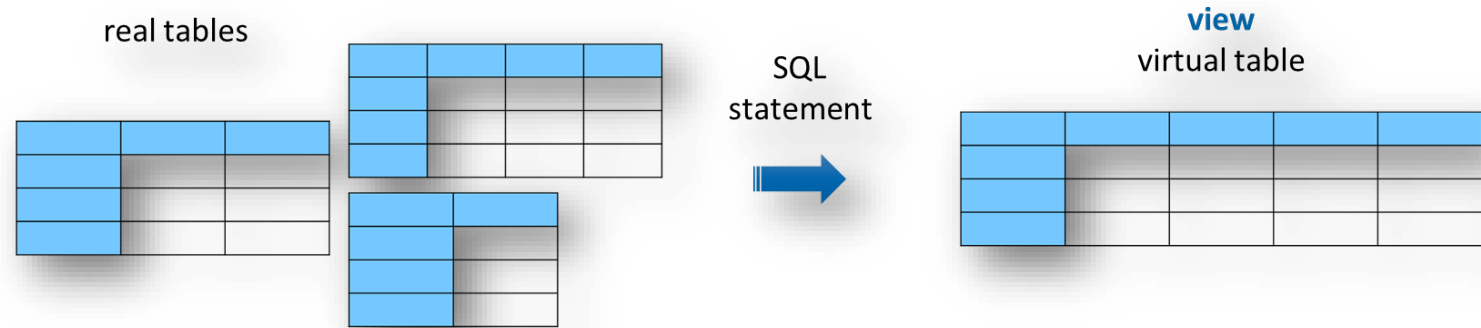
```

/* Get all employees with a salary larger than the salary of the
   employee with the ID=35 */
SELECT Employees.LastName, Salaries.Salary FROM Employees
INNER JOIN Salaries ON Employees.ID=Salaries.EmployeeID
WHERE Salaries.Salary>(SELECT Salary
                        FROM Salaries
                        WHERE Salaries.EmployeeID=35);
    
```

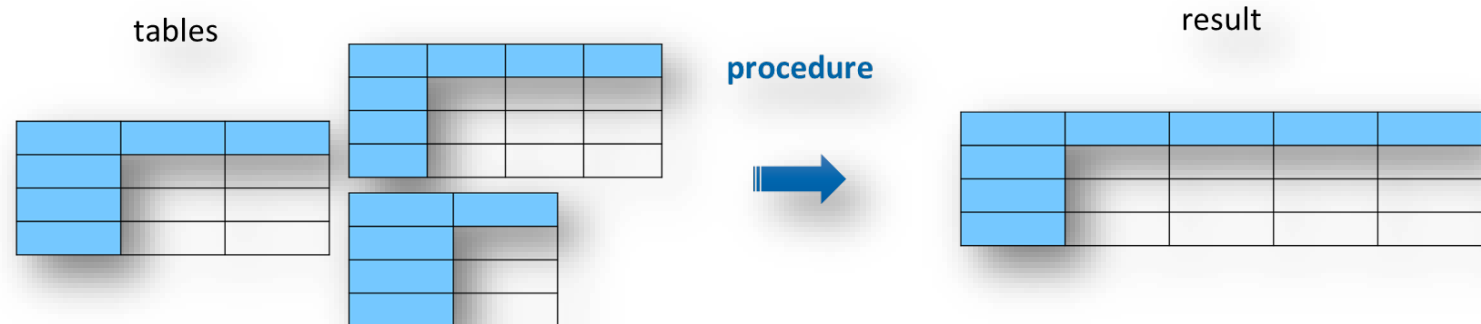
Review (3/4)

Views – Procedures

A **view** is a **virtual table** based on the result of an SQL query.



A **procedure** is a **subroutine** that can be called with parameters.



Review (4/4)

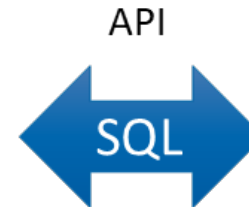
SQLite

your code

```

1 #include <stdio.h>
2 #include <sqlite3.h>
3
4 static int callback(void *NotUsed, int argc, char **argv, char **azColName){
5     int i;
6     for(i=0; i<argc; i++){
7         printf("%s = %s\n", azColName[i], argv[i] ? argv[i] : "NULL");
8     }
9     printf("\n");
10    return 0;
11 }
12 int main(int argc, char **argv){
13     sqlite3 *db;
14     char *zErrMsg = 0;
15     int rc;
16
17     if( argc!=3 ){
18         fprintf(stderr, "Usage: %s DATABASE SQL-STATEMENT\n", argv[0]);
19     }
20 }

```



database in a file



Content

• Review		5'
• Project – Task 1	🎓	20'
• Python and MySQL	🎓	65'
• Python and SQLite	🎓	45'
• Exercises	🔧	90'



Project

Scenario

1 The Lab

The project takes place in a lab for chemical research. Not much is known about the company or the place the lab is located. The only information available is the setup of the laboratory benches.

The lab contains three pairs of laboratory benches (six benches in total) that are suited for three different classes of experiments.

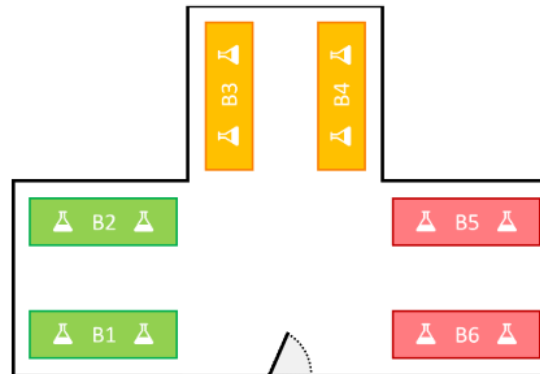


Figure 1: Depiction of the six laboratory benches.

The green, orange, and red benches are designed for experiments of class 1, 2, and 3, respectively.

Project

Task 1 – Creating the Database

MSc ACLS Databases and Data

[Dashboard](#) / [My courses](#) / [ACLS DDAS SS20](#) / [Project](#) / [Task 1 - Creating the database](#)

Task 1 - Creating the database

 [p1_database.pdf](#) +

4 March 2020, 11:14 AM

Due on 22nd of March

ACLS | DDAS

Project

Project – Task 1

Database

1 Goal

The goal of the first task is to create the database for the lab. Therefore, the entity-relationship model should be designed and a SQL-Script that creates a MySQL database and the required tables (see below 4. Tables). A description of the lab is given in the document [p0_scenario.pdf](#).

2 Entity-Relationship Model

2.1 Task

☐ Design the ER-Model of the database that should be created for the lab. Hand-in the ER-Model as a PDF on Moodle.

2.2 Grading

The ER-Model is worth 10 points according to the following list:

- 3 points for correct Entities
- 3 points for useable attribute types (i.e. int, varchar, datetime, etc.)
- 3 points for correct Relationships
- 1 point for a nice¹ arranging of the Entities and Relationships

3 SQL-Script

3.1 Task

☐ Write a SQL-script that creates a MySQL database and the tables according to the ER-Model. Hand-in the SQL-script (.sql-File) on Moodle. *NOTE: The tables should not be filled with records. All tables should be empty after running the script.*

3.2 Grading

The SQL script is worth 10 points as listed below:





- 1 point for running without an error
- 2 points for the correct tables
- 2 points for the correct attributes in all tables
- 3 points for setting the keys correctly
- 2 points for commenting the statements

¹ No crossings of relationships if possible

SS 2020

Robert Vorbürger

Content

• Review		5'
• Project – Task 1		20'
• Python and MySQL		65'
• Python and SQLite		45'
• Exercises		90'







Content

• Review		5'
• Project – Task 1	🎓	20'
• Python and MySQL	🎓	65'
• Python and SQLite	🎓	45'
• Exercises	🔧	90'



Content

• Review		5'
• Project – Task 1		20'
• Python and MySQL		65'
• Python and SQLite		45'
• Exercises		90'

MSc ACLS Databases and Data

[Dashboard](#) / [My courses](#) / [ACLS DDAS SS20](#) / [Exercises](#) / [Week 04](#)

Week 04

solutions

categories.csv

nobel_prize_winners.csv

python_mysql.pdf

python_sqlite.pdf