HOCHSCHULE FÜR TECHNIK STUTTGART
UNIVERSITY OF APPLIED SCIENCES

MASTER'S COURSE **S**oftware **T**echnology
MASTERSTUDIENGANG **M**athematik

EXAMINATION in **SUMMER SEMESTER 2016**

MODULE: **Databases II** NAME:

DATE: **6 July 2016** SEMESTER:

TIME: **14.00 - 16.00 Uhr**

EXAMINER: **Prof. D. Koch**

ALLOWED AIDS: All materials that were distributed in class and in the Moodle, all your own notes, two text books of your choice, plus an English dictionary.
NOT ALLOWED: Mobile Phones, laptop, watches and other communication devices
APPENDIX: Schema and tuples of the universitydb database from the class notes.

Please write your name on each sheet that you turn in.

Turn in the problem sheets as well!

Problem	**1**	**2**	**3**	**4**	**5**	**6**	Σ
Maximum points	20	20	20	20	20	20	120
Achieved points							
Grade							

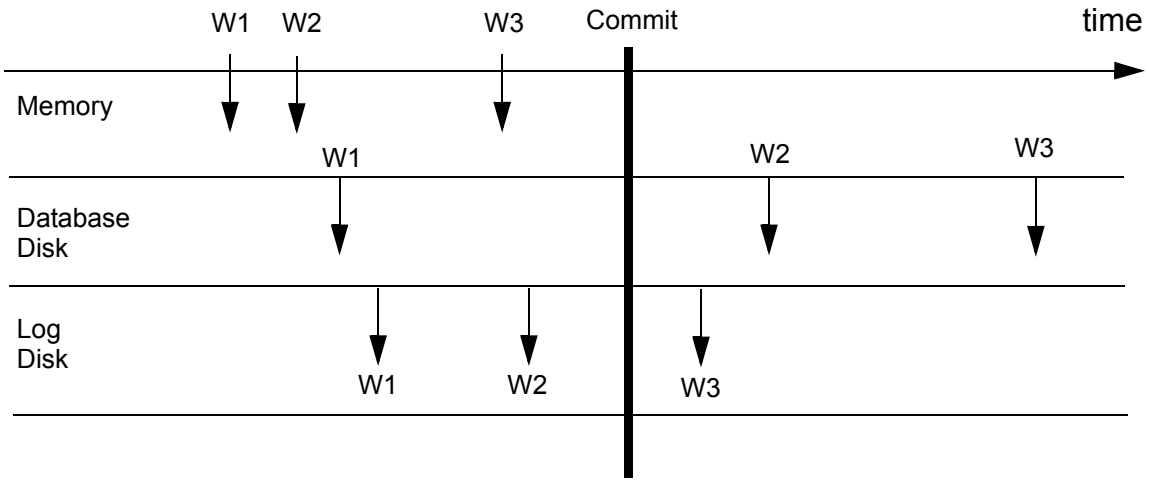
Problem 1. (2 times 10 points = 20 points)

Consider a database system that uses a steal/no force strategy and logfiles on a separate disk.
In the following diagrams you see update operations made by a committed transaction. Each arrow indicates the point in time when a piece of information is written to some type of media:
Wi: The operation of writing the updated value number i on some type of media.
Example: W2 associated with an arrow means: The second value that was updated by the transaction was written to the respective media (main memory, data disk or log disk).

Your tasks:

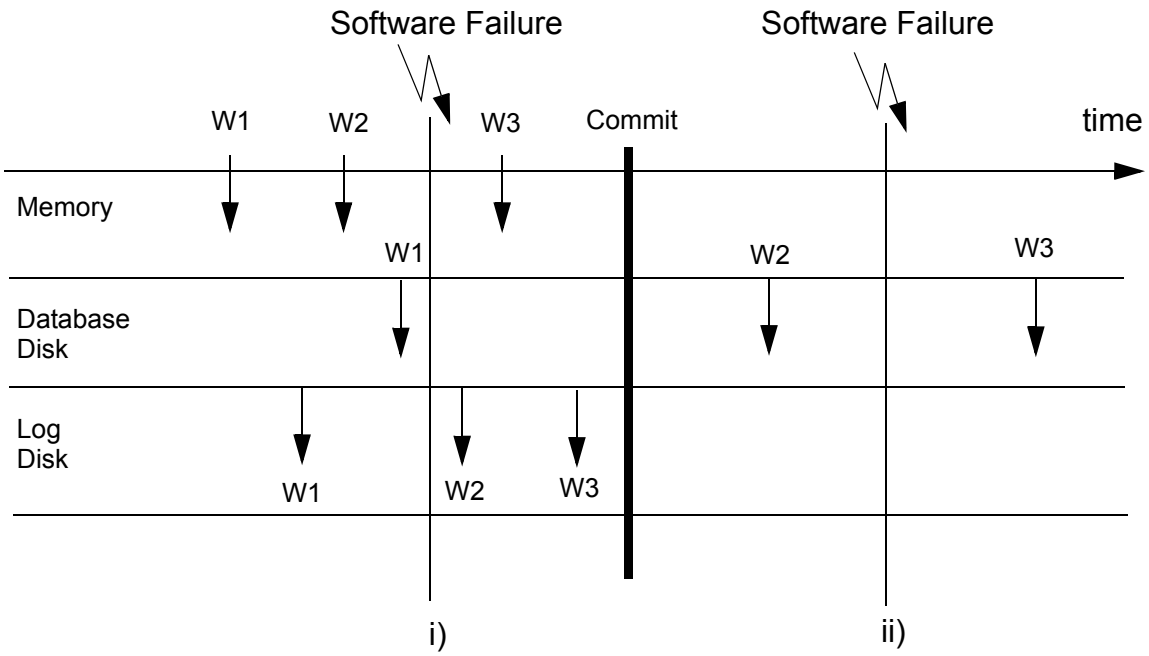
a) (10 points)

Evaluate the scenario in the diagram: Does it follow the principles for logging that ensure that the correctness of the database will be granted even when the system fails?
If the answer is yes, then explain how problems are prevented by these principles.
If the answer is no, then name the principle(s) that is/are violated and explain one example for each violated principle of what could go wrong.



b) (10 points)
Now consider a similar scenario which is outlined in the next diagram. Assume that the database disk is unharmed and that the last checkpoint did not affect the transaction described in the scenario.
Case i): a software failure happens at the time denoted by the line marked by i)
Case ii): a software failure happens at the time denoted by the line marked by ii).

Explain for case i) and for case ii) which steps must be taken for recovery of the transaction and why.
(Case i) and case ii) do not happen in the same universe, of course. These are two different questions.)



Problem 2. (8 points + 12 points = 20 points)

Consider the following Java code that a programmer has written in order to import SQL query results from a MySQL database into a program: You can assume that the class KeyboardInput.java for reading input that the user types (here the SQL query) exists exactly as in our lab exercises and presents no problems.

```
 1 import java.io.*;
 2 import java.sql.*;
 3
 4 public class Queries {
 5
 6   public static void main (String[] args)
 7         throws SQLException, IOException {
 8
 9      try {Class.forName("com.mysql.jdbc.Driver");
10           /* forName returns the class or interface object
11              with the name specified by the parameter string.
12            This line here doesn't do anything with it, but just
13            by referring to it, the class is loaded into the
14            JVM => load the JDBC driver for Oracle */
15      }
16      catch (ClassNotFoundException e) {
17         System.out.println("Driver could not be loaded.");
18      }
19
20        String query = KeyboardInput.readString();
21        Statement stmt = Conn.createStatement();
22        ResultSet rs = stmt.executeQuery(query);
23        ResultSetMetaData rsmeta = rs.getMetaData();
24        int numCols = rsmeta.getColumnCount();
25
26        for (int i=1; i <= numCols; i++) {  // print schema info
27            System.out.println("Column #" + i + ": " +
28                               rsmeta.getColumnName(i) + " of type "
29                               + rsmeta.getColumnTypeName(i));
30        }// for
31        while (rs.next()) {
32            for (int i=1; i<=numCols; i++){
33                if (rsmeta.getColumnTypeName(i).equals("VARCHAR")){
34                    String value1 = rs.getString(i);
35                    System.out.print(value1 + "\t");
36                } //if
37
38                if (rsmeta.getColumnTypeName(i).equals("INT")){
39                    int value2 = rs.getInt(i);
40                    System.out.print(value2 + "\t");
41                } //if
/**** Comment by Prof. Koch: imagine similar if-statements here for the
      other data types. I'm not printing them here for space reasons  ****/
42                System.out.println();
43            } //for
44        } //while
45   } //main
46} //Queries
```

This code contains a problem that will prevent the program from running.
See next page for your task =>

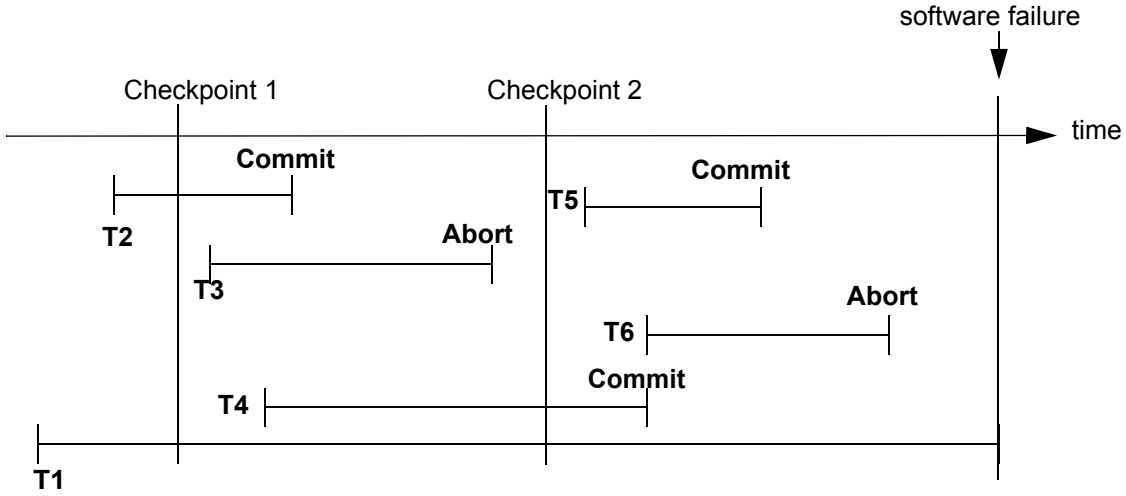
Your task for problem 2:

a) (8 points)
Identify the problem and explain why the program cannot run like this. (Hint: It is a severe problem in the programming logic, not just a missing parenthesis or similar small error).

b) (12 points)
Fix the problem by writing the necessary lines of code. Write before your lines the number of the row after which you would insert them.

Problem 3. (2 times 10 points = 20 points)

Consider the following transaction execution scenario with a log file where sharp checkpointing is used:



a) (10 points)
List which transactions the recovery manager must undo, which ones it must redo, and for which ones no action is necessary, when the software failure interrupts the system. Briefly explain your choice. In particular: explain which checkpoints took care of which actions concerning which transaction.

b) (10 points)
Assume again the situation in the diagram where checkpoint 2 has completed successfully. Now briefly explain the necessary actions of the recovery manager if the software failure happens in parallel with a crash of the database disk.
In particular explain what needs to be redone and what needs to be undone. What is the role of the checkpoints in this case?

Problem 4. (6 + 14 points = 20 points)

Consider the table schema of our university database (see Appendix). Imagine the table "Class" were extended by an additional attribute "enrolled" which describes how many students haven taken the class (i.e. how many tuples with grades for this class are in the table Takes).

Your task:

a) (6 points)

How many triggers are necessary to keep this attribute always up to date? Points are only given for the explanation.

b) (12 points)

Write the SQL code of one of these triggers.

Problem 5. (3 + 3 + 14 points = 20 points)

Consider the following schedule of two transactions A and B:

$A_{read\ (y)}$ $B_{read\ (x)}$ $A_{read\ (z)}$ $A_{read\ (x)}$ $B_{write\ (x)}$ $B_{read\ (y)}$ $A_{read\ (x)}$ $A_{write\ (y)}$ $B_{write\ (y)}$ $A_{write\ (z)}$

a) (3 points)

Is the schedule serial? (No explanation required).

b) (3 points)

Is the schedule serializable? (No explanation required).

c) (12 points)

If the schedule is serializable, explain why that is so (approx. 4 sentences).

If the schedule is not serializable,

- name the types of problems that occur and mention which data items and transactions they concern
- and rearrange the operations to form a serializable, but not serial schedule of the transactions A and B.

Problem 6. (2 times 10 points = 20 points)

Consider two parallel transactions A and B in a system that uses the 2 phase locking protocol with record level locking. Transaction A uses isolation level "serializable"; B uses "read uncommited".
The data items X and Y are in different tables.
The following table shows the first actions of the transactions in chronological order from top to bottom.

	Transaction A (serializable)	Transaction B (read uncommitted)
1	**Start transaction**	
2		**Start transaction**
3		**read x = 800**
4	**read y = 2000**	
5		**write x:= 700**
6	**write y = 1000**	
7	**read x = 700**	
8		**read y**

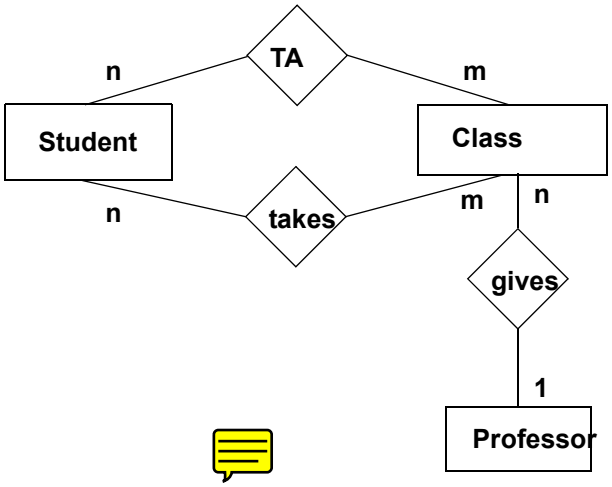
a) (10 points)
Explain the status of each transaction directly after line 8. (Approx. 3 sentences per transaction).

b) (10 points)
Assume that transaction B wants to commit as the next step. Briefly explain whether it can do that or not. What would be the consequence for the status of transaction A and why? (3 - 4 sentences).

Appendix:
Our universitydb database in MySQL, installed in the lab.

Consider the following ER model and relational model for a university database. For simplicity's sake, the attributes are omitted in the ERM.

Class			
classNr	**room**	**day**	**pName**
DTB-SS93	212	monday	Langes
BSY-SS93	114	monday	Müller
STP-WS92	212	tuesday	Müller
LIA-WS92	114	friday	Wagner



TA			
matNr	**classNr**	**hours**	**taSalary**
1000	LIA-WS92	40	1000
1234	LIA-WS92	35	1000
2345	DTB-SS93	80	2000
1000	STP-WS92	40	1000

Student	
matNr	**sName**
1234	Schmidt
2345	Schmidt
1000	Reinhard
4000	Wagner

Takes		
matNr	**classNr**	**grade**
1234	STP-WS92	1
1234	LIA-WS92	2
2345	STP-WS92	3
2345	LIA-WS92	4
1000	STP-WS92	3
1000	LIA-WS92	2
4000	STP-WS92	3
4000	LIA-WS92	1
1000	DTB-SS93	NULL
1000	BSY-SS93	NULL
1234	DTB-SS93	NULL

Professor	
pName.	**pSalary**
Langes	2000
Müller	3000
Wagner	2000
Schmidt	1000