UNIVERSITY OF TWENTE.

Written Exam

Advanced Database Systems
course code: 192110902
Wednesday 5 November 2014 (08:45 – 11:45), CR 3B
Mena Habib and Djoerd Hiemstra

Remarks:

- The exam consists of 6 questions.
- Motivate yours answers. The motivation / argumentation plays an important role in grading the assignments.
- You may not consult books or notes, but only one page of A4 size, double-sided. The page may contain text (typed or hand-written) and (possibly reduced) images (copied from the book, other sources or hand-made).
- For each assignment, the number of points is given. They add up to 90. You get 10 points for showing up at the exam. The grade for the exam is determined by dividing the number of points by 10.
- The final grade for the course is determined by 0.7 × 'Grade written exam' + 0.3 × 'Grade project'. Both grades need to be ≥ 5.5 and the MiniLab assignment has to be completed satisfactorily.

1 Using SQL in applications (15 points)

Below you find a fragment of example code that produces an RSS feed.

```
(…)
10   Connection db = DriverManager.getConnection(url, login, passw);
11   Query = "SELECT * FROM news WHERE source = ?";
12   PreparedStatement ps = db.prepareStatement(Query);
13   ps.setString(1, source);
14   ResultSet rs = ps.executeQuery();
15   System.out.println("<rss version=\"2.0\"><channel>");
16   System.out.println("<title>Atak Poppodium</title>");
17   System.out.println("<link>http://www.atak.nl/</link>");
18   System.out.println("<description>Informatie van... </description>");
19   while(rs.next()) {
20     System.out.println("<item>");
21     System.out.println("  <title>" + rs.getString("title") + "</title>");
22     System.out.println("  <link>http://www.atak.nl/pub/link.php?id=" +
23                        rs.getString("id") + "</link>");
24     System.out.println("  <description>" + rs.getString("description") +
25                       "</description>");
26     System.out.println("</item>");
27   }
28   System.out.println("</channel></rss>");
(…)
```

Answer the following questions:

(a) Is the programme code using a so-called “Statement-Level Interface” or a “Call-Level Interface” to include the SQL constructs? Explain your answer.

(b) Is this form of embedding SQL statements referred to as (answer with “yes” or “no”):

i. dynamic SQL
ii. static SQL
iii. embedded SQL
iv. SQL/PSM
v. a stored procedure

(c) Which line numbers contain the so-called “cursor”? In this case, the cursor is insensitive. What is an “insensitive cursor”?

(d) Suppose we remove line 13 and replace line 11 with

```
Query = "SELECT * FROM news WHERE source = '" + source + "'";
```

i. Would this work as well? Explain why/why not.

ii. Would the line introduce a security risk? If so, explain why.

2 Indexing (12 points)

Consider a relation *R* with attributes *A* and *B* with the following characteristics:

- 5,000 tuples with 10 tuples per page
- A 2-level B+ tree index on attribute *A* with up to 100 index entries per page
- Attribute *A* is a candidate key of *R*
- The values that the attribute *A* takes in relation *R* are uniformly distributed in the range 1 to 100,000.

Answer the following questions

(a) Assuming that the aforesaid index on *A* is unclustered, estimate the number of page fetches needed to compute the query $\sigma_{A>1000 \text{ AND } A<6000}(R)$. Explain your reasoning.

(b) What would be the cost if the above index were clustered? Again, explain your reasoning.

3 Query Processing (18 points)

Consider two relations *R* and *S*. *R* is stored in 2000 pages, 20 tuples per page. *S* is stored in 5000 pages, 5 tuples per page. The main memory available for processing the queries is 402 pages. There is a clustered B+-tree index on attribute *B* of the relation *S* with index depth 2. The weight of *B* in *S* is 1. Answer the following questions (ignore the costs of writing the results to disk).

(a) Calculate the cost of $R \bowtie_{A=B} S$ if we process it using a naïve *nested loop*, that is, using only two pages of main memory. Explain your answer.

(b) Answer the same question when using a *block-nested loop*.

(c) Answer the same question if an *index-nested loop* is used.

4 2-Phase Commit Protocol (15 points)

Suppose one single distributed transaction is running using the 2-Phase Commit protocol involving 3 cohorts A, B, and C under the supervision of coordinator M.
Consider the following possible cases:

(a) Cohort A crashes. After restarting, the recovery procedure found a prepare record but no commit or abort records in its log.

(b) Cohort C crashes. After restarting, the recovery procedure found a commit record in its log.

(c) Coordinator M crashes. After restarting, the recovery procedure found a commit record but not a complete record in its log.

For each of the aforementioned cases: Explain what are the possible timeouts that could have occurred? Explain what actions should be taken by the coordinator and/or the cohorts to maintain atomicity and durability?

5 Serializability (15 points)

Consider the following schedule:

r2(y) w2(x) r1(x) r3(x) w1(y) c1 w2(y) c2 w3(y) c3

(a) Draw the serialization graph of this schedule. Is it serializable or not? Why?

(b) In what order are the operations of the schedule executed if it is fed to a pessimistic concurrency controller with immediate update approach? Explain in terms of operation by operation.

(c) Considering an optimistic concurrency controller with deferred update approach, which transactions will commit and which will abort? Why?

6 Recovery (15 points)

Suppose a crash took place on a server containing a database that uses no-force commit policy with pessimistic immediate update concurrency controller. After restart the recovery procedure found the following situation:

Log:

	13	14	15	16	17	18	19	20	21	22	23	24
	U	B	U	CK	C	U	B	U	CL	A	U	C
…	T2	T3	T3	T1,	T1	T2	T4	T4	T3	T3	T2	T2
	X		Y	T2,		Z		W	Y		X	
	1, 2		‘A’,‘B’	T3		12, 13		4, 5	‘B’,‘A’		2,3	

Where each record contains LSN, Transaction type (U=Update, B=Begin Transaction, CK= Sharp Checkpoints, C=Commit, A=Abort, CL= Compensating Log Record), Transaction ID, Item ID, Before and After Images.

DB Pages:

Page 50	Page 51
LSN:20	LSN:15
w=5	Y=‘B’
X=2	
z=13	

(a) The recovery protocol reconstructs a consistent database state. Describe in details the three phases of the recovery protocol and show the database pages after each phase.

(b) The no force commit policy may cause two problems: (i) database pages containing updates of uncommitted transaction, (ii) database pages do not contain the updates of committed transactions. Which of the two problems take place in the aforementioned case?

(c) What is the role of the Compensating Log Record. What would happen if it was not inserted in the log?