Umeå University
Department of Computing Science
5DV052 — Advanced Data Models and Systems
Examination: June 01, 2011

Name (printed): ______________________________

Swedish ID number: ______________________________

Computer User-ID: ______________________________

Signature: ______________________________

Secret code number: ______________________________

Instructions: / Instruktioner:

This examination will be graded anonymously. This page will be removed before the instructor receives the examination for grading. The secret code number given above must therefore be written on every answer page which you turn in to the examination proctor.

Denna skrivning rättas kodad. Detta blad kommer att avskiljas innan läraren får skrivningen för rättning. Ovanstående kod måste därför finnas på samtliga svarsblad när du lämnar skrivningen till skrivvakten.

To the proctor of the examination: / Till skrivningsbevakaren:

Detach this cover sheet from the examination and put it in the envelope which is addressed to Yvonne Löwstedt, Department of Computing Science.

Avskilj detta försättsblad och stoppa i kuvert som skickas till Yvonne Löwstedt, Datavetenskap.

Umeå University
Department of Computing Science
5DV052 — Advanced Data Models and Systems
Examination: June 01, 2011

Secret code number: ____

1. All answers must be written in English.
2. A pocket engineering calculator (miniräknare) and English/X - X/English dictionary may be used. No other help materials are allowed.
3. All answers must be written on the printed sheets containing the questions. Do not append any additional answer sheets. Write the secret code assigned to your examination on the top of each page.
4. You must turn in this cover sheet and all answer sheets.
5. The examination has a total of 600 points, which constitute 60% of the course grade. The other 40% is based upon the obligatory exercises, as explained in the course syllabus. In order to pass the course, you must obtain at least 300 points on the examination, regardless of how many points you earn on the obligatory exercises. You must also obtain a total of no less than 500 points combined on the examination and the obligatory exercises.
6. For problems 2–6, you have the choice, for each part, to do the problem, or to skip it for partial credit. In the table below, place an X in the position for any problem for which you have attempted a solution, and which you wish to have graded. It is extremely important that you fill in this table properly, because of the following option. For any box which is left blank, the associated question will not be graded, and you will instead be awarded 10% of the points for that question. Your decision to leave a box blank is definitive, so be very careful. For example, If you leave box 4(b) blank, your answer to that question will not be graded, even if it is completely correct. On the other hand, if you place an X in box 4(b), but provide no answer whatsoever to that question, you will not receive 10% of the points for that question. It is strongly recommended that you use a pencil, in case you change your mind!

   This option does not apply to problem 1.

Prob	1	2	3	4	5	6
(a)						
(b)						
(c)						
(d)						
(e)						

Secret code number: ______________________________

(1. 100 points total; 5 points for each question) For each of the following statements, circle the **T** is the assertion is true, and the **F** if it is false. The point total will be computed as $5 \times C - 2 \times I$ with C = number of correct answers and I = number of incorrect answers; however, the score may never be less than zero. A blank answer is neither in C nor in I.

T **F** (a) If a scheduler enforces the read-committed isolation level, then every schedule is necessarily recoverable.

T **F** (b) In most real database systems, the isolation level which is labelled `serializable` is actually *snapshot isolation*.

T **F** (c) Every view-serializable schedule is also conflict serializable.

T **F** (d) The two-phase-locking protocol (2PL) guarantees recoverable schedules.

T **F** (e) In the authorization mechanism of standard SQL, a user loses a privilege whenever any user who granted that privilege revokes it.

T **F** (f) In the absence of blind writes, view serializability and conflict serializability are equivalent concepts.

T **F** (g) In the context of multigranular locking, before a transaction T_i may obtain a write lock on an object x, it is sufficient to obtain an intention-to-write lock on every data object x' below and including x in the hierarchy. (Below = more specialized in the hierarchy.)

T **F** (h) In the no-force strategy, cache pages which are not yet committed may be written to the stable database.

T **F** (i) A practical way to deal with transaction aborts in the logging process is to represent them as undo operations using compensation records.

T **F** (j) In a database cache, the pin bit is set whenever the page needs to be written to the stable database.

T **F** (k) In general, row stores are more suitable to OLAP than column stores.

T **F** (l) In the context of database management, OLTP stands for Ongoing Local Termination Process.

T **F** (m) Google's Map-Reduce runs over BigTable.

T **F** (n) Linestrings, Polygons and Circular Arcs are types of raster data in GIS.

T **F** (o) DTDs are written in XML.

T **F** (p) The original NoSQL system (Strossi, 1998) worked only over UNIX systems.

T **F** (q) PostGIS uses GIST indexes.

T **F** (r) `GROUP BY CUBE(x,y,z)` precomputes all rollups over {x}, {y}, {z}, {x, y}, {x, z}, {y, z}, and {x, y, z}.

T **F** (s) Facebook originally used Casandra.

T **F** (t) The "property as table" approach as applied to RDF (or semantic networks) leads to relations of three attributes.

Secret code number: ______________________________

(2. 100 points total; 25 points for each part) The four critical properties of a transaction are described by the acronym *ACID*. In the spaces below, identify the word which each of these four letter stands for, and describe briefly the meaning of the associated concept.

A______________________

C______________________

I______________________

D______________________

Secret code number: ____________________

(3. 100 points) A database contains four data objects, A, B, C, and D. Six transactions are given on these objects, specified as follows.

T_1	T_2	T_3	T_4	T_5	T_6
Read(A)	Read(C)	Read(A)	Read(D)	Read(C)	Read(B)
Write(A)	Write(C)	Read(C)	Write(D)	Write(C)	Read(C)
Read(B)	Read(D)	Write(C)		Read(B)	Write(B)
Write(B)	Write(D)	Write(A)		Write(B)	
Read(C)					
Write(C)					

Shown to the right is a schedule for this set of transactions, including a marked point at which a crash occurs. Using this schedule, answer the questions given on the following two pages. The following points are relevant:

- In the simple model here, only the pure immediate-update protocol and the pure deferred-update protocol will be considered. There is no DBMS cache, and so no need for checkpoints.
- Assume that all active transactions abort at the time of a crash, but that the log is preserved without damage. Regarding the state of the database itself, there are two scenarios. In the first (parts (a) and (b)), the state of the database is restored to its value before these transactions began, and recovery begins from that point. In the second (parts (c) and (d)), the database state remains exactly as it was at the point of the crash, and recovery must proceed from that point.
- The choice "No further action is necessary" (parts (c) and (d) only) is appropriate when the transaction had started before the time of the crash, but no special action on it is necessary for crash recovery. If this choice of no further action is appropriate, it is to be preferred to all other choices, even though others may also produce correct results.
- To *re-do* a transaction means to extract the write operations which the transaction executed from the log and apply them to the database. To *re-run* a transaction means to re-execute its associated programs and to update the database accordingly. When possible, a re-do of a transaction from log entries is always to be preferred to a re-run. Thus, when both will result in a correct recovery, the correct answer is "Re-do from log entries".
- There is no credit for showing work. The grade on this problem is determined entirely by which boxes are checked on the following two pages.

T_1:	Begin
T_1:	Read(A)
T_1:	Write(A)
T_2:	Begin
T_2:	Read(C)
T_3:	Begin
T_3:	Read(A)
T_2:	Write(C)
T_2:	Read(D)
T_2:	Write(D)
T_2:	Commit
T_3:	Read(C)
T_3:	Write(C)
T_1:	Read(B)
T_1:	Write(B)
T_4:	Begin
T_4:	Read(D)
T_5:	Begin
T_5:	Read(C)
T_5:	Write(C)
T_1:	Read(C)
T_1:	Write(C)
T_1:	Commit
T_4:	Write(D)
T_4:	Commit
T_6:	Begin
T_6:	Read(B)
T_6:	Read(C)
T_6:	Write(B)
T_6:	Commit
T_5:	Read(B)
T_5:	Write(B)
T_5:	Commit
	Crash
T_3:	Write(A)
T_3:	Commit

Secret code number: ______________________________

In answering parts (a) and (b) below, use exactly schedule shown on page 3, assuming that at the point of the crash, all active processes abort and the database state is restored to the value which it had just before T_1 began.

(a: 25 points) To recover from the system crash in the context of the *pure immediate-update protocol*, for each transaction, indicate which of the following actions is applicable by placing an × in the appropriate ☐, as so ☒. For each transaction, only one choice is correct.

T_1: ☐ Re-do from log entries.
☐ Re-run; no rollback necessary.

T_2: ☐ Re-do from log entries.
☐ Re-run; no rollback necessary.

T_3: ☐ Re-do from log entries.
☐ Re-run; no rollback necessary.

T_4: ☐ Re-do from log entries.
☐ Re-run; no rollback necessary.

T_5: ☐ Re-do from log entries.
☐ Re-run; no rollback necessary.

T_6: ☐ Re-do from log entries.
☐ Re-run; no rollback necessary.

(b: 25 points) To recover from the system crash in the context of the *pure deferred-update protocol*, for each transaction, indicate which of the following actions is applicable by placing an × in the appropriate ☐, as so ☒. For each transaction, only one choice is correct.

T_1: ☐ Re-do from log entries.
☐ Re-run; no rollback necessary.

T_2: ☐ Re-do from log entries.
☐ Re-run; no rollback necessary.

T_3: ☐ Re-do from log entries.
☐ Re-run; no rollback necessary.

T_4: ☐ Re-do from log entries.
☐ Re-run; no rollback necessary.

T_5: ☐ Re-do from log entries.
☐ Re-run; no rollback necessary.

T_6: ☐ Re-do from log entries.
☐ Re-run; no rollback necessary.

Secret code number: ____________________

In answering parts (c) and (d) below, use exactly schedule shown on page 3, assuming that at the point of the crash, all active processes abort but that the database state remains as it was at the time of the crash.

(c: 25 points) To recover from the system crash in the context of the *pure immediate-update protocol*, for each transaction, indicate which of the following actions is applicable by placing an × in the appropriate ☐, as so ☒. For each transaction, only one choice is correct.

T_1: ☐ No further action is necessary. ☐ Re-do from log entries.
☐ Re-run; no rollback necessary. ☐ Re-run; rollback necessary.

T_2: ☐ No further action is necessary. ☐ Re-do from log entries.
☐ Re-run; no rollback necessary. ☐ Re-run; rollback necessary.

T_3: ☐ No further action is necessary. ☐ Re-do from log entries.
☐ Re-run; no rollback necessary. ☐ Re-run; rollback necessary.

T_4: ☐ No further action is necessary. ☐ Re-do from log entries.
☐ Re-run; no rollback necessary. ☐ Re-run; rollback necessary.

T_5: ☐ No further action is necessary. ☐ Re-do from log entries.
☐ Re-run; no rollback necessary. ☐ Re-run; rollback necessary.

T_6: ☐ No further action is necessary. ☐ Re-do from log entries.
☐ Re-run; no rollback necessary. ☐ Re-run; rollback necessary.

(d: 25 points) To recover from the system crash in the context of the *pure deferred-update protocol*, for each transaction, indicate which of the following actions is applicable by placing an × in the appropriate ☐, as so ☒. For each transaction, only one choice is correct.

T_1: ☐ No further action is necessary. ☐ Re-do from log entries.
☐ Re-run; no rollback necessary. ☐ Re-run; rollback necessary.

T_2: ☐ No further action is necessary. ☐ Re-do from log entries.
☐ Re-run; no rollback necessary. ☐ Re-run; rollback necessary.

T_3: ☐ No further action is necessary. ☐ Re-do from log entries.
☐ Re-run; no rollback necessary. ☐ Re-run; rollback necessary.

T_4: ☐ No further action is necessary. ☐ Re-do from log entries.
☐ Re-run; no rollback necessary. ☐ Re-run; rollback necessary.

T_5: ☐ No further action is necessary. ☐ Re-do from log entries.
☐ Re-run; no rollback necessary. ☐ Re-run; rollback necessary.

T_6: ☐ No further action is necessary. ☐ Re-do from log entries.
☐ Re-run; no rollback necessary. ☐ Re-run; rollback necessary.

Secret code number: ______________________________

(4. 100 points total; 20 points for each part) Provide <u>concise</u> but correct answers to the following questions regarding database transactions.

(a) In snapshot-isolation, conflicts are typically resolved with a protocol with the acronym *FCW*. Explain what this acronym stands for and describe briefly how the underlying process of conflict resolution works.

(b) Every database recovery algorithm must satisfy the *write-head recovery protocol*. Explain clearly what this protocol is.

Secret code number: ______________________________

(c) Every database recovery algorithm must satisfy the *commit rule*. Describe this rule clearly.

(d) Explain what a *database checkpoint* is and state the steps which are taken when a checkpoint is executed. (You need describe only basic checkpointing not, fuzzy checkpointing.)

Secret code number: ____________________

(e) Explain briefly how *fuzzy checkpointing* differs from ordinary checkpointing as you have described above.

Secret code number: ____

(5. 100 points total) Consider the data cube `Sales(Item, Store, Date)` in which a given cube cell contains, for the associated item, the number sold at a given store on a given date. Each data field may be aggregated as follows.

- `Items` (*e.g.*, Hamburger) may be grouped by `Type` (*e.g.*, Meat), which in turn may be grouped by `Category` (*e.g.*, Food).
- Each `Store` is represented by an ID (*e.g.*, Store_{100}) and is located in a `City` (*e.g.*, Stockholm), which is located in a `Country` (*e.g.*, Sweden), which is located in a `Region` (*e.g.*, Scandinavia), which is located on a `Continent` (*e.g.*, Europe). For simplicity, assume that the name of a city identifies it uniquely; *i.e.*, no two distinct cities have the same name.
- `Dates` may be grouped into `Weeks` (*e.g.*, Week_{35}), which in turn may be grouped into `Quarters` (*e.g.*, Quarter_3). For simplicity, assume that all dates are for a fixed calendar year.

There are 1000 `Stores`, 20000 `Items` and 365 `Dates`.

(a: 10 points) Give the number of cells in the data cube. ______________.

(b: 10 points) Give the number of cells in the data cube assuming that all cross tabs (rollups) are precomputed in the cube.

______________.

(c: 25 points) Describe the `PIVOT`, `ROLL-UP`, and `SLICE/DICE` operations necessary to generate a single-dimensional vector of the numbers of items of type toothpaste that are sold in all Swedish stores for each week of the year. (Give a simple but clear characterization; do *not* use SQL here.)

Secret code number: ______________________________

(d: 30 points) Define a ROLAP snowflake schema (hint: snowflake schema implies 3NF) for the above data cube and the given aggregation hierarcies.

Secret code number: ____________________

(e: 25 points) Give the SQL query over the schema in part (d) which is necessary to obtain the answer in part (c). (Use standard SQL; do *not* use the special operation such as `PIVOT`, GROUP BY CUBE, and GROUP BY ROLLUP which are presented in the textbook.)

Secret code number: ______________________________

(6. 100 points total) Answer the following questions.

(a: 25 points) Explain why compression is particularly effective in column stores.

(b: 25 points) Explain what horizontal scalability is in the context of distributed database systems.

Secret code number: ______________________________

(c: 50 points) List at least three criticisms that NoSQL advocates lodge against relational databases.