Umeå University
Department of Computing Science
TDBC86 — Principles of Database Systems
Examination: January 09, 2006

Name : _____

Personnummer (Swedish ID number): _____

UNIX user ID: _____

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. The examination consists of two parts. In Part I, which includes Problem 1, on pages 1–2, the answers **must** be written on the sheets provided. In general, only the answers will be checked for these questions; derivations need not be provided and will not be evaluated. In Part II, which includes Problems 2–10, the answers must be written on separate sheets.
4. You must turn in this cover sheet, the answer sheets for Part I (pages 1–2), and the pages containing your answers for Problems 2–10. Collate the sheets in numerical order of the problems, and write on only one side of the paper. Write your name and Swedish ID number on the top of each sheet. For questions 2–10, write the question number on the top of each sheet as well.
5. The examination has a total of 800 points, which constitute 80% of the course grade. The other 20% is based upon the obligatory exercises, as explained in the course syllabus.
6. For Problems 2–10, 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 8(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 8(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! *Note that Problem 1 does not carry this option. No points are awarded for skipping it.*

Prob	2	3	4	5	6	7	8	9	10
(a)									
(b)	■								
(c)	■			■					
(d)	■			■		■	■	■	
(e)	■		■	■		■	■	■	
(f)	■	■	■	■		■	■	■	■

Name and ID number: _____

(1: 120 points total; 10 points for each part) Important: The answers to this question must be indicated below; do not write the answers to this question on a blank sheet.

The following relational schema with relation $R[ABCDEFG]$ and functional dependencies $\mathcal{F} = \{A \rightarrow CDEF,\ D \rightarrow AF,\ E \rightarrow FG,\ BG \rightarrow CD\}$ applies to all twelve parts of this problem.

Given below are eight projections of $R[ABCDEFG]$. Assume that each such projection is constrained by exactly the functional dependencies in $\mathcal{F}^+$ which embed in the projection. In the space provided, give all of the candidate keys for the projection, as well as a minimal cover (basis) of the functional dependencies from $\mathcal{F}^+$ which embed in it. Also, circle *all* of the normal-form properties which the projection possesses (even if some are implied by others). Circle the word "None" just in case none of the normal-form properties holds.

(a) Projection: $R[ABCDEFG]$

Candidate keys: _____

Minimal FD cover (basis): _____

Normal forms: None 2NF 3NF BCNF

(b) Projection: $R[ACDEFG]$

Candidate keys: _____

Minimal FD cover (basis): _____

Normal forms: None 2NF 3NF BCNF

(c) Projection: $R[ACD]$

Candidate keys: _____

Minimal FD cover (basis): _____

Normal forms: None 2NF 3NF BCNF

(d) Projection: $R[AEF]$

Candidate keys: _____

Minimal FD cover (basis): _____

Normal forms: None 2NF 3NF BCNF

(e) Projection: $R[BDG]$

Candidate keys: _____

Minimal FD cover (basis): _____

Normal forms: None 2NF 3NF BCNF

Name and ID number: _____

(f) Projection: $R[BFG]$

Candidate keys: _____

Minimal FD cover (basis): _____

Normal forms: None 2NF 3NF BCNF

(g) Projection: $R[BEFG]$

Candidate keys: _____

Minimal FD cover (basis): _____

Normal forms: None 2NF 3NF BCNF

(h) Projection: $R[BCEG]$

Candidate keys: _____

Minimal FD cover (basis): _____

Normal forms: None 2NF 3NF BCNF

Shown below are four decompositions of the schema on the previous page. For each decomposition, indicate whether or not it has the properties of *lossless* and *dependency preserving* by circling the appropriate word(s). Circle the word *neither* just in case it is neither lossless nor dependency preserving.

In checking the lossless property, use only pairwise reconstructions (*i.e.*, join trees). Do not test for cyclic decompositions (*i.e.*, decompositions which are lossless, but the join can not be reconstructed pairwise) [It was not explained in the course how to test for such decompositions, so if you do not understand what this means, just use the techniques which were presented in the course.]

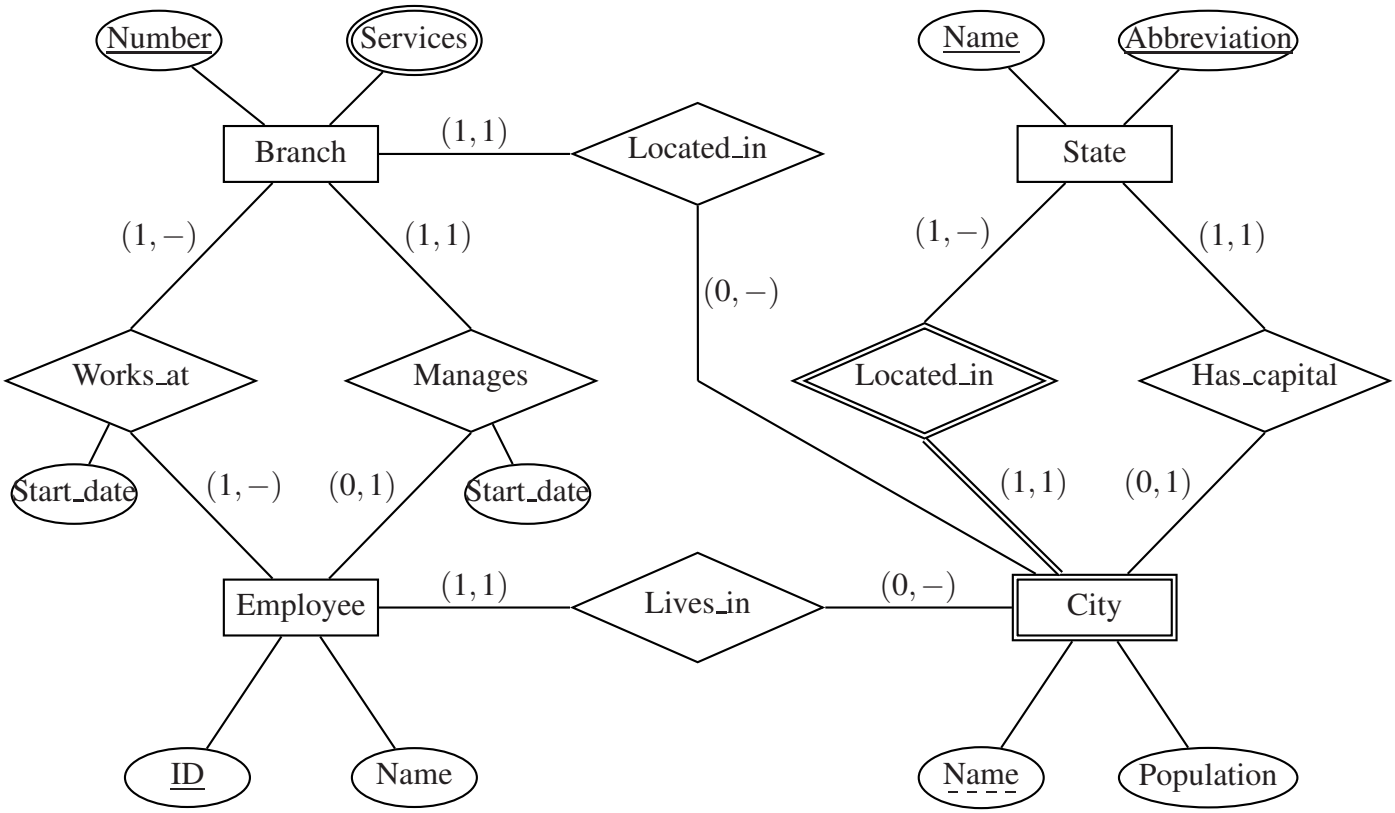
(i) Decomposition: $\{R[ACD], R[AEF]\ R[BDG]\}$.
Neither Lossless Dependency preserving

(j) Decomposition: $\{R[ACD], R[AEF], R[BEFG]\}$.
Neither Lossless Dependency preserving

(k) Decomposition: $\{R[ACD], R[AEF]\ R[BDG], R[BCEG]\}$.
Neither Lossless Dependency preserving

(l) Decomposition: $\{R[ACDEFG], R[BFG]\}$.
Neither Lossless Dependency preserving

(2: 80 points) Shown below is an ER-diagram for a banking database. Using the techniques developed in the course, map this diagram into an equivalent relational schema. Show all keys, primary and foreign, and link foreign keys to their primary partners.



Important: If a primary and/or foreign key consists of more than one attribute, make sure that your notation identifies and links these composite keys as groups.

Notes:

- The (x, y) notation gives the minimum and maximum number of times that a given instance of the entity may participate in the relationship. Thus, $(1, 1)$ means exactly one, and $(0, -)$ means any number.
- An employee may work at several branches in this model. This is not a mistake.
- A *state* is an administrative unit (*delstat* in Swedish) roughly corresponding to *län* in Sweden.

(3: 80 points total; 16 points for each part) Answer the following questions regarding the manner in which the database recovery manager handles read and write commands from transactions.

(a) In the context of *pure immediate update*, specify precisely which entries must be made in the database system log for:

(i) a read of data item x by transaction T.

(ii) a write of data item x by transaction T.

(b) In the context of *pure deferred update*, specify precisely which entries must be made in the database system log for:

(i) a read of data item x by transaction T.

(ii) a write of data item x by transaction T.

(c) In the context of pure immediate update, define precisely (in terms of the reads, writes, and commit times of the transactions) what is meant by a *recoverable schedule*.

(d) In the context of pure immediate update, define precisely (in terms of the reads, writes, and commit times of the transactions) the necessary and sufficient conditions which will ensure that a schedule is *cascadeless*; that is, which will ensure that cascading rollback will never be necessary, regardless of when a transaction aborts.

(e) In the context of recovery from a system crash, explain briefly the difference between *re-do* and *re-run* of a transaction.

(4: 80 points total; 20 points for each part) Answer the following questions concerning the *DBMS buffer* (also called the *DBMS cache*).

(a) In the context of writing the buffer to the permanent database, explain the difference between a *steal strategy* and a *no-steal strategy*.

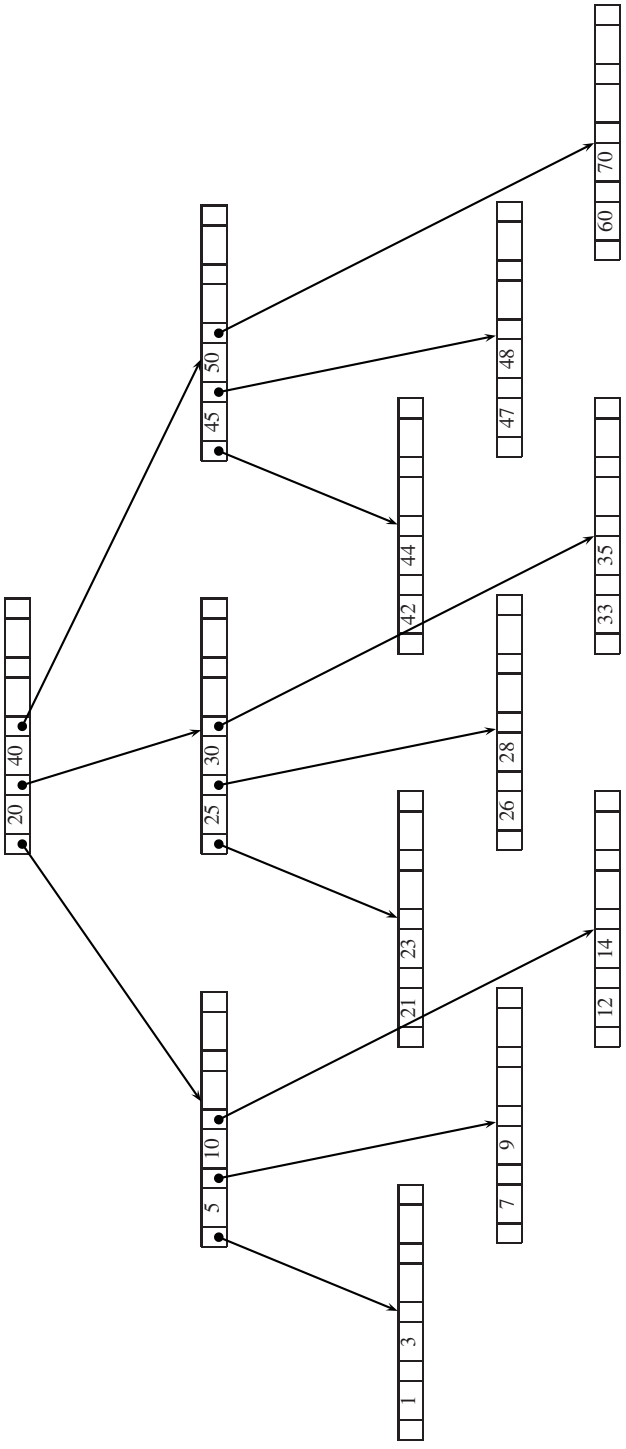
(b) In the context of writing the buffer to the permanent database, explain the difference between a *force strategy* and a *no-force strategy*.

(c) Explain briefly how the process of *checkpointing* is affected by the use of a steal strategy, as opposed to the use of a no-steal strategy.

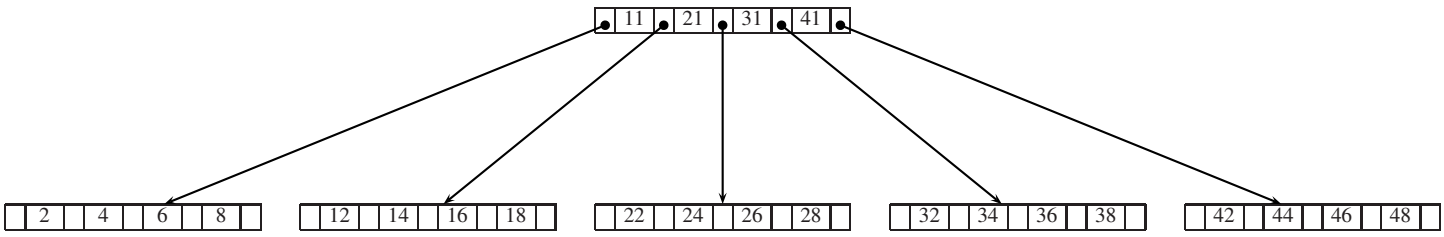
(d) Explain briefly how the process of *checkpointing* is affected by the use of a no-force strategy, as opposed to the use of a force strategy.

(5: 80 points; 40 points for each part)

(a) Shown below is a B-tree of order four, with keys taken from the natural numbers under the standard ordering. Give the B-tree which results from deletion of the record with key 70. To obtain credit, the tree which results from application of the standard deletion algorithm must be provided; it is not correct simply to provide an arbitrary B-tree with the correct key values.



(b) Shown below is a B-tree of order four, with keys taken from the natural numbers under the standard ordering. Give the B-tree which results from insertion of the record with key 10. To obtain credit, the tree which results from application of the standard deletion algorithm must be provided; it is not correct simply to provide an arbitrary B-tree with the correct key values.



(6: 80 points total) The following parameters apply to a relation which is to be stored in a B^+-tree. Assume that each node, interior or leaf, is stored in one page, and that each tuple is stored as one record.

Number of tuples in the relation	1×10^{10}
Size of one tuple	120 bytes
Size of pointer to a B^+-tree node	4 bytes
Size of internal B^+-tree key	24 bytes
Page size	2048 bytes
Sequential-access pointers in the leaf nodes	8 bytes total

Answer the following. To receive credit, you must show your work, including the formulas used. (If you show just numerical computations, without an underlying formula, full credit will not be awarded.)

(a: 10 points) Compute the maximum number of indices per non-leaf node.

(b: 10 points) Compute the maximum number of records per leaf node.

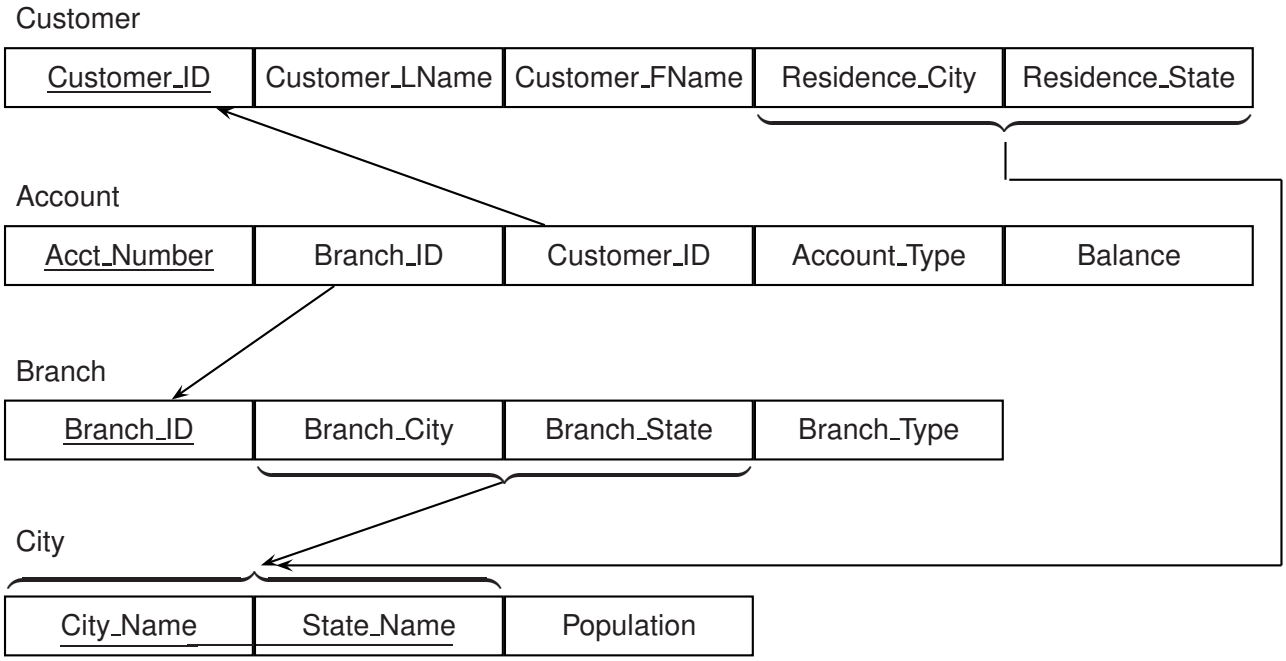
(c: 10 points) Compute the maximum depth that such a B^+-tree may have. (The depth is defined as the length of a path from the root to a leaf.)

(d: 10 points) Compute the minimum depth that such a B^+-tree may have.

(e: 20 points) For a B^+-tree of the minimum depth computed in (d), compute the average number of indices in the interior nodes other than the root, assuming that the root contains the maximum possible number of indices, and that the leaf nodes contain the maximum possible number of records. State whether or not this value is consistent with the constraints on the minimum and maximum number of indices in an interior node of a B^+-tree. In addition, if this value is consistent, compute the number of interior nodes and the number of leaf nodes, as well as the average number of records in each leaf node.

(f: 20 points) Repeat (e), this time assuming that the leaf nodes contain the minimum number of records possible. All other parameters remain the same as specified in (e).

The following schema applies to Problems 7 through 10. In each case, the primary key is underlined, and the arrows run from foreign keys to their primary counterparts.



The following queries apply to Problems 7 through 9, and reference the schema defined above.

(a) Find the names (last and first) of those customers who have accounts in both the state with State_Name VT and the state with State_Name NH.

(b) Find the names (last and first) of those customers who have an account in every branch which is located in a city with City_Name Burlington.

(c) Find the names (last and first) of those customers who do have exactly two accounts in branches which are located in the state with State_Name NH.

Note: In the above schema, the attributes Customer_LName and Customer_FName identify the last and first names of the customer, respectively.

(7: 60 points total; 20 for each part) Solve each of the three queries (a) - (c) in the relational algebra. Functional operators, such as count and average, may not be used. Use the join notation described in the course notes: (The symbol $\bowtie$ is used to denote join; in the absence of subscripts it denotes the natural join; any other join conditions must be specified explicitly using subscripts.)

(8: 60 points total; 20 for each part) Solve each of the three queries (a) - (c) in the tuple relational calculus. Functional operators, such as count and average, may not be used.

(9: 60 points total; 20 for each part) Solve each of the three queries (a) - (c) using SQL. To keep the solutions simple and consistent, the following restrictions apply:

(i) Directives involving the keyword JOIN may not be used. Express the join conditions directly using conditions in the WHERE clause.

(ii) The solution must consist of a single SQL directive. Creation and subsequent use of temporary tables is not permitted.

(iii) Subqueries involving SELECT are allowed only in the WHERE and HAVING fields of queries and subqueries, and in the SET fields of UPDATE commands.

(iv) Aggregate operators, such as COUNT, SUM, AVG, MAX, and MIN may **not** be used.

(10: 100 points total; 20 points for each part) Provide solutions, in SQL, to the following additional queries. Restrictions (i)-(iii) from Problem 7 apply here as well. On the other hand, aggregate operations may be used, but only in the SELECT and HAVING clauses of queries and subqueries.

(a) For each Branch_ID in the Branch relation, list both the total number of accounts held at that branch, as well as the number of distinct account types held at that branch. All Branch_IDs must be listed in the result, even those which house no accounts at all.

(b) For each (City_Name,State_Name) pair in the City relation, list the average balance of all accounts which are held by customers who live in (*i.e.*, have residence at) that (City_Name,State_Name) pair. If there are no accounts associated with a given (City_Name,State_Name) pair, list the average as zero.

(c) Define a *home account* of a customer to be one which is held in a branch which is located in the residence city and state of that customer, and the *home balance* of a customer to be the sum of all balances of the home accounts of that customer. Display the name and customer ID of that customer who has the greatest home balance. In case of a tie, display all such customers.

(d) Move all checking accounts which are housed in a branch located in the (City_Name,State_Name) pair (Burlington,VT) to the branch whose ID is 1234.

(e) Delete all customers who do not have any accounts from the Customer relation.

Appendix

A useful formula for uniform (m,r,d) B-trees:

$$R(m,r,d) = (m+1) \times (r+1)^d - 1$$

with

$$\begin{aligned} m &= \text{the total number of indices at the root node.} \\ r &= \text{the total number of records in each non-root node.} \\ d &= \text{the depth of the tree; } \textit{i.e.}, \\ &\quad \text{the length of a path from the root to a(ny) leaf node.} \\ R(m,r,d) &= \text{the total number of records in the tree.} \end{aligned}$$

A useful formula for uniform (m,q,r,d) B^+-trees:

$$R(m,q,r,d) = (m+1) \times (q+1)^{d-1} \times r$$

with

$$\begin{aligned} m &= \text{the total number of indices at the root node.} \\ q &= \text{the total number of indices in each non-root, non-leaf node.} \\ r &= \text{the total number of records in each leaf node.} \\ d &= \text{the depth of the tree; } \textit{i.e.}, \\ &\quad \text{the length of a path from the root to a(ny) leaf node.} \\ R(m,q,r,d) &= \text{the total number of records in the tree.} \end{aligned}$$