Umeå University
Department of Computing Science
TDBC86 – Database Concepts
Examination: December 20, 1999

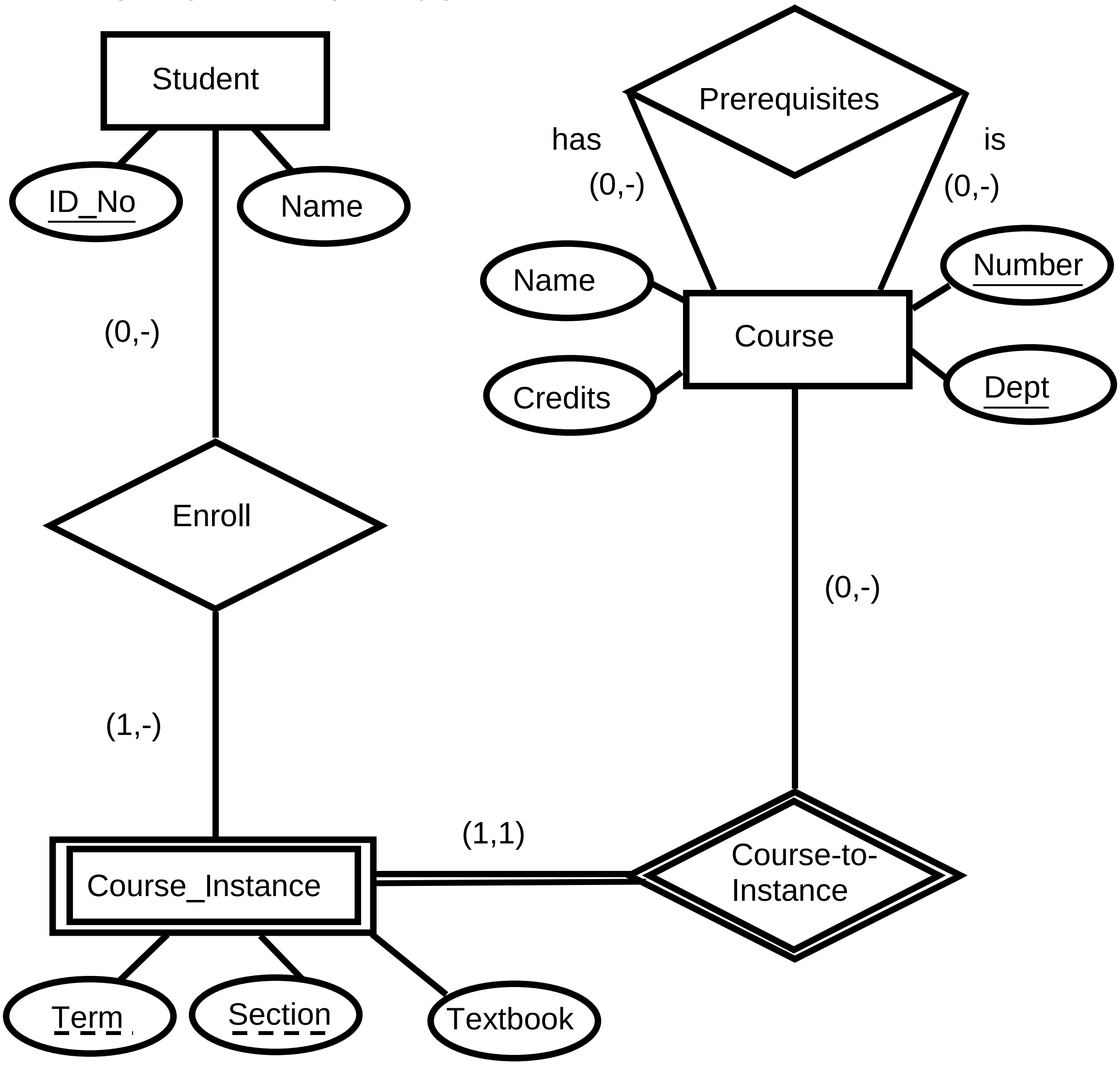
Name and ID number: ________________________________

Signature:____

1. All answers must be written in English.
2. A pocket engineering calculator and English/X – X/English dictionary may be used. No other help materials are allowed.
3. Solutions to all problems, except Problem 7, must be written on the special examination paper, which is provided. Solutions to problem 7 must be written on the special answer sheet, which is included as page 11 of htis examination. Write your name, ID number, and the problem number on each solution sheet. Collate the sheets in numerical order of the problems. Please write on only one side of the paper.
4. The examination has a total of 800 points, which constitutes 80% of the course grade. The other 20% is based upon the obligatory exercises, as explained in the course syllabus.
5. 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 20% 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 3(b) blank, your answer to that question will not be graded, even if it is completely correct. Similarly, if you place an X in box 3(b), but provide no answer whatsoever to that question, you will not receive 20% of the points for that question. It is strongly recommended that you use a pencil, in case you change your mind!

Prob.	1	2	3	4	5	6	7	8	9	10	11	12	13
(a)													
(b)													
(c)													
(d)													
(e)													
(f)													

1. (60 points) Shown below is an ER-diagram for a university course 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:

- Prerequisites = förkunskapskrav; Credits = poäng; Dept = department.
- A course may have several offerings (instances) during a term. A *section* is a number identifying a particular offering of a course during a given term.
- The (x,y) notation gives the minimum and maximum number of participants in a relationship. Thus, (1,1) means exactly one, and (0,-) means any number.

The following depicts a simplified database, based upon the Company example from the textbook. It, together with the following five queries, are the common context in which Problems 2, 3, and 4, below, should be solved.

Employee

SSN	Name	Salary	Home_Dept_No

Department

Dept_No	Dept_Name	Manager_SSN

Project

Project_Name	Project_Location	Controlling_Dept_No

Works_On

Employee_SSN	Project_Name	Hours_per_Week

Note: “No” is an abbreviation for “Number.”

(a) Find the names of those employees who work on either the project with name Alpha or the project with name Beta, or both.

(b) Find the names of those employees who work on both the project with name Alpha and the project with name Beta.

(c) Find the names of those employees who work on more than one project.

(d) Find the names of those employees who work on all projects which are controlled by the Research department.

(e) Find the names of those employees who work on some project which is controlled by their home department.

2. (60 points total; 12 for each part) Solve each of the five queries (a) – (e) in the relational algebra. Functional operators, such as count and average, may not be used.

3. (60 points total; 12 for each part) Solve each of the five queries (a) – (e) in the tuple relational calculus. Functional operators, such as count and average, may not be used.

4. (60 points total; 12 for each part) Solve each of the five queries (a) – (e) using SQL. For this part only, you may use SQL functional operations, such as count and average.

5. (60 points total) Using the database schema of Problems 2-4, provide solutions, in SQL, to the following additional queries. All SQL operations may be used, including aggregation operations.

(a: 15 points) For each department, find the minimum, maximum, and average salary of all employees with that department as home. The result must list both the department name and number, as well as the three salary values.

(b: 15 points) For each employee, find the number of distinct projects which that individual works on. The result must list the SSN and name of the employee, as well as the number of projects.

(c: 15 points) For each project, find the highest salary of all employees who work on that project. The result must list the project name, employee SSN and name, and salary. (Note: There may be several employees with the same salary. List information for all relevant employees.)

(d: 15 points) Find the set of all employees whose salary is higher than the average salary over all employees who have the same home department. The result must list the SSN and name of those employees satisfying the query.

6. (60 points total) Consider the relational schema **R** with relation R(A,B,C,D,E,F) and functional dependencies {$A \rightarrow B$, $B \rightarrow C$, $DE \rightarrow F$, $F \rightarrow D$}.

(a: 20 points) List all of the candidate keys of R.

(b: 20 points) List all of the superkeys of R.

(c: 0 points) Designate one of your candidate keys as the primary key. List it here. (You must do this to answer part (d), even though this question by itself carries no points.)

(d: 20 points) Suppose now that **R** contains a second relation S(G,H,I,J,K,L). Identify precisely the conditions which make a subset of {G,H,I,J,K,L} a foreign key for R. Illustrate with a concrete example, using the data identified in (a)-(c) above, and a suitably selected subset of the attributes of S.

7. (70 points total) **Important: The answers to this question must be indicated on page 11 of this examination, which is to be turned in with your solutions. Do not write the answers to this question on a blank sheet.**

Consider again the relational schema **R** with relation R(A,B,C,D,E,F) and functional dependencies $\{A \rightarrow B, B \rightarrow C, DE \rightarrow F, F \rightarrow D\}$.

Shown below are seven decompositions of **R**, together with a list of six properties. For each decomposition, identify those properties which the decompositions possesses. Do this on page 11 by circling (drawing a ring around) the names of the properties which that decomposition has. Circle the word *none* just in the case that none of the other five properties applies. (Thus, it is never correct to circle nothing at all, and it is never correct to circle the word *none* together with something else.)

For this problem, you must either choose to attempt all seven parts, or else none at all. Grading is such that random guessing is unlikely to produce more than 20% of the points. Each part is worth 10 points, and will be graded independently of the others.

Explanations of answers are not required, and will not be graded. This is a multiple-choice question.

For reference, here are the six decompositions to be considered. Remember, you must use page 11 of this examination for your answers.

(a) The decomposition { R[ABC], R[DEF] }

(b) The decomposition {R[AB], R[BC], R[DEF] }

(c) The decomposition { R[ABC], R[DEF], R[AEF] }

(d) The decomposition { R[AB], R[BC], R[ADE], R[DEF] }

(e) The decomposition { R[AB], R[BC], R[DF], R[EF] }

(f) The decomposition { R[DF], R[AB], R[AEF], R[BC], R[EF] }

(g) The decomposition { R[ABCDEF] }

For reference, here is the list of the six candidate properties:

none 2NF 3NF BCNF lossless dependency-preserving

Note on notation: R[ABC] is an abbreviation for the relation obtained by projecting R onto the attributes in {A,B,C}; *i.e.,* $\pi_{\{A,B,C\}}$(R(A,B,C,D,E,F)).

The following table applies to problems 8 and 9.

Number of tuples in the relation	3000000000 (3×10^9)
Size of one tuple	273 bytes
B^+-tree pointer size	6 bytes
B^+-tree internal key size	16 bytes
Page size	4K (4096 bytes)
Sequential-access pointers in the leaf nodes	16 bytes total

8. (70 points total) Suppose that a relation will 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. Answer the following. You *must* show work to obtain credit.

(a: 10 points) Compute the maximum number of indices (keys) per non-leaf node. Assume that such a node contains only indices and pointers.

(b: 12 points) Compute the maximum number of records per leaf node. Assume that such nodes contain only records and the sequential-access pointers.

(c: 12 points) Compute the maximum height that such a B^+-tree may have. (The height is defined as the length of a path from the root to a leaf.)

(d: 12 points) For a tree with the height computed in (c), assuming that it has the minimum possible density of records in the leaves, and a uniform distribution of keys in the interior nodes (other than the root), compute the average number of keys per (non-root) interior node. The root must contain the minimum number of keys consistent with these constraints. The answer need not be an integer.

(e: 12 points) Compute the number of leaf nodes in the B^+-tree.

(f: 12 points) Compute the number of interior (index) nodes in the B^+-tree.

9. (60 points total) A B^+-tree is again used as the data structure. Assume that each node, interior or leaf, is stored in one page, and that each tuple is stored as one record. Answer the following. You *must* show work to obtain credit.

(a: 12 points) Compute the minimum height that such a B^+-tree may have. (The height is defined as the length of a path from the root to a leaf.)

(b: 12 points) For a tree with the height computed in (a), assuming that it has the maximum possible density of records in the leaves, and a uniform distribution of keys in the interior nodes (other than the root), compute the average number of keys per (non-root) interior node. The root must contain the maximum number of keys consistent with these constraints. The answer need not be an integer.

(c: 12 points) Compute the number of leaf nodes in the B^+-tree. .

(d: 12 points) Compute the number of interior (index) nodes in the B^+-tree.

(e: 12 points) Repeat part (b), this time assuming that all interior nodes, including the root, have the same distribution of keys. If such a tree is not possible for the minimum height computed in part (a), explain why.

10. (60 points total) A simple database system has three data objects, named x, y, and z. The following four transactions are given.

$T_1 =$ $r_1(x)$ $r_1(z)$ $w_1(x)$ $w_1(z)$
$T_2 =$ $w_2(x)$ $r_2(z)$ $w_2(z)$
$T_3 =$ $r_3(y)$ $r_3(x)$ $w_3(x)$
$T_4 =$ $r_4(y)$ $w_4(z)$ $w_4(y)$

Shown below are two schedules for this set of transactions.

(i) $r_3(y)$ $r_3(x)$ $w_2(x)$ $r_4(y)$ $w_3(x)$ $r_1(x)$ $r_2(z)$ $w_2(z)$ $r_1(z)$ $w_1(x)$ $w_1(z)$ $w_4(z)$ $w_4(y)$

(ii) $r_4(y)$ $r_3(y)$ $r_3(x)$ $w_3(x)$ $r_1(x)$ $r_1(z)$ $w_1(x)$ $w_1(z)$ $w_4(z)$ $w_2(x)$ $r_2(z)$ $w_2(z)$ $w_4(y)$

Perform the following steps for each of these schedules:

(a: 30 points; 15 for each schedule) Draw the conflict graph, and based upon that information, determine whether or not the schedule is conflict serializable. Draw the entire conflict, graph, even if you can determine conflict serializability from a part which you have already drawn.

(b: 30 points; 15 for each schedule) Determine whether or not the schedule is view serializable. Justify your answer with a clear analysis.

11. (60 points total) Give concise answers to the following. Note that while a graph may support a definition, a graph, by itself, is not an adequate answer.

(a: 20 points) Give a precise definition of the general two-phase locking protocol (2PL). Within the context of transaction scheduling, explain why 2PL is such an important concept.

(b: 20 points) Give a precise definition of the conservative 2PL protocol. Identify and discuss the advantages and disadvantages of this extension over the general 2PL protocol.

(c: 20 points) Give a precise definition of the strict 2PL protocol. Identify and discuss the advantages and disadvantages of this extension over the general 2PL protocol.

12. (60 points total) Assume that the pure *immediate update* strategy is used for database recovery. Answer the following questions.

(a: 30 points) Assume that all log entries are deleted once a transaction has committed. Using a **simple** example, show that the crash of a transaction may lead to a situation in which full recovery is not possible.

(b: 30 points) Identify sufficient conditions on the schedule of transactions which will ensure that full recovery is always possible, even in the case that log entries for a transaction are deleted when the transaction commits. Your solution must retain the properties of the pure immediate update strategy.

13. (60 points total) One of the most important concepts in object-oriented database systems is that of *persistence*.

(a: 30 points) Give a concise explanation of this concept. In particular, contrast its applicability in database systems to its applicability in programming languages.

(b: 30 points)There are two main methods of achieving object persistence in an object-oriented database system. Identify them by name, and provide a concise description of each.

Appendix:

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

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

m = the total number of indices at the root node.
r= the total number of records in each non-root node.
d = the depth of the tree; *i.e.,* the length of a path from the root to a(ny) leaf node.
R(m,r,d) = the total number of records in the tree.

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

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

m = the total number of indices at the root node.
q = the total number of indices in each non-root, non-leaf node.
r= the total number of records in each leaf node.
d = the depth of the tree; *i.e.,* the length of a path from the root to a(ny) leaf node.
R(m,q,r,d) = the total number of records in the tree.

Name and ID number:______________________________________

Answer sheet for problem 7:

Instructions: Circle (draw a ring around) the names of the properties which that decomposition has. Circle the word *none* just in the case that none of the other five properties applies. (Thus, it is never correct to circle nothing at all, and it is never correct to circle the word *none* together with something else.)

Explanations of answers are not required, and will not be evaluated. This is a multiple-choice question.

(a) The decomposition { R[ABC], R[DEF] }:

none 2NF 3NF BCNF lossless dependency-preserving

(b) The decomposition {R[AB], R[BC], R[DEF] }:

none 2NF 3NF BCNF lossless dependency-preserving

(c) The decomposition { R[ABC], R[DEF], R[AEF] }:

none 2NF 3NF BCNF lossless dependency-preserving

(d) The decomposition { R[AB], R[BC], R[ADE], R[DEF] }:

none 2NF 3NF BCNF lossless dependency-preserving

(e) The decomposition { R[AB], R[BC], R[DF], R[EF] }:

none 2NF 3NF BCNF lossless dependency-preserving

(f) The decomposition { R[DF], R[AB], R[AEF], R[BC], R[EF] }:

none 2NF 3NF BCNF lossless dependency-preserving

(g) The decomposition { R[ABCDEF] }:

none 2NF 3NF BCNF lossless dependency-preserving