Umeå University
Department of Computing Science
TDBC86 – Database Concepts
Examination: January 10, 1998

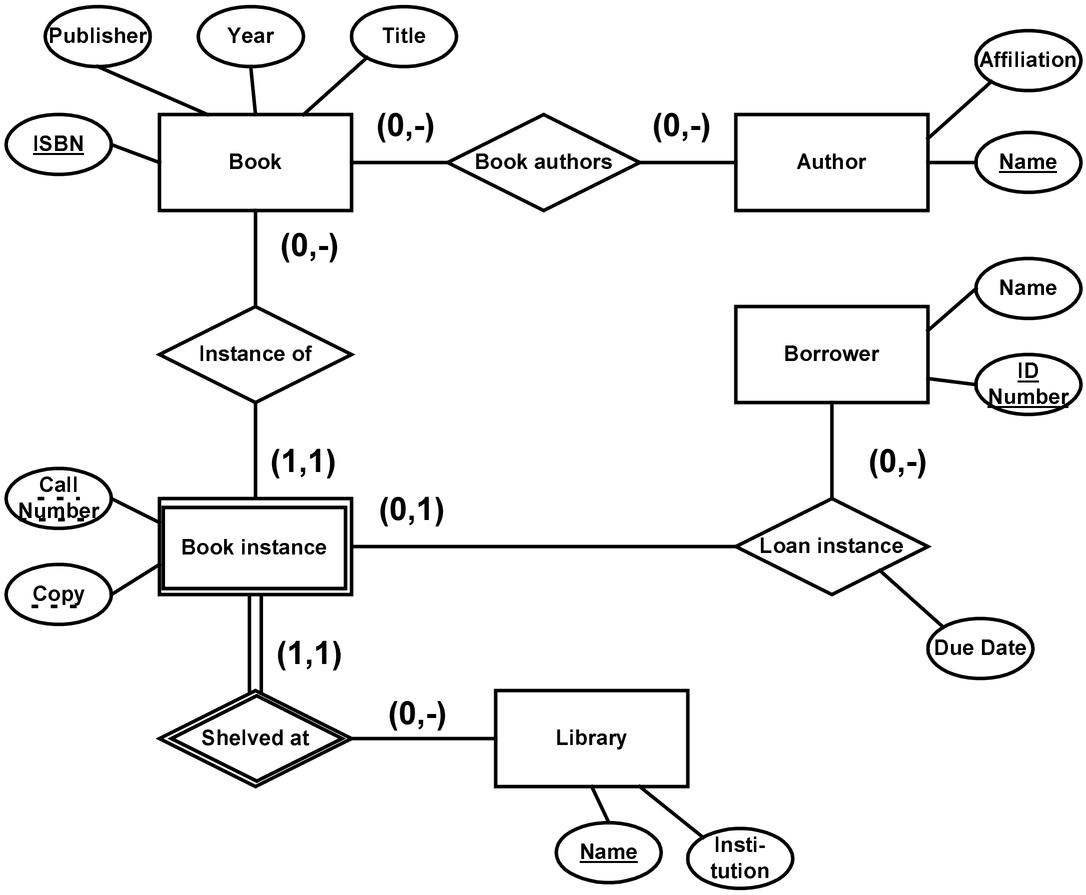
Name and ID number: ________________________________

Signature:____

1. All answers must be written in English.
2. A pocket calculator may be used. No other help materials are allowed.
3. With the exception of Problems 15 and 16, solutions to all problems must be written on the special examination paper, which is provided. 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 solutions to Problems 15 and 16 must be written on the special forms which are attached to this examination.
5. The examination has a total of 900 points. Your point total will be divided by 10, and then your average points from the obligatory exercises will be added on, to obtain a final score between 0 and 100, upon which your grade will be based.
6. 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 "no B.S." 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.	(a)	(b)	(c)	(d)	(e)	(f)	(g)	(h)	(i)	(j)
1		XXX	XXX	XXX	XXX	XXX	XXX	XXX	XXX	XXX
2							XXX	XXX	XXX	XXX
3							XXX	XXX	XXX	XXX
4							XXX	XXX	XXX	XXX
5					XXX	XXX	XXX	XXX	XXX	XXX
6										
7						XXX	XXX	XXX	XXX	XXX
8						XXX	XXX	XXX	XXX	XXX
9							XXX	XXX	XXX	XXX
10							XXX	XXX	XXX	XXX
11							XXX	XXX	XXX	XXX
12							XXX	XXX	XXX	XXX
13							XXX	XXX	XXX	XXX
14							XXX	XXX	XXX	XXX
15		XXX	XXX	XXX	XXX	XXX	XXX	XXX	XXX	XXX
16		XXX	XXX	XXX	XXX	XXX	XXX	XXX	XXX	XXX

1. (80 points) Shown below is an ER-diagram. Using the techniques developed in the course, map this diagram into an equivalent relational schema. Show all keys, primary and foreign.



Notes:

(1) A *book* is a generic object; *e.g., Fundamentals of Database Systems* by ElMasri and Navathe. A *book instance* is a specific copy of a book; *e.g.* the copy of this book which is held by the library at Umeå University.

(2) A *loan instance* is an instance of a person borrowing a book instance from a library.

(3) The notation (a,b) indicates the minimum and maximum number of times that the entity may participate in the relation. A dash (-) indicates that the number of times is indeterminate. (The textbook uses M or N instead of dash.)

(4) The *call number* is the number by which the library identifies the book instance. The call number and copy number together uniquely identify a given book instance within a given library.

The following simple relational database schema,

Customer		
Cust_ID	Name	Acct_No

Account		
Acct_No	Type	Balance

together with the following six queries, are the common context in which Problems 2, 3, and 4, below, should be solved.

(a) Find the name and ID of all customers who have an account of type "savings."

(b) Find the name and ID of all customers who have both an account of type "savings" and an account of type "salary."

(c) Find the name and ID of all customers who have an account of type "salary" but no account of type "savings."

(d) Find the name and ID of all customers who only have accounts of type "salary."

(e) Find the name and ID of all customers who have an account of every type which is listed in the database.

(f) Find the name and ID of all customers who have exactly two different types of accounts.

2. (50 points total) Solve each of the five queries (a) – (e) in the relational algebra. Functional operators, such as count and average, may not be used. You need not provide a solution for (f), since it requires an operation not covered in the course (renaming). Each of the five parts is worth 10 points.

3. (60 points total) Solve each of the six queries (a) – (f) in the tuple relational calculus. Functional operators, such as count and average, may not be used. Each of the six parts is worth 10 points.

4. (60 points total) Solve each of the six queries (a) – (f) using SQL. For this part only, you may use SQL functional operations, such as average. Each of the six parts is worth 10 points.

5. (40 points total) Using the database schema of Problem 2, provide solutions, in SQL, to the following additional queries. All SQL operations may be used, including aggregation operations.

(a) (10 points) Find the sum of the balances of all accounts owned by the customer named “William Gates.”

(b) (10 points) Find the average balance of all accounts of type “salary.”

(c) (10 points) Find the customer name and ID for all customers who have accounts with a balance over 100000.

(d) (10 points) Find the average balance for each type of account. (The average is taken over all customers with such accounts.)

6. (80 points total; 8 points for each part) Let **A** be a finite set of attributes, and let **R** be a relational database schema with a single relation R[**A**]. Assume that R is constrained by a set $\mathfrak{F}$ of functional dependencies. Within this context, define the following concepts. Your definitions should be formal enough to convey precise meaning.

(a) Superkey

(b) Candidate key

(c) Primary key

(d) Third normal form (3NF) *

(e) Boyce-Codd normal form (BCNF) *

Now assume further that $\mathbf{A}_1, \mathbf{A}_2, .., \mathbf{A}_k \subseteq \mathbf{A}$, and that $\{R_1[\mathbf{A}_1], R_2[\mathbf{A}_2], .., R_k[\mathbf{A}_k]\}$ is a decomposition of R[**A**] into projections. Define the following properties for the decomposition.

(f) Dependency preserving decomposition

(g) Lossless decomposition

(h) 3NF decomposition *

(i) BCNF decomposition *

Finally, let **A** and **B** be finite sets of attributes, and let R[**A**] and S[**B**] be relations in a database schema.

(j) Define what it means for a set of attributes of R[**A**] to be a foreign key of S[**B**].

* Note: (d) and (e) ask for the definition of what it means for a single schema to be in the identified normal form, while (h) and (i) ask for the definition of what it means for a decomposition to have that property.

7. (60 points total) Consider the relational schema **R** with relation R(A,B,C,D,E) and functional dependencies {AB $\rightarrow$ C, C $\rightarrow$ DE, D $\rightarrow$ E}.

(a) (10 points) Find all candidate keys for **R**.

(b) (10 points) Give a minimal cover for the set of functional dependencies.

(c) (10 points) Find a dependency-preserving lossless decomposition of the schema into 3NF.

(d) (10 points) Indicate any BCNF violations of the answer to (c).

(e) (20 points) Further decompose your answer to part (c) into a BCNF decomposition, and indicate whether or not this new decomposition is dependency preserving. (If your answer to (c) is already in BCNF, and it is correct, all you need do is indicate that no further decomposition is necessary to get credit for this part.)

8. (60 points) Repeat parts (a) – (e) of Problem 7 for the relational schema **R** with relation R(A,B,C,D,E) and functional dependencies {AB $\rightarrow$ C, C $\rightarrow$ ADE, D $\rightarrow$ E}. The parts carry the same number of points as in Problem 7.

9. (60 points total) A B-tree is used as the data structure for a relation. The following parameters apply:

Number of tuples in the relation	2000000
Size of one tuple	500 bytes
B-tree pointer size	4 bytes
Page size	2K (2048 bytes)
Average fullness of a B-tree node	80%
Time to access a single page	10 ms.

Assume that each B-tree node is stored in one page, and that each tuple is store as one record. Answer the following.

(a) (10 points) Compute the number of nodes in the B-tree. Assume that a node contains only records and pointers to child nodes.

(b) (10 points) Compute the percentage of nodes which will be leaves.

(c) (10 points) Compute the path length from the root to a leaf. (The path from the root to the root has length 0, not 1).

(d) (10 points) Compute the approximate time to retrieve a record, assuming that there is one disk access per B-tree node access.

Assume further that the first two levels of the tree (the root and one more level) are maintained in main memory, and that main-memory access time is negligible in comparison to disk access time.

(e) (10 points) Compute how much memory will be required to store these levels in main memory.

(f) (10 points) Compute the approximate time needed to retrieve a record under these conditions.

10. (60 points total) Suppose now that the same relation will be stored in a B^+-tree.

Number of tuples in the relation	2000000
Size of one tuple	500 bytes
B^+-tree pointer size	4 bytes
Page size	2K (2048 bytes)
Average fullness of a leaf node	80%
Average fullness of an interior node	80%
Time to access a single page	10 ms.
Sequential-access pointers in the leaf nodes	8 bytes total

Assume that each node, interior or leaf, is stored in one page, and that each tuple is store as one record. Answer the following.

(a) (10 points) Compute the number of leaf nodes in the B^+-tree. Assume that a node contains only records and the sequential-access pointers.

(b) (10 points) Compute the number of interior (index) nodes in the B^+-tree. Assume that such a node contains only pointers.

(c) (10 points) Compute the path length from the root to a leaf. (The path from the root to the root has length 0, not 1).

(d) (10 points) Compute the approximate time to retrieve a record, assuming that there is one disk access per B^+-tree node access, regardless of whether the node is a leaf or an interior node.

Now assume that all index nodes (interior nodes) are maintained in main memory, and that main-memory access time is negligible in comparison to disk access time.

(e) (10 points) Compute how much memory will be required to store this index in main memory.

(f) (10 points) Compute the approximate time needed to retrieve a record under these conditions.

A simple database system has three data objects, named x, y, and z. The following four transactions are given.

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

These four transactions, together with the following six schedules, form the common context for the solution of Problems 11, 12, and 13 below.

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

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

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

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

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

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

11. (30 points total; 5 points for each part) For each of six schedules (a) – (f), determine whether or not it is a serial schedule.

12. (60 points total; 10 points for each part) For each of the six schedules (a) – (f), determine whether or not it is view serializable. In the case that it is view serializable, give an equivalent serial schedule. In the case that it is not, explain why.

13. (60 points total; 10 points for each part) For each of the six schedules (a) – (f), determine whether or not it is conflict serializable. Support your answer by providing the associated conflict graph (precedence graph).

14. (40 points total)

(a) (7 points) Explain what is meant by a *two-phase locking (2PL) protocol*.

(b) (6 points) State the relationship between a 2PL protocol and the serializability of schedules.

(c) (6 points) Explain what is meant by a *conservative* 2PL protocol.

(d) (7 points) Explain why conservative 2PL protocols are sometimes employed. Identify the problems which such schedules overcome, and the limitations which they impose.

(e) (6 points) Explain what is meant by a *strict* 2PL protocol.

(f) (7 points) Identify the advantages of employing a strict 2PL protocol, as opposed to a general one. Identify the disadvantages as well.

The schedule below forms the common context in which Problems 15 and 16 are to be solved. **To simplify grading, two blank copies of the chart shown below are attached to this exam. Please use these preprinted sheets for your solutions**.

Line	**Action**	**Database Values**		
		x	**y**	**z**
	Initial values:	10	20	30
1	Begin(T_1)			
2	Read(T_1,x)			
3	Write(T_1,x,15)			
4	Begin(T_2)			
5	Read(T_2,y)			
6	Write(T_2,y,25)			
7	Begin(T_3)			
8	Read(T_3,x)			
9	Write(T_3,x,35)			
10	End(T_3)			
11	Begin(T_4)			
12	Read(T_4,y)			
13	Read(T_4,z)			
14	Write(T_4,z,45)			
15	Write(T_4,y,55)			
16	End(T_4)			
17	Write(T_1,z,65)			
18	End(T_1)			
19	End(T_2)			

15. (50 points) For the schedule of actions shown above, indicate the values which will be stored in the database at the conclusion of each step, assuming that the *immediate update strategy* is employed. You only need show values which have changed since the previous step. Write your answers on the copy on page 11 of this exam, and turn in this page as part of your solution.

16. (50 points) For the schedule of actions shown above, indicate the values which will be stored in the database at the conclusion of each step, assuming that the *deferred update strategy* is employed. You only need show values which have changed since the previous step. Write your answers on the copy on page 12 of this exam, and turn in this page as part of your solution.

Name and ID number: ____

Solution to problem 15:

Line	**Action**	**Database Values**		
		x	**y**	**z**
	Initial values:	10	20	30
1	Begin(T_1)			
2	Read(T_1,x)			
3	Write(T_1,x,15)			
4	Begin(T_2)			
5	Read(T_2,y)			
6	Write(T_2,y,25)			
7	Begin(T_3)			
8	Read(T_3,x)			
9	Write(T_3,x,35)			
10	End(T_3)			
11	Begin(T_4)			
12	Read(T_4,y)			
13	Read(T_4,z)			
14	Write(T_4,z,45)			
15	Write(T_4,y,55)			
16	End(T_4)			
17	Write(T_1,z,65)			
18	End(T_1)			
19	End(T_2)			

Name and ID number: ____

Solution to problem 16:

Line	Action	Database Values		
		x	**y**	**z**
	Initial values:	10	20	30
1	Begin(T_1)			
2	Read(T_1,x)			
3	Write(T_1,x,15)			
4	Begin(T_2)			
5	Read(T_2,y)			
6	Write(T_2,y,25)			
7	Begin(T_3)			
8	Read(T_3,x)			
9	Write(T_3,x,35)			
10	End(T_3)			
11	Begin(T_4)			
12	Read(T_4,y)			
13	Read(T_4,z)			
14	Write(T_4,z,45)			
15	Write(T_4,y,55)			
16	End(T_4)			
17	Write(T_1,z,65)			
18	End(T_1)			
19	End(T_2)			