Umeå University
Department of Computing Science
TDBC86 – Database Concepts
Examination: December 20, 2000

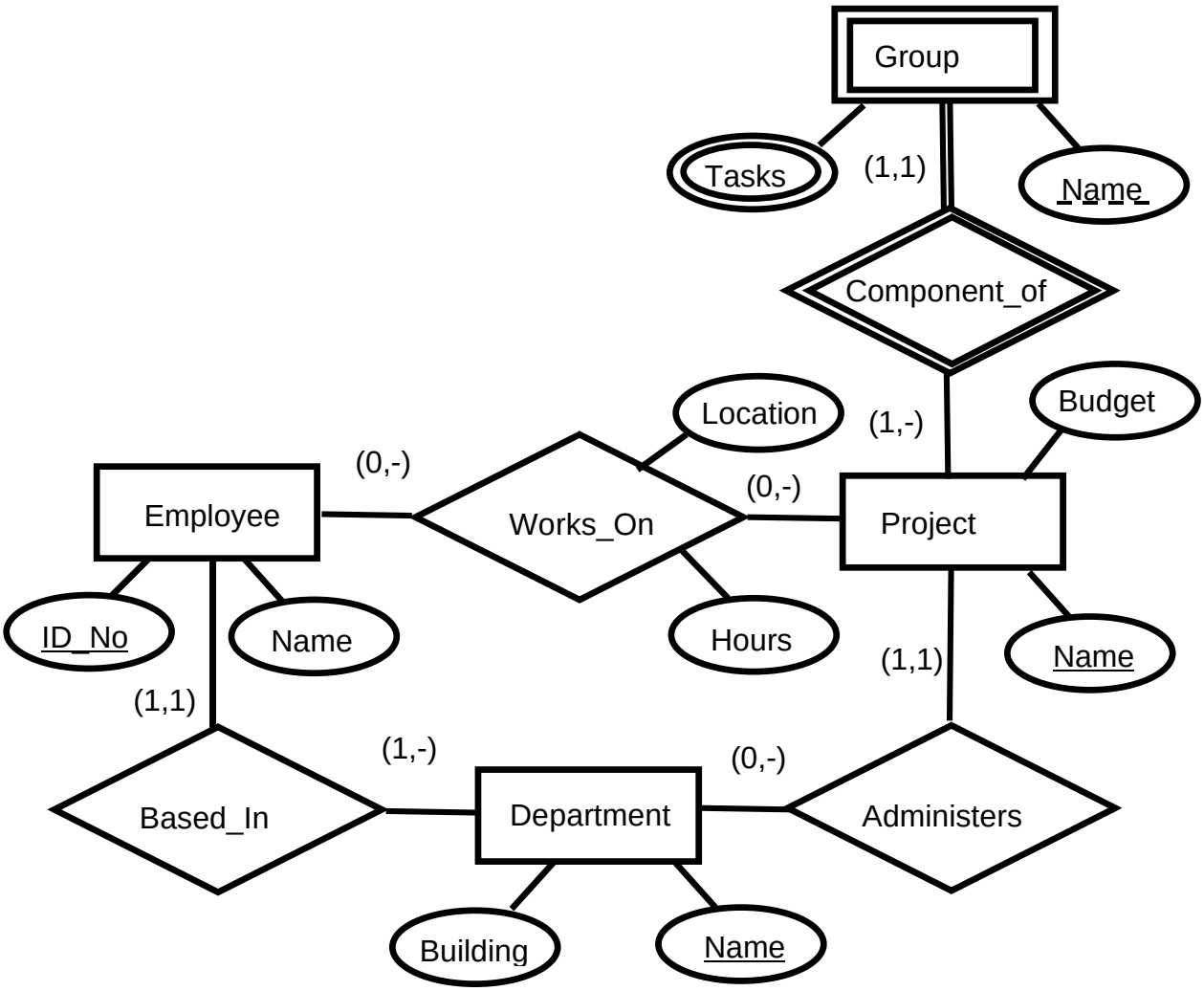
Name and ID number: _____

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 Problems 2(a) and 8, must be written on the special examination paper, which is provided. Solutions to Problems 2(a) and 8 must be written on the special answer sheets, which are included at the end of this examination. Write your name, ID number, and the problem number on each solution sheet. Collate the sheets in numerical order of the problems, and write on only one side of the paper. Attach this sheet as a cover to your solutions.
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 15% 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 15% 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)													
(g)													
(h)													
(i)													
(j)													

(1:60 points total) Shown below is an ER-diagram for a corporate 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.*

Note:

- 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.

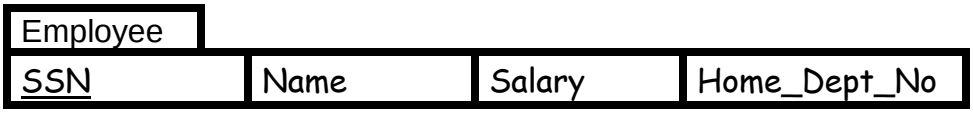
(2: 40 points total) This problem continues with the schema introduced in Problem 1.

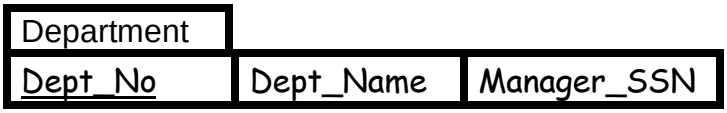
(a: 20 points) Augment this schema to include additional information: Every group has exactly one *group leader*, who is an employee. A given employee may be group leader for at most three groups. Use the answer sheet at the end of this examination to record your answer.

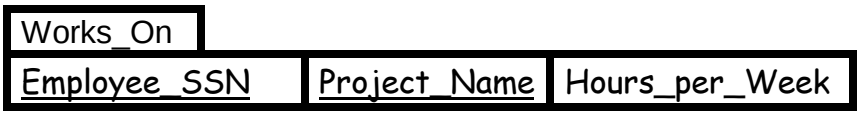
(b: 20 points) Show clearly how your answer to Problem 1 would be augmented to represent the additional information of part (a) of this problem in the relational setting. Redraw as much of the answer to Problem 1 as necessary to provide a clear answer. Do **not** combine your answers to Problem 1 and part (b) of this problem into one.

Note: If you do not do part (a) correctly, or reasonably so, you will not obtain credit for part (b). Thus, do not attempt part (b) unless you are reasonably certain of your answer to part (a).

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 3, 4, and 5, below, should be solved.







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

(a) Find the names of those employees who work on either a project located in Boston or a project located in Paris, or both.

(b) Find the names of those employees who work both on a project located in Boston and on a project located in Paris.

(c) Find the names of those employees who manage more than one department.

(d) Find the names of those employees who work on all projects which are located in Oslo.

(e) Find the names of those employees who work on some project which is also worked on by the manager of that employee.

3. (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.

4. (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.

5. (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.

6. (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 project, find the minimum, maximum, and average salary of all employees who work on that project. The result must list the project name as well as the three salary aggregates.

(b: 15 points) For each manager, find the number of employees which that individual manages. The result must list the SSN and name of the manager, as well as the number of employees.

(c: 15 points) Find the average salary of all managers.

(d: 15 points) For each employee, list the total number of projects on which that employee works, together with the average number of hours per project which that employee works. If the employee does not work on any projects, list that average as 0.

7. (70 points total; 7 points for each part) Let $\mathbf{A}$ be a finite set of attributes, and let $\mathbf{R}$ be a relational database schema with a single relation R[$\mathbf{A}$]. Assume that R is constrained by a set $\mathfrak{F}$ of functional dependencies. Within this context, define the following concepts. Your definitions must be formal enough to convey precise meaning.

(a) Superkey

(b) Candidate key

(c) Primary key

(d) Prime attribute

(e) Second normal form (2NF)

(f) Third normal form (3NF)

(g) 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[$\mathbf{A}$] into projections. Define the following properties for the decomposition.

(h) Dependency preserving decomposition

(i) Lossless decomposition

Finally, let $\mathbf{A}$ and $\mathbf{B}$ be finite sets of attributes, and let R[$\mathbf{A}$] and S[$\mathbf{B}$] be relations in a database schema.

(j) Define what it means for a set $\mathbf{K}$ of attributes of R[$\mathbf{A}$] to be a foreign key of R[$\mathbf{A}$] which references S[$\mathbf{B}$].

(8: 70 points total) **Important: The answers to this question must be indicated on page 14 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,G) and functional dependencies {A $\rightarrow$ B, B $\rightarrow$ C, DE $\rightarrow$ F, F $\rightarrow$ DG}.

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 14 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 15% 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[DEFG] }

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

(c) The decomposition { R[BC], R[ACDEFG] }

(d) The decomposition { R[AB], R[BCDEF], R[FG] }

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

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

(g) The decomposition { R[ABCDEFG] }

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.,* $\pi_{\{A,B,C\}}(R(A,B,C,D,E,F,G))$.

(9: 80 points total) A B^+-tree is used as the data structure to represent a relation. Assume that each node, interior or leaf, is stored in one page, and that each tuple is stored as one record. Specific information regarding the physical parameters are given in the following table.

Number of tuples in the relation	4000000000 (4×10^9)
Size of one tuple	200 bytes
B^+-tree pointer size	4 bytes
B^+-tree internal key size	20 bytes
Page size	2K (2048 bytes)
Time to access a single page	9 ms.
Sequential-access pointers in the leaf nodes	8 bytes total

(a: 10 points) Compute the maximum number of records per interior node.

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

(c: 10 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.)

(d: 10 points) For such a tree of minimum height, assume that the tree has the maximum possible density of records in the leaves, and a uniform distribution of keys in the interior nodes (other than the root). The root must contain the maximum number of keys consistent with these constraints. Under these assumptions, compute the number of keys in the root node, as well as the average number of keys per (non-root) interior node. The latter answer need not be an integer.
(Hint: If a completely full root does not yield a possible key distribution, compute the maximum number of keys in the root under the assumption of minimum key density in the other interior nodes. Then, use this key density for the root to compute the actual key density for the other nodes.)

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

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

(g: 10 points) Assume that the index nodes (interior nodes) for the first four levels of the tree are maintained in main memory, and that main-memory access time is negligible in comparison to disk access time.

(i) Compute the number of nodes in the first four levels.
(ii) Compute how much memory will be required to store this index.
(iii) Compute the approximate time needed to retrieve a record under these conditions.

(h: 10 points) Repeat part (b) of problem 5, 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) Shown below is a transaction for a system with four data objects: x, y, z, and w.

```
Begin(T)
        Read(T,x)
        Write(T,x)
        Read(T,y)
        Write(T,y)
        Write(T,w)
        Read(T,z)
        Write(T,z)
Commit(T)
```

(a: 30 points) For each of the following locking protocols, insert statements of the form Lock(T,d) and Unlock(T,d), with d $\in$ {x, y, z, w}, into (a copy of) the above transaction, such that the resulting augmented transaction satisfies the condition identified below. If no such locking is possible, so state and explain why.

(i) Not 2PL. (That is, fails to satisfy the conditions of 2PL).
(ii) 2PL, but neither strict nor conservative 2PL.
(iii) Strict 2PL, but not conservative 2PL.
(iv) Conservative 2PL but not strict 2PL.
(v) Both conservative 2PL and strict 2PL.

(b: 10 points) Briefly explain the importance of the two-phase locking (2PL) protocol.

(c: 10 points) Briefly explain the added advantage of a conservative 2PL strategy.

(d: 10 points) Identify a situation in which it is an advantage to enforce a strict 2PL strategy (over simple 2PL). Justify your answer with a brief but clear explanation.

Note: Assume that there is just one form of lock, which is an exclusive read and write lock. Do not use special read locks.

(11: 60 points total)

(a: 12 points) Briefly explain the *pure deferred update strategy* for database transaction processing.

(b: 12 points) Identify the major drawback of the pure deferred update strategy which makes it unsuitable for practical systems.

(c: 12 points) Describe and contrast the *no-steal* and *steal* strategies of database cache management.

(d: 12 points) Describe and contrast the *no-force* and *force* strategies of database cache management.

(e: 12 points) Suppose that it is desired to improve the performance of the pure deferred update strategy. Of the four possibilities *no-steal+no-force, steal+no-force, no-steal+force, steal+force,* identify the combination which has the most potential for improvement of performance, and explain why.

(12: 60 points total) Joe Politician (JP) works at a small firm with ten employees. He is also a representative in the local legislature (kommunfullmäktige). You are a newspaper reporter who is writing a story about JP, and you need to know his salary at the firm. He claims that such information is confidential, and the firm is not required to divulge (avslöja) such information. Fortunately, a local ordinance (förordning) does require the firm to allow statistical queries to its corporate database, provided they aggregate at least 30%, but not more than 70%, of the total database. This database consists of a single relation, with four attributes, SSN (a unique ID), Salary, Age, and Department. You do know that JP is the only person of age 50 who works in the Maintenance Department. You do not know his SSN.

Your task is to write a statistical tracker which determines the salary of JP. The constraints are that all queries, which must be written in SQL, must return purely aggregate information (*e.g,* Count, Average, etc.), and must summarize over at least three, but not more than seven, tuples. Your solution must be based upon a general tracker, and your queries *must* be accompanied by an explanation of what they do. You may have a final step which does some simple computations on results returned by the SQL queries.

The actual instance of the database is shown below. It is provided only so that you can use it to identify a suitable general tracker (since it is impossible to provide interactive access to the database during the examination). You may use this instance only to verify that your general tracker selects an appropriate number of tuples.

Employee	SSN	Salary	Age	Department
	0000000000	50000	30	Finance
	1111111111	35000	40	Maintenance
	2222222222	60000	40	Research
	3333333333	90000	60	Administration
	4444444444	250000	50	Maintenance
	5555555555	45000	40	Finance
	6666666666	55000	30	Research
	7777777777	65000	28	Administration
	8888888888	30000	45	Finance
	9999999999	40000	37	Personnel

(13: 60 points total) Within the general context of ODBC hosted within the programming language C, there are at least five distinct notions of data type, including the following:

- C_type
- ODBC type
- ODBC encoded type
- SQL type
- SQL encoded type

Give a contrastive explanation of the nature and rôle of each of these types. In particular, clarify the nature and use of each type (a type in the underlying programming language, a type in the underlying query language, an encoding, etc.). Illustrate with examples using types based upon characters and strings of characters (*e.g.*, SQL `Char` and `Varchar`).

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:

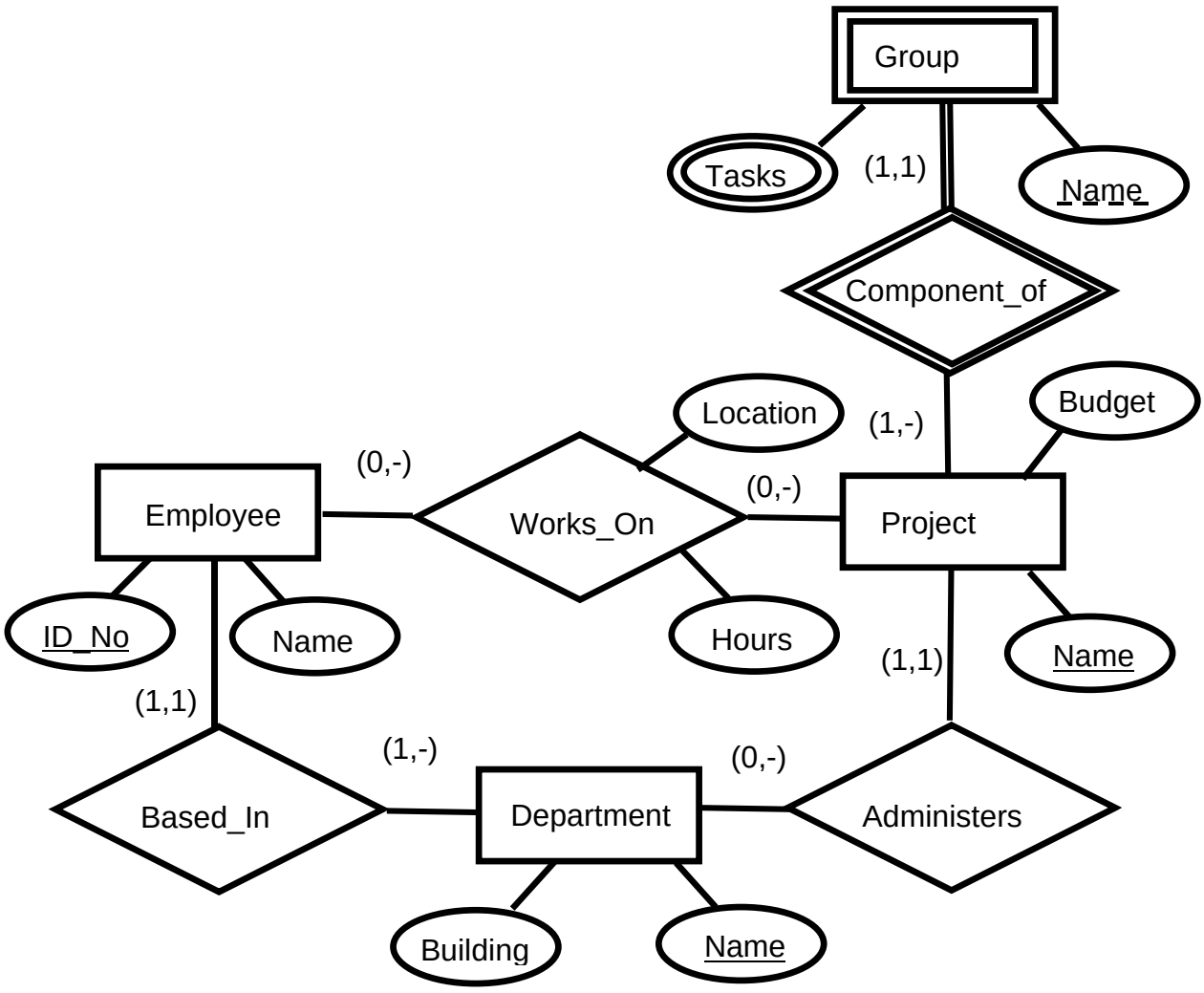
$$R(m,q,r,d) = (m+1) \cdot (q+1)^{d-1} \cdot r$$

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 2(a):

Draw your augmentation of the ER-schema on this sheet.



Name and ID number:_____

Answer sheet for Problem 8:

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[DEFG] }:

none 2NF 3NF BCNF lossless dependency-preserving

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

none 2NF 3NF BCNF lossless dependency-preserving

(c) The decomposition { R[BC], R[ACDEFG] }:

none 2NF 3NF BCNF lossless dependency-preserving

(d) The decomposition { R[AB], R[BCDEF], R[FG] }:

none 2NF 3NF BCNF lossless dependency-preserving

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

none 2NF 3NF BCNF lossless dependency-preserving

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

none 2NF 3NF BCNF lossless dependency-preserving

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

none 2NF 3NF BCNF lossless dependency-preserving