University of Canterbury

End Of Year Examinations 2017

Prescription Number(s): COSC265

Paper Title: Relational Database Systems

Time Allowed: 2 hours

Number of Pages: 5

Instructions

- Answer *all* questions.

- This test is worth a total of 100 marks.

- The exam is run under the closed book conditions.

- Look through the entire exam quickly first. Check carefully the number of marks allocated to each question. This suggests the degree of detail required in each answer, and the amount of time you should spend on the question.

- In all answers be sure to provide clear and concise explanations of your reasoning. No marks will be given for material not directly relevant to the question.

- Use the separate answer booklet for answering all questions.

- No form of collaboration is permitted.

- Do not write your answers in pencil.

- Marks are most frequently lost for several reasons:

    - The question has not been read carefully.

    - The answer contains poor writing and/or rough diagrams.

    - The correct answer contains multiple parts or thoughts, but the given answer only contains one part or thought.

1 [11 marks for the whole question] EER diagrams can model scenarios and concepts at many levels of abstraction. Here we have an EER diagram that models a small part of a relational database management system. This diagram thus represents some of the metadata of that DBMS.

Map this diagram to the equivalent relational database schema. Specify all attributes and primary keys, and note any foreign keys for each relation as well.
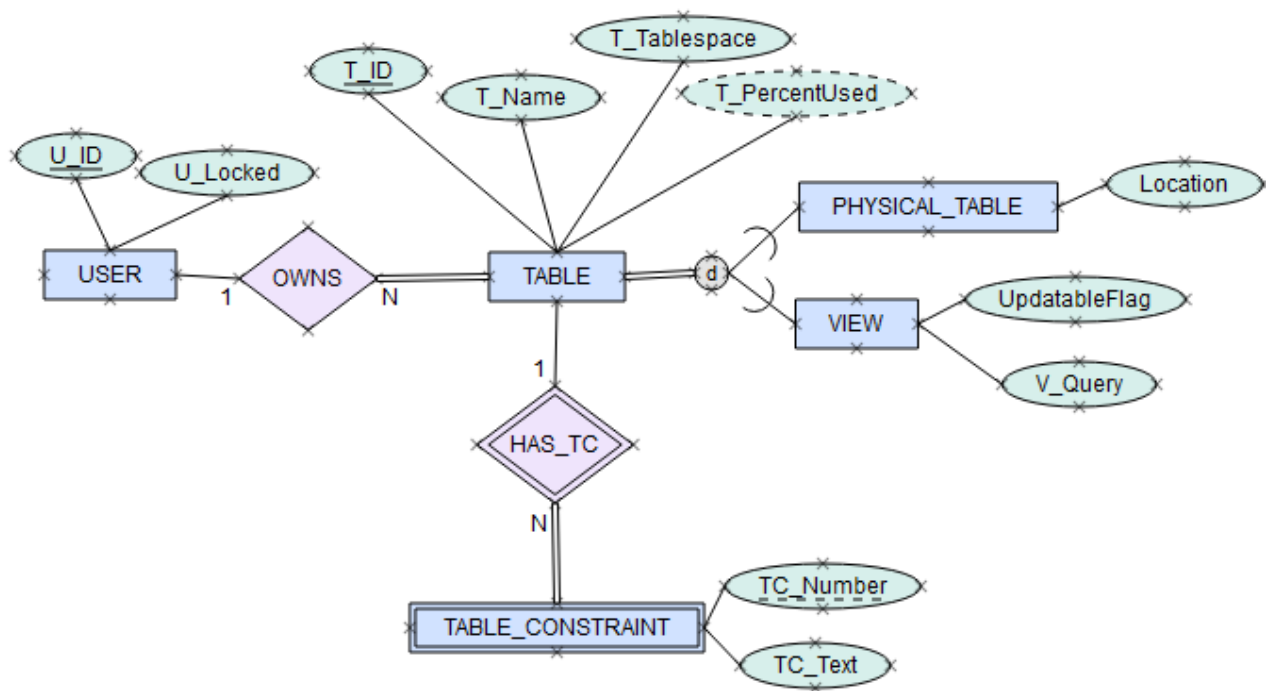
2 [17 marks for the whole question] Relation R (A, B, C, D, E, F) is given, as well as a minimal set of functional dependencies F = { E → A , B → C, B → F, D → A, D → E }.

a) **[5 marks]** Identify all candidate keys for relation R. Justify your answer.

b) **[5 marks]** Identify the highest normal form that R satisfies. Justify your answer.

c) **[7 marks]** If R is not in 3NF, transform it into a set of 3NF relations.

3 [40 marks for the whole question] In the lab test, you worked with a small database storing data for a creature/achievement/skill scenario. The five (slightly modified) tables here have the following structure, with the relational tables and columns named as follows:

```
Creature (C_id, C_name, C_Type, reside_t_id)
                    -- reside_t_id: residence town for that creature
Skill (S_code, S_desc, S_weight)
Achievement (C_id, S_Code, score, test_t_id)
                    -- test_t_id : testing town for that achievement
Town (T_id, T_name)
JobSkill (Job_ID, S_Code, Job_Name)
```

a) **[15 marks]** Draw an EER diagram that reflects all tables and relationships from the schema described above.

b) **[4 marks]** Write a single SQL statement to find the name of each creature that has had at least one skill achievement that was tested in the town of Rotorua.

c) **[4 marks]** State the result of the following SQL query in English:
```
SELECT DISTINCT T_name
FROM Town T
WHERE T.T_id IN
    (SELECT C.reside_t_id
     FROM Creature C
     JOIN Achievement A ON A.C_id = C.C_id
     WHERE A.test_t_id = C.reside_t_id);
```

d) **[7 marks]** Given an SQL query as follows:
```
SELECT C.C_ID, A.S_Code
FROM Creature C
JOIN Achievement A on C.C_id = A.C_id
WHERE A.Score = 1;
```

Draw relational algebra query trees for **two** different relational algebra queries corresponding to this SQL query, with one of the two query trees being the canonical tree. Discuss how efficient each query tree would be to execute, in relation to various options for the physical design.

e) **[3 marks]** Is the JobSkill table in 3rd Normal Form? Briefly explain why or why not.

f) **[4 marks]** Write a single SQL statement to create a new Job table that holds a Job_ID (as its key) and the job name. Note that we are separating out these two columns from the Job_Skill relation.

g) **[3 marks]** If we would populate and use this new Job table (NOTE: you don't have to populate/use it here), what normalization change(s) would we make to our existing tables?

4 [18 marks for the whole question]

a) **[4 marks]** Briefly describe why the SQL query you send to the DBMS may not be executed by the DBMS in the same form that you wrote it.

b) **[4 marks]** Why is cost-based query optimization preferable to heuristic query optimization?

c) **[6 marks]** Briefly describe how B+ trees are different than binary trees.

d) **[4 marks]** What are the two major types of indexing that are mostly commonly used for implementing relational DBMS indexes?

5 [14 marks for the whole question]

a) **[4 marks]** What is the purpose of a *transaction* in database systems?

b) **[4 marks]** In regard to applying transactions, would most DBMSs be considered *optimistic* (i.e. applies each transaction tentatively, performs an undo if necessary) or *pessimistic* (i.e. holds each transaction tentatively, only applies the transaction if all work is completed and committed)? Briefly justify your answer.

c) **[3 marks]** Briefly describe the purpose of a Data Access Object (DAO) when writing software to programmatically access a database system.

d) **[3 marks]** Briefly describe how SQL injection can allow malformed input to lead to security issues such as unauthorized access to a database system.

END OF PAPER