Database Systems Homework 4

(Due by 10pm on April 19, 2022 to Brightspace Homework 4 Submission Folder)

Please remember to include the following statement at the beginning of your submitted assignment and SIGN it. Your assignment won't be graded with the signed statement.
"I have done this assignment completely on my own. I have not copied it, nor have I given my solution to anyone else. I understand that if I am involved in plagiarism or cheating I will have to sign an official form that I have cheated and that this form will be stored in my official university record. I also understand that I will receive a grade of 0 for the involved assignment and my grade will be reduced by one level (e.g., from A to A- or from B+ to B) for my first offense, and that I will receive a grade of "F" for the course for any additional offense of any kind."

1. Consider the two tables Courses and Classes used in Project 2. Test the following with real SQL statements in your Oracle account and report the result of each test case (including the SQL statements used). [Don't forget to re-create and re-populate all tables again after you are done with this question by running the table creation script again.]

   (a) (6 points) Try to drop table Courses using "drop table courses;". Report what happened (show the actual SQL statement and the returned message. Explain the returned message (why you have this message).

   (b) (14 points) Try to drop table Courses using "drop table courses cascade constraints;"). Did the table Courses get dropped? Did any tuples from table Classes get deleted? Show the actual SQL statements and the tuples of Courses and Classes before and after executing "drop table courses cascade constraints;", and explain why. Did "on delete cascade" with the foreign key in Classes get activated?

2. (20 points) The Referential Integrity Constraint says that any value under a foreign key must either be a null value or a value matching a value under the corresponding primary (candidate) key. When a foreign key FK in table R1 has two attributes (A, B), there are two possible ways to define the meaning of a null value: (i) both A and B have null values, i.e., (null, null) and (ii) at least one of the values under A and B is a null value, i.e., (null, b), (a, null) or (null, null), where a and b are non-null values. In Project 2, the foreign key (dept_code, course#) in table Classes has two attributes that reference the primary key of table Courses. Use the two tables and design an experiment in Oracle to answer the following question. Report your queries and results for each experiment.

   Which of the above two meanings, i.e., (i) and (ii), is adopted by Oracle? Note that for this experiment, the schema for table Classes needs to be modified to allow null values for both dept_code and course# (just remove "not null" for the two attributes for this table in the script file for Project 2). To make the experiment somewhat uniform, please do the following:
   (a) Show the contents of Courses and Classes.
   (b) Try to insert a tuple into Classes with values for (dept_code, course#) that do not match anything in Courses. This is to show that the system is enforcing the referential integrity constraint.
   (c) Try to insert a tuple into Classes with null values for both dept_code and course#.
   (d) Try to insert a tuple into Classes with a non-null value for dept_code and a null value for course#.
   (e) Try to insert a tuple into Classes with a null value for dept_code and a non-null value for course#.
   (f) If you have not already done so, try to insert a tuple into Classes with a null value for dept_code and a non-existent non-null value for course# (i.e., the value is not in Courses).
   (g) Show the content of Classes again to see what tuples have been inserted.
   (h) Answer the question regarding which of (i) and (ii) is adopted by Oracle. Discuss the results – did any result surprise you?

3. Create a simple view CS_Courses on table Courses from Project 2 in your Oracle account. CS_Courses logically contains only CS courses from Courses. Perform the following tasks and save your work in one or more spool files, and include them in your solution.

   (a) (12 points) Try to insert a tuple into and delete a tuple from table Courses through view CS_Courses. Report whether you succeeded by showing the contents of the table and view before and after the insertion and deletion.

   (b) (8 points) Try to insert a new math course into Course via CS_Courses. Note that this new course does not logically belong to CS_Courses. Report whether you succeeded by showing the contents of the table and view before and after the insertion and deletion.

4. (10 points) Use an example to explain when and why a primary index on an attribute A of relation R can result in better performance (i.e., lower I/O cost) than a secondary index on the same attribute.

5. Suppose relation R has 4,000,000 tuples such that each tuple occupies 200 bytes. The tables are stored in consecutive storage spaces on disk. We want to create a B+ tree index based on attribute A of R. Suppose the type of A is char(6), each address/pointer occupies 4 bytes, the size of each page is 2KB (2,048 bytes), and each node in the B+ tree has an initial fill-factor of 80% (i.e., only 80% of each node/page in the B+ tree can be used initially, making the usable size of each page to be 2048 * 0.80 = 1638). **Note that the 80% fill-factor is only used for nodes (both leaf nodes and internal nodes) in the B+ tree, and pages used to store the tuples are all filled up to the fullest capacity possible.**

   (a) (10 points) How many levels this B+ tree will have? Compute the number of nodes at each level.

   (b) (5 points) Suppose your answer to question (a) is *k* (i.e., the B+ tree has *k* levels), what would be the maximum possible number of tuples this *k*-level B+ tree can accommodate? For this question, the 80% fill factor still applies. (hint: make the root as full (as close to 1638 bytes) as possible)

   (c) (10 points) Consider a selection condition "A between a1 and a2", where a1 and a2 are constants. Suppose the number of distinct values under attribute A between a1 and a2 (including a1 and a2) is 10. Suppose 100 tuples in R satisfy this condition. What would be the **maximum possible** number of pages that need to be brought into the memory in order to find these 100 qualified tuples if the B+ tree is a primary index? Please justify your answer. (It is assumed that initially both the B+ tree and the table are stored on disk. We also assume that indirection pages are used to handle repeating values and for simplicity, we assume ten indirection pages are used for these 100 tuples.)

   (d) (5 points) The same question as in (c) except that this time the B+ tree is a secondary index.