**Database Systems Homework 4**

“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.”

Sincerely,

Shahu S Ronghe

B00908937

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

CS532: Homework 4

**Name** - Shahu S Ronghe

**B-Number** - B00908937

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.]
2. (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).

**SQL> drop table courses;**

**drop table courses**

**\***

**ERROR at line 1:**

**ORA-02449: unique/primary keys in table referenced by foreign keys**

**The course# key of courses is used in prerequisite and classes tables as foreign key for dept\_code, therefore we cannot drop the table without removing the foreign key constraint.**

1. (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 id “on delete cascade” with the foreign key in Classes get activated?

**Yes, the table courses got dropped.**

**None of the tuples from classes got deleted.**

**Spool File: 1b.txt (both before and after)**

**SQL> drop table courses cascade contraints;**

**drop table courses cascade contraints**

**\***

**ERROR at line 1:**

**ORA-00905: missing keyword**

**SQL> drop table courses cascade constraints;**

**Table dropped.**

**SQL> select \* from courses;**

**select \* from courses**

**\***

**ERROR at line 1:**

**ORA-00942: table or view does not exist**

1. (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:

* 1. Show the contents of Courses and Classes.

**Spool File: 2a.txt**

* 1. 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.

**SQL> insert into classes values ('c0008', 'MS', 700, 1, 2021, 'Spring', 10, 9, 'LH 005');**

**insert into classes values ('c0008', 'MS', 700, 1, 2021, 'Spring', 10, 9, 'LH 005')**

**\***

**ERROR at line 1:**

**ORA-02291: integrity constraint (SRONGHE1.SYS\_C00460910) violated - parent key not found**

* 1. Try to insert a tuple into Classes with null values for both dept\_code and course#.

**SQL> insert into classes values ('c0008', null, null, 1, 2021, 'Spring', 10, 9, 'LH 005');**

**1 row created.**

* 1. Try to insert a tuple into Classes with a non-null value for dept\_code and a null value for course#.

**SQL> insert into classes values ('c0009', 'CS', null, 1, 2021, 'Spring', 10, 9, 'LH 005');**

**1 row created.**

* 1. Try to insert a tuple into Classes with a null value for dept\_code and a non-null value for course#.

**SQL> insert into classes values ('c0010', null, 532, 1, 2021, 'Spring', 10, 10, 'LH 005');**

**1 row created.**

* 1. 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).

**SQL> insert into classes values ('c0011', null, 522, 1, 2021, 'Spring', 10, 10, 'LH 005');**

**1 row created.**

* 1. Show the content of Classes again to see what tuples have been inserted.

**Spool File: 2g.txt**

* 1. Answer the question regarding which of (i) and (ii) is adopted by Oracle. Discuss the results – did any result surprise you?

**(ii) is adopted by Oracle where at least one of the values in a and b is null. I got surprised that oracle allowed to have a non-existing non-null value while inserting a new tuple.**

1. 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.
   1. (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.

**Spool File: 3a.txt**

* 1. (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.

**Spool File: 3b.txt**

1. (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.

A picture containing text, whiteboard

Description automatically generated Diagram

Description automatically generated

**Let's look at the two primary and secondary index examples above. If you need to extract a particular tuple for a query, you need to fetch more blocks of memory into main memory because it can't be sorted, as opposed to sorting the primary index, which stores everything in order. You can easily reach the conclusion that there is. Fewer I / O operations and all related tuples are in main memory. This is the main advantage of the primary index over the secondary index.**

1. 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.
2. (10 points) How many levels this B+ tree will have? Compute the number of nodes at each level.

**Total = 6 + 4 = 10 bytes.**

**1638/10 = 163 cells (after fill factor)**

**4000000 requires 24540 pages at bottom level**

**Number of cells at next level = 151 pages**

**So, 151 needs 1 page at each next level.**

1. (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)

**K^2 tuples can be accommodated.**

1. (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 possiblenumber 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.)
2. (5 points) The same question as in (c) except that this time the B+ tree is a secondary index.