Problem 1: Relational Model and SQL

Following are the relational schemas of a SRTP (Student Research Training Program) project database.

    student (<u>sId</u>, sName, dId)
    teacher (<u>tId</u>, tName, dId)
    department (<u>dId</u>, dName)
    project (<u>pId</u>, pName, tId, startTime, endTime)
    participate (<u>pId, sId</u>, role)

The underlined attributes are primary keys, and foreign keys are listed as follows:

    "dId" in "student" references "department";
    "dId" in "teacher" references "department";
    "tId" in "project" references "teacher";
    "pId" and "sId" in "participate" reference "project" and "student", respectively.

In "participate", only two different roles are permitted: "leader" and "member". Based on the above relational schemas, please answer the following questions:

(1) Write a relational algebra expression to find the names of the projects that are instructed by a teacher from the department "Computer Science".

(2) Write SQL statements to create tables project and participate with all the necessary constraints (Note: Tables student, teacher, and department have already been created and can be referenced).

(3) Write a SQL statement to find the names of the teachers that instruct at least one project started in the year 2020.

(4) Write a SQL statement to find the names of the students participating more than 2 projects.

Problem 2: E-R Model

Based on the SRTP project management scenario in Problem 1, some new requirements are added as follows:

(1) There are two kinds of SRTP projects, i.e., school-level projects and national-level projects, and a project is either school-level or national-level.

(2) National-level projects have budget information, and school-level projects have mid-term check information.

(3) A school-level project is associated with exactly a department that is in charge of the management of the project.

Please draw an E-R diagram for the scenario.

Problem 3: Relational Formalization

For relation schema R (A, B, C, D, E, F) with functional dependencies set F = {A->B, A->C, B->C, D->E, D->F, EF->D}. Answer the following questions:

(1) Find all the candidate keys.

(2) Find the canonical cover Fc.

(3) If R is not in BCNF, decompose it into BCNF schemas. Is this decomposition dependency preserving?

Problem 4: B+ -Tree and Query Processing

Table student in Problem 1 is stored sequentially on sId. The following B+-tree is built for the table on attribute dId. Please answer the following questions:
(1) Is the built index a primary index? Why?
(2) Draw the B+-tree after inserting entry 14.
(3) Draw the B+-tree after deleting entry 19 from the original B+-tree.
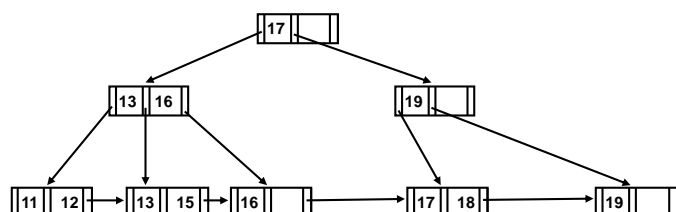
Problem 5: Query Processing

There are two relations r (100 blocks) and s (20 blocks), and hash-join algorithm is used to perform natural join between these two relations (memory size M=6 blocks). Please answer the following questions:
(1) How many partitions can be constructed? Why?
(2) Which relation is best to choose as the build relation? Why?
(3) Is recursive partition needed? Why?
(4) Please compute the cost (numbers of seeks and block transfers) of the hash-join.

Problem 6: Concurrency Control

Given the following schedule, please answer the following questions:
(1) Draw the precedence graph for the schedule.
(2) Is the schedule conflict serializable? Why?
(3) Is it possible that the schedule is generated by the 2PL protocol with lock conversions? Explain.
(4) Which conditions should be satisfied if we want the schedule to be recoverable?

T1	T2	T3
	read C	
read B		
	write C	
	read A	
		read C
	write A	
read A		
		write C
write B		
		read B

Problem 7: Recovery

Given the following log file that supports logical undo, please answer the following questions:
(1) The system crashes just after the last log record. What are the values of B and C in the database after system crash?
(2) Which transactions should redo and undo, respectively?
(3) What are the start and end points for redo and undo, respectively?
(4) What are the log records added during recovery?

1 $<T_0$ start$>$

2 $<T_0$, B, 2000, 2050$>$

3 $<T_1$ start$>$

4 $<T_1$, B, 2050, 2100$>$

5 $<T_1$, O_1, operation-begin$>$

6 $<$checkpoint $\{T_0, T_1\}>$

7 $<T_1$, C, 700, 400$>$

8 $<T_0$ commit$>$

9 $<T_1$, O_1, operation-end, (C, +300)$>$

10 $<T_2$ start$>$

11 $<T_2$, O_2, operation-begin$>$

12 $<T_2$, C, 400, 300$>$

13 $<T_2$, O_2, operation-end, (C, +100)$>$

14 $<T_2$, commit$>$