CSE-6421

Assignment #3

---

1. (15 points) **Join Enumeration.** *Down for the count.* SHORT ANSWER

---

a. [3pt] Name two advantages of left-linear join trees compared with join trees generally.

---

b. [3pt] Name a disadvantage of left-linear join trees when many joins are involved.

Consider a query with five tables, **A**, **B**, **C**, **D**, and **E**, such that there is a join condition between each pair.

c. [3pt] How many left-linear join trees are possible?

d. [3pt] At least how many plans involving three tables are carried forward from stage 3 to stage 4 of System R's join enumeration algorithm?

e. [3pt] At least how many plans will System R have enumerated before choosing the final one?

2. [20pts] **Query Planning & Optimization.** *This is the last time I enrol!* EXERCISE

Schema:

Student(sid, sname, startdate, major, advisor)
FK (advisor) refs **Prof** (pid)
Class(cid, dept, number, section, term, year, room, time, pid, ta)
FK (pid) refs **Prof**
FK (ta) refs **Student** (sid)
Enrol(sid, cid, date, grade)
FK (sid) refs **Student**
FK (cid) refs **Class**
Prof(pid, pname, pdept, office)

Assume no attribute is nullable. The attribute pid in **Class** refers to the the professor / instructor for the class. The attribute ta in **Class** refers to the teaching assistant for the class. The attribute advisor in **Student** refers to the student's academic advisor.

Statistics:

- **Student**: 50,000 records on 1,000 pages
  - advisor: 2,500 distinct values
- **Enrol**: 2,000,000 records on 20,000 pages
  - sid: 50,000 distinct values
  - cid: 80,000 distinct values
- **Class**: 80,000 records on 1,600 pages
  - pid: 4,000 distinct values
  - ta: 5,000 distinct values
- **Prof**: 4,000 records on 40 pages

Indexes:

- **Student**:
  - clustered tree index on sid (200 data entries per page)
- **Enrol**:
  - clustered tree index on cid, sid (167 data entries per page)
  - unclustered tree index on sid, cid (167 data entries per page)
- **Class**:
  - clustered tree index on cid (200 data entries per page)
- **Prof**:
  - clustered tree index on pid (200 data entries per page)

All indexes are of alternative #2. For each tree index, the index pages are 3 deep, except for the index on **Prof**.pid which is 2 deep.

Consider the following query.

```
select sid, sname, dept, number, section, term, year, pid, pname
    from Student S, Enrol E, Class C, Prof P
    where S.sid = E.sid and E.cid = C.cid
      and C.pid = P.pid and S.sid = C.ta;
```

a. [5pt] Estimate the cardinality of the query; that is, how many records are returned.

b. [12pt] Devise a good query plan for the query. Show the query tree, *fully* annotated with the chosen algorithms and access paths.

You have an allocation of 50 buffer-pool frames.

Estimate the cost of your plan. For full credit, you should have a plan that costs less than 20,000 I/O's.

c. [3pt] For Question 2b, if we additionally had an unclustered tree index for **Class** on cid, ta, would a better (less expensive) query plan be possible? Why or why not?

3. [5pt] **Index Usage.** *The DBA playoffs.* Analysis

Consider the following schema.

Employee(e#, name, salary, d#)
FK (d#) refs **Department**
Department(d#, name, location, budget)

It is important that the following queries be fast to be evaluated.

A. Find the location where a user-specified employee works.

B. List all the departments such that the sum of the employees' salaries who belong to the department exceeds the department's budget.

What indexes would you make to benefit queries **A** and **B**? Indicate which are clustered and their types (B+ tree, hash).

Briefy explain how they would be useful.