

Hw 7 Solution

For each SQL query given below use the attached relational diagram to help you do the following:

- Write out the corresponding relational algebra expression. Your expression must match the SQL query given (i.e. do not change a join to a Cartesian product).
- Use this expression to create a matching query parse tree. **Do not optimize this parse tree.**
- Assume every primary key has a primary index. List what primary indexes, if any, will speed up the query and list what part of the query they will help. Do not assume your database will use indexes on partial keys when the primary key is composite (e.g. if the primary key is {a, b} do not assume there is an index on just a)
- List what secondary indexes, if any, you could add to speed up the query and list what part of the query they will help.

Submit a single PDF file to canvas. Your submission must be typed and drawn with software using correct symbols -- no hand drawn submissions will be given credit. Verify your upload after submission -- corrupted or incorrect files will not get credit.

```
1)
SELECT DISTINCT ID, first, last
FROM PROFESSOR join SECTION on ID = prof_ID
WHERE room_num = 'RVR1008';
```

- $\pi_{ID, first, last}(\sigma_{room_num='RVR1008'}(PROFESSOR \bowtie_{ID=prof_ID} SECTION))$
-



c) Assuming that we read PROFESSOR first, no primary index on SECTION will help.

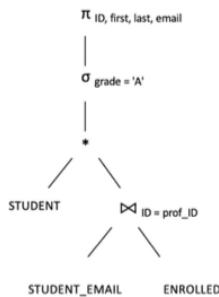
d) Assuming that we read PROFESSOR first, a secondary index on SECTION.prof_ID will help with the join.

```
2)
SELECT DISTINCT ID, first, last, email
FROM STUDENT natural join STUDENT_EMAIL
```

```
JOIN ENROLLED on ID = student_id
WHERE grade = 'A';
```

a) $\pi_{ID, first, last, email}(\sigma_{grade='A'}(STUDENT * STUDENT_EMAIL \bowtie_{ID=student_ID} ENROLLED))$

b)



c) Note: the natural join will be on ID. If STUDENT_EMAIL is read first, no primary index will help with ENROLLED. The primary index of STUDENT will help with the natural join.

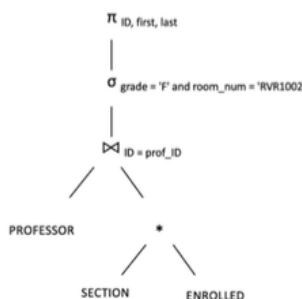
d) If STUDENT_EMAIL is read first, a secondary index on `ENROLLED.prof_ID` will help. STUDENT will use the primary index.

3)
 SELECT DISTINCT ID, first, last
 FROM PROFESSOR join SECTION on ID = prof_ID
 NATURAL JOIN ENROLLED
 WHERE grade = 'F' and room_num = 'RVR1002';

a)

$\pi_{ID, first, last}(\sigma_{grade='F' \text{ and } room_num='RVR1002'}(PROFESSOR \bowtie_{ID=prof_ID} SECTION * ENROLLED))$

b)



c) (note: the natural join is on catnum, sectnum, and semester) If SECTION is read first, the primary index of ENROLLED will not help. The primary index of PROFESSOR will help with the equijoin.

d) If SECTION is read first, a secondary index on `ENROLLED(catnum, sectnum, semester)` will help. PROFESSOR is using a primary index, so no secondary index is needed.

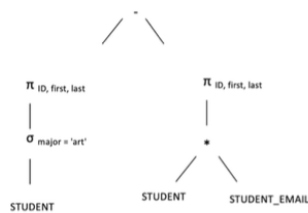
```

4)
(SELECT ID, first, last
FROM STUDENT
WHERE major = 'art')
EXCEPT
(SELECT ID, first, last
FROM STUDENT NATURAL JOIN STUDENT_EMAIL);

```

a) $\pi_{ID, first, last}(\sigma_{major = 'art'}(STUDENT)) - \pi_{ID, first, last}(STUDENT * STUDENT_EMAIL)$

b)



c) (note: the natural join is on ID) If the right-hand STUDENT is read first, the primary index on STUDENT_EMAIL will not help.

d) For the left-hand STUDENT a secondary index on major will help. If the right-hand STUDENT is read first, a secondary index on `STUDENT_EMAIL(ID)` will help.