

[\[Instructions\]](#) [\[Notes\]](#) [\[PostgreSQL\]](#) [\[C\]](#)  
[\[Q1\]](#) [\[Q2\]](#) [\[Q3\]](#) [\[Q4\]](#) **[\[Q5\]](#)** [\[Q6\]](#) [\[Q7\]](#) [\[Q8\]](#)

### Question 5 (9 marks)

Consider the following database schema:

```
create table R (  
    id      integer primary key,  
    rval    text unique,  
    tid     integer -- references T(id)  
);  
create table S (  
    id      integer primary key,  
    sval    text,  
    rid     integer references R(id)  
);  
create table T (  
    id      integer primary key,  
    tval    text,  
    sid     integer references S(id)  
);  
alter table R add foreign key (tid) references T(id);
```

and the following join query on this schema

```
select * from R, S, T  
where  R.id = S.rid and S.id = T.sid and T.id = R.tid  
       and R.rval = 'abc' and S.sval < 'xyz';
```

The query optimiser would represent the query as a tree of join operations, and try variations on the order of applying the joins. You may assume that the query optimiser will push the selections below the appropriate join. You should also assume that the left relation in each join is always treated as the "outer relation" when evaluating the join, so that  $(R \text{ join } S)$  has a different execution behaviour from  $(S \text{ join } R)$  even though they produce the same result.

- Show at least three different ways of accomplishing the above joins. Write your answer in the form  $X \text{ join } (Y \text{ join } Z)$  where the join condition is implied and the parentheses force the order in which the joins are carried out. There is no need to show selection operations; assume that they are carried out before any join operations.
- Based on the answer given in (a), along with any other assumptions that you think are necessary, which join order is likely to lead to the most efficient query evaluation? Explain your answer. You should include the costs and benefits of the selection operations in your discussion. You can assume that the selection condition on  $S$  returns only half of the tuples in the relation.

- c. For the join tree you selected in part (b), write a series of relational assignment statements. Each assignment statement should have the name of a temporary relation on the left-hand side, and a *single* relational algebra operation on the right-hand side.

Show all working.

**Instructions:**

- Type your answer to this question into the file called q5.txt
- Submit via: **give cs9315 sample\_q5 q5.txt**  
or via: Webcms3 > exams > Sample Exam > Submit Q5 > Make Submission

*End of Question*

**Assignment Project Exam Help**

**<https://powcoder.com>**

**Add WeChat powcoder**