

The assignment is to be turned in before Midnight (by 11:59pm) on February 8 , 2018. You should turn in the solutions to this assignment as a pdf file through the TEACH website. The solutions should be produced using editing software programs, such as LaTeX or Word, otherwise they will not be graded.

1: Query optimization (2 points)

Consider the following relations:

Product (name, production-year, rating, company-name)

Company (name, state, employee-num)

Assume each product is produced by just one company, whose name is mentioned in the *company-name* attribute of the *Product* relation. Attributes *name* are the primary key for both relations *Product* and *Company*. Attribute *rating* shows how popular a product is and its values are between 1-5.

The following query returns the products with rating of 5 that are produced after 2000 and the states of their companies.

```
SELECT p.name, c.state
FROM Product p, Company c
WHERE p.company-name = c.name and p.production-year > 2000 and p.rating = 5
```

Suggest an optimized logical query plan for the above query.

The optimized logical query plan for the above query is the following. The plan first selects table *Product* according to the conditions on *production-year* and *rating*. It then projects *Product* on *company-name* and *p.name*. It also projects table *Company* on attributes *c.name* and *state*. Finally, it joins the projected *Product* and *Company* relations and projects the resulting table on attributes *p.name* and *state*.

2: Query optimization (2 point)

For the four base relations in the following table, find the best join order according to the dynamic programming algorithm used in System-R. You should give the dynamic programming table entries for evaluate the join orders. Suppose that we are only interested in left-deep join trees and join trees without Cartesian products. Note that you must use the Selinger-style formulas to compute the size of each join output. The database system uses hash join to compute every join. Assume that there is enough main memory to perform the hash join for every pairs of relations. Each block contains at most 5 tuples of a base or joint relation. If relations are joined on multiple attributes, you can compute the selectivity factor of the full join by multiplying the selectivity factors of joining on each attribute.

R(A,B,C)	S(B,C)	W(B,D)	U(A,D)
T(R)=4000	T(S)=3000	T(W)=2000	T(U)=1000
V(R,A) =100 V(R,B) =200 V(R,C) =100	V(S,B) =100 V(S,C) = 300	V(W,B) =100 V(W,D) =50	V(U,A) =100 V(U,D) =100

3: Concurrency control (3 points)

Consider the following classes of schedules: serializable and 2PL. For each of the following schedules, state which of the preceding classes it belongs to. If you cannot decide whether a schedule belongs to a certain class based on the listed actions, explain briefly your reasons.

The actions are listed in the order they are scheduled and prefixed with the transaction name. If a commit or abort is not shown, the schedule is incomplete; assume that abort or commit must follow all the listed actions.

1. T1:R(X), T2:R(Y), T3:W(X), T2:R(X), T1:R(Y)
2. T1:R(X), T1:R(Y), T1:W(X), T2:R(Y), T3:W(Y), T1:W(X), T2:R(Y)