

The assignment is to be turned in before Midnight (by 11:59pm) on March 14. You should turn in the solutions to this assignment as a PDF file through Canvas. The solutions should be produced using editing software programs, such as LaTeX or Word, otherwise they will not be graded. The assignment should be done in groups of two students. Each group must submit only one file that contains the full name, OSU email, and ONID of every member of the group.

---

### 1: Query optimization (2 points)

---

Consider the following relations:

R(A, B, C, D, E)

S(E, D)

T(G, B, D, H)

U(I, J, K)

V(L, J, M)

W(L, J, N)

Suggest an optimized logical query plan for the above query and explain why your proposed plan may be faster than other possible plans. You should find the best guess(es) for the join order in your plan without knowing the statistics of the join attributes and base relations.

```
SELECT B,C
```

```
FROM R, S, T, U, V, W
```

```
WHERE R.D = S.D and R.D = T.D
```

```
and R.B = T.B and U.J <= V.J and U.J = W.J and R.E <= 200 and W.N <= 100
```

Please note that your answer may not always be more efficient than other plans, but it should run faster than other plans for most input relations.

---

### 2: Query optimization (4 points)

---

For the four 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 evaluating the join orders. The cost of each join is the number of I/O accesses the database system performs to execute the join. Assume that the database system uses the two-pass sort-merge join algorithm to perform the join operations. Each block contains 4 tuples and tuples of all relations have the same size. We are interested only in left-deep join trees. Note that you should use the System-R optimizer formula to compute the size of each join output.

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: Query optimization (2 points)**

---

What is the time-complexity of the dynamic programming algorithm you used in question (3) of this assignment? You should explain your answer.

---

**4: Serializability and 2PL (5 points)**

---

(a) 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 in a certain class based on the listed actions, explain briefly. Also, for each 2PL schedule, identify whether a cascading rollback (abort) may happen. A cascading rollback will happen in a schedule if a given transaction aborts at some point in the schedule, and at least one other transaction must be aborted by the system to keep the database consistent. (4 points)

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)
3. T1:W(X), T2:R(X), T1:W(X)
4. T1:R(X), T2:W(X), T1:W(X), T3:R(X)

(b) Consider a database DB with relations R1 and R2. The relation R1 contains tuples t1 and t2 and the relation R2 contains tuples t3, t4, and t5. Assume that the database DB, relations, and tuples form a hierarchy of lockable database elements. Explain the sequence of lock requests and the response of the locking scheduler to the following schedule. You may assume all lock requests occur just before they are needed, and all unlocks occur at the end of the transaction, i.e., EOT. (1 point)

- T1:R(t1), T2:W(t2), T2:R(t3), T1:W(t4)

---

**5: Degrees of Consistency (4 points)**

---

(a) Consider the schedule shown in Table 1.

What are the maximum degrees of consistency for T1 and T2 in this schedule? You must find the maximum degrees of consistency for T1 and T2 that makes this schedule possible. (2 points)

(b) Consider a transaction that reads the information about a set of accounts from a CSV file and writes them in a database. What degree of consistency will you choose for this transaction?

	T1	T2
0	start	
1	read X	
2	write X	
3		start
4		read X
5		write X
6		Commit
7	read Y	
8	write Y	
9	Commit	

Table 1: Transaction schedule

You should justify your answer. Next, consider another transaction in a banking system that reads the balances of all bank accounts in a branch and computes their average. What degree of consistency will you choose for this transaction? You should justify your answer. (2 points)

---

### 6: Serializability (6 points)

---

Consider the following protocol for concurrency control. The database system assigns each transaction a unique and strictly increasing id at the start of the transaction. For each data item, the database system also keeps the id of the last transaction that has modified the data item, called the transaction-id of the data item. Before a transaction  $T$  wants to read or write on a data item  $A$ , the database system checks whether the transaction-id of  $A$  is greater than the id of  $T$ . If this is the case, the database system allows  $T$  to read/write  $A$ . Otherwise, the database system aborts and restarts  $T$ .

- (a) Does this protocol allow only serializable schedules for transactions? If not, you may suggest a change to the protocol so that all schedules permitted by this protocol are serializable. You should justify your answer. (3 points)
- (b) Propose a change to this protocol or the modified version you have designed for part (a) that increases its degree of concurrency, i.e., it allows more serializable schedules. (3 points)