**Assignment 2**

**CS540- Advanced Database Management Systems**

**Arash Termehchy**

**Question 1:**

Using B+ trees with degrees of two (2) whose keys are integer values, give an example of a B+ tree whose height changes from 2 to 3 when the value 25 is inserted. Show your structure before and after the insertion. A B+ tree with a single node has height of 1.

**Answer:**

Before Inserting:

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | 13 |  | 18 |  | 24 |  | 40 |  |

2,3,5,7 🡪 13,15,16,17 🡪 18,20,21,22 🡪 24,28,35,39 🡪40,42,43,44

After Inserting 25:

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | 13 |  | 18 |  | 24 |  | 40 |  |

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | 28 |  |  |  |  |  |  |  |

2,3,5,7 🡪13,15,16,17 🡪18,20,21,22🡪 🡪40,42,43,44

24,25,\_,\_ 🡪 28,35,39,\_

**Question 2**:

Consider the join of the relation R and S on attributes R.a=S.b, given the following information about the relations to be joined. Relation R contains 10,000 tuples and has 10 tuples per block. Relation S contains 2000 tuples and also has 10 tuples per block. Attribute b of relation S is the primary key for S. Neither relation has any indexes built on it. 52 buffer pages are available in main memory. What is the cost of joining R and S using a sort-merge join? You should use a version of sort-merge algorithm that provides the minimum cost. The cost metric is the number of block I/Os.

**Answer**:

B(r) = 10,000/10 = 1000

B(s) = 2000/10 = 200

M = 52

Total cost = sorting + 2B(r)+2B(s) – B(r)

Sorting cost(r) = No. of passes\*2\*B(r)

Sorting cost(s) = No. of passes\*2\*B(s)

No. of passes = 1+ log [B(r)/M] to the base M-1

Sorting Cost(r) = 2\*1000\*ceil of [1+log20 to base 51]

= 2000\*[1+1] = 4000

Sorting Cost(s) = 2\*200\*ceil of [1+log4 to base 51]

= 400\*[1+1] = 800

Total cost = 2000+400+800+4000-1000

=6200

Optimized:

When B(r) + B(s) <= M2

Total Cost = 3.B(r) + 3.B(s)

1000+200<= 52\*52

1200 < 2704 🡪 True

Total Cost = 3\*1000 + 3 \*200

= 3600 🡪 Lowest I/O Cost

**Question 3**:

Consider the following relational schema and SQL query: Suppliers (sid, sname, city), Supply (sid, pid), Parts(pid, pname, price). SELECT S.sname, P.pname FROM Suppliers S, Parts P, Supply Y WHERE S.sid = Y.sid AND Y.pid = P.pid. How many different join orders, assuming that cross products are disallowed, does a System R style query optimizer consider when deciding how to process the given query? List the join orders.

**Answer**:

The only possible left deep join orders are:

1. (Suppliers X Supply) X Parts

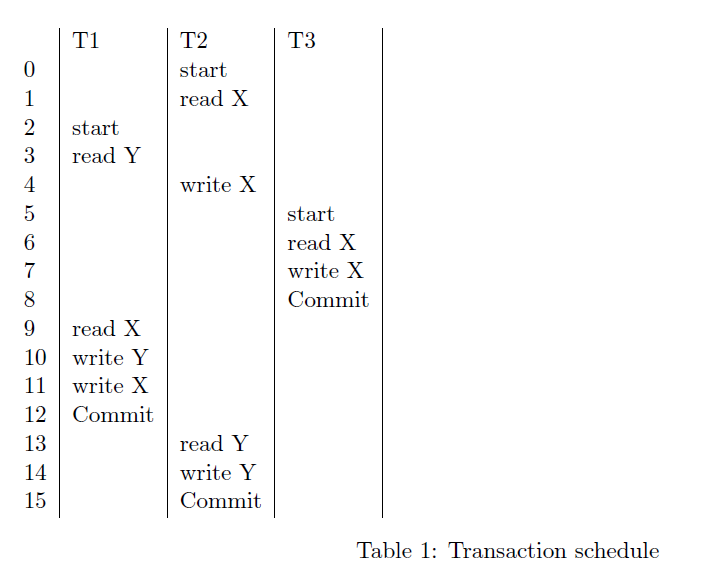
2. (Supply X Parts) X Suppliers

**Question 4:**

Consider the schedule shown at Table 1.

(a) What is the equivalent serialization order for this schedule? If no order is possible, you may state so.

(b) When all transactions run in the above schedule, identify the transactions with degree 3 consistency. Answer the same question when transaction T3 does not exist in the schedule.



**Answer**:

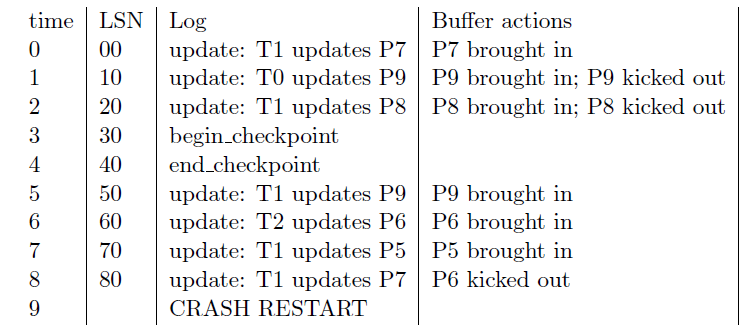
1. There is no Serialization schedule possible. Because there is a loop in the Serialization Graph from T1 to T2.

T2 T3 T1

1. T2 can be of degree 3 consistency because If we apply locks on T2, T2 holds long shared read locks on X and long exclusive write locks on X which will prevent other transactions from dirty any data.

**Question 5**:

In this problem, you need to simulate the actions taken by ARIES. Consider the following log records and buffer actions:



1. For the actions listed above, show Transaction Table (XT) and Dirty Page Table (DPT) after each action. Assume that DPT holds pageID and recLSN, and XT contains transID and lastLSN.
2. Simulate Analysis phase to reconstruct XT and DPT after crash. Identify the point where the Analysis phase starts scanning log records and show XT and DPT after each action.
3. Simulate Redo phase: First identify where the Redo phase starts scanning the log records. Then, for each action identify whether it needs to be redone or not.
4. Simulate Undo phase: identify all actions that need to be undone. In what order will they be undone?

**Answer**:

1. **Transaction Table** **Dirty Page Table**

**Trans ID** **Last LSN** **Page ID** **rec LSN**

Step-0 T1 00 P7 00

Step-1 T1 00 P7 00

T0 10 P9 10 Kicked Out

Step-2 T1 20 P7 00

T0 10 P8 20 Kicked Out

Step-3 Start CheckPoint

Step-4 End Checkpoint

Step-5 T1 50 P7 00

T0 10 P9 50

Step-6 T2 60 P6 60

T1 50 P7 00

T0 10 P9 50

Step-7 T1 70 P5 70

T2 60 P6 60

T0 10 P9 50

P7 00

Step-8 T1 80 P5 70

T2 60 P9 50

T0 10 P7 00

1. Analysis Phase: We start from Last checkpoint and check which transactions committed/failed since checkpoint.

Step-3 Start CheckPoint

Step-4 End Checkpoint---- Reconstruct the state at start check point

T1 20 P7 00

T0 10

Step-5 T1 50 P7 00

T0 10 P9 50

Step-6 T2 60 P6 60

T1 50 P7 00

T0 10 P9 50

Step-7 T1 70 P5 70

T2 60 P6 60

T0 10 P9 50

P7 00

Step-8 T1 80 P5 70

T2 60 P9 50

T0 10 P7 00

1. REDO Phase: Start from Smallest recLSN in Dirty Page Table

Transaction Table Dirty Page Table

T1 80 P5 70

T2 60 P9 50

T0 10 P7 00

Step-0 Redo T1 00 because recLSN of P7 <= LSN

Step-1 T1 00

No Redo T0 10 because recLSN of P9 in DPT i.e. 50 > LSN i.e 10

Step-2 No Redo T1 20 because P8 not in Dirty Page Table after Analysis

T0 00

Step-3 Begin Checkpoint

Step-4 End checkpoint

Step-5 Redo T1 50 because recLSN of P9 <= LSN

T0 00

Step-6 No Redo T2 60 because P6 not in Dirty Page Table after Analysis.

T1 50

T0 00

Step-7 Redo T1 70 because recLSN of P5 <= LSN

T0 00

Step-8 Redo T1 80 because recLSN of P7<=LSN

T0 00

1. UNDO Phase: Start from end, Undo all Transactions active during crash.

Step-8 Undo T1 80 write CLR in Log

Step-7 Undo T1 70 write CLR in Log

Step-6 Undo T2 60 write CLR in Log, end T2

T1 70

Step-5 Undo T1 50 write CLR in Log

T2 60

Step-4 End Checkpoint

Step-3 Start Checkpoint

Step-2 Undo T1 20 write CLR in Log

T2 60

Step-1 Undo T0 10 write CLR in Log, end T0

T1 20

T2 60

Step-0 Undo T1 00 write CLR in Log, end T1

T0 10

T2 60

**Submitted By**:

Lakshman Madhav Kollipara

ONID Name-kollipal