**CS 411 Database Systems Homework 4**

**Spring 2023**

**Submitted by: Payal Mantri (payaljm2)**

1. **Searching B+ Trees**
   1. SELECT \* FROM table WHERE ID = 12   
      **Solution : B0, B1, B3**
   2. SELECT \* FROM table WHERE ID <= 41   
      **Solution : B0, B1, B3, B4, B5**
   3. SELECT \* FROM table WHERE ID > 68

**Solution : B0, B2, B6, B7, B8**

* 1. SELECT \* FROM table WHERE ID > 80 AND ID < 120   
     **Solution: B0, B2, B7, B8**

1. **Isolation Level and Locking**
   1. **T1: Repeatable Read T2: Read Committed**

**S1:** *SLOCK1(A); R1(A); SLOCK2(C); R2(C); REL2(C); SLOCK1(B); R1(B); XLOCK1(C); W1(C); SLOCK2(A); R2(A); REL2(A); XLOCK2(B); DENIED2(B); R1(A); R1(B); REL1(A, B, C); SLOCK2(A); R2(A); REL2(A); XLOCK2(B); W2(B); REL2(B)*

**S2:** *SLOCK2(A); R2(A); REL2(A); SLOCK1(B); R1(B); SLOCK2(C); R2(C); REL2(C); SLOCK1(A); R1(A); XLOCK1(C); W1(C); REL1(A, B, C); SLOCK2(C); R2(C); REL2(C); XLOCK2(B); W2(B); SLOCK2(A); R2(A); REL2(A, B);*

* 1. **T1: Repeatable Read T2: Repeatable Read**

**S1:** *SLOCK1(A); R1(A); SLOCK2(C); R2(C); SLOCK1(B); R1(B); XLOCK1(C); DENIED1(C); SLOCK2(A); R2(A); XLOCK2(B); DENIED2(B); R1(A); R1(B); R2(A).*

*Now W1(C) and W2(B) are in queue but it is a deadlock state as T1 is waiting for T2 to finish and release shared lock on C SLOCK2(C) and T2 is waiting for T1 to finish and release shared lock on B SLOCK1(B).  
Hence this is infeasible*

**S2:** *SLOCK2(A); R2(A); SLOCK1(B); R1(B); SLOCK2(C); R2(C); SLOCK1(A); R1(A); XLOCK1(C); DENIED1(C); R2(C); XLOCK2(B); DENIED2(B); R2(A);*

*W1(C) ; W2(B) – these are remaining in end of queue and lead to deadlock state as T1 is waiting for T2 to finish and release shared lock on C SLOCK2(C) and T2 is waiting for T1 to finish and release shared lock on B SLOCK1(B).*

*Hence this is infeasible*

|  |  |
| --- | --- |
| Transaction1 T1 | Transaction2 T2 |
| START TRANSACTION | START TRANSACTION |
|  | **SELECT \* FROM Temperatures WHERE cityName=’Portland’;**  **Result:**   |  |  |  | | --- | --- | --- | | tempId | **temperature** | **cityName** | | **1** | **50** | **Portland** | |
| INSERT INTO Temperatures (tempId, temperature, cityName) VALUES (10001, 60, ‘Portland’); |  |
| COMMIT; |  |
|  | **SELECT \* FROM Temperatures WHERE cityName=’Portland’;**  **Result:**   |  |  |  | | --- | --- | --- | | tempId | **temperature** | **cityName** | | **1** | **50** | **Portland** | | **10001** | **60** | **Portland** | |

1. **Deletion from a B+ Tree**
   1. **Deletion of 40 – traverse B0->B1->B4  
        
      Diagram, engineering drawing

      Description automatically generated**
   2. **Remove 69- Traverse B0->B1->B6  
      Diagram, engineering drawing

      Description automatically generated  
      Since B6 has only one value , it will borrow from right sibling  
        
      After Rotation  
        
      Diagram, engineering drawing

      Description automatically generated**
   3. **Remove 18 from B3  
      Diagram, engineering drawing

      Description automatically generated  
        
      Delete 35 from B4**Diagram, engineering drawing

      Description automatically generated **Delete 40 – B4 becomes unbalanced**Diagram, engineering drawing

      Description automatically generated **Rotation not possible. So Nodes B4 and B5 are merged.  
      Parent Node is updated**Diagram, engineering drawing

      Description automatically generated

**Now B1 is unbalanced. Hence merge B1, B0, B2**

**Diagram

Description automatically generated**

1. **Extensible Hash Table**
   1. **Insert 0001 – Step 1: increase buckets i=2  
      Diagram

      Description automatically generated  
        
      Step 2: Insert 0001  
      Diagram

      Description automatically generated**
   2. **Insert 0111  
      Diagram

      Description automatically generated**
   3. **Insert 1110  
      Diagram

      Description automatically generated  
        
      Need to add a new block   
      Diagram

      Description automatically generated**
   4. **Insert 0101  
        
      Diagram

      Description automatically generated  
        
        
      Need to extend the buckets as block is full, make i=3**Diagram

      Description automatically generated
2. **Linear HashTable**

* **Insert 11000**

**r = 5 n=3 ; r/n = 5/3 = 1.67> 1.65   
Hence introduce a new bucket 11 . Rearrange the entries from MSB flipped bucket (01) and add the new entry 11000 in 00 bucket as overflow block .  
  
  
  
Diagram

Description automatically generated**

* **Insert 00101   
  r=6 n=4; r/n = 6/4= 1.5 <1.65 . Hence directly insert in 01 bucket**

**Diagram

Description automatically generated**

* **Insert 00001  
  r=7; n=4; r/n = 7/4 =1.75 >1.65  
  hence need to extend bucket array to add a new bucket 100. Rearrange entries from 000 - move 00100 from 000 to 100 bucket.**

**Now insert 00001 to bucket 001**

Diagram

Description automatically generated

* **Insert 01111  
  r=8 n=5; r/n = 8/5 = 1.6 < 1.65  
  As bucket 111 isn’t present , flip MSB to get 011 and now insert it in bucket 011**

Diagram

Description automatically generated