**LAB 8**

Submitted by: Simran

100377444

1. The lost update anomaly is said to occur if a transaction Tj reads a data item, then another transaction Tk writes the data item (possibly based on a previous read), after which Tj writes the data item. The update performed by Tk has been lost, since the update done by Tj ignored the value written by Tk . [5+5 Marks]

a. Give an example of a schedule showing the lost update anomaly.

ANS: Lost update anomaly will occur whenT1 reads then T2 writes that data item after which T1writes it as shown below and because of which the changes made by T2 are lost.

|  |  |
| --- | --- |
| T1 | T2 |
| read\_item(X);  X:=X-N;  Write\_item(X); | Read\_item(X);  X:=X+M  Write\_item(X); |

b. Give an example schedule to show that the lost update anomaly is possible with the read committed isolation level.

ANS: Lost Update anomaly is possible with read commited isolation as well like shown below in table because the changes made in values of Y by T5 are lost due to write of T3.

|  |  |
| --- | --- |
| T3 | T5 |
| lock(Y)  read\_lock(Y)  unlock(Y)  lock(Y)  write\_lock(Y)  unlock(Y)  commit | lock(Y)  read\_lock(Y)  write\_lock(Y)  unlock(Y)  commit |

2. You must explain how you reach to an answer. Writing a number without explanation is not accepted.

• Consider a disk with block size B = 512 bytes.

• A block pointer is P = 6 bytes long, and a record pointer is RP = 7 bytes long.

• A file has r = 30,000 EMPLOYEE records of fixed length.

• Each record has the following fields: Name (30 bytes), SSN (9 bytes), Department\_code (9 bytes), Address (40 bytes), Phone (10 bytes), Birth\_date (8 bytes), Sex (1 byte), Job\_code (4 bytes), and Salary (4 bytes, real number).

• An additional byte is used as a deletion marker.

a) Calculate the record size R in bytes. (A record is composed of fields. A field holds information about an entity. For example, Name and SSN are two fields of an employee’s record)

ANS: As record is collection of fields so the record size will be the sum of size of all the fields:

Record size(R) = 30 +9 + 9+ 40 +10 + 8 + 1+ 4 +4 = 155 bytes

b) Calculate the blocking factor bfr and the number of file blocks b, assuming an un-spanned organization.

ANS: Blocking factor(bfr) for an un-spanned organization is

Bfr = (B/R) = (512/155) = 4 records per block.

Now, number of file blocks b = (r/bfr) = (30000/4) = 7500 blocks.

c) Suppose that the file is ordered by the key field SSN and we want to construct a primary index on SSN.

Calculate:

1. the index blocking factor bfri (which is also the index fan-out fo);

ANS: To find bfri, we first need Ri. Now, ordering key field is SSN = 9 bytes and block pointer is P= 6 bytes.

So, size of each index entry= Ri = 9+6 = 15 bytes.

And, bfri = (B/RI) = 512/ 15 = 34 entries per block.

1. the number of first-level index entries and the number of first-level index blocks;

ANS: The total number of first-level index entries r1  = number of blocks in the file = 7500

The number of first level index blocks = b1 = (r1/bfri) = (7500/34) = (220.6) = 221 blocks

1. the number of levels needed if we make it into a multilevel index;

ANS: Now, to check that we need to check it step by step that is

The total number of first-level index entries r2  = The number of first level index blocks b1  = 221

The number of 2nd level index blocks b2 = ceil(r2/bfri  ) = ceil(221/34) = ceil(6.5) =7 blocks.

The number of 3RD level index entries r3 = The number of 2ND level index blocks b2 = 7

The number of 3rd level index blocks b3 = ceil(r3/bfri  ) = ceil(7/34) = ceil(0.2) =1 block.

Now, as 3rd level block have only 1 block so, this index have only three levels.

1. the total number of blocks required by the multilevel index;

ANS: bi = b1 +b2 +b3 = 221 + 7+1 = 229 blocks

1. The number of block accesses needed to search for and retrieve a record from the file—given its SSN value—using the one-level primary index and multilevel index.

ANS: Number of block accesses needed to search for a record = number of blocks +1 = 3+1= 4

3. Consider the following relational schema. An employee can work in more than one department; the pct time field of the Works relation shows the percentage of time that a given employee works in a given department.

Emp(eid: integer, ename: string, age: integer, salary: real) Works(eid: integer, did: integer, pct time: integer) Dept(did: integer, budget: real, managerid: integer)

Write SQL integrity constraints (domain, key, foreign key, or CHECK constraints) and SQL triggers to ensure each of the following requirements, considered independently.

1. Employees must make a minimum salary of $1000.

ANS:

CREATE TABLE Emp( eid INT PRIMARY KEY,

ename VARCHAR(155),

age INT,

salary REAL,

CHECK(salary>=1000)

)

1. Every manager must be also be an employee.

ANS:

CREATE Dept(did INT PRIMARY KEY,

budget REAL,

managerid INT,

CHECK( (COUNT(managerid)

FROM Dept

WHERE managerid NOT IN

(SELECT eid FROM Emp) = 0)

)

1. The total percentage of all appointments for an employee must be under 100%.

ANS: CREATE TABLE Works(eid INT,

did INT,

pct\_time INT,

PRIMARY KEY(eid, did),

CHECK( (SELECT COUNT(Works.eid)

FROM Works

GROUP BY Works.eid

HAVING

Sum(Works.pct\_time) > 100) =0

)

1. A manager must always have a higher salary than any employee that he or she manages.

ANS: CREATE FUNCTION ManagerHighSalEmp

RETURNS salary AS $$

BEGIN

CHECK (SELECT E.eid

FROM EMP E, Emp Mng, Works W, Dept D

WHERE E.eid = W.eid

AND W.did = D.did

AND D.managerid = Mng.eid

AND E.salary < Mng.salary

)

1. Whenever an employee is given a raise, the manager’s salary must be increased by the same percentage.

ANS:

CREATE TRIGGER IncSalMng

AFTER UPDATE ON Emp

WHEN old.salary < new.salary

FOR EACH ROW

BEGIN

UPDATE Emp Mng

SET Mng.Salary = new.salary

WHERE Mng.salary < new.salary

AND Mng.eid IN (SELECT Dept.managerid

FROM Emp , Works , Dept

WHERE Emp.eid = new.eid

AND Emp.eid = Works.eid

AND Works.did = Dept.did);

END