Implementation of Databases
Assignment 7

Participants:
(sorted in last name order)
Ulfet CETIN
Shreya KAR
Samuel ROY

Assignment 7

Exercise 7.1 (Serializability)

1. Is conflict serializability guaranteed? Why or why not?

Ans. Yes, it is guaranteed because schedules are Two Phase Locking.

2. Is cascading rollback possible? If not, explain why not?

Ans. Not possible since it Transactions follow Strict Two Phase Locking.

3. Is deadlock possible?

Ans. Yes Deadlock is possible and T2 would roll back.

4. Explain Dirty Read and Phantom Read with example?

Ans. **Dirty read** occurs when one transaction is changing the record, and the other transaction can read this record before the first transaction has been committed or rolled back. This is known as a dirty read scenario because there is always the possibility that the first transaction may rollback the change, resulting in the second transaction having read an invalid data.

Example:

Transaction A begins.

UPDATE EMPLOYEE SET SALARY = 10000 WHERE EMP_ID= '123';

Transaction B begins.

SELECT * FROM EMPLOYEE;

(Transaction B sees data which is updated by transaction A. But, those updates have not yet been committed.)

Phantom read occurs where in a transaction execute same query more than once, and the second transaction result set includes rows that were not visible in the first result set. This is caused by another transaction inserting new rows between the execution of the two queries. This is like a non-repeatable read, except that the number of rows is changed either by insertion or by deletion.

Example:

Transaction A begins.

SELECT * FROM EMPLOYEE WHERE SALARY > 10000 ;

Transaction B begins.

INSERT INTO EMPLOYEE (EMP_ID, FIRST_NAME, DEPT_ID, SALARY) VALUES (‘111’, ‘Jamie’, 10, 35000);

Transaction B inserts a row that would satisfy the query in Transaction A if it were issued again.

Exercise 7.4

5. **Which synchronization problem can be avoided by index locking?**

Ans. Phantom Problem can be avoided by index locking.

6. **Describe briefly the 3 phases of ARIES recovery method. What are log sequence numbers (LSNs) in ARIES? How are they used?**

Ans. Algorithms for Recovery and Isolation Exploiting Semantics, or ARIES is a recovery algorithm designed to work with a no-force, steal database approach.

The ARIES recovery procedure consists of three main steps:

Analysis

The analysis step identifies the dirty (updated) pages in the buffer and the set of transactions active at the time of the crash. The appropriate point in the log where the REDO operation should start is also determined

REDO

The REDO phase reapplies updates from the log to the database. Generally, the REDO operation is applied to only committed transactions. However, in ARIES, this is not the case. Certain information in the ARIES log will provide the start point for REDO, from which REDO operations are applied until the end of the log is reached. In addition, information stored by ARIES and in the data pages will allow ARIES to determine whether the operation to be redone has actually been applied to the database and hence need not be reapplied. Thus only the necessary REDO operations are applied during recovery.

UNDO

During the UNDO phase, the log is scanned backwards and the operations of transactions that were active at the time of the crash are undone in reverse order. The information needed for ARIES to accomplish its recovery procedure includes the log, the Transaction Table, and the Dirty Page Table. In addition, check pointing is used. These two tables are maintained by the transaction manager and written to the log during check pointing.

Log sequence number: It refers to a pointer used to identify the log records. As with any record id, we can fetch a log record with one disk access given the LSN. Further, LSNs should be assigned in increasing order; this property is required for the ARIES recovery algorithm. If the log is a sequential file, in principle growing indefinitely, the LSN can simply be the address of the first byte of the log record'