K. J. Somaiya College of Engineering, Mumbai-77
(Autonomous College Affiliated to University of Mumbai)

Batch: A2	**Roll No.: 1911027**
Experiment / assignment / tutorial No. 10	
Grade: AA / AB / BB / BC / CC / CD /DD	

Title: Implementation of Concurrency Control Protocols

Objective: To understand Transaction, Transaction Control Protocols and its implementation.
Implement Lock based protocol.

Expected Outcome of Experiment:
CO 5: Formulate and demonstrate the transaction, concurrency and recovery techniques

Books/ Journals/ Websites referred:

1. Dr. P.S. Deshpande, SQL and PL/SQL for Oracle 10g.Black book, Dreamtech Press
2. www.db-book.com
3. Korth, Slberchatz, Sudarshan : "Database Systems Concept", 5th Edition , McGraw Hill
4. Elmasri and Navathe,"Fundamentals of database Systems", 4th Edition,PEARSON Education.
5. https://dev.mysql.com/doc/refman/8.0/en/innodb-transaction-isolation-levels.html

Resources used:

Theory

In a multiprogramming environment where multiple transactions can be executed simultaneously, it is highly important to control the concurrency of transactions.

Concurrency control is provided in a database to:

i. enforce isolation among transactions.

ii. preserve database consistency through consistency preserving execution of transactions.
iii. resolve read-write and write-read conflicts.

Concurrency control protocols can be broadly divided into two categories −

1. Lock based protocols
2. Time stamp based protocols

Lock-based Protocols

Database systems equipped with lock-based protocols use a mechanism by which any transaction cannot read or write data until it acquires an appropriate lock on it. Locks are of two kinds −

1. Binary Locks − A lock on a data item can be in two states; it is either locked or unlocked.
2. Shared/exclusive − This type of locking mechanism differentiates the locks based on their uses. If a lock is acquired on a data item to perform a write operation, it is an exclusive lock. Allowing more than one transaction to write on the same data item would lead the database into an inconsistent state. Read locks are shared because no data value is being changed.

Timestamp Ordering Protocol

The timestamp-ordering protocol ensures serializability among transactions in their conflicting read and write operations. This is the responsibility of the protocol system that the conflicting pair of tasks should be executed according to the timestamp values of the transactions.

The timestamp of transaction Ti is denoted as TS(Ti).
Read time-stamp of data-item X is denoted by R-timestamp(X).
Write time-stamp of data-item X is denoted by W-timestamp(X).

Timestamp ordering protocol works as follows −

If a transaction Ti issues a read(X) operation −
If TS(Ti) < W-timestamp(X)
Operation rejected.
If TS(Ti) >= W-timestamp(X)
Operation executed.
All data-item timestamps updated.

If a transaction Ti issues a write(X) operation −
If TS(Ti) < R-timestamp(X)
Operation rejected.
If TS(Ti) < W-timestamp(X)
Operation rejected and Ti rolled back.
Otherwise, operation executed.

Implementation of Lock Protocol:

1) Applying WRITE lock on transactions table:

Transactions table:

Trans_ID	C_ID	P_ID	M_ID	Quantity	Total_Cost	Phar_Name	Pres_ID	D_FName	D_LName
100	3	14	2	20	2000	NonePhrama	35	Kritarth	Jain
201	3	14	1	1	15	NonePhrama	35	Kritarth	Jain
202	3	14	1	1	15	NonePhrama	35	Kritarth	Jain
203	3	14	2	5	500	NonePhrama	35	Kritarth	Jain
206	3	14	1	1	15	NonePhrama	35	Kritarth	Jain
NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

Pharmacy table:

Pharma_ID	Pharma_Name	Area	Pin	street
12	Kritarth Pharma	Ghatkopar	400086	MG Road
13	Hussein Pharma	Ghatkopar	400016	PG Road
14	Nayan Pharma	Ghatkopar	400077	KG Road
15	Manan Pharma	Ghatkopar	400077	LG Road
16	Daisy Pharma	Bandra	400099	LT Road
NULL	NULL	NULL	NULL	NULL

LOCK TABLE transactions WRITE;

insert into Transactions values(204,3,14,1,1,100,'NonePhrama',35,'Kritarth','Jain');

SELECT * FROM transactions;

Trans_ID	C_ID	P_ID	M_ID	Quantity	Total_Cost	Phar_Name	Pres_ID	D_FName	D_LName
100	3	14	2	20	2000	NonePhrama	35	Kritarth	Jain
201	3	14	1	1	15	NonePhrama	35	Kritarth	Jain
202	3	14	1	1	15	NonePhrama	35	Kritarth	Jain
203	3	14	2	5	500	NonePhrama	35	Kritarth	Jain
204	3	14	1	1	15	TrickPhrama	35	Kritarth	Jain
206	3	14	1	1	15	NonePhrama	35	Kritarth	Jain
NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

insert into pharmacy values(17,"Smeet Pharma","Ghatkopar",400077,"LG Road");

Error Code: 1100. Table 'pharmacy' was not locked with LOCK TABLES

UNLOCK TABLES;

insert into pharmacy values(17,"Smeet Pharma","Ghatkopar",400077,"LG Road");

Result Grid | Filter Rows: | Edit:

Pharma_ID	Pharma_Name	Area	Pin	street
12	Kritarth Pharma	Ghatkopar	400086	MG Road
13	Hussein Pharma	Ghatkopar	400016	PG Road
14	Nayan Pharma	Ghatkopar	400077	KG Road
15	Manan Pharma	Ghatkopar	400077	LG Road
16	Daisy Pharma	Bandra	400099	LT Road
17	Smeet Pharma	Ghatkopar	400077	LG Road
NULL	NULL	NULL	NULL	NULL

2) Applying READ lock on medicine table:

LOCK TABLE medicine READ;

SELECT * FROM medicine;

Result Grid | Filter Rows: | Edit: | Export/Import: | Wrap Cell Content:

Med_ID	Med_Name	Med_Company	Quantity	Price	Med_Type	Pharma_Name	Exp_Date	Phar_ID	Cus_ID
1	Crocin	intel	196	15	headache	Kritarth Pharma	2022-12-12	12	1
2	Paracetemol	AMD	5	100	Antibiotic	HusseinPharma	2023-12-12	13	4
3	Crocin	intel	200	10	headache	Nayan Pharma	2024-12-12	14	1
4	Crocin	intel	200	8	headache	Manan Pharma	2024-12-12	15	1
5	ALDOX	intel	200	8	headache	Manan Pharma	2024-12-12	15	1
7	Benadryl	intel	20	8	Cough Syrup	Manan Pharma	2024-12-12	15	1
8	Phospha	AWS	50	9	Heart Problem	Kritarth Pharma	2024-12-12	12	1
9	CoughSyr	kin	60	11	Cough Syrup	Hussein Pharma	2024-12-12	13	1
NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

select * from pharmacy;

Error Code: 1100. Table 'pharmacy' was not locked with LOCK TABLES

UNLOCK TABLES;

select * from pharmacy;

Pharma_ID	Pharma_Name	Area	Pin	street
12	Kritarth Pharma	Ghatkopar	400086	MG Road
13	Hussein Pharma	Ghatkopar	400016	PG Road
14	Nayan Pharma	Ghatkopar	400077	KG Road
15	Manan Pharma	Ghatkopar	400077	LG Road
16	Daisy Pharma	Bandra	400099	LT Road
17	Smeet Pharma	Ghatkopar	400077	LG Road
NULL	NULL	NULL	NULL	NULL

3) Applying READ lock on customer table and WRITE lock on pharmacy table:

Customer table:

Cust_ID	C_FName	C_MName	C_LName	Gender	Street	Pin	Area
1	Kri	Tar	Th	1	MG Road	400086	Ghatkopar
2	Na	Ya	N	1	KG Road	400077	Ghatkopar
3	Hus	Sei	N	1	PG Road	400033	Ghatkopar
4	Pi	Na	Ki	0	KL Road	401033	Ghatk
5	K	an	K	0	RS Road	111033	Ghatkop
6	Kri	Tar	Th	0	RS Road	111033	Ghatkop
NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

Pharmacy table:

Pharma_ID	Pharma_Name	Area	Pin	street
12	Kritarth Pharma	Ghatkopar	400086	MG Road
13	Hussein Pharma	Ghatkopar	400016	PG Road
14	Nayan Pharma	Ghatkopar	400077	KG Road
15	Manan Pharma	Ghatkopar	400077	LG Road
16	Daisy Pharma	Bandra	400099	LT Road
17	Smeet Pharma	Ghatkopar	400077	LG Road
NULL	NULL	NULL	NULL	NULL

LOCK TABLE customer READ, pharmacy WRITE;

SELECT * FROM customer;

Cust_ID	C_FName	C_MName	C_LName	Gender	Street	Pin	Area
1	Kri	Tar	Th	1	MG Road	400086	Ghatkopar
2	Na	Ya	N	1	KG Road	400077	Ghatkopar
3	Hus	Sei	N	1	PG Road	400033	Ghatkopar
4	Pi	Na	Ki	0	KL Road	401033	Ghatk
5	K	an	K	0	RS Road	111033	Ghatkop
6	Kri	Tar	Th	0	RS Road	111033	Ghatkop
NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

select * from transactions;

Error Code: 1100. Table 'transactions' was not locked with LOCK TABLES

insert into pharmacy values(19,"Nishit Pharma","Ghatkopar",400077,"LG Road");

select * from pharmacy;

Pharma_ID	Pharma_Name	Area	Pin	street
12	Kritarth Pharma	Ghatkopar	400086	MG Road
13	Hussein Pharma	Ghatkopar	400016	PG Road
14	Nayan Pharma	Ghatkopar	400077	KG Road
15	Manan Pharma	Ghatkopar	400077	LG Road
16	Daisy Pharma	Bandra	400099	LT Road
17	Smeet Pharma	Ghatkopar	400077	LG Road
19	Nishit Pharma	Ghatkopar	400077	LG Road
NULL	NULL	NULL	NULL	NULL

insert into customer values (8,"Niren","Kanak","Singhania",0,"KL Road",401033,"Ghatk");

Error Code: 1099. Table 'customer' was locked with a READ lock and can't be updated

UNLOCK TABLES;

select * from transactions;

Trans_ID	C_ID	P_ID	M_ID	Quantity	Total_Cost	Phar_Name	Pres_ID	D_FName	D_LName
100	3	14	2	20	2000	NonePhrama	35	Kritarth	Jain
201	3	14	1	1	15	NonePhrama	35	Kritarth	Jain
202	3	14	1	1	15	NonePhrama	35	Kritarth	Jain
203	3	14	2	5	500	NonePhrama	35	Kritarth	Jain
204	3	14	1	1	15	TrickPhrama	35	Kritarth	Jain
206	3	14	1	1	15	NonePhrama	35	Kritarth	Jain
NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

insert into customer values (8,"Niren","Kanak","Singhania",0,"KL Road",401033,"Ghatk");

select * from customer;

Result Grid | Filter Rows: | Edit: | Export/Import:

Cust_ID	C_FName	C_MName	C_LName	Gender	Street	Pin	Area
1	Kri	Tar	Th	1	MG Road	400086	Ghatkopar
2	Na	Ya	N	1	KG Road	400077	Ghatkopar
3	Hus	Sei	N	1	PG Road	400033	Ghatkopar
4	Pi	Na	Ki	0	KL Road	401033	Ghatk
5	K	an	K	0	RS Road	111033	Ghatkop
6	Kri	Tar	Th	0	RS Road	111033	Ghatkop
8	Niren	Kanak	Singhania	0	KL Road	401033	Ghatk
NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

Post Lab Questions:

1. **Explain pitfalls of 2PL (Two Phase Locking) Protocol**

ANS) To guarantee serializablity, we must follow some additional protocol concerning the positioning of locking and unlocking operations in every transaction. This is where the concept of Two Phase Locking(2-PL) comes in the picture, 2-PL ensures serializablity.

Two Phase Locking – A transaction is said to follow Two Phase Locking protocol if Locking and Unlocking can be done in two phases.

Growing Phase: New locks on data items may be acquired but none can be released.

Shrinking Phase: Existing locks may be released but no new locks can be acquired.

Let's see a transaction implementing 2-PL:

	T_1	T_2
1	lock-S(A)	
2		lock-S(A)
3	lock-X(B)	
4	.....	.....
5	Unlock(A)	
6		Lock-X(C)
7	Unlock(B)	
8		Unlock(A)
9		Unlock(C)
10	.....	.....

This is just a skeleton transaction which shows how unlocking and locking works with 2-PL. Note for:

Transaction T1:

Growing Phase is from steps 1-3.

Shrinking Phase is from steps 5-7.

Lock Point at 3

Transaction T2:

Growing Phase is from steps 2-6.

Shrinking Phase is from steps 8-9.

Lock Point at 6

Deadlock in 2-PL –

Consider this simple example, it will be easy to understand.Say we have two transactions T1 and T2.

Schedule: Lock-X1(A) Lock-X2(B) Lock-X1(B) Lock-X2(A)

Drawing the precedence graph, you may detect the loop. So Deadlock is also possible in 2-PL.

Two-phase locking may also limit the amount of concurrency that occur in a schedule because a Transaction may not be able to release an item after it has used it. This may be because of the protocols and other restrictions we may put on the schedule to ensure serializability, deadlock freedom and other factors. This is the price we have to pay to ensure serializability and other factors, hence it can be considered as a bargain between concurrency and maintaining the ACID properties.

Conclusion (In your own words): By performing this experiment understood concurrency control in database. Also implemented read lock and write lock on our created database with unlocking as well. Also checked the error that is obtained after executing query on another tables on which lock was not there.