SQL 11 : TRANSACTIONS -2

27/05/24

AGENDA

Isolation Levels	Problems
1. Read Uncommitted	Dirty Read
2. Read Committed	Non-Repeatable Read
3. Repeatable Read	Phantom Read
4. Serializable	Deadlock.

DB locks

1. Shared (S) - Multiple Transcations can read a row.
2. Exclusive(X) - Transactions can "update" or "delete" a row.

T1	T2
Begin;	Begin
Xclusive lock. Update where id=1.	X update where id=1;
Commit;	. commit;

READ COMMITTED

1. Transactions will read last committed data.
2. Dirty Reads will not happen.

Ques

Table: students

s_id	s_name	psp	email_sent	
1	A	60	False	True
2	B	~~75~~ 80	False	True.
3	C	70	False	True
4	D	45	false	True
5	E	90	false.	

RU → RC → RR → S

Send email to all students with PSP < 80%.

① List<Students> = SELECT * from STUDENTS WHERE PSP<80;

② Send Emails (A, C, D)

③ Update "email_sent" field to "True".

```
Update Students
SET email_sent = True
WHERE PSP < 80;
```

D - True

Non-Repeatable Read.

REPEATABLE READ

Performance poor -> because it takes a snapshot at the beginning of transaction and uses that as a reference during the transaction.

Phantom → (ghost)

BREAK TILL - 8:23 AM.

PHANTOM READ

Read something that does not exist.

Isolation Level :- RR.

T1 (RR)	T2
① Begin;	① Begin
② Read the last row; "XYZ"	② Insert 1 row; "ABC"
③ Read the last row.	③ Commit;
④ Commit;	

SERIALIZABLE

→ Transactions executed in a particular order. Serially, even though they occured in parallel.

SELECT . . . "S"

Deadlock:

T1	T2
Select * film_id in (1,2)	Select * film_id in (2,3)
S 1	S 2
S 2	S 3
update id=2	update=2.
S id=2	S id=2
T1	T2
X	X

InnoDB

Serializable.

T1

Serializable

T2

Insert(" ");

Performance goes down ↓
Consistency goes up ↑

Summary

Read	Transaction Level Data
RU	No need to maintain any version of data.
RC	Maintain a version of all committed data. — till the current start in transaction.
RR	Maintain a history (Snapshot till the start of transaction.
S	Acquire a shared lock for read also.

Queries run in class -

```
-- SQL 11 TRANSACTIONS 2
-- T 1
USE SAKILA;
SET SESSION TRANSACTION ISOLATION LEVEL READ UNCOMMITTED;
SET SESSION TRANSACTION ISOLATION LEVEL READ COMMITTED;
SET SESSION TRANSACTION ISOLATION LEVEL REPEATABLE READ;
SET SESSION TRANSACTION ISOLATION LEVEL SERIALIZABLE;
SHOW VARIABLES LIKE '%ISOLATION%';
-- LOCKS
SELECT ENGINE_TRANSACTION_ID,  LOCK_TYPE, LOCK_MODE, LOCK_STATUS, LOCK_DATA
FROM PERFORMANCE_SCHEMA.DATA_LOCKS;
-- RUNNING TRANSACTIONS
SELECT * FROM information_schema.innodb_trx;


SET AUTOCOMMIT=0;
BEGIN;
SELECT * FROM FILM WHERE FILM_ID=1;

UPDATE FILM
SET TITLE = "ABCD"
WHERE FILM_ID = 1;

COMMIT;

-- Error Code: 2013. Lost connection to MySQL server during query
```

```
-- READ COMMITTED
-- T1
SET AUTOCOMMIT=0;
BEGIN;
SELECT * FROM FILM WHERE FILM_ID=1;

UPDATE FILM
SET TITLE = "ABCD"
WHERE FILM_ID = 1;

COMMIT;

-- T1
-- NON REPEATABLE READ
SET AUTOCOMMIT=0;
BEGIN;
SELECT * FROM FILM WHERE FILM_ID=1;
COMMIT;

-- T1
-- REPEATABLE READ
SET AUTOCOMMIT=0;
BEGIN;
SELECT * FROM FILM WHERE FILM_ID=1;
COMMIT;

-- T1
-- PHANTOM READ
SET AUTOCOMMIT=0;
BEGIN;
SELECT * FROM FILM
ORDER BY FILM_ID DESC LIMIT 1;

SELECT COUNT(*) FROM FILM;

UPDATE FILM
SET TITLE = "ABCD"
WHERE FILM_ID = 1020;
COMMIT;

-- LOCKS
SELECT ENGINE_TRANSACTION_ID,  LOCK_TYPE, LOCK_MODE, LOCK_STATUS, LOCK_DATA
FROM PERFORMANCE_SCHEMA.DATA_LOCKS;
-- RUNNING TRANSACTIONS
SELECT * FROM information_schema.innodb_trx;
SET SESSION TRANSACTION ISOLATION LEVEL SERIALIZABLE;
SHOW VARIABLES LIKE '%ISOLATION%';
```

```
-- SERIALIZABLE , DEADLOCK
-- DEADLOCK
-- T1
SET AUTOCOMMIT=0;
BEGIN;
SELECT * FROM FILM WHERE FILM_ID IN (1,2);
UPDATE FILM
SET TITLE = "OMG"
WHERE FILM_ID = 2;

COMMIT;



-- SESSION 2 -- T2

- -- TRANSACTION-2
-- T 2
USE SAKILA;
SET SESSION TRANSACTION ISOLATION LEVEL READ UNCOMMITTED;
SET SESSION TRANSACTION ISOLATION LEVEL SERIALIZABLE;
SHOW VARIABLES LIKE '%ISOLATION%';
-- LOCKS
SELECT ENGINE_TRANSACTION_ID,  LOCK_TYPE, LOCK_MODE, LOCK_STATUS, LOCK_DATA
FROM PERFORMANCE_SCHEMA.DATA_LOCKS;
-- RUNNING TRANSACTIONS
SELECT * FROM information_schema.innodb_trx;

SET AUTOCOMMIT=0;
BEGIN;
SELECT * FROM FILM WHERE FILM_ID=1;

UPDATE FILM
SET TITLE = "PK"
WHERE FILM_ID = 1;

-- Error Code: 2013. Lost connection to MySQL server during query
COMMIT;

-- READ COMMITTED
-- T2
SET AUTOCOMMIT=0;
BEGIN;
SELECT * FROM FILM WHERE FILM_ID=1;
UPDATE FILM
SET TITLE = "MOM"
WHERE FILM_ID = 1;
COMMIT;
```

```
UPDATE FILM
SET TITLE = "MAA"
WHERE FILM_ID = 1;
COMMIT;

SET AUTOCOMMIT=0;
BEGIN;
SELECT * FROM FILM WHERE FILM_ID=1;
UPDATE FILM
SET TITLE = "RRR"
WHERE FILM_ID = 1;
COMMIT;


-- T2
SET AUTOCOMMIT=0;
BEGIN;

INSERT INTO film
VALUES (default, 'IT', 'GREAT FILM', 2008, 1,
NULL, 3, 4.99, 152, 19.99, 'PG-13', 'Trailers', default);
COMMIT;

SELECT * FROM FILM
ORDER BY FILM_ID DESC LIMIT 1;

SELECT COUNT(*) FROM FILM;

-- T2
-- LOCKS
SELECT ENGINE_TRANSACTION_ID,  LOCK_TYPE, LOCK_MODE, LOCK_STATUS, LOCK_DATA
FROM PERFORMANCE_SCHEMA.DATA_LOCKS;
-- RUNNING TRANSACTIONS
SELECT * FROM information_schema.innodb_trx;
SET SESSION TRANSACTION ISOLATION LEVEL SERIALIZABLE;
SHOW VARIABLES LIKE '%ISOLATION%';

-- DEADLOCK
-- T2
SET AUTOCOMMIT=0;
BEGIN;

SELECT * FROM FILM WHERE FILM_ID IN (2,3);
UPDATE FILM
SET TITLE = "PAA"
WHERE FILM_ID = 2;

COMMIT;
```

-- Error Code: 1213. **Deadlock found when trying to get lock; try restarting transaction**