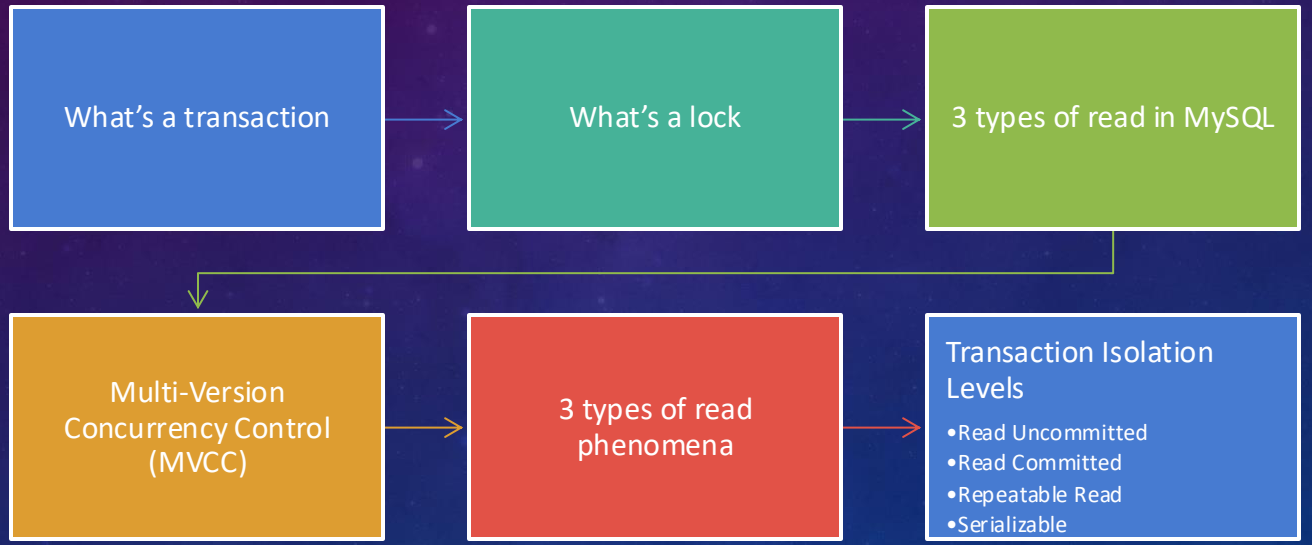
HOW NOT TO GO BANKRUPT (AND LOOK FOOLISH)

MASTERING TRANSACTIONS IN MYSQL

MIKAËL FRANCOEUR

CONTENTS


What's a transaction
What's a lock
3 types of read in MySQL
Multi-Version Concurrency Control (MVCC)
3 types of read phenomena
Transaction Isolation Levels
•Read Uncommitted
•Read Committed
•Repeatable Read
•Serializable

WHAT'S A TRANSACTION

- Sequence of statements that is atomic WRT persistence and that can be isolated from other transactions.
- In MySQL, everything runs in a transaction

WHAT'S A LOCK

- An access right
- Resource types
  - Table
  - Row
  - Row + preceding gap
- Are kept until the end of a transaction
- Prevent concurrent modification of a resource
- All updated/inserted/deleted rows
- All rows scanned in UPDATE/DELETE
- All rows scanned if in REPEATABLE READ or higher in
  - INSERT INTO … SELECT
- SELECT … FOR SHARE, SELECT … FOR UPDATE

3 TYPES OF READ

- Consistent Nonlocking Read
  - Plain SELECT
  - Read data from a single snapshot/timestamp
  - See results of earlier INSERT/UPDATE/DELETE statements in the same transaction
- Locking Reads
  - Take locks
  - Prevent concurrent modification
  - Always see the latest committed version of a row
- Plain Read
  - Only in READ UNCOMMITTED

MULTI-VERSION CONCURRENCY CONTROL (MVCC)

- Every row contains a pointer to part of an “undo log” that contains information on how to rebuild previous version of the row.
- Prevents certain read phenomena without locking/contention.

3 READ PHENOMENA

- Defined in SQL-92
- Non-Repeatable Read
  - A transaction reads a row, then reads it again but the rows has changed or was deleted.
- Phantom Reads
  - A transaction reads a set of rows, then reads it again but more rows are returned.
- Dirty Reads
  - A transaction reads uncommitted data from another transaction.

TRANSACTION ISOLATION LEVELS

- Prevent or allow certain read phenomena as a performance tradeoff.
- Can be set globally, by session, or by transaction.
- Implemented with
  - MVCC
  - Locking
- Four levels
  - SERIALIZABLE
  - REPEATABLE READ
  - READ COMMITTED
  - READ UNCOMMITTED

REPEATABLE READ

- MVCC
  - The first SELECT from an InnoDB table in a transaction establishes a snapshot that all other queries in the same transaction will read from.
    - Does not apply to locking reads
    - More resource-intensive
- Locking
  - Does gap locking
  - Able to lock the absence of a row

Non-Repeatable Read	Phantom Read	Dirty Read	Read Consistency
No*	No*	No	Transaction

READ COMMITTED

- MVCC
  - Every consistent read gets its own snapshot
  - Less use of undo logs
- Locking
  - No gap locking

Non-Repeatable Read	Phantom Read	Dirty Read	Read Consistency
Yes	Yes	No	Statement

READ UNCOMMITTED

- Read phenomena
  - Transaction sees uncommitted data from other ongoing transactions
  - Does not use MVCC
- Use it only if the goal of your query is to view uncommitted data
  - Useful to debug integration tests

Non-Repeatable Read	Phantom Read	Dirty Read	Read Consistency
Yes	Yes	Yes	No

SERIALIZABLE

- Similar to REPEATABLE READ
- Every SELECT is transformed into a SELECT ... FOR SHARE, except if autocommit is ON and not in an explicit transaction
- More contention
  - NO WAIT
  - SKIP LOCKED

Non-Repeatable Read	Phantom Read	Dirty Read	Read Consistency
No	No	No	Transaction

TRANSACTION ISOLATION LEVELS

Level	Non-Repeatable Read	Phantom Read	Dirty Read	Read Consistency
SERIALIZABLE	No	No	No	Transaction
REPEATABLE READ	No*	No*	No	Transaction
READ COMMITTED	Yes	Yes	No	Statement
READ UNCOMMITTED	Yes	Yes	Yes	Statement

jepsen.io

Questions?