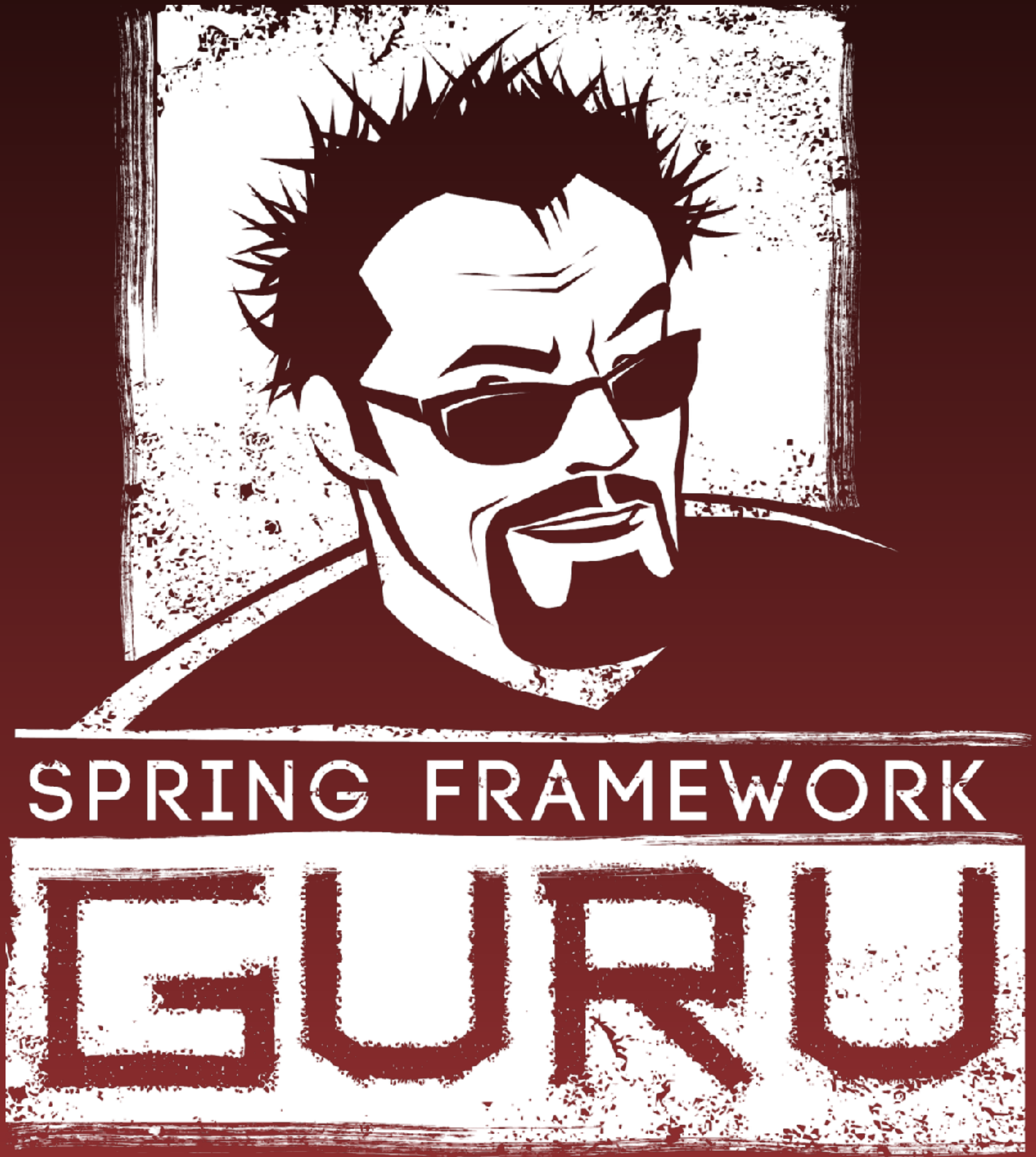

SPRING FRAMEWORK
GURU

SQL Beginner to Guru

MySQL Edition

Introduction to SQL Transactions

SQL Database Transactions - ACID

- ACID
  - **Atomicity** - All operations are completed successfully or database is returned to previous state.
  - **Consistency** - Operations do not violate system integrity constraints.
  - **Isolated** - Results are independent of concurrent transactions.
  - **Durable** - Results are made persistent in case of system failure (ie written to disk)

Important Terms

- **Transaction** - A unit of work. One or more SQL operations
  - Typically DML (and **Not** DDL) statements which alter data.
  - Can be just one; can be hundreds or thousands.
- **Commit** - Indicates the end of the transaction and tells database to make changes permeant.
  - More efficient to do multiple operations in a transaction. There is a 'cost' with commits.
- **Rollback** - Revert all changes of the transaction
- **Save Point** - Programatic point you can set, which allows you to rollback to (ie rollback part of a transaction)

MySQL Transactions

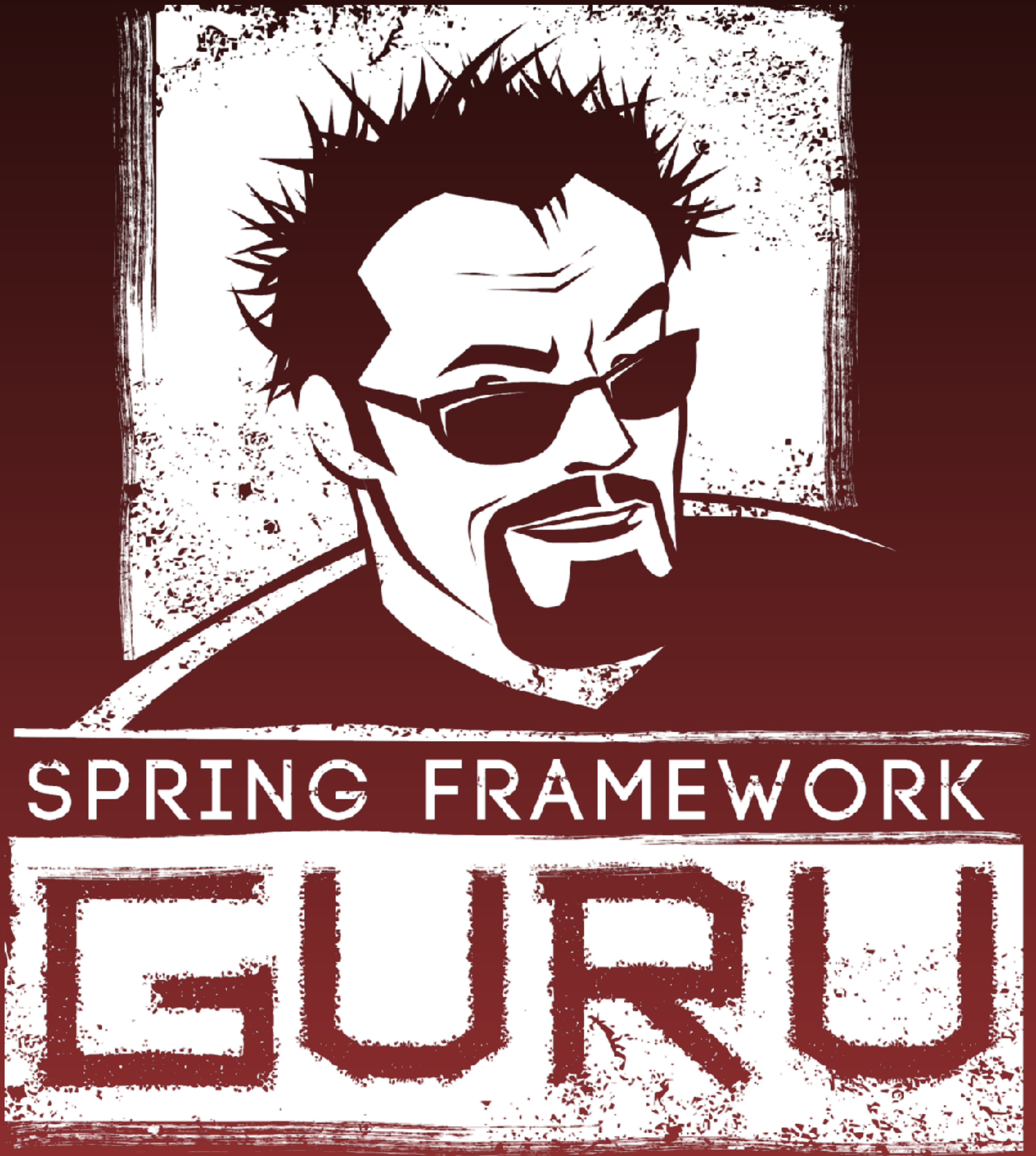
- Auto-Commit - Typically enabled by default
- **SET AUTOCOMMIT** - enables or disables auto commit for your session
- **START TRANSACTION** or **BEGIN** - Starts a transaction
- **COMMIT** - Commits the current transaction
- **ROLLBACK** - **ROLLBACK TO** <*save point name*> - Rolls back current transaction
- **SAVEPOINT** <*name*> - Sets a save point

Database Locks

- The database will lock the records, (in some cases whole table or database) to prevent other processes from changing data
  - ACID compliance
- Within a transaction the following DML statements will lock records of the affected rows:
  - `SELECT FOR UPDATE; UPDATE; DELETE`
- During the transactions other sessions attempting to modify locked records will by default wait for the lock to be released. (ie interactively it will seem like things are hanging)
- Deadlock - Occurs where two transactions lock each other and can never complete.
  - Both fail and roll back.

Transaction Isolation Levels

- **Repeatable Read** - Default Isolation Level. Your statement receives a consistent view of the database, even if other transactions are committed during your transaction.
  - Your transaction gets a snapshot of the data, which does not change.
- **Read Committed** - Reads within your transaction will receive a fresh snapshot of the data.
- **Read Uncommitted** - Reads are not consistent, but may avoid additional database locks.
  - aka - “Dirty Read”
- **Serializable** - Similar to Repeatable Read, but may lock rows selected in transaction.



SPRING FRAMEWORK

GURU