トランザクションの分離レベル(Isolation Level)について

はじめに

極力、自信がないところは書かないように気を付けましたが、
間違っているところが有るかもしれないので、
このスライドの内容を鵜呑みせずに、
くわしくはググるか↓の記事を見てください。

このスライドはおもに↓の記事を参照しながら書きました。

http://qiita.com/kumagi/items/1dc1a91ec007365ac694
http://gyouza-daisuki.hatenablog.com/entry/2013/11/19/150838
https://ja.wikipedia.org/wiki/%E3%83%88%E3%83%A9%E3%83%B3%E3%82

分離レベルとは

トランザクションが複数同時に行われた時に、
一つのトランザクションがほかの処理からどれだけ独立しているか、
ほかの処理に影響をどれだけ影響をあたえるのか、
を設定するもの。

分離レベルとは(2)

ANSIの定義だと、4つの種類がある。

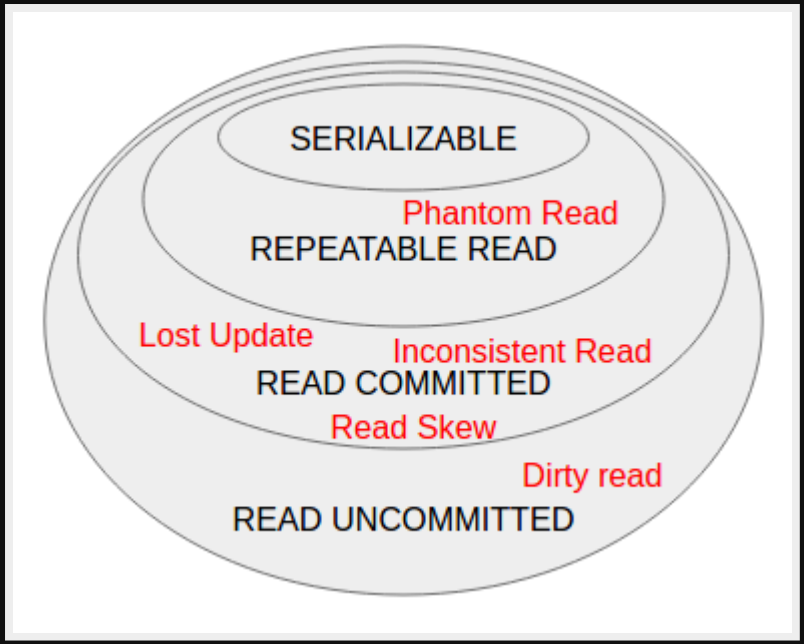
- READ UNCOMMITTED
- READ COMMITTED
- REPEATABLE READ
- SERIALIZABLE

READ UNCOMMITTED が一番独立性が低い、下に行くほど高い

(Web開発者が知っておくべき八つの分離レベル という話もある)

READ UNCOMMITTED	
READ COMMITTED	PostgreSQL, Oracle, SQL Server
REPEATABLE READ	Mysql, MariaDB
SERIALIZABLE	

READ UNCOMMITTEDは性能は良いが独立性が低い、
SERIALIZABLEは独立性は高いが性能が低い



http://qiita.com/kumagi/items/1dc1a91ec007365ac694 より

READ COMMITTED

コミット済みのものはreadする。

- ファジーリード/Non-Repeatable Read Anomaly/Inconsistent Read Anomaly
  - xを2回readしようとした時に、間に他トランザクションでxのwriteがあった時に、readした内容が異なる
- Read Skew Anomaly
  - x, yをreadしようとした時に、他トランザクションでyのwriteがあった時に、readしたx, yに一貫性がない
- Lost Update Anomaly

REPEATABLE READ

一度読み込んだデータはそのトランザクション内では何度読んでも同じ値になる。
ほかのトランザクションでwriteのコミットがされても、write前のデータをreadしていれば、write後にreadしても同じデータが読まれる。

- ファントムリード
  - 2回範囲に対する操作をした時に、間にinsertが発生していると、結果が異なる
  - 2回countしようとした時に、間で1件insertされてた場合、2回目は1件多い数字になる
- READ COMMITTEDで発生してたAnomalyは発生しない

REPEATABLE READ(2)

同じtransaction内で2回readしようとしても結果は同じ。
=> しかし、この仕組みのことを知らずに、
最新のデータを取るぞ！って思って2回目のreadをしても1回目のreadと同じデータなので、
事故る可能性がある。

REPEATABLE READ(2)

MysqlのInnnoDBでは、ネクストキーロックという仕組みでファントムリードが起きないようにしている。
が、そのせいでギャップロックが発生する
https://dev.mysql.com/doc/refman/5.6/ja/innodb-next-key-locking.html

言いたかったこと

Railsを使っているとActiveRecordのお陰で、DBサーバーがMysqlなのか
PostgreSQLなのかを気にせず使えてとても便利。
だが、こういう違いが有ることを意識しておかないと事故る可能性がある。

最後に

極力、自信がないところは書かないように気を付けましたが、
間違っているところが有るかもしれないので、
このスライドの内容を鵜呑みせずに、
くわしくはググるか↓の記事を見てください。

このスライドはおもに↓の記事を参照しながら書きました。

http://qiita.com/kumagi/items/1dc1a91ec007365ac694
http://gyouza-daisuki.hatenablog.com/entry/2013/11/19/150838
https://ja.wikipedia.org/wiki/%E3%83%88%E3%83%A9%E3%83%B3%E3%82