

MySQL手动提交事务



为什么我的mysql在read-uncommitted也会加行锁，排他更新



在 MySQL 中，隔离级别 `READ UNCOMMITTED` 主要是允许脏读，也就是说，它允许一个事务读取另一个事务未提交的更改。然而，**即便在 `READ UNCOMMITTED` 隔离级别下，MySQL 依然可能会加锁，尤其是在执行更新操作时。**这与 MySQL 的事务实现、行锁机制以及更新操作的行为有关。

为什么 `READ UNCOMMITTED` 也会加行锁？

在 `READ UNCOMMITTED` 隔离级别下，**并不是所有的查询都会加锁**，只有在执行 **数据修改操作**（如 `UPDATE`、`INSERT`、`DELETE`）时，MySQL 才会加锁。

1. 行锁和数据修改操作

- 当你执行 `UPDATE` 或 `DELETE` 操作时，MySQL 会在数据修改时加 **排他锁（exclusive lock）**，即使是在 `READ UNCOMMITTED` 隔离级别下。这样做是为了防止并发冲突，确保数据一致性，并防止其他事务修改正在被当前事务修改的行。
- **排他锁（Exclusive Lock）** 确保了只有当前事务能够修改该行数据，直到事务提交（`COMMIT`）或回滚（`ROLLBACK`）。如果其他事务也试图修改该行数据，它们将被阻塞，直到当前事务释放锁。

2. `READ UNCOMMITTED` 的行为

- 在 `READ UNCOMMITTED` 隔离级别下，事务A 允许读取事务B 未提交的数据（即脏读）。但是，这并不意味着所有操作都不会加锁，特别是 **修改数据** 时，MySQL 仍然会加锁（即使隔离级别较低）。
- 在 `READ UNCOMMITTED` 隔离级别下，MySQL 加锁的主要目的是为了 **确保更新操作的原子性和一致性**。即使隔离级别较低，锁的使用仍然是为了防止并发事务之间的数据冲突。

3. `READ UNCOMMITTED` 不等于“不加锁”

- **查询操作**：在 `READ UNCOMMITTED` 隔离级别下，**读取操作不会加锁**，其他事务可以修改读取的行数据。
- **更新操作**：对于 `UPDATE` 或 `DELETE` 等修改操作，MySQL 会加 **排他锁**，确保其他事务无法同时修改该行数据。

4. 为什么排他锁在 `READ UNCOMMITTED` 级别下仍然存在

排他锁用于保护数据一致性，在某些情况下即使在较低的隔离级别下（如 `READ UNCOMMITTED`），MySQL 仍需要加锁。以下是一些原因：

- **并发冲突**：如果事务A更新了某个数据行，其他事务B在同一时间也想更新该行数据，那么数据库必须通过排他锁确保只有一个事务能够修改该行数据，防止数据冲突。

- **修改操作的原子性**：即使隔离级别较低，数据库依然需要保证每个事务中的修改操作是原子性的。为了实现这一点，MySQL 会在执行 `UPDATE`、`DELETE` 等修改操作时加锁。
- **锁的优化**：MySQL 会加锁以提高并发性能和减少数据冲突。在 `READ UNCOMMITTED` 隔离级别下，虽然可以读取未提交的数据，但当数据被修改时，仍然会加排他锁，以减少事务间的冲突。

5. 示例

```
sql

-- 设置为 READ UNCOMMITTED 隔离级别
SET SESSION TRANSACTION ISOLATION LEVEL READ UNCOMMITTED;

-- 事务A：读取数据，且不加锁
START TRANSACTION;
SELECT * FROM accounts WHERE account_id = 1;

-- 事务A：更新数据，需加排他锁
UPDATE accounts SET balance = balance - 100 WHERE account_id = 1;

-- 事务B：读取事务A未提交的数据（脏读）
SELECT * FROM accounts WHERE account_id = 1;
```

在这个示例中：

- **事务A的读取操作**（`SELECT`）不会加锁，它允许其他事务读取未提交的数据（脏读）。
- **事务A的更新操作**（`UPDATE`）会加排他锁，确保只有事务A能够修改该行数据，直到事务A提交或回滚。即使隔离级别是 `READ UNCOMMITTED`，更新操作依然会加锁。

6. 总结

- `READ UNCOMMITTED` 隔离级别允许脏读，即事务可以读取其他事务未提交的数据。
- **行锁** 在 `READ UNCOMMITTED` 隔离级别下，**不会**在读取操作时加锁，但是 **会在修改数据（如 `UPDATE`）时加锁**，加的是 **排他锁**，确保数据一致性和原子性。
- 即使在较低的隔离级别下，MySQL 也会使用排他锁来避免并发修改时的冲突，因此无法完全避免加锁。