

题目 01- 完成 ReadView 案例，解释为什么 RR 和 RC 隔离级别下看到查询结果不一致

要求：

- 完成案例 01- 读已提交 RC 隔离级别下的可见性分析
- 完成案例 02- 可重复读 RR 隔离级别下的可见性分析
- 用通俗易懂的方式记录整个案例过程，可以画图与截图
- 做完案例给出结论，并对结论进行分析

回答范式：

准备一张表 t,里面数据如图：

id	score	name
1	90	张三

1. 案例 01- 读已提交 RC 隔离级别下的可见性分析
- 目标

通过两个会话对同一数据进行更新操作后查询，来判断结果是否符合 RC 的可见性判断。

○ 操作步骤

会话 1-txd=100	会话 2-txd=110
t1 时刻	
begin; select * from t where id = 1; update t set name = '李四' where id = 1;	
	t2 时刻
	begin; select * from t where id = 1;
t3 时刻	
commit;	
	t4 时刻
	select * from t where id = 1;

准
会
级
为 RC

	commit;

备会话 1,
话 2, 隔离
别 均 设置

○ 实践过程

1.t1 时会话 1 查询结果, 并更新

id	score	name
1	90	张三

2.t2 时刻会话 2 查询结果

id	score	name
1	90	张三

3.t4 时刻会话 2 查询结果

id	score	name
1	90	李四

○ 结论

在 t1 时刻, 会话 1 更新'张三'为'李四', 但是事务没有提交。此时会话 2 在 t2 时刻查询时, 生成的 ReadView 中, txd=100 的会话 1 事务处于活动状态, 对于会话 2 来说不可见, 在 undo 日志中会追溯但最近可见版本'张三'。

在 t3 时刻, 会话 1 提交更新后, t4 时刻会话 2 查询时会生成新的 ReadView, 此时 txd=100 属于过去的事务, 可以查询到更新, 因此读取到'李四'。

2. 案例 02- 可重复读 RR 隔离级别下的可见性分析

○ 目标

通过两个会话对同一数据行进行更新操作后查询, 来判断结果是否符合 RR 的可见性判断。

○ 操作步骤

准备会话 1, 会话 2, 隔离级别均设置为 RC, 后续操作步骤如案例 1

○ 实践过程

1.t1 时刻会话 1 查询

id	score	name
1	90	张三

2.t2 时刻会话 2 查询

id	score	name
1	90	张三

3.t4 时刻会话 2 查询

id	score	name
1	90	张三

○ 结论

在 t1 时刻, 会话 1 更新'张三'为'李四', 但是事务没有提交。此时会话 2 在 t2 时刻查询时, 生成的 ReadView 中, txd=100 的会话 1 事务处于活动状态, 对于会话 2 来说不可见, 在 undo 日志中会追溯但最近可见版本'张三'。

在 t3 时刻, 会话 1 提交更新后, t4 时刻会话 2 查询时仍然使用 t2 会话第一次查询的 ReadView,txd=100 的会话 1 事务处于活动状态, 对于会话 2 来说不可见, 在 undo 日志中会追溯但最近可见版本'张三'。

3. 结论分析

A:mysql 中的 innodb 引擎对于 rc 与 rr 隔离级别的支持是基于 undo 日志版本链和 readview 的视图来实现的。rc 与 rr 的主要区别是: rc 每次查询都会生产一个新的 readview, 而 rr 只会在事务第一次读生成 readview,后续都基于这个 readview 来判断数据可见性。

题目 02- 什么是索引?

要点:

1. 优点是什么?

A:加快数据的检索效率, 减少磁盘 IO,提高数据库的查询速度。

2. 缺点是什么?

A:索引树会占用一定的磁盘空间; 数据更新需要维护索引树。

3. 索引分类有哪些? 特点是什么?

A:

根据数量分:

单列索引: 单个列上建立的索引。

组合索引: 在两个及以上的列上建立的索引。

根据是否主键分:

聚集索引: 主键上建立的索引。

辅助索引: 在非单个主键上建立的索引。

4. 索引创建的原则是什么?

频繁出现在 where, order by ,group by 上的字段建立组合索引;

频繁出现在 select 后的字段建立组合索引;

多表 join, on 条件上的字段建立索引;

5. 有哪些使用索引的注意事项?

表记录少不需要建立索引;

一个表的索引不要太多;

尽量建立组合索引;

频繁更新的字段不建议建立索引;

区分度低的字段不建议建立索引;

很长的字段不建议建立索引;

无序的字段不建议建立索引;

6. 如何知道 SQL 是否用到了索引?

可以通过 explain 分析 sql 语句的执行计划, 看是否使用了索引。

7. 请你解释一下索引的原理是什么? 「重点」

- 说清楚为什么要用 B+Tree

A: 索引的主要目的是通过空间换时间的理念, 尽可能减少磁盘 io, 提高数据的检索效率。

索引的数据结构一般需要满足两个基本的需求: 等值查询与范围查询;

使用 b+树建立索引, 查询的时间效率是 $\log n$, 而且因为树的层次不高, 磁盘 io 的次数会减少。

B+树的叶子结点形成的有序双向链表结构, 也能极大提高范围查询的效率。并且每个表的数据都在主键索引的叶子结点上, 减少了磁盘空间的消耗。

题目 03- 什么是 MVCC?

要点:

1. Redo 日志

A:redo 日志是 mysql 的重做日志, 能够降低磁盘 IO, 提供高性能的并发能力。在数据库崩溃时也能用于数据恢复, 保证数据的完整性和一致性。

2. ReadView

A:ReadView 是 Innodb 引擎中用来实现 RC 和 RR 事务隔离级别的基础。总的来说, 就是在查询的时候, 会生成对当前活动的事务 Id、当前活动事务 Id 的最小值、最大值、生成本 ReadView 的事务 id 的视图。

3. 如何判断可见性

A:mysql 中的 RC 和 RR 的可见性是通过 Undo 日志与 ReadView 来实现的。

总体上讲:

如果 undo 版本链中的记录的事务 Id < ReadView 中活动事务 Id 的最小值, 表明这是过去已经提交的事务, 可见。

如果 undo 版本链中的记录的事务 Id > ReadView 中活动事务 Id 的最大值, 表明这是未来的事务, 不可见。

如果 undo 版本链中的记录的事务 Id = ReadView 中本事务 Id, 则可见。

如果 undo 版本链中的记录的事务 Id 在 ReadView 活动事务 Id 的最小值与最大值之间, 则判断是否是当前活动的事务 Id, 是则不可见; 不是则可见。