**第七周作业**

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：

1. 案例 01- 读已提交 RC 隔离级别下的可见性分析
   * 目标

假设3个事务过程如下表所示，隔离级别时RC, 求事务3在T6,T8,T10时的返回值

* + 操作步骤

|  |  |  |  |
| --- | --- | --- | --- |
| 时 间 | 事务01 【db\_trx\_id=100】 | 事务02 【db\_trx\_id=200】 | 事务03【db\_trx\_id=300】 |
| T1 | 开启事务 | 开启事务 | 开启事务 |
| T2 | 更新为关羽 |  |  |
| T3 | 更新为张飞 |  |  |
| T4 |  | 更新为赵云 |  |
| T5 |  | 更新为诸葛亮 |  |
| T6 |  |  | SELECT01，id=1， |
| T7 | 提交事务01 |  |  |
| T8 |  |  | SELECT02，id=1， |
| T9 |  | 提交事务02 |  |
| T10 |  |  | SELECT03，id=1， |

* + 实践过程

T6 时刻事务3产生readview

|  |  |  |  |
| --- | --- | --- | --- |
| m\_low\_limit\_id | m\_ids | m\_up\_limit\_id | m\_creator\_trx\_id |
| 1 | 1,2，3 | 4 | 3 |

Undo log

|  |  |  |  |
| --- | --- | --- | --- |
| trx\_id | Rollover\_pointer (指向下一行) | id | name |
| 1 |  | 1 | 张飞 |
| 1 |  | 1 | 关羽 |
| 0 |  | 1 | 刘备 |

因为trx\_id = 1是在m\_ids里面，并且大于等于最小trx\_id,小于最大trx\_id,所以trx\_id为1的数据不可见，只能用0的数据，所以返回刘备。

T8 时刻事务3因为隔离级别是RC,产生新的readview

|  |  |  |  |
| --- | --- | --- | --- |
| m\_low\_limit\_id | m\_ids | m\_up\_limit\_id | m\_creator\_trx\_id |
| 2 | 2，3 | 4 | 3 |

Undo log

|  |  |  |  |
| --- | --- | --- | --- |
| trx\_id | Rollover\_pointer (指向下一行) | id | name |
| 2 |  | 1 | 诸葛亮 |
| 2 |  | 1 | 赵云 |
| 1 |  | 1 | 张飞 |
| 1 |  | 1 | 关羽 |
| 0 |  | 1 | 刘备 |

因为trx\_id = 2是在m\_ids里面，并且大于等于最小trx\_id,小于最大trx\_id,所以trx\_id为2的数据不可见，但trx\_id=1小于最小x\_id,所以x\_id为1的数据可见，所以返回张飞。

T10 时刻事务3因为隔离级别是RC,产生新的readview

|  |  |  |  |
| --- | --- | --- | --- |
| m\_low\_limit\_id | m\_ids | m\_up\_limit\_id | m\_creator\_trx\_id |
| 3 | 3 | 4 | 3 |

Undo log

|  |  |  |  |
| --- | --- | --- | --- |
| trx\_id | Rollover\_pointer (指向下一行) | id | name |
| 2 |  | 1 | 诸葛亮 |
| 2 |  | 1 | 赵云 |
| 1 |  | 1 | 张飞 |
| 1 |  | 1 | 关羽 |
| 0 |  | 1 | 刘备 |

因为trx\_id=2小于最小trx\_id,所以trx\_id为2的数据可见，所以返回诸葛亮。

* + 结论

T6返回刘备，T8返回张飞，T10返回诸葛亮

1. 案例 02- 可重复读 RR 隔离级别下的可见性分析
   * 目标

假设3个事务过程如下表所示，隔离级别时RR, 求事务3在T6,T8,T10时的返回值

* + 操作步骤

|  |  |  |  |
| --- | --- | --- | --- |
| 时 间 | 事务01 【db\_trx\_id=100】 | 事务02 【db\_trx\_id=200】 | 事务03【db\_trx\_id=300】 |
| T1 | 开启事务 | 开启事务 | 开启事务 |
| T2 | 更新为关羽 |  |  |
| T3 | 更新为张飞 |  |  |
| T4 |  | 更新为赵云 |  |
| T5 |  | 更新为诸葛亮 |  |
| T6 |  |  | SELECT01，id=1， |
| T7 | 提交事务01 |  |  |
| T8 |  |  | SELECT02，id=1， |
| T9 |  | 提交事务02 |  |
| T10 |  |  | SELECT03，id=1， |

* + 实践过程

T6 时刻事务3产生readview

|  |  |  |  |
| --- | --- | --- | --- |
| m\_low\_limit\_id | m\_ids | m\_up\_limit\_id | m\_creator\_trx\_id |
| 1 | 1,2，3 | 4 | 3 |

1. Undo log

|  |  |  |  |
| --- | --- | --- | --- |
| trx\_id | Rollover\_pointer (指向下一行) | id | name |
| 1 |  | 1 | 张飞 |
| 1 |  | 1 | 关羽 |
| 0 |  | 1 | 刘备 |

因为trx\_id = 1是在m\_ids里面，并且大于等于最小trx\_id,小于最大trx\_id,所以trx\_id为1的数据不可见，只能用0的数据，所以返回刘备。

T8 时刻事务3因为隔离级别是RR,所以readview不变

|  |  |  |  |
| --- | --- | --- | --- |
| m\_low\_limit\_id | m\_ids | m\_up\_limit\_id | m\_creator\_trx\_id |
| 1 | 1，2，3 | 4 | 3 |

Undo log

|  |  |  |  |
| --- | --- | --- | --- |
| trx\_id | Rollover\_pointer (指向下一行) | id | name |
| 2 |  | 1 | 诸葛亮 |
| 2 |  | 1 | 赵云 |
| 1 |  | 1 | 张飞 |
| 1 |  | 1 | 关羽 |
| 0 |  | 1 | 刘备 |

因为trx\_id = 1是在m\_ids里面，并且大于等于最小trx\_id,小于最大trx\_id,所以trx\_id为1的数据不可见，只能用0的数据，所以返回刘备。

T10 时刻事务3因为隔离级别是RR所以readview不变

|  |  |  |  |
| --- | --- | --- | --- |
| m\_low\_limit\_id | m\_ids | m\_up\_limit\_id | m\_creator\_trx\_id |
| 1 | 1,2,3 | 4 | 3 |

1. Undo log

|  |  |  |  |
| --- | --- | --- | --- |
| trx\_id | Rollover\_pointer (指向下一行) | id | name |
| 2 |  | 1 | 诸葛亮 |
| 2 |  | 1 | 赵云 |
| 1 |  | 1 | 张飞 |
| 1 |  | 1 | 关羽 |
| 0 |  | 1 | 刘备 |

因为trx\_id = 1是在m\_ids里面，并且大于等于最小trx\_id,小于最大trx\_id,所以trx\_id为1的数据不可见，只能用0的数据，所以返回刘备。

* + 结论

T6,T8,T10返回的都是刘备

## 题目 02- 什么是索引？

**要点：**

1. 优点是什么？

提高查询效率，减少查询时I/O次数

1. 缺点是什么？

占用更多磁盘空间，降低数据修改效率

1. 索引分类有哪些？特点是什么？

有不同分类办法。

办法一，按索引是否和数据同顺序  
1. 聚簇索引。索引和数据同次序。每张表最多一个。

2. 辅助索引。索引和数据不同顺序。索引内容指向数据主键值。

办法二, 按索引列数

单列索引

1. 主键索引，必须用逐渐，所以一定是唯一的
2. 唯一性索引。不是主键，但也唯一。
3. 普通索引。不唯一。
4. 全文索引。用列中的字符串当索引。
5. 前缀索引。用列中的字符串前缀当索引。

组合索引

1. 有多列。推荐用最左前缀原则。同常推荐用组合索引。
2. 索引创建的原则是什么？

频繁出现在where, order, groupby，频繁出现在select中，多表关联中on的两边， 这些情况下的列建议用索引。

1. 有哪些使用索引的注意事项？

以下情况不要用索引：

数据少

更新频繁

区分度不高

无序字段

同时避免索引过多，而且主键尽量要自增。

1. 如何知道 SQL 是否用到了索引？

Show index from table\_name;

1. 请你解释一下索引的原理是什么？「重点」

索引的原理是通过索引尽量减少查询时I/O次数。抽象在数据结构层面就是键值对。常用实现键值对的数据结构有hash, binary tree, red black tree, B-tree, B+tree

* + 说清楚为什么要用 B+Tree

Hash table: 单个查询快，但无法做区间查询。

Binary tree 和red black tree: 树的高度太高，没法用到I/O时读的数据是按块读的，一个块可以包含多个键值。

B-tree:树的高度低，但跨叶子节点访问时每次都要从跟节点开始。

B+tree: 树的高度更低，跨叶子节点顺序访问时可以直接跳到下一个节点。

综上所述，B+tree综合性能最好。

## 题目 03- 什么是 MVCC？

**要点：**

1. Redo 日志

Redo日志包含事务对数据修改的过程。事务提交时所以redo日志都要先落盘，这样事务数据丢失时可以重播redo日志恢复数据。Redo日志要定期通过checkpoint删除。

Undo日志是用来实现MVCC的，包含每次对数据修改前的快照。这样可以根据不同的隔离级别读到不同的数据。

1. ReadView

每个select语句都会有一个相应的readview,可以知道当前活跃的事务的情况。结合undo日志，就可以算出每次看到的数据版本。

Readview 包含如下数据：

m\_ids：表示在生成ReadView时，当前系统中活跃的读写事务id列表

m\_low\_limit\_id：事务id下限，表示当前系统中活跃的读写事务中最小的事务id，m\_ids事务列表 中的最小事务id

m\_up\_limit\_id：事务id上限，表示生成ReadView时，系统中应该分配给下一个事务的id

m\_creator\_trx\_id：表示生成该ReadView的事务的事务id

1. 如何判断可见性
2. 如果被访问版本的 trx\_id 属性值，小于ReadView中的事务下限id，表明生成该版本的事务在生 成 ReadView 前已经提交，所以该版本可以被当前事务访问。
3. 如果被访问版本的 trx\_id 属性值，等于ReadView中的 m\_creator\_trx\_id ，可以被访问。
4. 如果被访问版本的 trx\_id 属性值，大于等于ReadView中的事务上限id，在生成 ReadView 后才产 生的数据，所以该版本不可以被当前事务访问。
5. 如果被访问版本的 trx\_id 属性值，在事务下限id和事务上限id之间，那就需要判断是不是在 m\_ids 列表中。 如果在，说明创建 ReadView 时生成该版本的事务还是活跃的，该版本不可以被访问； 如果不在，说明创建 ReadView 时生成该版本的事务已经被提交，该版本可以被访问。