

数据库中定义隔离级别的异常

ANSI SQL-92根据三种异常定义了四种隔离级别，其中，三种异常分别名为：**脏读**、**模糊读**、**幻读**。四种隔离级别分别名为：**读未提交**、**读已提交**、**可重复读**、**可串行化**。本文整理了其中对异常的描述。

主要来源：[A Critique of ANSI SQL Isolation Levels](#)

ANSI SQL-92对以上三种异常的描述：

脏读：事务T1修改了一个数据项，事务T2在T1提交或回滚操作前读了这个数据项。如果之后T1回滚，T2即读到了一个没有提交的、不存在的值。

形式化为： $w1(x) \dots r2(x) \dots (a1 \text{ and } c2 \text{ in either order})$

模糊读：事务T1读了一个数据项，事务T2在T1提交前修改或者删除了这个数据项并且提交。如果之后T1重新读该数据项，会读到一个与事务内前一次读不一致的结果。

形式化为： $r1(x) \dots w2(x) \dots c2 \dots r1(x) \dots c1$

幻读：事务T1读了满足某一谓词条件的数据项集合，事务T2在T1提交前创建(插入)了一个新的满足该谓词条件的数据项并提交。如果之后T1再次根据同样的条件读数据集合，会得到与之前不一样(更大)的数据集。

形式化为： $r1(P) \dots w2(y \text{ in } P) \dots c2 \dots R1(P) \dots c1$

Critique对上述三种异常的补充：

在ANSI SQL-92的上述描述中，没有强调T1一定会回滚或者一定会重读，只是简单地陈述了“如果发生”会怎样。

脏读：

考虑这样一个历史： $r1(x=50) \ w1(x=10) \ r2(x=10) \ r2(y=50) \ c2 \ r1(y=50) \ w1(y=90) \ c1$ ，T1在x和y两个账户之间做一个转账40的操作。对应到SQL-92描述，T1没有回滚，但是T2的读也是一种异常，因为两个账户余额总和是错误的。

- 因此有必要考虑将该异常的形式化放宽为： $w1(x) \dots r2(x) \dots ((c1 \text{ OR } a1) \text{ and } (c2 \text{ OR } a2) \text{ in any order})$

模糊读：

考虑这样一个历史： $r1(x=50) \ r2(x=50) \ w2(x=10) \ r2(y=50) \ w2(y=90) \ c2 \ r1(y=90) \ c1$ ，T1对x和y两个账户读取余额。T2在两个账户之间做一个转账40的操作。对应到SQL-92描述，T1没有重新读x这个账户的数据项，但是T1的读也是一种异常，因为读到的账户余额总和是错误的。

- 因此有必要考虑将该异常的形式化放宽为： $r1(x) \dots w2(x) \dots ((c1 \text{ OR } a1) \text{ and } (c2 \text{ OR } a2) \text{ in any order})$

幻读：

考虑这样一个历史：r1(P) w2(insert y to P) r2(z=10) w2(z=11) c2 r1(z) c1，T1针对谓词条件P读取数据项集合，之后再读取满足条件的数据项集合的大小z。对应到SQL-92描述，T1没有重读数据项集合，但是读到的大小z是错误的。

- 因此有必要考虑将该异常的形式化放宽为：r1(P)...w2(y in P)...((c1 OR a1) and (c2 OR a2) in any order)

Critique又补充了另外三种异常(以定义SQL-92四种隔离级别之外新的隔离级别)：

脏写：事务T1修改了一个数据项，事务T2在T1提交或回滚前又写了这个数据项。之后如果T1或者T2回滚，将不确定回滚后的正确值。

直接理解的形式化为：w1(x)...w2(x)...(a1 or a2)

但考虑这样一个历史：w1(x=2000) w2(x=2007) w2(y=10) c2 w1(y=17) c1，假设x和y分别表示年份和年龄，即存在约束x+y=2017。对应到SQL-92描述，T1或T2都没有回滚，但是最后x和y的值不满足约束。

- 因此将该异常的形式化放宽为：w1(x)...w2(x)...((c1 or a1) and (c2 or a2) in any order)

更新丢失：事务T1读了一个数据项，事务T2在T1提交之前(基于读)修改了这个数据项并提交，之后T1(基于之前的读)修改这个数据项并提交，T2的更新就丢失了。

考虑这样一个历史：r1(x=100) r2(x=100) w2(x=120) c2 w1(x=130) c1，x最终的值仅包含T1增加的30。

- 形式化为：r1(x)...w2(x)...w1(x)... c1

读倾斜：事务T1读了一个数据项x，事务T2在T1提交或回滚前更新了x和y并提交。如果之后T1读y，可能读到一个不满足约束的错误状态。

考虑这样一个历史：r1(x=100) w2(x=90) w2(y=90) c2 r1(y=90) c1，假设数据项x和y存在一个约束x=y，T1读到了不满足约束的错误状态。

- 形式化为：r1(x)...w2(x)...w2(y)...c2...r1(y)...(c1 or a1)

写倾斜：事务T1读了数据项x和y，事务T2在T1提交或回滚前也读了x和y，并写了x。如果之后T1写y，T1和T2提交后，x和y的约束可能会被打破。

考虑这样一个历史：r1(x=50) r1(y=50) r2(x=50) r2(y=50) w2(x=-40) w1(y=-40) c1 c2，假设数据项x和y存在一个约束x+y>0，在银行的场景中可能出现，只要账户x和y的加和非负，账户余额允许为负。T1和T2提交后，数据库的状态是错的。

- 形式化为：r1(x)...r2(y)...w2(x)...w1(y)...(c1 AND c2)

extracted by bingo