

在Oracle中使用自治事务保存日志表条目 PDF转换可能丢失图片或格式，建议阅读原文

https://www.100test.com/kao_ti2020/467/2021_2022__E5_9C_A8Oracle_E4_c67_467985.htm 数据库事务是一种单元操作，要么是全部操作都成功，要么全部失败。在Oracle中，一个事务是从执行第一个数据管理语言(DML)语句开始，直到执行一个COMMIT语句，提交保存这个事务，或者执行一个ROLLBACK语句，放弃此次操作结束。事务的“要么全部完成，要么什么都没完成”的本性会使将错误信息记入数据库表中变得很困难，因为当事务失败重新运行时，用来编写日志条目的INSERT语句还未完成。针对这种困境，Oracle提供了一种便捷的方法，即自治事务。自治事务从当前事务开始，在其自身的语境中执行。它们能独立地被提交或重新运行，而不影响正在运行的事务。正因为这样，它们成了编写错误日志表格的理想形式。在事务中检测到错误时，您可以在错误日志表格中插入一行并提交它，然后在不丢失这次插入的情况下回滚主事务。因为自治事务是与主事务相分离的，所以它不能检测到被修改过的行的当前状态。这就好像在主事务提交之前，它们一直处于单独的会话里，对自治事务来说，它们是不可用的。然而，反过来情况就不同了：主事务能够检测到已经执行过的自治事务的结果。要创建一个自治事务，您必须在匿名块的最高层或者存储过程、函数、数据包或触发的定义部分中，使用PL/SQL中的PRAGMA AUTONOMOUS_TRANSACTION语句。在这样的模块或过程中执行的SQL Server语句都是自治的。触发无法包含COMMIT语句，除非有PRAGMA

AUTONOMOUS_TRANSACTION标记。但是，只有触发中的语句才能被提交，主事务则不行。列表A展示了对一个简单但灵活的错误日志表格所作的CREATE TABLE和CREATE SEQUENCE语句。列表B是一个独立的存储过程，用于更新错误日志表格。该过程接受最多三个数字和三个文本变量，然后将它们和时间戳以及调用过程的用户一起储存在表格中。

列表A：CREATE TABLE errorlog (errorlog_id NUMBER,logged_on TIMESTAMP DEFAULT SYSTIMESTAMP,logged_by VARCHAR2(30) DEFAULT USER,num1 NUMBER,num2 NUMBER,num3 NUMBER,text1 VARCHAR2(1000),text2 VARCHAR2(1000),text3 VARCHAR2(1000)).CREATE SEQUENCE errorlog_seqSTART WITH 1INCREMENT BY 1.

列表B：CREATE OR REPLACEPROCEDURE log_error (n1 IN NUMBER:=NULL, t1 IN VARCHAR:=NULL,n2 IN NUMBER:=NULL, t2 IN VARCHAR:=NULL,n3 IN NUMBER:=NULL, t3 IN VARCHAR:=NULL)ISPRAGMA AUTONOMOUS_TRANSACTION.BEGININSERT INTO errorlog(errorlog_id,num1, num2, num3, text1, text2, text3)VALUES(errorlog_seq.NEXTVAL, n1, n2, n3, t1, t2, t3).COMMIT.END.

要测试这个过程，就要更新(UPDATE)或删除(DELETE)表格中的某些行.这就引发了主事务。然后执行存储过程，将您选择的要记入日志的数据传递给它。最后，重新运行主事务，选择(SELECT)错误日志表格，您的日志条目仍会在那儿。

100Test 下载频道开通，各类考试题目直接下载。详细请访问 www.100test.com