

Oracle中通过触发器来追踪用户的活动 PDF转换可能丢失图片或格式，建议阅读原文

[https://www.100test.com/kao\\_ti2020/143/2021\\_2022\\_Oracle\\_E4\\_B8\\_AD\\_E9\\_c102\\_143066.htm](https://www.100test.com/kao_ti2020/143/2021_2022_Oracle_E4_B8_AD_E9_c102_143066.htm)

从Oracle8i开始，Oracle引入了特殊的触发器，这些触发器并不是和特殊的DML事件相关联的（DML事件，如，INSERT,UPDATE和DELETE）。这些系统级别的触发器包括数据库启动触发器，DDL触发器和最终用户登陆/注销触发器。当Oracle提供了这些新的触发器的功能后，却还不清楚该如何使用这些触发器来追踪系统的使用情况。这篇文章描述了我是如何使用创建终端用户登陆/注销的方法来追踪终端用户的活动的。刚开始实行的追踪终端用户系统级触发器是非常新的，就因为它非常的新，所以它在功能上还是不够完善。虽然用户登陆/注销触发器会告诉你用户登陆和用户注销的准确时间，可是代码却无法获取在用户会话阶段所进行活动的任何信息。这些用户的登陆/注销触发器对于使用时间戳用户的应用程序而言是非常有用的。所谓的时间戳用户，就是那些在访问应用程序的时候被给定一个唯一的Oracle用户ID的人。不使用时间戳Oracle用户ID的应用程序使用这些登陆/注销触发器可能没有什么用。既然我们知道了基本的知识，那么，让我们看看该如何设计用户审计表来追踪用户的活动。设计一个用户审计表 第一步就是创建一个用于存储终端用户登陆/注销触发器信息的Oracle表。为了合理地设计这些触发器，让我们先来看看系统级的触发器。首先，我们在登陆的时候就会获取以下信息：用户 ID用户的ID，用于执行登陆操作；会话 IDOracle为用户控制会话的ID；主机计算机的名称；登陆时间一个用户登陆时间一样

的Oracle时间数据类型，精确到0.001秒。现在，我们将获得仅仅优先于用户注销的信息。当用户在注销的时候，Oracle系统级触发器就可以提供这个用户当前的会话状态和活动信息：

- 最后的程序用户在注销时执行的最后一个程序的名称。
- 最后的活动用户在会话中执行的最后一个活动。
- 最后的模块用户在注销前访问的最后一个模块的名称。
- 注销时间一个和用户实际注销时间一样的一个Oracle时间数据类型，精确到0.001秒。

现在，我们知道了登陆和注销的信息，那么我们如何搜集这些信息，并管理它们呢？让我们来看看可用的选择。用户表的标准化因为用户的登陆/注销触发器是两个分离的实体，所以我们在设计支持存放这些信息的表的时候就可以有多种选择。我们可以设计两个隔离的表格：一个用户登陆表和一个用户注销表。如果我们这样做了，那么在把这两个表连接起来的时候和判断哪个用户的登陆引起了哪个用户的注销等等事情的时候就会有困难。这个投机取巧的方法可能会给错误的产生制造机会。诸如，用户24在登陆前是如何注销的？等等。现在，让我们考虑一个更好的选择。为了很好的使用表的信息，我们可以创建一个数据库，它里面包含一个单一的表，用于记录登陆和注销时间信息。这个方法就省去了连接表和关联数据的麻烦。另外，我们还将创建一个区域，用于计算每个用户在特定会话中所用去的时间。这个计算预先由触发器做好，这样就可以节省时间，并且可以创建一个丰富的报表，这个我们后面会提到。设计一个登陆触发器一旦表格设计好了以后，接下来的步骤就是创建一个系统级的登陆触发器，它可以在发生登陆时间的时候记录尽可能多的信息。我创建的登陆审计触发器。正像你看到的一

样，我在这个表中创建了一些在登陆时可以使用到的信息值：  
：用户建立Oracle会话的Oracle用户ID 会话ID使用Oracle的SYS context功能从v\$session 表中直接获取Oracle的会话ID。主机使用Oracle的SYS context功能，在Oracle会话被创建的时候获取主机名。请注意，对于使用Oracle并行服务器或者实际应用群的时候，获取主机名是非常重要的，因为我们可能同时和许多不同的远程主机都有会话连接。登陆时间获取实际工作登陆的时间，精确到0.001秒。请注意我们是如何把登陆时间划分成两个区域的。产生一个登陆日期和一个登陆时间可以给最后的报表带来很强的可读性。既然登陆触发器已经创建好了，我们就需要创建一个注销触发器来获取用户完成会话的所有信息。 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](http://www.100test.com)