

通过缓存数据库结果提高PHP性能之一 PDF转换可能丢失图片或格式，建议阅读原文

https://www.100test.com/kao_ti2020/142/2021_2022__E9_80_9A_E8_BF_87_E7_BC_93_E5_c102_142908.htm 众所周知，缓存数据库查询的结果可以显著缩短脚本执行时间，并最大限度地减少数据库服务器上的负载。如果要处理的数据基本上是静态的，则该技术将非常有效。这是因为对远程数据库的许多数据请求最终可以从本地缓存得到满足，从而不必连接到数据库、执行查询以及获取结果。但当您使用的数据库与 Web 服务器位于不同的计算机上时，缓存数据库结果集通常是一个不错的方法。不过，根据您的情况确定最佳的缓存策略却是一个难题。例如，对于使用最新数据库结果集比较重要的应用程序而言，时间触发的缓存方法（缓存系统常用的方法，它假设每次到达失效时间戳记时就重新生成缓存）可能并不是一个令人满意的解决方案。这种情况下，您需要采用一种机制，每当应用程序需要缓存的数据库数据发生更改时，该机制将通知该应用程序，以便该应用程序将缓存的过期数据与数据库保持一致。这种情况下使用“数据库更改通知”（一个新的 Oracle 数据库 10g 第 2 版特性）将非常方便。“数据库更改通知”入门“数据库更改通知”特性的用法非常简单：创建一个针对通知执行的通知处理程序 一个 PL/SQL 存储过程或客户端 OCI 回调函数。然后，针对要接收其更改通知的数据库对象注册一个查询，以便每当事务更改其中的任何对象并提交时调用通知处理程序。通常情况下，通知处理程序将被修改的表的名称、所做更改的类型以及所更改行的行 ID（可选）发送给客户端监听程序，以便客户端应用程序

序可以在响应中执行相应的处理。为了了解“数据库更改通知”特性的作用方式，请考虑以下示例。假设您的 PHP 应用程序访问 OE.ORDERS 表中存储的订单以及 OE.ORDER_ITEMS 中存储的订单项。鉴于很少更改已下订单的信息，您可能希望应用程序同时缓存针对 ORDERS 和 ORDER_ITEMS 表的查询结果集。要避免访问过期数据，您可以使用“数据库更改通知”，它让您的应用程序方便地获知以上两个表中所存储数据的更改。您必须先将 CHANGE NOTIFICATION 系统权限以及 EXECUTE ON DBMS_CHANGE_NOTIFICATION 权限授予 OE 用户，才能注册对 ORDERS 和 ORDER_ITEMS 表的查询，以便接收通知和响应对这两个表所做的 DML 或 DDL 更改。为此，可以从 SQL 命令行工具（如 SQL*Plus）中执行下列命令。

```
CONNECT / AS SYSDBA. GRANT CHANGE NOTIFICATION TO oe. GRANT EXECUTE ON DBMS_CHANGE_NOTIFICATION TO oe.
```

确保将 init.ora 参数 job_queue_processes 设置为非零值，以便接收 PL/SQL 通知。或者，您也可以使用下面的 ALTER SYSTEM 命令：ALTER SYSTEM SET "job_queue_processes"=2. 然后，在以 OE/OE 连接后，您可以创建一个通知处理程序。但首先，您必须创建将由通知处理程序使用的数据库对象。例如，您可能需要创建一个或多个数据库表，以便通知处理程序将注册表的更改记录到其中。在以下示例中，您将创建 nfresults 表来记录以下信息：更改发生的日期和时间、被修改的表的名称以及一个消息（说明通知处理程序是否成功地将通知消息发送给客户端）。CONNECT oe/oe. CREATE TABLE nfresults (operdate

DATE, tblname VARCHAR2(60), rslt_msg VARCHAR2(100)).
在实际情况中，您可能需要创建更多表来记录通知事件以及所更改行的行 ID 等信息，但就本文而言，nfresults 表完全可以满足需要。使用 UTL_HTTP 向客户端发送通知您可能还要创建一个或多个 PL/SQL 存储过程，并从通知处理程序中调用这些存储过程，从而实现一个更具可维护性和灵活性的解决方案。例如，您可能要创建一个实现将通知消息发送给客户端的存储过程。“清单 1”是 PL/SQL 过程 sendNotification。该过程使用 UTL_HTTPPL 程序包向客户端应用程序发送更改通知。

清单 1. 使用 UTL_HTTP 向客户端发送通知

```
CREATE OR REPLACE PROCEDURE sendNotification(url IN VARCHAR2, tblname IN VARCHAR2, order_id IN VARCHAR2) IS req UTL_HTTP.REQ. resp UTL_HTTP.RESP. err_msg VARCHAR2(100). tbl VARCHAR(60). BEGIN tbl:=SUBSTR(tblname, INSTR(tblname, ' . ', 1, 1) 1, 60). BEGIN req := UTL_HTTP.BEGIN_REQUEST(url||order_id|| ' & . ' || ' table= ' ||tbl). resp := UTL_HTTP.GET_RESPONSE(req). INSERT INTO nfresults VALUES(SYSDATE, tblname, resp.reason_phrase). UTL_HTTP.END_RESPONSE(resp). EXCEPTION WHEN OTHERS THEN err_msg := SUBSTR(SQLERRM, 1, 100). INSERT INTO nfresults VALUES(SYSDATE, tblname, err_msg). END. COMMIT. END. /
```

如“清单 1”所示，sendNotification 以 UTL_HTTP.BEGIN_REQUEST 函数发出的 HTTP 请求的形式向客户端发送通知消息。此 URL 包含 ORDERS 表中已更改行的 order_id。然后，它使用 UTL_HTTP.GET_RESPONSE 获取

客户端发出的响应信息。实际上，sendNotification 并不需要处理客户端返回的整个响应，而是只获取一个在 RESP 记录的 reason_phrase 字段中存储的简短消息（描述状态代码）。

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