

数据库应用程序常见T-SQL和PL_SQL PDF转换可能丢失图片或格式，建议阅读原文

https://www.100test.com/kao_ti2020/167/2021_2022__E6_95_B0_E6_8D_AE_E5_BA_93_E5_c102_167123.htm 许多软件公司都理解开发不依赖于特定数据库类型(例如，Oracle，SQL Server，DB2)的应用程序的重要性，它可以让客户们选择自己习惯的平台。一般来说，软件开发人员都能够识别出他们的负责数据库维护的客户，和必须使用现有平台和个性化的客户。关于Oracle和SQL Server之间的区别，已经有很多篇文章从企业的角度和数据库管理员的角度出发描写了两者之间的一般差别。在这篇文章里面，我将会从应用程序的角度向你描述SQL Server 和Oracle平台之间的区别，并且将会讨论开发不依赖于数据库环境的应用程序的几种可能的方法。与此同时，我不会再讨论对于应用程序来说，两个平台之间显而易见的区别，例如表的分割和索引。定义通用接口和语言 有很少的几种通用语言和接口可以让应用程序不依赖于数据库，想来也可以以同样的方式应用在关系型数据库上面: ANSI是美国国家标准局定义的，这是一家志愿者成员的组织(用私人基金运转)，他们在有关设备和程序等广泛领域内开发了国家承认的标准。在数据库领域，ANSI定义了编写SQL 命令的标准，假设命令可以运行在任何的数据库上，而不需要更改命令的语法。ODBC是开放数据库连接(ODBC)接口，微软定义的，它可以让应用程序访问数据库管理系统(DBMS)中的数据，使用SQL 作为访问数据的标准。ODBC允许最大的互联性，这意味着一个单个的应用程序可以访问不同的数据库管理系统。然后，应用程序终端用户可以添加开放数据库连

接(ODBC)数据库驱动来将应用程序与他们所选择的数据库管理系统链接上。OLEDB，是ODBC的继承者，是一组允许例如基于VB，C，Access等任何可以连接到类似SQL Server，Oracle，DB2，MySQL等后台的“前台”的软件组件。在许多种情况下，OLEDB组件都比原有的ODBC提供了好得多的性能。JDBC(Java数据库连接)应用程序接口是Java编程语言和广泛范围的数据库，SQL数据库和其他表列数据源(例如电子表格或者普通文本文件)之间，进行不依赖于数据库的连接的行业标准。JDBC应用程序接口为基于SQL的数据库访问提供了调用级别的应用程序接口。真实世界中的通用接口不幸的是，并不是所有数据库级别的命令都是ANSI，每个数据库平台都有自己的扩展功能。ANSI，或者通用接口，一般来说都代表着几本功能，因此也可能意味着丧失了性能方面的竞争力。对于小型数据库和小型应用程序来说，要维护对数据库的通用访问是简单的，但是当数据库和/或应用程序变得越来越大，越来越复杂，你就不得不向代码中添加功能。示例: 在两个平台上以同种方式编写的命令: Insert into Table_1 values (1,Michelle) Update Table_2 set Col_1 = 2 Delete from Table_3 where Col_3 like Michelle% 在两个平台上不是以同种方式编写的命令: Select getdate() Select case Fld when 1 then a When 2 then b Else c End From Table_4 Oracle Select sysdate from dual Select DECODE (Fld, 1, a, 2, b, c) From Table_4 下面的两篇文章包括了对Oracle PL/SQL命令和T-SQL命令比较的列表: 从Oracle移植到SQL Server 开始学SQL: SQL Server和Oracle之间的区别 可能的解决方案: 我曾见过几种对数据库互操作性问题的可能的解决方案: # 解决方案描述 优点 缺点 1 处理两种版本的应用程序

一个是针对Oracle的，另一个是针对SQL Server的。

- 1.不需要处理SQL命令的版本
1. 复制代码必须在两个版本上应用所有的变化。
- 2 尽可能地使用通用语言（ANSI/ODBC/OLEDB/...），并且在应用程序用IF命令处理不同的命令
1. 处理单个应用程序
1. 如果有非ANSI的命令，代码就会变得越来越大, 这会影响应用程序的性能
2. 代码可能会因为很多的IF 语句变得复杂。
- 3 把数据库命令保存在数据库或者INI文件中，当应用程序开始的时候将其读入缓存。
1. 应用程序中不需要 IF命令了。
2. SQL 命令可以任意修改，并且不需要在修改之后重新编译应用程序。
1. SQL命令管理可能变得更加复杂。

选择哪个解决方案? 这个问题的答案依赖于应用程序的特性和平台。每个解决方案都易于实现，这里没有最好的解决方案。结论: 如果你想要将你的应用程序开发成为不依赖数据库的，你就应该仔细的计划解决方案。考虑到应用程序在数据库级别上的复杂性，还有所需代码的总量。在计划过程中，考虑到应用程序未来的发展是至关重要的。

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