经验之谈：使用Oracle的TDE特性加密 PDF转换可能丢失图片或格式，建议阅读原文 https://www.100test.com/kao_ti2020/645/2021_2022__E7_BB_8F_E9_AA_8C_E4_B9_8B_E8_c102_645627.htm 使用作为 Oracle 高级安全选件（版本 10.2 及更高版本）的一部分引入的 Oracle 数据库透明数据加密（TDE），可以有选择地对保留在数据库底层数据文件中的敏感数据库数据以及所有下游文件组件（如联机重做日志、归档重做日志和数据库备份）进行加密。 TDE 的基本目标是保护在这些原始操作系统文件中发现的敏感数据，防止不怀好意的人访问磁盘或备份磁带时对这些数据进行窥探，然后尝试还原数据库或扫描原始操作系统文件中的数据，如个人可识别信息或信用卡信息。 作为我的咨询惯例的一部分，我已经实施 TDE 多次。但是，在其中一个最近的合约之前，我一直使用 TDE 对现有表中的新列或属于全新表的列进行加密。在这两种情况下使用 TDE 非常简单，因为目标列为空，因此由于缺乏数据和现有应用程序相关性而不会涉及较大的风险。 我最近实施 TDE 的体验有所不同。我帮助一家大型公司对一个已超过一百万行的表中的现有列进行加密。还有一个依赖于列的关键任务应用程序，因此，您可以设想一下，在开始工作之前有很多重要的事情要考虑。在 Internet 上搜索可提供经验的类似情形之后，我发现只有几个优秀的资源可以帮助我。 本文概述了我在通过使用 TDE 对现有数据进行加密的过程中总结出的经验教训。如果您尝试对现有列数据使用 TDE，我希望此处提供的信息可帮助您迅速有效地开展类似工作。 确定可能的限制 研究客户的系统时，我做的第一件事情就是查找与目标列有关的将禁止我们

对列加密的数据模型特征，或者查找可能对现有操作产生负面影响的有关列的事项。该研究包括查找列索引和完整性约束。 正如 Oracle 文档明确声明，当您想对具有索引的某个列进行加密时，需要了解很多限制条件。Oracle 不允许对具有位图索引的列进行加密，这与我们的情况没有密切关系。但是，目标列具有多个普通的（B 树）索引。尽管 Oracle 允许对具有普通索引的列进行加密，但是 Oracle 禁止对索引列进行“ salt 处理 ”加密。Salt 处理通过在加密之前向数据添加随机字符串来提高重复数据的安全性，因此窃贼使用模式匹配识别技术更加难于破解加密的数据。总而言之，经过这个最初的分析之后，我们会遇到一种情况，那就是我们可以对列进行加密，但不能进行 salt 处理。 对列索引进行分析后，我本可以到此为止，但是我想回答的下一个问题是“ 使用这些索引合适吗？ ”我的思考过程是这样：如果索引没有用，那么我会将其删除，从而减少维护索引条目所必需的系统开销，尤其是考虑到加密的额外负担。要判断索引是否有用，我使用 Oracle 数据库的索引监视特性。我发现，实际上索引正处于使用当中，因此我们必须对其继续进行维护。 接下来，我查看了引用完整性约束条件中是否涉及目标列。由于每个表都具有其自己的加密密钥，因此 Oracle 不允许您使用 TDE 对外键关系中涉及的列进行加密。在我们的情况下，引用完整性约束条件中未涉及目标列。 评估性能开销 我的客户询问的第一组问题之一就是“ TDE 对我的应用程序的一般性能影响如何？ ”Oracle 文档中有一小部分论述了一般情况下 TDE 对相关应用程序性能的影响。但是我的客户希望获得一些具体的统计信息，以帮助他们了解 TDE 如何影响日常进行的有

严格时间要求的数据加载过程。 为了满足客户需求，我计算了每天在有严格时间要求的过程中插入到目标表中的平均行数。然后，我在客户端的相同沙箱环境中创建了一个类似的测试表和索引，测量在加密目标列前后插入相同数量的行所花费的时间。时间消耗上的差别让我们更好地了解了在该过程中对列数据进行加密所造成的“ 性能损失 ”。列表 1 是我如何使用 SQL*Plus 执行该操作的示例。 