

企业数据管理解决方案

2008年4月

IBM Information Management software



## 改进应用测试的企业战略

---

目录

---

**2 执行摘要**

**3 交付可靠的应用**

**5 应用测试为何成为了挑战?**

**11 支持您的应用测试需求**

**12 IBM Optim 可以应对挑战**

**15 创建目标测试环境**

**18 通过改进测试数据管理创造**

**执行摘要**

保证关键业务应用的准确性、可靠性和质量从未变得像今天这样重要。原因何在? 因为各个行业公司依赖于任务关键型企业应用来驱动他们的业务计划。反过来, 这些应用也依赖于关系数据库来保存和管理基本的企业数据。增强、维护、定制和升级这些完善应用的能力是实现长期业务目标的关键。很多公司正在加快部署可靠的高质量应用的速度, 同时严格控制预算。

现在, 公司在为企业应用设计有效和高效的测试战略时面临着前所未有的挑战。测试数据不完整或有缺陷意味着测试结果不准确, 这会导致应用故障和业务中断。构建测试环境更常用的方法包括克隆应用生产环境和编写定制的提取程序。然而, 这些方法是劳动密集型的, 容易出错而且代价不菲。没有公司愿意冒由于交付的应用测试不彻底, 从而导致丢失客户、市场份额、品牌资产或收入的风险。因此, 端到端的应用测试在整个应用开发生命周期里具有战略意义上的优先地位。

那么, IT 企业如何提高测试效率并降低拥有和维护企业应用的总体成本呢? 本白皮书描述久经考验的测试数据管理功能如何帮助您交付可靠的应用, 并从应用投资中获得最大的业务价值。

### 交付可靠的应用

如今的应用担负着推动收入和让复杂的市场计划取得满意效果的责任。这些面对客户的任务关键型应用具有巨大的战略性影响。结果，对于应用的准确性、可靠性和性能的关注不断升级，以便让宕机时间最小化并提高客户忠诚度。

根据 **Computerworld** 杂志的说法，“漏洞百出的软件造成的损失将可能高达数十亿美元。”<sup>1</sup> 软件缺陷将直接影响公司的底线收入。无法在各个开发阶段准确地测试应用可能导致软件停止工作。您的业务能够承受由于应用故障而引起的收入下降和客户丢失吗？

改进管理测试环境中的企业应用数据的方式可以带来很好的效果，包括可靠性提高，部署时间缩短和开发成本降低。

- **加快部署速度。**及时部署可靠的应用、增强功能和升级可以为公司带来竞争优势。缩短升级和开发周期是公司想要看到的结果，但是可靠性测试欲速则不达。这两个业务目标之间的传统斗争通常意味着，部署计划与投入使用日期不符，或者质量测试的水平不够。如果能在不牺牲准确性和可靠性的前提下，加快应用面市的速度，这对于业务的重要性是不言而喻的。
- **控制开发和质量保证的成本。**降低开发、测试和交付可靠应用的总体成本，可以在应用开发的整个生命周期内节约大量成本。

通过减少 DBA 和质量保证测试人员的工作量，让他们能够腾出时间解决其他业务问题，公司可以直接降低成本和提高生产力。如果能够降低开发成本和更有效地使用有用的 IT 资源，您的业务就能够从中获益。

- **尽早找出并解决应用错误。**在生产环境中排除应用错误的相关成本，是在应用开发和测试过程早期排除这些错误的成本的 10 到 100 倍，甚至更糟，因为您的业务用户和客户是最先发现应用错误的，所以您会面临收入减少和客户忠诚度受损的风险。您的公司是否能够承受在发布不完善的软件后，直接费用和非直接的机会成本都升高的后果呢？
- **降低基础设施成本。**在应用开发与测试环境中，一般备有 4 份以上原始生产数据库的副本。复制数据时，成本就会按比例增加。使用“规模适中的”而非完整的生产数据库副本，有助于节约容量资源。另外，通过减小测试数据库的大小，还可以减少平台测试行为所需的处理能力。使用较小的、规模适中的应用测试数据库能否给你的业务带来好处？

IT 企业意识到需要更多计划才能实现严格的测试目标，这是交付可靠的应用升级与增强的必要条件。设计一种全面的测试战略通常被视为是理所当然的。企业还会认识到，它们需要久经考验的测试数据管理功能，才能以较低的成本改进质量保证过程。

很多企业都不愿意在测试上投入大量资金，而更愿意将有限的预算资源投入到其他方面。然而，如果在质量保证方面没有保障，业务后果将会很严重。必须了解当应用的准确度和可靠性不可接受时，会给关键业务目标造成的影响，比如在产生收入和确保客户满意度方面。

使用有效的测试数据管理功能和可重复的测试过程可以保护您在定制和打包应用中的投资，最终将提高您的投资回报。在现今的市场中，可靠性和质量是保持竞争力的主要竞争优势，因此必须把在部署之前彻底测试应用摆在首位。

#### **应用测试为何成为了挑战？**

通常，每次开发新应用或修改现有应用时，都要创建新的测试数据库。因为使用“真实”测试数据的效果最好，测试数据库往往就是对生产数据库的克隆副本。克隆整个生产数据库会增加运行测试用例所需的时间，因为数据量更大了。如果使用数量较少但能准确反映生产数据的真实数据子集进行测试，不仅速度更快，而且不会增加测试过程的开销。另外，克隆后的数据可能不支持有效测试需要的特定错误和边界条件。在测试开始之前可能要加入特殊的测试用例。

从单元测试到系统整合和验收测试的每个阶段，它们的要求和复杂程度都是各不相同的。迭代测试的过程包括使用测试数据库执行应用，以及通过检查结果来验证应用的功能是否符合设计。在测试继续之前，所发现的任何问题都必须得到解决，而且必须刷新测试数据。

此过程在所有测试阶段（单元、整合、系统、负载、衰退和验收测试）都将重复，直到应用迁移到生产环境中。最后，在执行测试后，验证结果没有轻松或自动的方法。除了在部署前找出并解决应用错误之外，另一个重要的目标是创建一个可重复的测试过程，能够改进应用质量，缩短面市时间并让成本降到最低。

使用关系数据进行测试的复杂性。大多数应用都依赖于关系数据库技术的这个事实，给处于测试过程中的企业带来了一个很大的挑战。应用数据模型可以包含几十、数百或者甚至数千个表，它们之间的相互关系的数量级也一样。数据模型的复杂性并不只限于大型系统。即使是几个表所包含的关系也会让定位数据模型变得困难。

如果没有正确的解决方案，开发人员通常需要编写复杂的提取程序才能创建测试数据。在众多表、行和列之间来回导航，以创建、操作和刷新目标数据库子集，这是一个很大的挑战。开发一个充分考虑到应用定义和实施的关系的提取程序也绝不轻松。最后，只要更新应用或数据库，提取程序也必须随之更新。

支持各种数据库管理系统和不同数据模型也增添了复杂性。例如，CRM 数据可能在 Oracle® 数据库中管理，而相关的帐单数据则在 IBM® DB2® 数据库中管理。显然，全面的测试功能必须支持这些关系复杂性，并“记住”在每次提取、比较或更新操作中对它们进行说明。

**测试数据来自多个数据库。**大多数企业都会把数据保存在各种关系数据库中，比如 DB2、Oracle、Microsoft® SQL Server®、Sybase® 和 IBM Informix®。另外，数据也可以以层次或非关系格式保存，比如 IBM VSAM® 文件和 IBM IMSTM 数据库。测试复杂的应用时，要求测试数据同时来自关系和非关系数据源的多个关系数据库的情况很常见。所有数据库管理系统处理数据的方法都不同。各种系统的整合是一个典型的需求。质量测试还必须包括无缝的功能，可以处理来自运行在不同平台上的不同数据库管理系统的数据库。

**测试来自同构数据库的数据。**在同构的数据库环境中，技术都是相同或兼容的。在同构数据库环境中创建测试数据的难点之一，就是提取相关数据的完整子集并保持这些数据的引用完整性。必须导航数据模型中的所有关系，无论它们是定义在数据库还是应用中（图 1）。

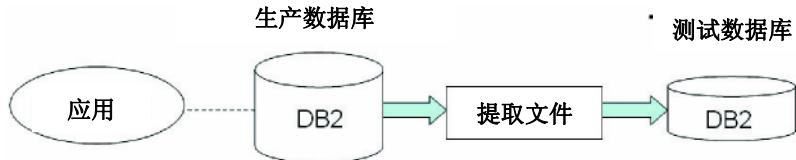


图 1. 使用同构测试数据测试应用

**测试来自异构数据库的数据。**异构数据库环境可能包含不同的硬件、操作系统和数据模型，以及语法和语义上的差别。从异构数据库创建测试数据增加了数据复杂性和转换问题。通常，将应用移动到不同的数据库或测试环境中时会发生这种情况。例如，可能必须先从 **DB2** 生产数据库中提取数据子集，然后再将数据插入到 **Oracle** 测试数据库中。这种转变可能导致特殊注册程序或日期与时间戳出现错误，因为每个 DBM 对它们的处理方式各不相同（图 2）。

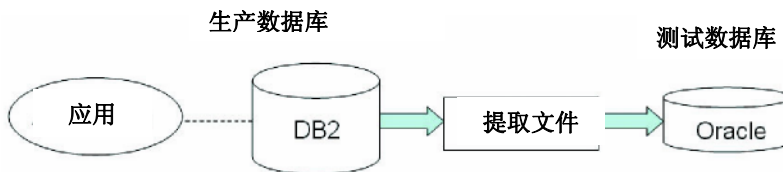


图 2. 使用异构测试数据测试应用

除了提取引用上完整的数据子集之外，处理异构数据库时，还需要另外的功能来自动管理兼容性和转换差异的问题。

**来自多个关系数据库的联合测试数据。**联合环境通常需要提取保存在多个关系数据库中的相关测试数据，这又增加了一层复杂性。例如，工资数据在 **DB2** 数据库中管理，而员工数据则在 **Oracle** 数据库中管理（图 3）。



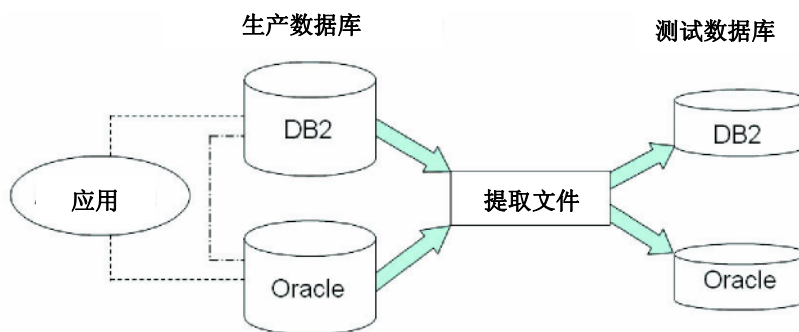


图 3. 使用联合测试数据测试应用.

除了维护引用完整性和处理数据兼容性与转换问题之外，创建实际测试数据现在还需要联合数据访问功能，这样才能使用同一个提取过程从多个数据库中提取相关数据。

**来自非关系与关系数据库的联合测试数据。**关系数据库天生具有复杂性，如果要测试的应用同时引用了非关系和关系数据库，将会更加复杂。例如，一个订单输入应用可能需要 VSAM 库存文件中的产品数据和 DB2 订单项表中的客户数据。VSAM 文件或 IMS 数据库中保存的数据具有不同的结构，无法与关系数据轻松整合（图 4）。

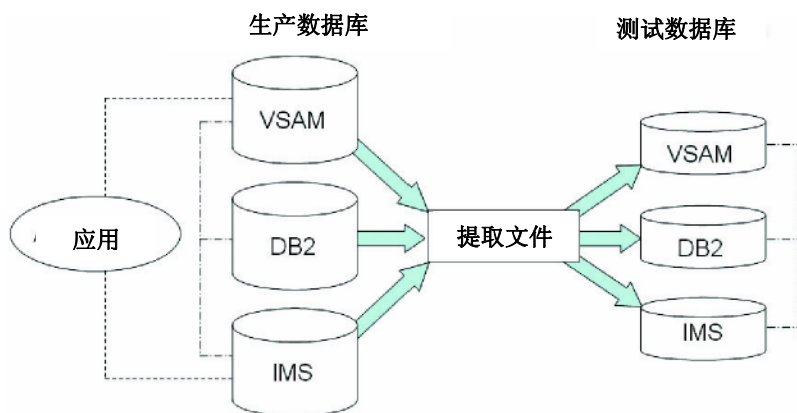


图 4. 使用联合的非关系与关系测试数据测试应用

在这些场景中，一种有效的测试战略必须能够为来自非关系与关系数据库的真实联合测试数据创建精确子集。因为数据库结构显著不同，跨多个复杂的非关系与关系数据源关联数据成为了一个很大的难题。要解决这个难题，需要一种能够访问 VSAM 和 IMS 数据，并通过容易“理解”和管理的方式关联这些数据的工具。

因为 VSAM 和 IMS 数据库的灵活性不够，如果不能指定简单或复杂的选择标准，很难从这些数据库中提取出数据子集。如果能够动态地应用选择标准来提取 VSAM 或 IMS 数据，就能极大地减少处理开销并提高测试数据的质量。

### 支持您的应用测试需求

那么，如何才能改进应用测试呢，尤其是在测试数据的质量至关重要的时候？

- **创建规模适中的测试数据库。**使用来自多个数据源的、引用上完整的真实生产数据子集，可以提高测试的准确性和有效性，并降低对容量的要求。子集化功能允许您创建的真实测试数据库既小到足以支持测试的快速运行，同时又大到足以准确反映生产数据的多样性。在整个测试周期内以迭代方式处理预定义的真实数据子集和刷新测试数据，可以提高测试和总体应用的质量。保存和重用处理规范的能力不仅简化了测试过程，同时提供了一个一致的和易于管理的测试环境。
- **使用真实的测试数据。**构建和填充真实的测试数据库，要求有足够容量，可以基于规范提取和移动相关数据库的精确子集。维护引用完整性的百分之百准确是基本要求，特别是对于最复杂的应用数据模型而言。在提取过程中包含元数据可以复制测试环境，并快速而准确地适应数据模型的变化。
- **屏蔽敏感数据以保护隐私。**保护应用开发和测试环境中的数据隐私，对于防止个人可识别信息的误用和遵守全球数据隐私法规和行业标准来说是有必要的。去标识化机密数据的功能允许您保护隐私，同时为开发人员和质量保证人员提供真实的测试数据。

- **提高测试效率。**由于在迭代测试过程中要修改测试数据库的内容，它与基准测试数据的分歧将进一步拉大，从而导致测试环境无法达到最佳状态。如果没有保存和刷新测试数据的功能，重用测试数据库来提高后续测试运行结果的准确性就会更加困难。保存和重用提取、插入和加载处理规范的功能，有助于简化测试过程和维护一个一致且易于管理的测试环境。
- **创建目标测试场景。**创建测试数据以测试错误和边界条件，需要智能浏览功能和关系编辑功能。关系浏览功能增强了清晰地查看数据关系和快速找出并解决问题的能力。插入行和编辑数据库表的功能直接提高了生产力和准确度。多层次撤消功能也是必不可少的。
- **自动比较测试结果。**如果能够在测试期间找出数据的异常和不一致，对于应用的总体质量是大有帮助的。真正实现这个目标的唯一途径是部署一项自动功能，用于比较基准测试数据和连续测试的运行结果。速度和准确性都是必要的。自动比较处理不仅节省了时间，而且有助于找出原本检测不到的问题。

#### **IBM Optim 可以应对挑战**

IBM® Optim™ 测试数据管理解决方案使企业能够应对甚至最复杂的应用测试的挑战，它提供了有效的测试数据管理战略的所有基本组件：

- 提取准确无误且在引用上完整的数据子集，以创建真实的测试数据库，无论涉及到的表和关系数量有多少。

- 插入或加载相关数据的子集，以便快速构建真实的测试数据库。一致地更新或刷新测试数据以保持测试环境的完整性。
- 在测试环境中去标识化敏感数据，以便符合数据隐私的制度遵从要求。转换测试数据以满足特定测试用例的要求。
- 浏览和编辑测试数据以发现错误条件和解决问题。在数据的关系业务上下文中检查数据可以更清晰地了解数据模型。
- 在测试应用之前和之后比较测试数据，以便精准地自动验证期望的测试结果和找出异常。
- 整合来自多个关系数据库和数据库管理系统（DB2、Oracle、Sybase、SQL Server 和 Informix）的数据。
- 整合关系与非关系数据，以创建一个联合测试环境。
- 将测试数据管理与自动化测试整合，以便提供全面的测试功能。

当这些功能就绪之后：

- 开发人员可以在单元测试中验证新的应用功能是否达到预期，并在整合测试中确认修改没有引入问题。
- 质量保证人员可以验证整个系统是否按照预期正常运转，以及与其他系统的交互是否正确。

- 业务单元用户可以在验收测试中确保系统在功能和性能方面都达到了他们的预期。

开发人员和质量保证人员可以花费较少的时间提取数据，用于创建和维护开发与测试环境，而将更多时间用在性能调整、备份和恢复处理，以及管理生产数据库上。

**Optim** 的独特优势在哪里？**Optim Relationship Engine™** 是一项独一无二的技术，可以理解和处理来自多个表的相关数据，并保证每个测试子集始终是引用完整和逻辑完整的。例如，在客户数据的一个子集中，一个客户可能有一项订货不足，而另一个客户没有。对于不同的客户，从任意数量的表检索到的行数都会发生变化。但是 **Relationship Engine** 始终能为正确的客户找到正确的项，而且每次都完整无缺。

**Optim** 使用一个活动存储库来保存用户定义的业务规则。该库将自动捕获为数据库定义的关系信息。其他数据模型信息可以从第三方的字典或数据建模软件进行填充。最后，用户可以定义存在但 **DBM** 不知道的关系，比如应用管理的关系。

使用活动存储库时，它的工作成员可以定义、共享和重用不同的访问定义，用于从本质上指定数据库的不同子集——来自很多表的相关行。**Optim** 的 **Relationship Engine** 技术将来自每个表的所有相关行组装为一个引用上完整的子集。

**Optim** 功能允许用户迁移、浏览、插入、加载、转换、编辑和比较相关数据的完整子集。企业可以设计全面的测试战略，其中包括真实的测试数据，同时提高生产力和总体的应用质量，并降低开发成本（图 5）。

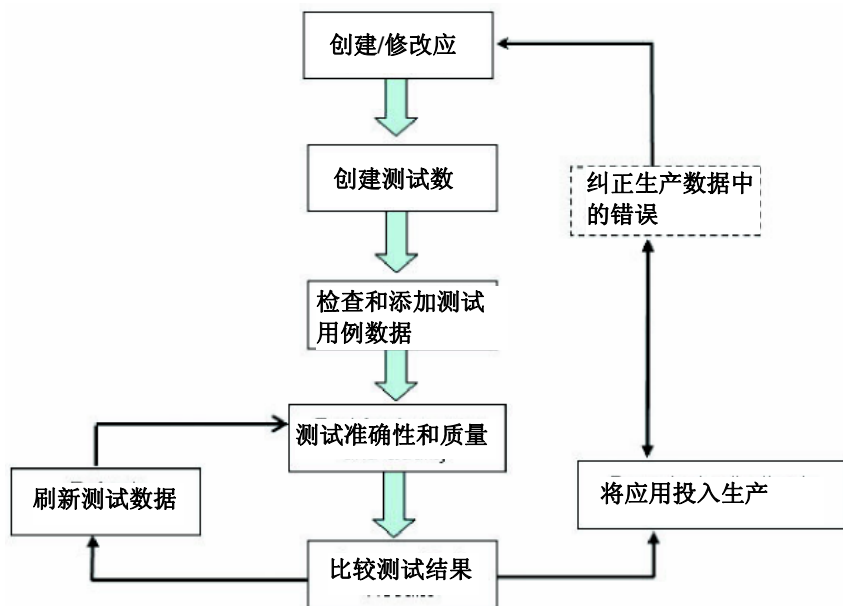


图 5. IBM Optim 改进了应用测试过程的每个阶段

### 创建目标测试环境

通常，仅仅为了测试目的而克隆包含数百个高度相关的表的整个生产数据库是不切实际的。首先，容量就是个问题。其次，还有质量问题——使用大型的测试数据库时，开发人员会发现很难跟踪和验证特定测试用例。

创建真实的测试数据。无需克隆副本，Optim 的子集化功能支持创建真实的、规模适中的、易于管理和维护的测试数据库。借助 Optim，您可以从真实的应用测试数据中提取出引用上完整的子集，而无需编写定制程序。只要指定能够在测试特定应用功能时提取出所需相关数据的精确子集的选择标准，使用能够准确反映应用生产数据的、较小而且真实的子集进行测试的速度要快得多，同时不会增添测试过程的开销。

**保持测试数据的引用完整性。** Optim 遵从数据库和应用内实施的引用完整性 (RI) 规则。一般而言, 应用实施的 RI 更为复杂。例如, 应用包含的关系可能使用兼容但不完全一样的数据类型, 组合和部分列, 以及数据驱动的关系。Optim 处理每类关系的功能已经经过了实践检验。Optim 提供了足够的灵活性来构建关注的测试数据库, 以便验证需要测试的应用模块的功能和性能。这些功能可以支持您从单元测试到系统整合和验收测试的每个阶段的需求。

**检查错误和边界条件。** 从生产数据库创建相关测试数据的真实子集是一个合理的开始, 但是有时候必须编辑数据才能满足特定的错误条件或者验证特定的处理功能。Optim 包含了全面的关系编辑功能, 不仅可以简化创建这些特殊数据所必需的任务, 而且还让浏览数据和解决应用问题变得更加轻松。在其关系或业务上下文中浏览数据的能力允许用户清楚地查看数据关系和应用数据模型的结构。

**屏蔽和转换测试数据。** 随着对于数据隐私的越来越重视, 在开发与测试环境中转换和去标识化敏感数据的能力, 对于防止数据损坏和出现严重后果是至关重要的。Optim 提供了映射源列和目标列的技术, 以及使用各种转换函数和算法的功能。

例如, 只要应用一个随机数函数, 就能轻松去标识化客户的身份证号码。更加完善的屏蔽功能允许您使用子字符串、序列号、日期失效、币种换算或表查找功能函数。包含用户定义的数据转换程序的功能甚至提供了更高的灵活性, 以满足复杂或网站特定的要求。



上面描述的所有方法都是去标识化测试数据的有效技术。然而，在测试关系数据库应用的过程中，还存在另一个复杂之处。具体来讲，用户需要能够将父表内的已屏蔽数据元素传播给数据库中所有相关子表。为了维持转换后数据的引用完整性，键传播是必不可少的。否则，父表与子表之间的关系将被切断，测试数据变得不准确，而应用测试也会产生无效的结果。

键传播还可以提供“生产”新的测试数据集合的能力。当测试需要另外的数据时，这种技术就可以派上用场，例如测试新应用或者当在数据模型中加入新表之后测试应用的时候。

验证测试结果。如果没有在测试运行之前和之后比较测试结果的功能，验证测试结果并找出差异也就无从谈起。首先，存在各种变化：插入、删除和更新将在数以百计的表上展开。其次，可能存在意料之外的问题（例如，孤儿）和其他在手动分析的情况下检测不到的异常。

Optim 具有自动比较相关数据的子集并找出差异的功能。对于关系数据，这不仅意味着逐行比较，它还意味着使用数据模型只能比较相关的行集合。例如，开发一个新的成品率表后，您可以比较“之前”和“之后”的测试结果，以验证成品率的变化是否得到了正确应用。

**保持测试数据的质量。**测试环境的质量与测试数据的质量同样重要。测试数据库的内容在测试过程中将被修改。随着时间的推移，它与基准测试数据之间的差异进一步变大，从而导致测试环境无法达到最佳状态。Optim 提供保存和重用测试数据的功能，这样刷新测试数据库就会更容易和更快，从而保证后续测试运行结果的准确性。

Optim 的提取过程还包括元数据，这使得适应测试阶段数据模型中的变化变得更加容易。元数据是定义性的数据，用于提供在应用或环境中管理的数据结构的相关信息。例如，元数据将记录数据库的结构，包括表、列、关系、视图、触发器等。

在每个测试周期内循环使用预定义的真实数据子集和刷新测试数据，可以提高测试和总体应用的质量。在提取处理中包含元数据支持重新创建测试数据，以及快速而准确地适应数据模型的变化。保存和重用处理规范的能力简化了测试过程，并有助于获得一个一致和易于管理的测试环境。

#### **通过改进测试数据管理交付业务价值**

您在企业应用及其支持基础设施上投入了数百万美元的资金，目的是获得最佳的运营绩效，改善决策过程和赢得竞争优势。**Optim** 提供了企业数据管理的强大功能，让您能够推动企业交付最大的业务价值。

**Optim** 提供的解决方案使您能够在信息生命周期的每一个阶段管理企业应用数据。现在可以按照年龄和状态来评估和划分应用数据，还可以应用业务规则对企业数据实行存档、求子集、访问、保存、保留和保护等操作。**Optim** 功能的基础是一致且久经考验的数据管理方法，此方法与业务目标高度一致，可以扩展到各种应用、数据库、操作系统和硬件平台。

通过简化创建和管理测试环境的方式来加快应用部署。

提取数据的子集，然后进行迁移，以构建真实且规模适中的测试数据库。消除维护多个数据库副本的开销和工作。去标识化、屏蔽和转换机密数据，以保护隐私和让未经授权泄露的可能性降到最低。

Optim 支持所有主要的企业数据库和操作系统，包括 DB2、Oracle、Sybase、SQL Server、Informix、ISM、VSAM、Microsoft Windows®、Unix®、Linux® 和 IBM z/OS®。而且它支持目前广泛使用的重要 ERP 和 CRM 应用——Oracle® E-Business Suite、PeopleSoft® Enterprise、JD Edwards® EnterpriseOne、Siebel® 和 Amdocs® CRM，以及您定制和打包的应用。

#### 关于 IBM Optim

IBM® Optim™ 企业数据管理解决方案关注关键的业务问题，比如数据增长管理、数据隐私遵从性、测试数据管理、电子发现、应用升级、迁移和退役。Optim 使应用数据管理适应业务目标，以优化性能、降低风险并控制成本，同时交付在不同的企业应用、数据库和平台中伸缩的能力。如今，Optim 帮助全球行业范围内的企业利用在其生命周期的每个阶段管理企业应用数据的能力，发挥其企业应用和数据库的业务价值。

#### 更多信息

要了解 IBM Optim 企业数据管理解决方案的更多信息，请联系 IBM 销售代表，或者访问：[http://www-01.ibm.com/software/cn/data/data-management/optim-solutions/。](http://www-01.ibm.com/software/cn/data/data-management/optim-solutions/)



© 版权所有 IBM Corporation 2008

IBM Software Group  
111 Campus Drive Princeton, NJ  
USA, 08540-6400  
[www.optimsolution.com](http://www.optimsolution.com)

在美国印刷

2008 年 3 月

保留所有权利。

<sup>1</sup> Sue Hildreth, "Buggy Software: Up From a Low-Quality Quagmire," *Computerworld*, 2005 年 7 月 25 日。

DB2、IBM、IBM 徽标、IMS、Informix、Optim、Relationship Engine、VSAM 和 z/OS 是国际商业机器公司在美国和其他国家/地区的商标或注册商标。

Linux 是 Linus Torvalds 在美国和其他国家/地区的注册商标。UNIX 是 The Open Group 在美国和其他国家或地区的注册商标。SQL Server 和 Windows 是 Microsoft 公司在美国和其他国家/地区的商标。

所有其他公司或产品的名称都是其各自所有者的商标或注册商标。

本出版物中对 IBM 产品或服务的引用，不代表它们可用于所有 IBM 运营的国家/地区。

**TAKE BACK CONTROL WITH** **Information Management**