

《数据库重构》

图书基本信息

书名：《数据库重构》

13位ISBN编号：9787111202097

10位ISBN编号：7111202090

出版时间：2007-1

出版社：机械工业出版社发行室

作者：安布勒

页数：217

译者：王海鹏

版权说明：本站所提供下载的PDF图书仅提供预览和简介以及在线试读，请支持正版图书。

更多资源请访问：www.tushu111.com

《数据库重构》

内容概要

数据库重构，ISBN：9787111202097，作者：（加）安布勒（Ambler,S.W.），（美）塞得拉吉（Sadalage,P.J.）著，王海鹏译

《数据库重构》

作者简介

作者：(美)安布勒(Ambler S.W.) (加)塞得拉吉 (Sadalage P.J.) 译者：王海鹏 等 Scott W. Ambler 国际知名的软件过程改进顾问，技术领头人，敏捷建模、敏捷数据、企业统一过程、敏捷统一过程方法学的创始人。Scott 经常在 Software Development、JavaOne、OOPSLA 和 DAMA 等会议上进行主题演讲，他写作（或与人合著）出版的书还包括《Agile Modeling》、《Agile Database Technique》、《The Object Primer, Third Edition》、《The Elements of UML UML 2.0 Style》和《The Enterprise Unified Process》等。

Pramod J. Sadalage Thoughtworks 公司的顾问。在 1999 年用 Xp 方法开发一个大型 J2EE 应用时，他就率先实践了演进式数据库设计和数据库重构的过程。他目前正在进行有关演进式项目中的数据库管理以及在数据库设计和管理中使用演进式过程等主题的写作和演讲。

《数据库重构》

书籍目录

第1章演进式数据库开发1.1数据库重构1.2演进式数据库建模1.3数据库回归测试1.4数据库工件的配置管理1.5开发者沙盒1.6演进式数据库开发技术的障碍1.7本章小结第2章数据库重构2.1代码重构2.2数据库重构2.3数据库重构的分类2.4数据库味道2.5数据库重构在开发中的位置2.6使数据库schema的重构更容易2.7本章小结第3章数据库重构过程3.1验证数据库重构是否合适3.2选择最合适的数据库重构3.3让原来的数据库schema过时3.4前测试、中测试和后测试3.5修改数据库schema3.6迁移源数据3.7重构外部访问程序3.8运行回归测试3.9对工作进行版本控制3.10宣布此次重构3.11本章小结第4章部署到生产环境4.1在沙盒之间有效地部署4.2采用数据库重构包4.3制定部署时间窗口进度计划4.4部署系统4.5移除已过时的schema4.6本章小结第5章数据库重构策略5.1小的变更更容易进行5.2唯一地标识每一次重构5.3通过许多小变更实现一次大变更5.4建立数据库配置表5.5触发器优于视图或批量同步5.6选择一个足够长的转换期5.7简化数据库变更控制委员会策略5.8简化与其他团队的协商5.9封装对数据库的访问5.10能够容易地建立数据库环境5.11不要复制SQL5.12将数据库资产置于变更控制之下5.13注意机构中的政治斗争5.14本章小结5.15在线资源第6章结构重构6.1实现结构重构时的常见问题..6.2删除列6.3删除表6.4删除视图6.5引入计算列6.6引入替代键6.7合并列6.8合并表6.9移动列6.10列改名6.11表改名6.12视图改名6.13用表取代LOB6.14取代列6.15用关联表取代一对多关系6.16用自然键取代替代键6.17拆分列6.18拆分表第7章数据质量重构7.1实现数据质量重构时的常见问题7.2增加查找表7.3采用标准代码7.4采用标准类型7.5统一主键策略7.6删除列约束7.7删除缺省值7.8删除不可空约束7.9引入列约束7.10引入通用格式7.11引入缺省值7.12使列不可空7.13移动数据7.14用属性标识取代类型代码第8章参照完整性重构8.1增加外键约束8.2为计算列增加触发器8.3删除外键约束8.4引入层叠删除8.5引入硬删除8.6引入软删除8.7为历史数据引入触发器第9章架构重构9.1增加CRUD方法9.2增加镜像表9.3增加读取方法9.4用视图封装表9.5引入计算方法9.6引入索引9.7引入只读表9.8从数据库中移出方法9.9将方法移至数据库9.10用视图取代方法9.11用方法取代视图9.12使用正式数据源第10章方法重构10.1接口变更重构10.2内部重构第11章转换11.1插入数据11.2引入新列11.3引入新表11.4引入视图11.5更新数据附录UML数据建模表示法词汇表参考文献和推荐读物重构和转换列表

《数据库重构》

编辑推荐

《数据库重构》首次专门讨论数据库重构，向数据专业人员展示了如何运用重构、测试驱动及其他敏捷技术进行演进式数据库开发。书中通过许多实际例子，详细说明了数据库重构的过程、策略以及部署。《数据库重构》前第5章介绍了演进式数据库开发的基本思想和技术，后6章详细描述了每一类重构，包括结构、数据质量、参照完整性、架构、方法的重构；另外还描述了不属于重构范畴的转换技术。书中的示例代码是用Java、Hibernate和Oracle代码编写的，代码都很简单，读者可毫无困难地将它们转换成C#、C++或Visual Basic代码。

《数据库重构》

精彩短评

- 1、略读毕。数据库也可以拿来重构的想法第一次听。重构动机仍可作为学习db设计的好材料。不过和花的代价相比，真有那么大价值吗？
- 2、理论意义大于实践。
- 3、个人觉得还是缺乏可操作性。作者在这方面也是探索书中也少Process, Step和Tools但是有很多建议和创新，可以探讨最好结合作者的另外一本 Agile Data(敏捷数据) 一道服用
- 4、数据库的Migration和refactory的结合讲解，提出了很多实用的模式，因总结的许多模式很清晰，以此为基础，有希望出现数据库重构工具
- 5、同样是重构，人民邮电的《重构》制作让人赏心悦目，再看看这个，很多页都可以看出反页的字印，如此差的用纸，还要45的高价。内容虽然，可是不能用这种制作态度。
- 6、书中对各种修改数据库的情况进行了归纳，试图用敏捷的方式进行重构。
- 7、首次专门讨论数据库重构，向数据专业人员展示了如何运用重构、测试驱动及其他敏捷技术进行演进式数据库开发
- 8、和代码重构一样，数据库设计也需要重构。
- 9、这本书非常实用，里面有很多日常开发数据库改变策略，开发人员应该一读
- 10、主要是如何对数据库进行版本控制，就是简单的数据库变更的操作如何进行版本控制，对有数据库基础的人来说一个下午就可以看完完。在实际项目中可操作性不大

《数据库重构》

精彩书评

- 1、 <http://www.trydofor.com/a9w3-auhome/trydofor/article/2010/0607170631/body.htm>作者啰嗦点，译者外行点。数据和数据库重构，基本上都是组合拳，牵涉到所有角色。书中是分节动作，总结起来就是做事6要素：首先What，Why，How，然后Where，When，最后安排Who。
- 2、 买了5本书挑了这本做赠品。没想到倒是最先看完的，花了3、4天的业余时间。想来，把数据库重构的各方各面一一列举，比起代码重构，书中讲述最多的是如何来保证数据的安全。对于线上系统的重构尤其如此，一个完好的数据安全机制是本书最值得称赞的地方。
- 3、 有一点疑问,译者是学计算机的么
- 4、 无疑数据库重构一直以来就是很少人敢做的事，因为数据库schema、甚至是数据和系统的强耦合性，这本书和《重构》一样，可以在需要进行重构的时候来查找对应的方法。

章节试读

1、《数据库重构》的笔记-第1页

现代软件过程，也称为方法学，在本质上都是演进的，要求你以迭代和增量的方式工作。这些过程的例子包括：Rational统一过程（RUP）、极限编程（XP）、Scrum、动态系统开发方法（DSDM）、水晶方法系列、团队软件过程（TSP）、敏捷统一过程（AUP）、企业统一过程（EUP）、特征驱动开发（FDD）和快速应用开发（RAD），等等。

2、《数据库重构》的笔记-第6页

为了使数据库重构能够进行，你需要将下列工作置于配置管理之下：

1. 创建数据库schema的数据库定义语言（DDL）脚本；
2. 数据加载/抽取/迁移的脚本；
3. 数据模型文件；
4. 对象/关系映射元数据；
5. 参考数据；
6. 存储过程和触发器定义；
7. 视图定义；
8. 参照完整性约束；
9. 其他数据库对象，如序列、索引等；
10. 测试数据；
11. 测试数据产生脚本；
12. 测试脚本；

3、《数据库重构》的笔记-第15页

将数据库重构分为6类：

1. 结构重构：对一个或多个表或视图所作的变更；
2. 数据库质量重构：改进数据库中包含信息质量的一种变更；
3. 参照完整性重构：确保参照的行在另一个表中存在，并确保不再需要的行被相应的删除；
4. 架构重构：从总体上改变外部程序和数据库交互的方式；
5. 方法重构：改进方法（存储过程、存储函数或触发器）质量的一种变更；
6. 转换：不属于重构，变更数据库schema，改变了schema的语义；

数据库味道：

1. 多用途的列：如果一个列被用于多种用途，就有可能存在额外的代码来确保源数据以正确的方式使用，这些代码常常会检查一个列或更多其他列的值。举例：一个DateTime类型的列，如果用户是顾客的话，代表顾客的生日，如果用户是雇员，代表雇员的入职时间；
2. 多用途的表：一个表用于存放多种类型的实体。举例：使用Customer表存放用户和公司的信息；
3. 重复的数据：重复的数据对操作型（operational）数据库来说是一个严重问题，因为如果数据存放在多个地方，不一致的机会就增加了。举例：Customer表中存放了用户的手机号码Mobile，CustomerAddress表中也存放了用户的手机号码Mobile，如果这两个字段保持同步还好，如果不一致的话，在查询用户手机的时候，就不能判断到底哪个才是准确的。
4. 列太多的表：当一个表包含太多的列时，就说明这个表缺乏内聚---它试图存在来自几类实体的数据。譬如：Customer表中存在多种地址（发货地址、账单地址、季节性地址）或多种电话号码（家庭电话、工作电话、移动电话）等等，你就可能需要将这种结构进行标准化处理，加入Address和PhoneNumber表。
5. 行太多的表。大的表就说明有性能问题。如果一个表包含几百万行的数据，查询速度会很慢。你可

《数据库重构》

能需要对表进行垂直分割，将一些列移到另一个表中，或者进行水平分割，将一些行移动到另一个表中。这两种策略都会减小表的规模，可能改善性能。

6. “智能”列：譬如CustomerId是一串数字字符串，前4位代表客户的开户行，后面的数值代表客户ID；另一个列子是使用文本列来保存xml数据结构；

7. 害怕变化

《数据库重构》

版权说明

本站所提供下载的PDF图书仅提供预览和简介，请支持正版图书。

更多资源请访问:www.tushu111.com