Atitit db access req数据库访问规范jdo jdbc jpa pdo sql

目录

[1. 常见特性 1](#_Toc6986)

[1.1. 元数据 API 1](#_Toc17622)

[1.2. 分布式事务 vs事务中使用 Savepoint 1](#_Toc16189)

[1.3. 离线数据api 1](#_Toc18902)

[1.4. Ui数据绑定 2](#_Toc24525)

[1.5. 多数据集的支持 2](#_Toc21265)

[1.6. 分页访问数据的支持 2](#_Toc5758)

[1.7. 数据源的获取方式式多样 2](#_Toc32377)

[1.8. 数据库与数据对象化 2](#_Toc20976)

[2. BDE的性能比初期的ODBC 3](#_Toc3782)

[2.1. Jdbc 3](#_Toc26183)

[3. Jdo jpa 4](#_Toc23206)

[4. Sql 5](#_Toc24083)

[5. Atitit 数据库抽象层jdbc pdo ado.net等比较与异常点 7](#_Toc32419)

[6. Ref 7](#_Toc30495)

# 常见特性

## 元数据 API

## 分布式事务 vs事务中使用 Savepoint

ADO .NET不包括分布式事务的接口，ADO .NET的分布式事务是通过MS DTC统一管理的。JDBC本身就提供了对分布式事务支持的接口，不同的JDBC Driver实现了这一个接口，就支持分布式事务了。

## 离线数据api

ADO .NET最大的特性是对断开连接方式的全方位支持，其引入了DataSet、DataTable、DataRow等等对象，构建了一个“内存数据库”简化版本

DataAdapter把DataReader的数据填充到DataTable或者DataSet中，给使用者一个直观的使用方式

ADO .NET最大的优点是对断开连接访问数据库方式的强有力支持。相比起来，JDBC也引入类似的功能，RowSet，但是比起ADO .NET来，还是不够。

Php pdo好像没有离线数据api

## Ui数据绑定

ADO .NET还能够和ASP .NET和Windows Form的控件进行数据邦定，使得编写一些小程序很简单。

## 多数据集的支持

## 分页访问数据的支持

## 数据源的获取方式式多样

JDBC的数据源的获取方式多样：  
  a、直接通过DriverManager.getConnection获得连接  
  b、通过应用服务器的数据源获得连接。例如：

IMG_256  // Get the initial JNDI naming context  
IMG_257  Context ctx = new InitialContext();  
IMG_258  // Get the DataSource object associated with the logical name  
IMG_259  // "jdbc/AcmeDB" and use it to obtain a database connection  
IMG_260  DataSource ds = (DataSource)ctx.lookup("jdbc/AcmeDB");  
IMG_261  Connection con = ds.getConnection("user", "pwd");

 .NET中，没有类似的咚咚。

## **数据库与数据对象化**

关系数据库对象化、SQL99与JDBC3.0

随着许多关系数据库厂商开始提供对象化服务，各自的接口开始互不兼容，在经历一些麻烦之后，关系数据库厂商感觉到规范化的必要，因为当初关系数据库雄霸天下时SQL92标准起了很大作用，大家可以按照统一的编程方式来访问高性能的商用数据库。

关系数据库厂商集中起来，重新将对象化服务规范起来，形成了SQL99规范，将其中的对象结构等内容规范起来，开始一个崭新的面向对象的关系数据库（ORDBMS）的历程。

(···条消息)JDO之前世今生\_sun2bin的专栏-CSDN博客

# BDE的性能比初期的ODBC

当市场上出现众多的数据库产品之后，Borland和微软都发现自己拥有的数据库产品挺多，市场也不小，不同的产品给用户带来不同的配置任务，不利于所有产品的推广，于是，两者纷纷开始制定数据库访问的规范，微软推出了ODBC，其面向开发人员的亲和性，逐步获得了认可，同时，Borland纠集了IBM和Novell也推出了IDAPI数据库接口规范，也就是今天BDE的核心，不过后来Novell和IBM先后退出，只剩Borland独力支撑。不过Borland是一个技术实力雄厚的公司，其技术一向领先于微软，BDE的性能比初期的ODBC不知道要好多少倍，后来微软偷师学艺，把连接池等技术加到ODBC中，在Delphi3.0及其BDE在市场上风光无限的时候，逐步赶了上来并有超过。直到今天，BDE仍是Borland的产品线上的数据库访问标准，而微软如果不是将ODBC和多数数据库的客户端内嵌进Windows的话，估计BDE仍是市场的赢家。不过，微软是玩弄市场的老手，通过对操作系统的垄断，其数据库产品和ODBC标准终究占据了多数开发市场。

————————————————

## Jdbc

ava开始涉及数据库应用后，Sun就极力制定Java的数据库规范，JDBC API就是类似ODBC一样，对数据库访问的底层协议进行最基本的包装，然后形成一套统一的数据访问接口，数据库连接、SQL语句句柄、结果集，都带有ODBC的影子。以方便配置为目的，Sun极力推荐完全瘦客户端的TYPE 4型JDBC驱动，这是一个不需要安装数据库客户端的驱动规范，是现在使用最多的。当然，为了保持与旧的数据库兼容，JDBC规范中包括了专用于连接ODBC的TYPE 1驱动和需要安装数据库客户端的TYPE 2驱动，以及可以由厂商在数据库服务端专门提供面向JDBC的服务的TYPE 3驱动。

JDBC最早出现时，还不属于标准JDK的一部分，而是作为一个额外包提供下载。后来，随着Java编写的数据库应用的的增多，和JDBC规范本身的逐渐成熟，JDBC终于成为JDK1.1的一部分。

JDBC目前最新的是3.0版本，还有正在讨论中的4.0版本。实际上，在开发中使用得最多的还是1.0中的API，2.0中主要增加了可双向滚动的结果集、更新批处理等提高可用性和性能的API，3.0主要增加了连接池、可更新的结果集等特性。4.0将在可管理性、连接池规范化等方面再做改进。

————————————————

Atitit jdbc3.0 新特性 【JDBC发展史】从JDBC1.0到JDBC4.0.html

目录

1. jdbc2.0 1

1.1. 1）数据源（DataSource）。 1

1.2. 2）连接池（ConnectionPoolDataSource）。 1

2. 四、JDBC 3.0 1

2.1. 元数据 API 2

2.2. CallableStatements 中已命名的参数 2

2.3. 检索自动产生的关键字 2

3. 五.JDBC 4.0 3

3.1. 6．对SQL XML的支持 3

3.2. 自动加载驱动 3

3.3. SQLExcpetion 的增强 3

# Jdo jpa

JDO的全文是：Java Data Object，就是对象持久化的新的规范。也是一个用于存取某种数据仓库中的对象的标准化API。JDO提供了透明的对象存储，因此对开发人员来说，存储数据对象完全不需要额外的代码（如JDBC API的使用）。这些繁琐的例行工作已经转移到JDO产品提供商身上，使开发人员解脱出来，从而集中时间和精力在业务逻辑上。另外，JDO很灵活，因为它可以在任何数据底层上运行。JDBC只是面向关系数据库（RDBMS）JDO更通用，提供到任何数据底层的存储功能，比如关系数据库、文件、XML以及对象数据库（ODBMS）等等，使得应用可移植性更强

————————————————

2.2 数据库与数据对象化

数据库是企业级应用不可缺少的，因此，在面向对象流行的时候，数据库厂商也在进行着数据对象化的研究。这些研究在上个世纪八十年代就初现端倪。

数据库的对象化一般有两个方向：一个是在主流的关系数据库的基础上加入对象化特征，使之提供面向对象的服务，但访问语言还是基于SQL；另一个方向就是彻底抛弃关系数据库，用全新的面向对象的概念来设计数据库，这就是对象数据库ODBMS。

2.2.1 关系数据库对象化、SQL99与JDBC3.0

随着许多关系数据库厂商开始提供对象化服务，各自的接口开始互不兼容，在经历一些麻烦之后，关系数据库厂商感觉到规范化的必要，因为当初关系数据库雄霸天下时SQL92标准起了很大作用，大家可以按照统一的编程方式来访问高性能的商用数据库。

关系数据库厂商集中起来，重新将对象化服务规范起来，形成了SQL99规范，将其中的对象结构等内容规范起来，开始一个崭新的面向对象的关系数据库（ORDBMS）的历程。

JDBC3.0就是在这种情况下出台的，它将对关系数据库中的对象服务的访问API规范起来，为Java平台提供了访问ORDBMS的标准方式。当然，JDBC3.0对传统的SQL操作也进行了很多功能增强。

Oracle是一个传统的关系数据库厂商，在对象化的道路上，Oracle当然采取追加对象化特征的道路，以侵入数据对象化的市场，保持Oracle在数据库领域的领导地位。如果说Oracle7.4使Oracle走向全盛的话，从Oracle8开始，Oracle就成为关系数据库加对象类型的先驱。在Oracle8中，我们可以定义一些数据结构（Record），将普通的类型包装在其中成为数据元素，然后可以在客户端按Record结构进行访问，初步提供了面向对象的数据库服务。

2.2.2 对象数据库

对象数据库就是采用全新的面向对象概念来设计数据库的全新数据库类型。在这方面，主要以一些大学研究机构进行设计和开发，有些也形成了产品，不过由于市场方面的原因（主要是关系数据库的容易上手和市场绝对领导地位）和ODBMS先天的一些弱点（比如查询引擎很难优化），使ODBMS没有象关系数据库那样流行起来。

不过对象数据库的对象化特点还是令人割舍不下，目前还是有一些很好的产品在市场上，从商用的到免费的都用。目前在ODBMS领域占据领导地位的是Versant、FastObjects和ObjectStore等几大厂商，并且，市场份额也在逐步扩展。免费的产品包括C++编写的Ozone、纯Java的db4o等等。还有一些研究机构开发一些底层的面向对象数据库引擎，但只提供一些底层的API，不提供管理方面的功能，以及一些算法提供开放式接口，让厂商去选择和实现。比如美国威斯康新大学计算机系数据库组的SHORE引擎，就是一个非常出色的面向对象数据库引擎，现在还在积极的更新中，一些其它研究机构和数据库厂商采用它完成了自己的特别的对象数据库，比如专用于地理信息的数据库、专用于宇宙空间数据研究的数据库等等。

目前对象数据库最大的障碍是缺乏统一的规范，各个数据库厂商有各自的访问接口。对象数据库比起关系数据库来，不只是基本的几种数据类型那么简单，它还涉及继承处理、多态等一大堆面向对象特征的实现，规范化道路当然困难重重。这也是对象数据库无法普及的一个重要原因。

也有一些机构提出了一些建议的规范，比如制定Corba标准的OMG小组的一个分组ODMG提出的ODMG规范，目前已经是3.0版本，其中的OQL对象查询语言相当具有吸引力。还有一些中立的机构提出了其它的一些标准化的对象访问API，也可算是面向对象数据库的规范之一。象前面提到的FastObjects和Ozone就是符合ODMG3.0规范的。

————————————————

# Sql

8. Sql标准化 8

8.1. 1. Sql标准化的历史3 8

8.2. 1.1. Sql92标准4 8

8.3. 1.2. Sql99标准4 8

8.4. 1.3. SQL:2003为例，它包括以下9个部分 5 8

8.5. 1.4. Sql2006标准6 8

8.6. 1.5. Sql2008标准7 8

8.7. 1.6. SQL:2011 7 8

Atitit sql的标准化与新特性

目录

1. SQL标准分为十部分。 2

1.1. 第1部分：框架（SQL /框架）。它提供了逻辑概念。[41] 2

1.2. 第2部分：基础（SQL / Foundation）。它包含语言的最主要元素，并且包含强制性和可选性功能。 2

1.3. 第3部分：呼叫层介面（SQL / CLI）。它定义了接口组件（结构，过程，变量绑定），可用于执行以应用程序中的SQL语句 3

1.4. 第4部分：持久存储的模块（SQL / PSM） sp udf 3

1.5. 第9部分：外部数据管理（SQL / MED）。 3

1.6. 第10部分：对象语言绑定（SQL / OLB）。它定义了SQLJ的语法和语义，SQLJ是Java中嵌入的SQL（另请参见第3部分） 3

1.7. 第11部分：信息和定义架构（SQL / Schemata）。 4

1.8. 第13部分：使用Java TM编程语言（SQL / JRT） 4

1.9. 第14部分：与XML相关的规范（SQL / XML）。 4

1.10. 第15部分：多维数组（SQL / MDA）。 5

2. 废弃的步伐 5

3. Sql标准化的历史 5

3.1. 1986年 SQL-86 SQL-87 首先由ANSI规范化。 6

3.2. 1989年 SQL-89 ，增加了完整性约束 6

3.3. Sql92标准 7

3.3.1. 定义了新的数据类型：DATE，TIME，TIMESTAMP， 8

3.3.2. 新的标量运算，例如字符串连接和子字符串提取，日期和时间数学以及条件语句。 8

3.3.3. 1996年的SQL / PSM（存储过程） 9

3.4. Sql99标准 9

3.4.1. 使用Java编程语言存储过程 10

3.4.2. 触发器，对过程和流程控制语句的支持 10

3.4.3. ，非标量类型（数组）和一些面向对象的功能（例如，结构化类型）。 10

3.4.4. 支持在Java（SQL / OLB）中嵌入SQL，反之亦然（SQL / JRT）。 10

3.5. SQL:2003为例，它包括以下9个部分 10

3.5.1. 与XML相关的功能（SQL / XML） 11

3.5.2. 窗口功能 11

3.5.3. 两种新的列类型：自动生成的值和标识列 11

3.5.4. 新的MERGE语句 11

3.5.5. 对CREATE TABLE语句的扩展，以允许“ CREATE TABLE AS”和“ CREATE TABLE LIKE” 11

3.6. Sql2006标准 11

3.6.1. 可以使用XQuery 11

3.7. Sql2008标准 12

3.7.1. TRUNCATETABLE语句， 12

3.7.2. FETCH子句 12

3.7.3. INSTEAD 数据库触发器 12

3.7.4. 分区连接表， 12

3.7.5. 各种XQuery的正则表达式/模式匹配功能的支持，并 13

3.8. SQL:2011 13

3.9. 2016年 SQL：2016年 添加行模式匹配，多态表函数JSON。 14

3.10. 2019年 SQL：2019年 添加了第15部分，多维数组（MDarray类型和运算符）。 14

# Atitit 数据库抽象层jdbc pdo ado.net等比较与异常点

目录

1. 应该具有的功能 1

1.1. 元数据 API 1

1.2. 分布式事务 vs事务中使用 Savepoint 1

1.3. 离线数据api 1

1.4. Ui数据绑定 2

1.5. 多数据集的支持 2

1.6. 分页访问数据的支持 2

1.7. 数据源的获取方式式多样 2

2. 离线数据集合的填充方式 2

2.1. （1）从数据库直接获取数据 3

2.2. （2）用ResultSet填充 3

2.3. （3）用XML填充 3

2.4. （4）用其他方法填充 csv、excel、text、http等格式 4

3. 4、操作离线数据集的操作RowSet中的数据及元数据 4

3.1. 1）更新数据 4

3.2. （2）插入数据 4

3.3. （3）删除数据 4

3.4. （4）查询 4

3.5. （5）元数据 5

4. 类与方法对应表 5

5. Rf 5

# Ref

Atitit sql art 艺术 attilax著 v4 u55.docx

Atitit jdbc3.0 新特性 【JDBC发展史】从JDBC1.0到JDBC4.0.html

(···条消息)JDO与JPA哪个更好？\_java\_wu560130911的专栏-CSDN博客