

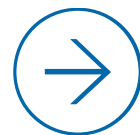
JDBC编程

SOFTCITS@2016



SoftCITS
SOFT CULTURE IT SALON

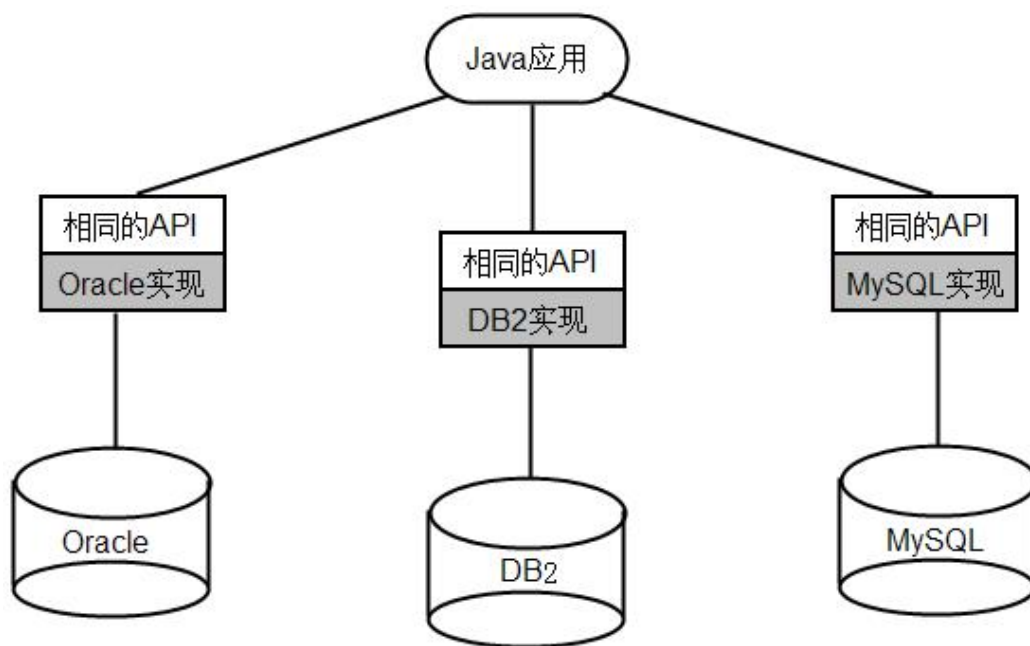
软件园IT文化沙龙



JDBC基础

JDBC的全程是Java Database Connectivity,及Java数据库连接。

特点：跨平台，跨数据库。



MySQL

1. 安装：下载 dev.mysql.com/downloads/mysql

2. 基本命令

`show databases;`

`create database [DB NAME];`

`drop database [DB NAME];`

`use [DB NAME];`

`show tables;`

`desc [TABLE NAME];`

`mysql -p [PWD] -u [ACCOUNT] -h [HOST] --default-character-set=utf8`

数据库对象

对象名称	对应关键字	描述
表	table	表是存储数据的逻辑单元，以行和列的形式存在；列就是字段，行就是记录。
数据字典		就是系统表，存放数据库相关信息的表。系统表里的数据通常由数据库系统维护。程序员通常不应该手动修改系统表及内部数据，程序员只可查看系统表的数据。
约束	constraint	执行数据校验的规则，用于保证了数据完整性的规则。
视图	view	一个或者多个数据表里数据的逻辑显示。视图并不存储数据。
索引	index	用于提高查询的性能，相当于书的目录。
函数	function	用于完成一次特定的计算，具有一个返回值。
存储过程	procedure	用于完成一次完整的业务处理，没有返回值，但可通过传出参数将多个值传给调用环境。
触发器	trigger	相当于一个事件监听器，当数据库发生特定事件后，触发器被触发，完成响应的处理。

对象操作-DDL

1. 表的创建，修改（增加字段，删除字段，修改字段类型，重命名字段），删除

2. 约束

NOT NULL：非空约束，指定某列不能为空。

UNIQUE：唯一约束，指定某列或者几列组合不能重复，但可以出现多个null值。

PRIMARY KEY：主键，指定该列的值可以唯一地表示该条记录。Not null+Unique

FOREIGN KEY：外键，指定该行记录从属于主表中的一条记录，主要用于保证参照完整性

数据操作-DML

数据添加、更新，删除，查找 ---- insert/update/delete/select

组查询-

count(*)组函数，统计记录行数

group by 只返回组相同的一条记录，所以很多数据库产品要求select中只能出现group by的指定字段。一般联合count用于统计。

group by 限定条件使用having,与where的区别：

- 1.where过滤行，having过滤组
- 2.使用组函数只能用having

数据操作-DML

多表连接查询

from table1,table2 where

inner join/left join/right join

union

子查询

可以把子查询结果当成一个 ‘视图’ 对象。

JDBC基础

- (1) 加载数据库驱动。
- (2) 通过DriverManager获取数据库连接。
- (3) 通过Connection对象创建Statement对象。
- (4) 使用Statement执行SQL语句。 executeQuery(查询)/executeUpdate(case1: DDL – 返回0, case2:DML-返回受影响记录数)
- (5) 操作结果集ResultSet。

示例代码-1.1

JDBC基础

PreparedStatement:预编译的Statement对象,是Statement的子接口,预编译SQL语句,以后每次只传递SQL的参数,避免数据库每次都需要编译SQL语句,因此性能更好。

示例代码 – 1.2

操作结果集

以默认方式打开的ResultSet是不可更新的，如果希望创建可更新的ResultSet，必须在创建Statement或PreparedStatement时传入额外的参数。

resultSetType：控制ResultSet的类型，该参数可以是如下三个值：

ResultSet.TYPE_FORWARD_ONLY：该常量控制记录指针只能向前移动。这是JDK1.4以前的默认值。

ResultSet.TYPE_SCROLL_INSENSITIVE：该常量控制记录指针可以自由移动（可滚动结果集），但底层数据的改变不会影响ResultSet的内容。

ResultSet.TYPE_SCROLL_SENSITIVE：该常量控制记录指针可以自由移动（可滚动结果集），而且底层数据的改变会影响ResultSet的内容。

resultSetConcurrency：控制ResultSet的并发类型，该参数可以接受如下两个值：

ResultSet.CONCUR_READ_ONLY：该常量指示ResultSet是只读的并发模式（默认）。

ResultSet.CONCUR_UPDATABLE：该常量指示ResultSet是可更新的并发模式。

示例代码-1.3

离线结果集与查询分页

传统ResultSet需要保持数据库强连接，一旦connection关闭时去读取ResultSet就会导致异常。

使用CachedRowSet可以将底层数据读入内存中，这样即使数据连接关闭也仍然可以读取数据。

示例代码-1.4.1

由于CachedRowSet会将数据装载内存中，如果数据过大，会导致内存溢出。为解决该问题，分页功能应运而生。

CachedRowSet提供分页方法：

setPageSize(int pageSize):设定每页包含记录数

Populate(ResultSet rs, int startRow)：返回从startRow开始的记录

示例代码-1.4.2

事务

事物由若干数据库操作组成，要么全部执行，要么全部放弃。

四个特性：

原子性(A)：事物是最小执行单位。

一致性(C)：数据库从一个一致性保持到另一个一致性。

隔离性(I)：事物之间互不干扰。

持久性(D)：可以持久化存储。

默认事物是关闭的，及操作自动提交

SET AUTOCOMMIT={0|1} 0关闭自动提交，1开启自动提交

Session间对事物的设置互不干扰

命令：

savepoint a;

rollback to a;

commit;

JDBC中的事物

Connection的setAutoCommit方法来关闭自动提交，开启事务，如下SQL语句所示：

```
conn.setAutoCommit(false);
```

程序可以调用Connection的commit方法来提交事务，如下代码所示：

```
conn.commit();
```

如果任意一条SQL语句执行失败，Connection的rollback来回滚事务，如下代码所示：

```
conn.rollback();
```

示例代码-1.5

连接池

数据库连接的建立和关闭非常耗费资源和时间，一个数据库连接均对应一个物理连接，每次都需要打开和关闭连接。

数据库连接池：当应用启动时系统建立足够连接并放到连接池中等待程序请求，程序请求完毕后不再关闭数据库连接而是归还给连接池。

目前流行产品:WebLogic,WebShpere和开源C3P0

示例代码-1.6

谢谢观看
See You !

SoftCITS
SOFT CULTURE IT SALON

软件园IT文化沙龙

