链滴

032 Java 数据库程序设计

作者：pzs233

原文链接：https://ld246.com/article/1537289505983

来源网站：链滴

本文为《Java 语言程序设计》第十版 章节笔记

32.1 引言

32.2 关系型数据库系统

SQL 是定义和访问数据库的标准数据库语言。

三大要素：结构、完整性、语言。结构定义了数据的表示，完整性给出了一些对数据的约束，语言提了访问和操纵数据的手段。

- 关系结构：一个关系实际上是一个没有重复行的表格
- 完整性约束：对表格强加一个条件，表中的所有合法值都必须满足该条件
    - 域约束（domain constraint）规定一个属性的允许值
    - 主键（primary key）约束：主键是最小的超键之一（超键是一个属性或一组属性，它唯一地标了一个关系）
    - 外键约束（foreign key constraint），两个表的共同属性就是外键

32.3 SQL

结构化查询语言（structured query language，SQL）是用来定义表格和完整性约束以及访问和操数据的语言。

SQL 是访问关系数据库系统的通用语言。

在 MySQL 上创建用户账户

```
mysql -uroot -p
//输入密码进入MySQL
use mysql;
create user 'userName'@'localhost' indentified by 'password';
grant select, insert, update, delete, create, create view, drop, execute, references on *.* to 'use
Name'@'localhost';
//grant all privileges on *.* to 'userName'@'%' identified by 'password';
//grant all privileges on *.* 'userName'@'ipAddress' identified by 'password';
exit;
```

在 window系统中，MySQL是自启的，net stop mysql 终止，net start mysql 启动。

创建数据库

```
mysql -uuserName -ppassword
create database databaseName;
use databaseName;
source script.sql;
```

创建和删除表

create/drop table TableName (column1 options, column2 options,...);

简单插入、更新和删除表中数据

插入：

insert into bableName [(column1, column2, ..., column)] values (value1, value2, ..., valuen);

更新：

update tableName set column1 = newValue1 [, column2 = newVlaue2, ...] [where contition];

删除：

delete from tableName [where contition];

简单查询

select column-list from table-list [where condition];

比较运算符和布尔运算符

=、<> or !=、<、<=、>、>=、not、and、or

操作符 like、between-and、is null

SQL 有一个可以用于模式匹配的操作符 like。

检查字符串 s 是否含有 p 的语法是：s like p或 s not like p。

在模式 p 中可以使用通配符 %和 _。%匹配零个或多个字符，_与 s 中的任何单个字符匹配。

例如，lastName like '_mi%'表示与第二个和第三个字符分别为 m 和 i 在任意任意字符串匹配；lastName not like '_mi%'。

运算符 between-and检查值 v 是否在值 v1 和 v2 之间，eg：v between v1 and v2；v not between v1 and v2。

运算符 is null 检查值 v 是否为 null（空），eg:v is null或v is not null。

列的别名

select columnName [as] alias

算术运算符

在 SQL 中可以使用算术运算符：+、-、*、/。

显示不同记录

SQL 中使用 distinct 关键字来去除输出重复的元组。

eg：select distinct subjectId as "Subject ID" from Course;

显示排好序的记录

SQL 提供对输出结果排序的 order by 子句，语法如下：

select column-list from table-list [where condition] [order by columns-to-be-sorted][desc];

默认按升序排列（asc），降序加关键字 desc。

联结表

经常需要从多个表中获得信息，如下查询所示：

查询一个学生的所有课程。

学生表（Student）：

ssn	lastName	mi	firstNmae
...	...	...	...

注册表（Enrollment）：

ssn	courseId	...
...	...	...

可以使用如下 SQL 语句编写：

```
select distinct lastName, firstName, courseId from
Student, Enrollment where Student.ssn = Enrollment.ssn and lastName = 'xxx' and firstName
 'xxx';
```

32.4 JDBC

开发数据库应用程序的 Java API 称为 JDBC。

使用 Java 开发任何数据库应用程序都需要 4 个主要接口：Driver、Connection、Statement、Resul Set。

```
//加载驱动程序
Class.forName("JDBCDriverClass");

//建立连接
Connection conection = DriverManager.getConnection(databaseURL, "userName", "password
);
//或建立和数据源的连接
//Connection connection = DriverManager.getConnection("jdbc:odbc:ExampleMDBDataSour
e");
```

```
//创建语句
Statement statementName = connection.createStatement();

//执行语句
ResultSet resultSet = statementName.executeUpdate(String sql)/executeQuery(String sql);

// 处理 ResultSet
while (resultSet.next()) {
    System.out.println(resultSet.getString(1) + " " +
        resultSet.getString(2) + " " + resultSet.getSring(3));
}

// 关闭连接
connection.close();
```

驱动程序是一个实现接口 java.sql.Driver 的具体类。访问不同的数据库必须加载它们各自的驱动程序。

数据库	驱动程序类	来源
Access	sun.jdbc.odbc.JdbcOdbcDriver	经在 JDK 中
MySQL	com.mysql.jdbc.Driver	ysql-connector-java-5.1.26.jar
Oracle	oracle.jdbc.driver.OracleDriver	jdbc6.jar

必须将驱动程序放至目标程序的类路径中。

databaseURL 的模式是：

- MySQL： jdbc:mysql://hostname/dbname
- Oracle： jadc:roacle:thin:@hostname:port#:oracleDBSID

结果集 ResultSet 维护一个表，该表的当前行可以获得。当前行的初始位置是 null。可以使用 next 法移动到下一行，可以使用 各种 getter 方法从当前行获取值。

32.5 PreparedStatement

PreparedStatement 可以创建参数化的 SQL 语句。

Statement 接口用于执行不含参数的静态 SQL 语句。PreparedStatement 接口继承自 Statement 口，用于执行含有或不含参数的预编译的 SQL 语句，由于 SQL 语句是预编译的，所以重复执行它们时效率更高。

PreparedStatement 对象是用 Connection 接口中的 preparedStatement 方法创建的。eg：

```
PreparedStatement preparedStatement = connection.prepraeStatement(
    "insert into Student (firstName, mi, lastName) values (?, ?, ?)");
```

三个问号表示三个占位符，代表一条记录的三个属性的值，使用如下方法设置这三个的值：

```
setX(int parameterIndex, X value) //X 是参数的类型，parameterIndex 是语句中参数的下标，从 1
```

开始。

设置参数后，使用如下方法查询或更新：

```
ResultSet rset = preparedStatement.executeQuery();
ResultSet rset = preparedStatement.eeecuteUpdate();
```

32.6 CallableStatement

CallableStatement 可以执行 SQL 储存过程。

32.7 获取元数据

可以使用 DatabaseMetaData 接口来获取数据库的元数据，例如数据库 URL、用户名、JDBC驱动序名称等。

使用 ResultSetMetaData 接口获取 结果集合的元数据，例如表的列数和列名等。