**分页技术**

# 授课内容

授课科目：JavaEE

授课内容：分页技术

授课类型：讲授+实践

授课时间：3+1学时

主讲教师：孟双英

# 教学目标要求

## 能力目标：

1. 培养学生融会贯通知识的能力；
2. 培养学生掌握分页的应用场景；
3. 培养学生掌握分页技术的实现方式及其原理。

## 知识目标：

1. 复习JDBC访问数据库；
2. 掌握分页技术的两种实现方式及其代码实现；
3. 了解各数据库产品中不同的分页SQL。

# 课件分析

概 述：大部分的Java企业应用中会利用数据库来存储数据和信息，本章基于一个图书商城的案例回顾JavaSE课程中学过的JDBC数据库访问技术，并由此引出当页面要显示的数据量庞大时分页技术的应用。常用的分页实现方案有两种，一种是一次从数据库中取出所有的数据，在内存中实现分页，另一种是多次访问数据库每次取出指定页码的数据，两种实现方案各有优略，重点介绍第二种，并介绍不同的数据库产品实现分页的SQL语句，要求同学们掌握MySQL数据库中的分页SQL语句。

教学重点：多次访问数据库实现分页技术。

教学难点：分页技术的实现流程和代码结构。

# 教学方法

回顾法、问题教学法、案例法、讲授法、演示法等。

# 教学过程

## 课前补充

（0分钟）

## 上节回顾

（0分钟）

## 作业点评

（0分钟）

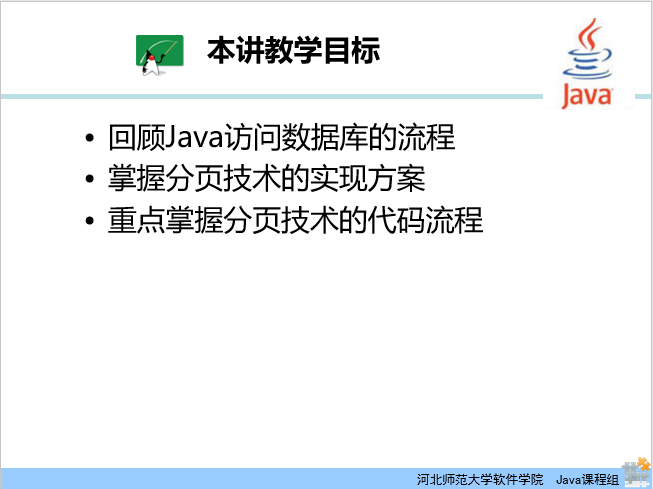
## 导入新课

（2分钟）

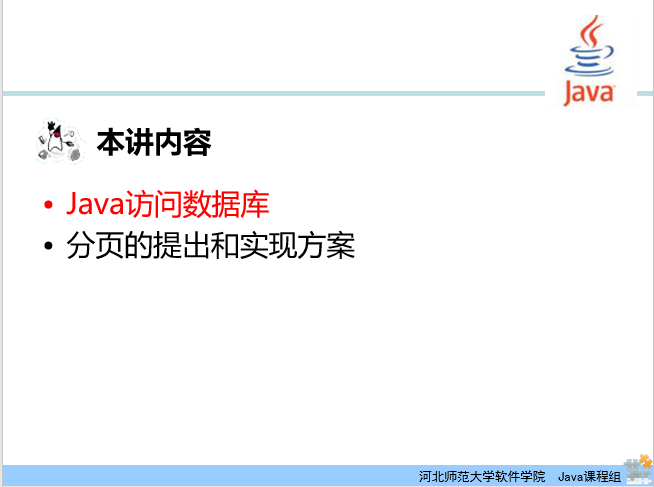
Java企业应用中大多数都会用到数据库来存储数据和信息，因此Java开发中对数据库操作必须熟练。本章内容的基础也是数据库操作，Java通过JDBC访问数据库，JDBC技术是必备技能，首先以一个案例的形式复习JDBC访问数据库技术，再进行分页技术的讲解和实现。

## 讲授新课

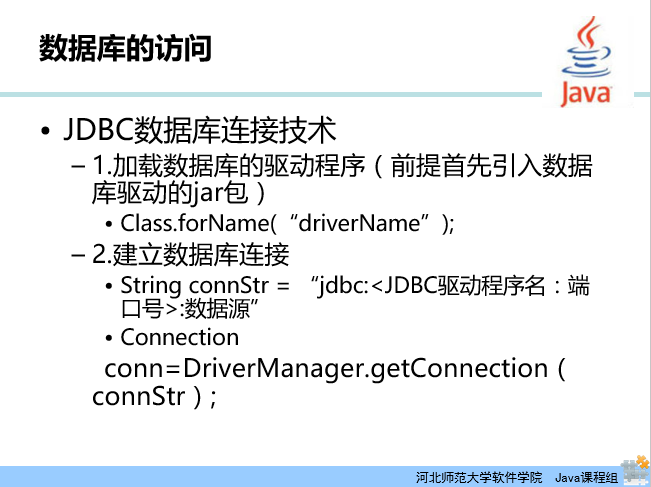
（130分钟）



本章教学目标：复习Java访问数据库的技术实现——JDBC，重点复习JDBC操作数据库的编码流程，介绍分页技术的两种实现方法，要求学生掌握分页的实现方案和实现流程。



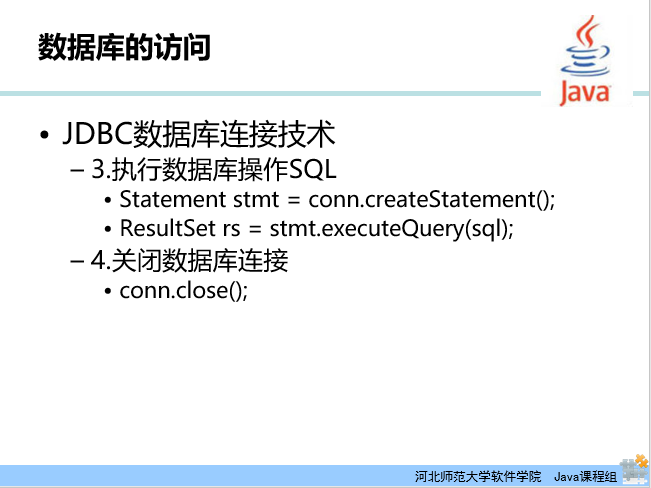
第一部分内容复习Java访问数据库技术——JDBC。



Java程序中访问数据库的技术称为JDBC，Java DataBase Connection，JDBC访问数据库的基本流程是：

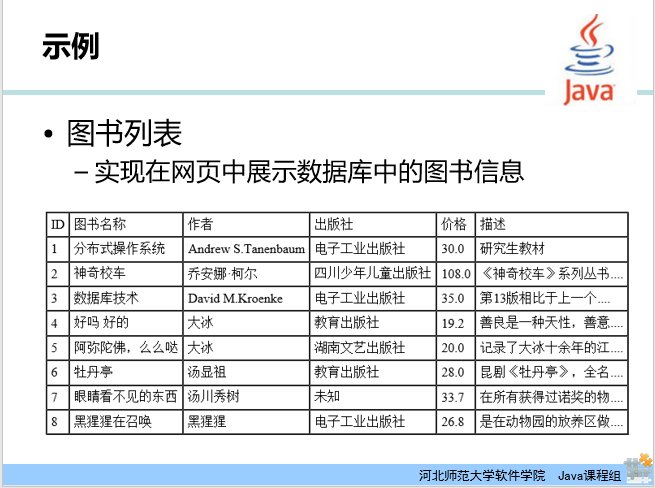
1.首先加载数据库的驱动程序，即引入数据库产品的jar包，在程序中通过反射Class.forName()将数据库的驱动类加载到内存中。

2.建立程序与数据库之间的连接（桥梁），有了连接才可与数据库之间进行通信，通过DriverManager.getConnection()方法获取数据库连接。



3.向数据库管理系统申请执行SQL语句，数据库管理系统执行SQL并返回结果，connection.createStatement();statement.executeQuery(sql)。

4.关闭数据库连接，connection.close()。



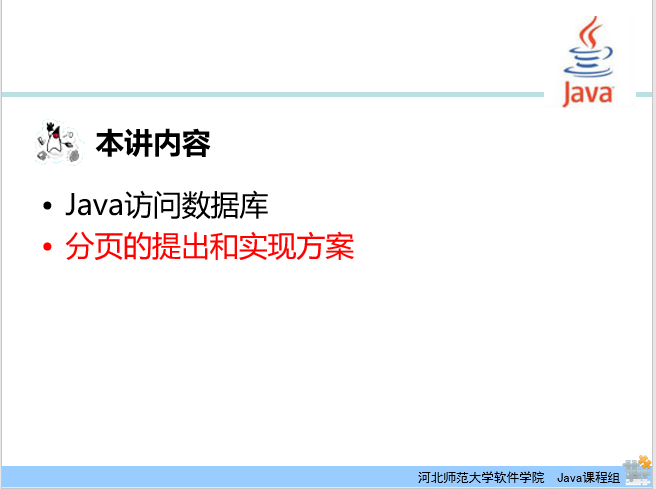
下面以一个图书商城案例中的展示图书列表的功能为例，复习在程序中实现数据库的访问，核心代码如下：

|  |
| --- |
| Dao层代码：  public ArrayList<Book> selectProduct() {  Connection conn = DB.getConn();  Statement stmt = null;  ResultSet rs = null;  ArrayList<Book> productList = new ArrayList<Book>();  try {  String sql = "select \* from book";  stmt = conn.createStatement();  rs = stmt.executeQuery(sql);  while (rs.next()) {  Book book = new Book();  book.setBook\_id(rs.getInt("book\_id"));  book.setBook\_name(rs.getString("book\_name"));  book.setBook\_price(rs.getDouble("book\_price"));  book.setBook\_auth(rs.getString("book\_auth"));  book.setBook\_publisher(rs.getString("book\_publisher"));    String desc = rs.getString("book\_description");  if (desc == null)  desc = "";  if (desc.length() > 10) {  book.setBook\_description(desc.substring(0, 10) + "....");  } else {  book.setBook\_description(desc);  }  book.setBook\_imgurl(rs.getString("book\_imgurl"));  productList.add(book);  }  } catch (SQLException e) {  e.printStackTrace();  } finally {  DB.close(rs);  DB.close(stmt);  DB.close(conn);  }  return productList;  }  获取数据库连接代码：  public static Connection getConn() {  Connection conn = null;  try {  Class.forName("com.mysql.jdbc.Driver");  conn = DriverManager.getConnection("jdbc:mysql://localhost:3306/bookshop?user=root&password=&characterEncoding=utf8");  } catch (ClassNotFoundException e) {  e.printStackTrace();  } catch (SQLException e) {  e.printStackTrace();  }    return conn;  } |

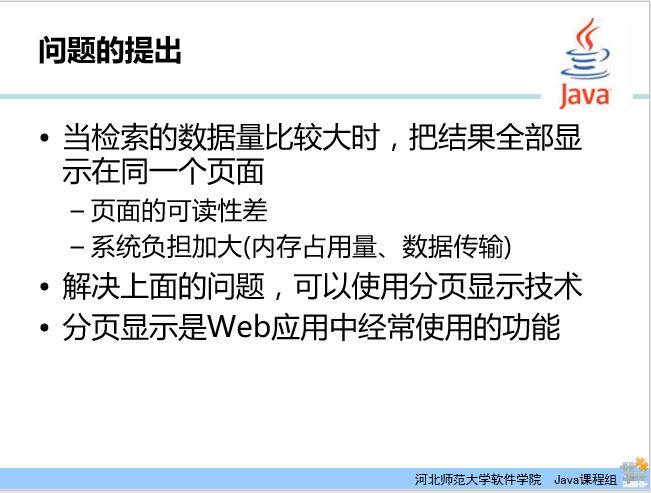
本例中不仅回顾了JDBC的使用流程，还介绍了Java企业应用开发中的代码分层（分包）惯例，一般的Java企业应用开发中，代码分为db、dao、service、servlet、domain 5层，其中db存放连接数据库的类和接口，dao存放数据访问的类和接口，service存放业务逻辑相关的类和接口，servlet存放接收请求、返回响应的Servlet相关的类，domain存放实体类。

课程扩展：

引入数据库连接池的概念和知识的学习，自定义数据库连接池实现对数据库连接的管理和维护。

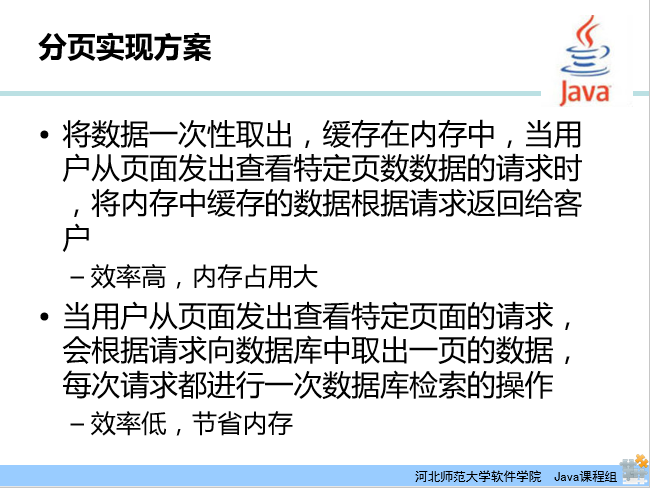


示例中在页面中显示的图书列表可以看出，图书数量非常多时，页面会很长，显示不够美观、也不利于用户查看需要的图书信息，因此需要对大量的图书的信息采用分页的方式来展示。



分页技术的提出是因为当大量数据需要显示在同一个页面时，页面的可读性较差，并且同时从数据库中取出大量的数据，不管是从内存的占用量上还是数据的传输上都会使得系统的负担过重。

分页显示数据是目前应用系统中经常使用的功能，需要重点理解和掌握。



分页的实现方案大体有两种，一种是将数据一次性取出（像不分页一样），将所有数据缓存在内存中，当用户从页面发出查看特定页数据的请求时，服务器端将内存中的数据分页并将指定的数据返回给客户端，这种实现分页的方式并没有解决之前提到的数据量大浪费内存的问题，因此有人称之为“假分页”，这种分页方式最大的缺点是，当数据库中的数据被更改时，内存中的数据无法实时响应，造成内存中数据陈旧。

另一种分页实现是数据库分页技术，当用户从页面发出一个查看特定页数据请求时，服务器端根据页码每次从数据库中取出指定的数据，并将数据返回给用户，这种分页方式能有效解决之前提到浪费内存的问题，所以有人把这种分页称为“真分页”，当然，每种实现方式都不是无缺点的，这种分页方式会频繁访问数据库，会降低应用系统的执行效率。

两种分页方式各有利弊，但从应用上来看，第二种数据库分页技术实现更为常用，也是目前大多数应用系统中采用的方式。

下面以图书信息展示为例，采用两种不同的分页技术来实现分页展示图书信息。

实现关键演示点：

1. Page类的封装，在分页实现中，需要设计表示页码的类，此类中包括：当前页码、总页数、总条数、每页条数，这些信息是在前端页面中显示页码时需要用到的，当服务器端向客户端返回信息时携带此Page对象即可；

2. 分页查询找的逻辑（对比两种分页技术实现的不同点）

a) 内存分页：当浏览器发送请求某页数据到服务器端时，服务器先判断内存中是否有图书信息列表，如果没有向数据库发送查找请求，查找出所有的图书信息列表；如果有直接从内存中取出图书信息列表，服务器端根据接收到的页码判断是否是合法的，封装Page对象信息，将Page对象以及对应页的图书信息响应给客户端，客户端接浏览器收到响应信息，显示图书列表和页码链接等页面信息。

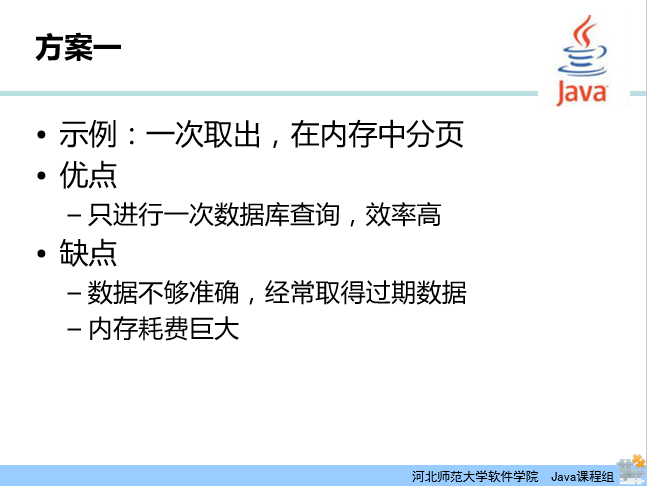
b)数据库分页：当浏览器发送请求某页数据到服务器端时，服务器向数据库发送查找图书总记录数的请求，数据库返回总记录数后，服务器根据总记录数和接收到的页码信息判断是否请求的页码是否是合理的页码，如果是，封装Page对象信息，并向数据库发送查找指定页码图书信息的请求，将Page对象以及对应页的图书信息响应给客户端，客户端接浏览器收到响应信息，显示图书列表和页码链接等页面信息。

3. 页面中显示的，首页、上一页、下一页、尾页等超链接的实现，页面显示分页相关的信息依据于服务器端响应回的Page对象信息，根据Page对象的信息显示，效果图如下图所示：

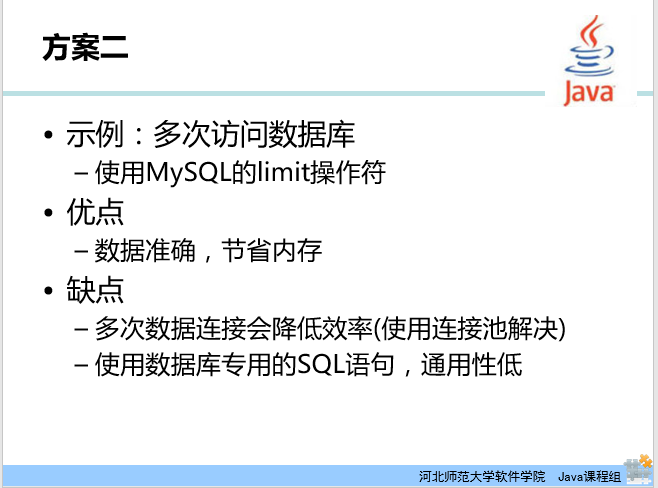


完整代码参考fenyetest项目。

下面来总结两种分页技术的实现。

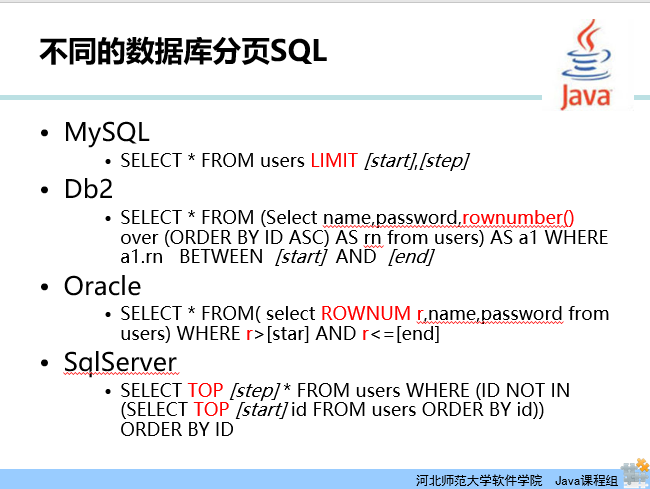


方案一，一次取出所有数据，在内存中分页，其优点是只进行一次数据库查询，效率高；缺点是数据可能会出现过期现象，并且如果数据量大会耗费巨大的内存量。



方案二，数据库分页技术，多次访问数据库，每次取出一页数据，优点是数据准确、节省内存；缺点是多次数据库连接会降低应用程序的执行效率，并且具体的分页实现依赖于底层的数据库产品，数据库产品必须支持分页才可，通用性较差。

不同的数数据库产品对分页技术的支持不同，下面介绍常用的数据库产品对分页技术的支持。



MySQL数据库：limit语句，limit start,step; 举例：查询第2页图书信息select \* from books limit 10,10;

SQLServer数据库：top语句，取前\*条，需要嵌套查询实现，举例：查询第2页图书信息select top 10 \* from books where id not in(select top (2-1)\*10 \* from books order by id) order by id;

Oracle数据库：rownum 行号判断，举例：查询第2页图书的信息select \* from (select rownum r,\* from books) where r>10 and r<=20;

## 课时小结

（10分钟）

1. 复习JDBC数据库访问流程。
2. 分页技术的实现。
   1. 一次取出，内存分页。
   2. 数据库实现分页，多次访问数据库

# 布置作业

（2分钟）

完成分页展示用户信息的实验。