**第十六章 回顾SQL99中的连接查询**

1）内连接

2）外连接

3）自连接

**第十七章 回顾hibernate多表开发**

1）一对一

2）一对多

3）多对多

**第十八章 mybatis一对一映射【学生与身份证】**

1）参见<<一对一图示.JPG>>

2）创建students.sql和cards.sql

|  |
| --- |
| drop table students;  drop table cards;  create table cards(  id int(5) primary key,  num varchar(20)  );  create table students(  id int(5) primary key,  name varchar(10),  cid int(5),  constraint cid\_fk foreign key(cid) references cards(id)  );  insert into cards(id,num) values(1,'111');  insert into students(id,name,cid) values(1,'哈哈',1); |

3）创建Students.java和Card.java

|  |
| --- |
| **public** **class** Card {  **private** Integer id;  **private** String num;  **private** Student student;  **public** Card(){}  **public** Integer getId() {  **return** id;  }  **public** **void** setId(Integer id) {  **this**.id = id;  }  **public** String getNum() {  **return** num;  }  **public** **void** setNum(String num) {  **this**.num = num;  }  **public** Student getStudent() {  **return** student;  }  **public** **void** setStudent(Student student) {  **this**.student = student;  }  }  **public** **class** Student {  **private** Integer id;  **private** String name;  **private** Card card;  **public** Student(){}  **public** Integer getId() {  **return** id;  }  **public** **void** setId(Integer id) {  **this**.id = id;  }  **public** String getName() {  **return** name;  }  **public** **void** setName(String name) {  **this**.name = name;  }  **public** Card getCard() {  **return** card;  }  **public** **void** setCard(Card card) {  **this**.card = card;  }  } |

4）创建StudentMapper.xml和CardMapper.xml

CardMapper.xml

|  |
| --- |
| <?xml version="1.0" encoding="UTF-8" ?>  <!DOCTYPE mapper PUBLIC "-//mybatis.org//DTD Mapper 3.0//EN"  "http://mybatis.org/dtd/mybatis-3-mapper.dtd">  <mapper namespace="cardNamespace">  <resultMap type="cn.itcast.javaee.mybatis.one2one.Card" id="cardMap">  <id property="id" column="id" />  <result property="num" column="num" />  </resultMap>  </mapper> |

StudentMapper.xml

|  |
| --- |
| <?xml version="1.0" encoding="UTF-8" ?>  <!DOCTYPE mapper PUBLIC "-//mybatis.org//DTD Mapper 3.0//EN"  "http://mybatis.org/dtd/mybatis-3-mapper.dtd">  <mapper namespace="studentNamespace">  <resultMap type="cn.itcast.javaee.mybatis.one2one.Student" id="studentMap">  <id property="id" column="id" />  <result property="name" column="name"/>  <association property="card" resultMap="cardNamespace.cardMap"/>  </resultMap>  <select id="findById" parameterType="int" resultMap="studentMap">  select s.id,s.name,c.id,c.num  from students s inner join cards c  on s.cid = c.id  and s.id = #{id}  </select>  </mapper> |

5）创建StudentCardDao.java

|  |
| --- |
| **public** **class** StudentCardDao {  /\*\*  \* 查询1号【学生】  \*/  **public** Student findById(**int** id) **throws** Exception{  SqlSession sqlSession = **null**;  **try**{  sqlSession = MybatisUtil.*getSqlSession*();  **return** sqlSession.selectOne("studentNamespace.findById",id);  }**catch**(Exception e){  e.printStackTrace();  **throw** e;  }**finally**{  MybatisUtil.*closeSqlSession*();  }  }  **public** **static** **void** main(String[] args) **throws** Exception{  StudentCardDao dao = **new** StudentCardDao();  Student student = dao.findById(1);  System.*out*.println(student.getId()+":"+student.getName());  System.*out*.println(student.getCard().getId()+":"+student.getCard().getNum());  }  } |

**第十九章 mybatis一对多映射【班级与学生】**

1）参见<<一对多图示.JPG>>

2）创建grades.sql和students.sql

|  |
| --- |
| drop table students;  drop table grades;  create table grades(  gid int(5) primary key,  gname varchar(10)  );  create table students(  sid int(5) primary key,  sname varchar(10),  sgid int(5),  constraint sgid\_fk foreign key(sgid) references grades(gid)  );  insert into grades(gid,gname) values(1,'java');  insert into students(sid,sname,sgid) values(1,'哈哈',1);  insert into students(sid,sname,sgid) values(2,'呵呵',1); |

3）创建Grade.java和Student.sql

|  |
| --- |
| /\*\*  \* 班级（单方）  \* **@author** AdminTC  \*/  **public** **class** Grade {  **private** Integer id;  **private** String name;  **private** List<Student> studentList = **new** ArrayList<Student>();  **public** Grade(){}  **public** Integer getId() {  **return** id;  }  **public** **void** setId(Integer id) {  **this**.id = id;  }  **public** String getName() {  **return** name;  }  **public** **void** setName(String name) {  **this**.name = name;  }  **public** List<Student> getStudentList() {  **return** studentList;  }  **public** **void** setStudentList(List<Student> studentList) {  **this**.studentList = studentList;  }  }  /\*\*  \* 学生（多方）  \* **@author** AdminTC  \*/  **public** **class** Student {  **private** Integer id;  **private** String name;  **private** Grade grade;  **public** Student(){}  **public** Integer getId() {  **return** id;  }  **public** **void** setId(Integer id) {  **this**.id = id;  }  **public** String getName() {  **return** name;  }  **public** **void** setName(String name) {  **this**.name = name;  }  **public** Grade getGrade() {  **return** grade;  }  **public** **void** setGrade(Grade grade) {  **this**.grade = grade;  }  } |

4）创建GradeMapper.xml和StudentMapper.xml

GradeMapper.xml

|  |
| --- |
| <?xml version="1.0" encoding="UTF-8" ?>  <!DOCTYPE mapper PUBLIC "-//mybatis.org//DTD Mapper 3.0//EN"  "http://mybatis.org/dtd/mybatis-3-mapper.dtd">  <mapper namespace="gradeNamespace">  <resultMap type="cn.itcast.javaee.mybatis.one2many.Grade" id="gradeMap">  <id property="id" column="gid" />  <result property="name" column="gname"/>  <collection property="studentList" resultMap="studentNamespace.studentMap"/>  </resultMap>  <select id="findGradeByName" parameterType="string" resultMap="gradeMap">  select g.gid,g.gname,s.sid,s.sname  from grades g,students s  where g.gid = s.sgid  and s.sname = #{name}  </select>  </mapper> |

StudentMapper.xml

|  |
| --- |
| <?xml version="1.0" encoding="UTF-8" ?>  <!DOCTYPE mapper PUBLIC "-//mybatis.org//DTD Mapper 3.0//EN"  "http://mybatis.org/dtd/mybatis-3-mapper.dtd">  <mapper namespace="studentNamespace">  <resultMap type="cn.itcast.javaee.mybatis.one2many.Student" id="studentMap">  <id property="id" column="sid" />  <result property="name" column="sname"/>  <association property="grade" resultMap="gradeNamespace.gradeMap"/>  </resultMap>  <select id="findAllByName" parameterType="string" resultMap="studentMap">  select s.sid,s.sname,g.gid,g.gname  from grades g,students s  where g.gid = s.sgid  and g.gname = #{name}  </select>  </mapper> |

5）创建GradeStudentDao.java

|  |
| --- |
| **public** **class** GradeStudentDao {  /\*\*  \* 查询java班级有哪些【学生】  \*/  **public** List<Student> findAllByName(String name) **throws** Exception{  SqlSession sqlSession = **null**;  **try**{  sqlSession = MybatisUtil.*getSqlSession*();  **return** sqlSession.selectList("studentNamespace.findAllByName",name);  }**catch**(Exception e){  e.printStackTrace();  **throw** e;  }**finally**{  MybatisUtil.*closeSqlSession*();  }  }  /\*\*  \* 查询哈哈属于哪个【班级】  \*/  **public** Grade findGradeByName(String name) **throws** Exception{  SqlSession sqlSession = **null**;  **try**{  sqlSession = MybatisUtil.*getSqlSession*();  **return** sqlSession.selectOne("gradeNamespace.findGradeByName",name);  }**catch**(Exception e){  e.printStackTrace();  **throw** e;  }**finally**{  MybatisUtil.*closeSqlSession*();  }  }  **public** **static** **void** main(String[] args) **throws** Exception{  GradeStudentDao dao = **new** GradeStudentDao();  List<Student> studentList = dao.findAllByName("java");  **for**(Student s : studentList){  System.*out*.println(s.getId()+":"+s.getName()+":"+s.getGrade().getId()+":"+s.getGrade().getName());  }  Grade grade = dao.findGradeByName("哈哈");  System.*out*.println(grade.getId()+":"+grade.getName());  }  } |

**第二十章 mybatis多对多映射【学生与课程】**

1）参见<<多对多图示.JPG>>

2）创建students.sql和courses.sql和middles.sql

|  |
| --- |
| drop table middles;  drop table students;  drop table courses;  create table students(  sid int(5) primary key,  sname varchar(10)  );  create table courses(  cid int(5) primary key,  cname varchar(10)  );  create table middles(  sid int(5),  cid int(5),  primary key(sid,cid)  );  insert into students(sid,sname) values(1,'哈哈');  insert into students(sid,sname) values(2,'呵呵');  insert into courses(cid,cname) values(1,'java');  insert into courses(cid,cname) values(2,'net');  insert into middles(sid,cid) values(1,1);  insert into middles(sid,cid) values(1,2);  insert into middles(sid,cid) values(2,1);  insert into middles(sid,cid) values(2,2);  select \* from students;  select \* from courses;  select \* from middles; |

3）创建Student.java和Course.java

|  |
| --- |
| /\*\*  \* 学生(多方)  \* **@author** AdminTC  \*/  **public** **class** Student {  **private** Integer id;  **private** String name;  **private** List<Course> courseList = **new** ArrayList<Course>();  **public** Student(){}  **public** Integer getId() {  **return** id;  }  **public** **void** setId(Integer id) {  **this**.id = id;  }  **public** String getName() {  **return** name;  }  **public** **void** setName(String name) {  **this**.name = name;  }  **public** List<Course> getCourseList() {  **return** courseList;  }  **public** **void** setCourseList(List<Course> courseList) {  **this**.courseList = courseList;  }  }  /\*\*  \* 课程(多方)  \* **@author** AdminTC  \*/  **public** **class** Course {  **private** Integer id;  **private** String name;  **private** List<Student> studentList = **new** ArrayList<Student>();  **public** Course(){}  **public** Integer getId() {  **return** id;  }  **public** **void** setId(Integer id) {  **this**.id = id;  }  **public** String getName() {  **return** name;  }  **public** **void** setName(String name) {  **this**.name = name;  }  **public** List<Student> getStudentList() {  **return** studentList;  }  **public** **void** setStudentList(List<Student> studentList) {  **this**.studentList = studentList;  }  } |

4）创建StudentMapper.xml和CourseMapper.xml

StudentMapper.xml

|  |
| --- |
| <?xml version="1.0" encoding="UTF-8" ?>  <!DOCTYPE mapper PUBLIC "-//mybatis.org//DTD Mapper 3.0//EN"  "http://mybatis.org/dtd/mybatis-3-mapper.dtd">  <mapper namespace="studentNamespace">  <resultMap type="cn.itcast.javaee.mybatis.many2many.Student" id="studentMap">  <id property="id" column="sid" />  <result property="name" column="sname"/>  </resultMap>  <select id="findStudentByName" parameterType="string" resultMap="studentMap">  select s.sid,s.sname  from students s,middles m,courses c  where s.sid = m.sid  and m.cid = c.cid  and c.cname = #{name}  </select>  </mapper> |

CourseMapper.xml

|  |
| --- |
| <?xml version="1.0" encoding="UTF-8" ?>  <!DOCTYPE mapper PUBLIC "-//mybatis.org//DTD Mapper 3.0//EN"  "http://mybatis.org/dtd/mybatis-3-mapper.dtd">  <mapper namespace="courseNamespace">  <resultMap type="cn.itcast.javaee.mybatis.many2many.Course" id="courseMap">  <id property="id" column="cid" />  <result property="name" column="cname"/>  </resultMap>  <select id="findCourseByName" parameterType="string" resultMap="courseMap">  select c.cid,c.cname  from students s,middles m,courses c  where s.sid = m.sid  and m.cid = c.cid  and s.sname = #{name}  </select>  </mapper> |

5）创建students.sql和courses.sql和middles.sql

|  |
| --- |
| **public** **class** StudentCourseDao {  /\*\*  \* 查询哈哈选学的【课程】  \*/  **public** List<Course> findCourseByName(String name) **throws** Exception{  SqlSession sqlSession = **null**;  **try**{  sqlSession = MybatisUtil.*getSqlSession*();  **return** sqlSession.selectList("courseNamespace.findCourseByName",name);  }**catch**(Exception e){  e.printStackTrace();  **throw** e;  }**finally**{  MybatisUtil.*closeSqlSession*();  }  }  /\*\*  \* 查询java课程有哪些【学生】  \*/  **public** List<Student> findStudentByName(String name) **throws** Exception{  SqlSession sqlSession = **null**;  **try**{  sqlSession = MybatisUtil.*getSqlSession*();  **return** sqlSession.selectList("studentNamespace.findStudentByName",name);  }**catch**(Exception e){  e.printStackTrace();  **throw** e;  }**finally**{  MybatisUtil.*closeSqlSession*();  }  }  **public** **static** **void** main(String[] args) **throws** Exception{  StudentCourseDao dao = **new** StudentCourseDao();  //List<Course> courseList = dao.findCourseByName("哈哈");  //for(Course c : courseList){  // System.out.println(c.getId()+":"+c.getName());  //}  List<Student> studentList = dao.findStudentByName("java");  **for**(Student s : studentList){  System.*out*.println(s.getId()+":"+s.getName());  }  }  } |

**第二十一章 spring + mybatis + mysql/oracle开发**

1）创建一个spring-mybatis-mysql这么一个javaweb或java工程

2）导入spring-ioc，spring-aop，spring-transaction，mybatis，c3p0，mysql/oracle相关的jar包和spring整合mybatis的jar包

3）创建students.sql

|  |
| --- |
| --mysql  create table students(  sid int(5) primary key,  sname varchar(10),  ssal double(8,2)  ); |

4）创建Student.java

|  |
| --- |
| /\*\*  \* 学生  \* **@author** AdminTC  \*/  **public** **class** Student {  **private** Integer id;//编号  **private** String name;//姓名  **private** Double sal;//薪水  **public** Student(){}  **public** Student(Integer id, String name, Double sal) {  **this**.id = id;  **this**.name = name;  **this**.sal = sal;  }  **public** Integer getId() {  **return** id;  }  **public** **void** setId(Integer id) {  **this**.id = id;  }  **public** String getName() {  **return** name;  }  **public** **void** setName(String name) {  **this**.name = name;  }  **public** Double getSal() {  **return** sal;  }  **public** **void** setSal(Double sal) {  **this**.sal = sal;  }  } |

5）创建StudentMapper.xml

|  |
| --- |
| <?xml version="1.0" encoding="UTF-8" ?>  <!DOCTYPE mapper PUBLIC "-//mybatis.org//DTD Mapper 3.0//EN"  "http://mybatis.org/dtd/mybatis-3-mapper.dtd">  <mapper namespace="studentNamespace">  <resultMap type="cn.itcast.javaee.mybatis.entity.Student" id="studentMap">  <id property="id" column="sid" />  <result property="name" column="sname"/>  <result property="sal" column="ssal"/>  </resultMap>  <insert id="insert" parameterType="cn.itcast.javaee.mybatis.entity.Student">  insert into students(sid,sname,ssal) values(#{id},#{name},#{sal})  </insert>  </mapper> |

6）创建StudentDao.java

|  |
| --- |
| **public** **class** StudentDao {  **private** SqlSessionFactory sqlSessionFactory;  **public** **void** setSqlSessionFactory(SqlSessionFactory sqlSessionFactory) {  **this**.sqlSessionFactory = sqlSessionFactory;  }  **public** **void** insert(Student student){  SqlSession sqlSession = sqlSessionFactory.openSession();  sqlSession.insert("studentNamespace.insert",student);  //int i = 10/0;  }  **public** **static** **void** main(String[] args) {  ApplicationContext ac = **new** ClassPathXmlApplicationContext(**new** String[]{"spring.xml"});  StudentDao studentDao = (StudentDao) ac.getBean("studentDaoID");  studentDao.insert(**new** Student(1,"哈哈",7000D));  }  } |

7）在src目录下创建mybatis.xml

|  |
| --- |
| <?xml version="1.0" encoding="UTF-8"?>  <!DOCTYPE configuration PUBLIC "-//mybatis.org//DTD Config 3.0//EN"  "http://mybatis.org/dtd/mybatis-3-config.dtd">  <configuration>  <mappers>  <mapper resource="cn/itcast/javaee/mybatis/entity/StudentMapper.xml"/>  </mappers>  </configuration> |

8）在src目录下创建spring.xml

|  |
| --- |
| <?xml version="1.0" encoding="UTF-8"?>  <beans  xmlns="http://www.springframework.org/schema/beans"  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"  xmlns:context="http://www.springframework.org/schema/context"  xmlns:aop="http://www.springframework.org/schema/aop"  xmlns:tx="http://www.springframework.org/schema/tx"  xmlns:mvc="http://www.springframework.org/schema/mvc"    xsi:schemaLocation="    http://www.springframework.org/schema/beans  http://www.springframework.org/schema/beans/spring-beans-3.0.xsd    http://www.springframework.org/schema/context  http://www.springframework.org/schema/context/spring-context-3.0.xsd    http://www.springframework.org/schema/aop  http://www.springframework.org/schema/aop/spring-aop-3.0.xsd    http://www.springframework.org/schema/tx  http://www.springframework.org/schema/tx/spring-tx-3.0.xsd    http://www.springframework.org/schema/mvc  http://www.springframework.org/schema/mvc/spring-mvc-3.0.xsd    ">  <!-- 配置C3P0连接池(即管理数据库连接) -->  <bean id="comboPooledDataSourceID" class="com.mchange.v2.c3p0.ComboPooledDataSource">  <property name="driverClass" value="com.mysql.jdbc.Driver"/>  <property name="jdbcUrl" value="jdbc:mysql://127.0.0.1:3306/mybatis"/>  <property name="user" value="root"/>  <property name="password" value="root"/>  </bean>      <!-- 配置SqlSessionFactoryBean(即替代MyBatisUtil工具类的作用) -->  <bean id="sqlSessionFactoryBeanID" class="org.mybatis.spring.SqlSessionFactoryBean">  <property name="dataSource" ref="comboPooledDataSourceID"/>  <property name="configLocation" value="classpath:mybatis.xml"/>  </bean>      <!-- 配置事务管理器(即使用JDBC事务管理器) -->  <bean id="dataSourceTransactionManagerID" class="org.springframework.jdbc.datasource.DataSourceTransactionManager">  <property name="dataSource" ref="comboPooledDataSourceID"/>  </bean>      <!-- 配置事务通知(即哪些方法需要事务) -->  <tx:advice id="tx" transaction-manager="dataSourceTransactionManagerID">  <tx:attributes>  <tx:method name="\*" propagation="REQUIRED"/>  </tx:attributes>  </tx:advice>    <!-- 配置事务切面(即哪些包中的类需要事务通知) -->  <aop:config>  <aop:pointcut id="pointcut" expression="execution(\* cn.itcast.javaee.mybatis.dao.\*.\*(..))"/>  <aop:advisor advice-ref="tx" pointcut-ref="pointcut" />  </aop:config>      <!-- 配置StudentDao类 -->  <bean id="studentDaoID" class="cn.itcast.javaee.mybatis.dao.StudentDao">  <property name="sqlSessionFactory" ref="sqlSessionFactoryBeanID"/>  </bean>    </beans> |

**第二十二章 jsp/js/jquery/easyui/json + @springmvc + spring + mybatis + mysql/oracle开发**

1. 员工管理系统--增加员工

**第二十三章 jdbc访问oracle存储过程和存储函数**

1）写一个计算个人所得税的应用

|  |
| --- |
| --定义过程  create or replace procedure get\_rax(salary in number,rax out number)  as  --需要交税的钱  bal number;  begin  bal := salary - 3500;  if bal<=1500 then  rax := bal \* 0.03 - 0;  elsif bal<=4500 then  rax := bal \* 0.1 - 105;  elsif bal<=9000 then  rax := bal \* 0.2 - 555;  elsif bal<=35000 then  rax := bal \* 0.25 - 1005;  elsif bal<=55000 then  rax := bal \* 0.3 - 2755;  elsif bal<=80000 then  rax := bal \* 0.35 - 5505;  else  rax := bal \* 0.45 - 13505;  end if;  end;  /  --调用过程  declare  --交税  rax number;  salary number := &salary;  begin  get\_rax(salary,rax);  dbms\_output.put\_line(salary||'元工资需要交'||rax||'元税');  end;  / |

|  |
| --- |
| **//Java调用过程**  **public** **class** TestCallOracleProc {  **public** **static** **void** main(String[] args) **throws** Exception{  String sql = "{call get\_rax(?,?)}";  Connection conn = JdbcUtil.*getConnection*();  CallableStatement cstmt = conn.prepareCall(sql);  cstmt.setInt(1,10000);  cstmt.registerOutParameter(2,Types.*INTEGER*);  cstmt.execute();  Integer rax = cstmt.getInt(2);  System.*out*.println("10000元需要交" + rax + "元税");  JdbcUtil.*close*(cstmt);  JdbcUtil.*close*(conn);  }  } |

查询7788号员工的的姓名，职位，月薪

|  |
| --- |
| --定义函数  create or replace function findEmpNameAndJobAndSal(pempno in number,pjob out varchar2,psal out number)  return varchar2  as  pename emp.ename%type;  begin  select ename,job,sal into pename,pjob,psal from emp where empno = pempno;  return pename;  end;  /  --调用函数  declare  pename emp.ename%type;  pjob emp.job%type;  psal emp.sal%type;  begin  pename := findEmpNameAndJobAndSal(7788,pjob,psal);  dbms\_output.put\_line('7788'||'--'||pename||'--'||pjob||'--'||psal);  end;  / |

|  |
| --- |
| **//Java调用函数**  **public** **class** TestCallOracleFunc {  **public** **static** **void** main(String[] args) **throws** Exception{  String sql = "{?=call findEmpNameAndJobAndSal(?,?,?)}";  Connection conn = JdbcUtil.*getConnection*();  CallableStatement cstmt = conn.prepareCall(sql);  cstmt.registerOutParameter(1,Types.*VARCHAR*);  cstmt.setInt(2,7788);  cstmt.registerOutParameter(3,Types.*VARCHAR*);  cstmt.registerOutParameter(4,Types.*INTEGER*);  cstmt.execute();  String ename = cstmt.getString(1);  String job = cstmt.getString(3);  Integer sal = cstmt.getInt(4);  System.*out*.println(ename+":"+job+":"+sal);  JdbcUtil.*close*(cstmt);  JdbcUtil.*close*(conn);  }  } |