一、mysql

DDL：数据定义语言

create drop alter

DCL:数据控制语言

grant revoke if

DML:数据操作语言

insert update delete

DQL:数据查询语言

select

insert into 表 values(?,?,?)

update 表 set 名1=值1 ,名2=值2 。。。 where id=?

delete from 表 where id=?

select from where group by having order by limit ?,?

二、多表操作

表关系：

一对一 一对多（多对一） 多对多

内联接

JOIN ON

```
select * from student s JOIN score c ON s.stuid=c.stuid;
select * from student s , score c where s.stuid=c.stuid;
```

外联接

左外

```
select * from student s LEFT JOIN score c ON s.stuid=c.stuid;
```

右外

```
select * from student s right JOIN score c ON s.stuid=c.stuid;
```

子查询

一个sql语句作为另一个sql语句的条件。

扩展：4表联查

-- 查询用户名、订单编号、收货人、商品名称、购买数量

```
SELECT u.name,o.oid,o.name,p.pname,oi.count FROM USER u ,orders o,orderitem
oi,product p
WHERE u.uid=o.uid AND o.oid=oi.oid AND oi.pid=p.pid AND
o.oid='351ac8980f7e4e8dbd4069bedf6560df'

SELECT u.name,o.oid,o.name,p.pname,oi.count FROM USER u
JOIN orders o ON u.uid=o.uid
JOIN orderitem oi ON o.oid=oi.oid
JOIN product p ON p.pid=oi.pid
WHERE o.oid='351ac8980f7e4e8dbd4069bedf6560df'
```

三、JDBC　连接池

//加载驱动

//创建连接

//得到执行sql语句的对象

//执行sql语句，并返回结果

//处理结果

//关闭资源

C3P0　DBCP

装饰者设计模式

口决：

编写一个类，实现与被包装类相同的接口
在类中声明一个被包装类类型的变量
通过构造方法给变量赋值
对于不需要修改的方法，调用原有的方法
对于需要修改的方法，编写自己的代码

四、Class 反射：乃框架设计之灵魂。

Student.class

stu.getClass();

Class.forName("com.itheima.domain.Student");

Method[] clazz.getMethods(); //得到所有公共的方法（包括继承的）
Method[] clazz.getDeclaredMethod(); //得到本类中的所有方法（包括私有的）
clazz.getMethod(String name,Object...);
clazz.getDeclaredMethod(String name,Object...);

五、http协议 tomcat服务器

请求：

请求行

请求头
请求体

响应：

响应行
响应头
响应体

六、Servlet

Servlet的作用：使用页面与java代码可以数据交互。

职责：

1、获得请求参数

2、调用业务
3、分发转向

Servlet
GenricSevlet
HttpServlet

ServletConfig
ServletContext
ServletRequest
ServletResponse
HttpServletRequest
HttpServletResponse

七、HttpRequest HttpResponse

request:得到请求数据
请求行
getMethod();
getRequestURL();
getRequestURI();
***** getContextPath();
请求头

请求体
getParameter();
getParameterValues();
getParameterMap();

getInputStream();

作为域对象的方法
setAttribute(String name,Object value);
getAttribute(String name);
removeAttribute(String name);

其它方法：
setCharacterEncoding("UTF-8"); //解决post方式的请求乱码

getRequestDisPatcher("/index.jsp").forward(request,response);

response:响应信息
响应行
http/1.1 200 OK
setStatus(302);
响应头
setHeader(String name,String value);
响应体

PrintWriter out = getWriter();
out.write("<h1>hello </h1>");
out.print();

ServletOutputStream getOutputStream();

其它方法
sendRedirect("url");

response.setCharacterEncoding("UTF-8");//告知服务器使用什么编码

response.setHeader("Content-Type","text/html;charset=UTF-8");

//即告知服务器又告知浏览器使用什么编码
setContentType("text/html;charset=UTF-8");

请求转发和重定向区别：
转发：
在服务器端执行
只执行一次
地址栏不变
可以传递数据
不能跳转到其它网站
路径不用写项目名

重定向：
在客户端由浏览器执行
执行两次
地址会变
不能传递数据
可以跳转到其它网站
路径前要写项目名

八、会话：
如同打电话。
***** 会话解决什么问题？
保存用户各自的数据。
会话使用的技术：
Cookie:
保存数据在客户端浏览器。不安全。只能保存字符串，且少量数据。
session:
保存数据在服务器。安全。可以保存对象类型数据。

如何使用：
对于Cookie的使用：
创建：
Cookie cookie = new Cookie(String name,String value);
常用的属性：
name:不能修改。
value:只能存字符串，不能存中文
path:设置Cookie数据的访问路径。
默认路径：是创建Cookie的Servlet的路径。
setPath("/");当前应用下的所有资源都可以访问此Cookie
maxAge:
-1：默认。Cookie数据保存到浏览器关闭之前。
0：立即删除
正数：保存Cookie数据到磁盘（时间为：秒）

写回到浏览器：
response.addCookie(cookie);

得到Cookie数组
request.getCookies();

对于HttpSession:
创建：
getSession();

setAttribute(String name,Object value);
getAttribute(String name);
removeAttribute(String name);

getId();

销毁：
默认30分钟。
invalidate();直接销毁
服务器非正常关闭。

***** session是依赖于Cookie的：
getSession(){
//1、获取Cookie中的JSESSIONID
//2、如果浏览器没有带过来JSESSIONID，创建session,并生成JSESSIONID，同时写回到浏览器
//3、如果浏览器有带过来JSESSIONID，
找到对应JSESSIONID的session直接使用
如果session销毁了，则执行步骤2
}

十、JSP ： java server pages
JSP就是Servlet。

执行原理：
翻译 编译 执行
.jsp .java .class

jsp中可以编写的内容：
1、静态内容。
2、脚本：
<% 小脚本 %>
<%= 表达式 %> out.print();
<%! 声明 （全局成员） %>
3、注释
<%-- jsp --%>
<!-- html -->
4、3个指令
page include taglib
5、6个动作
<jsp:forward
<jsp:param

<jsp:include
<jsp:useBean
<jsp:setProperty
<jsp:getProperty

6、9个内容对象
out
request
response
session
application
exception
page
pageContext
config

十一、开发模式：

解决的问题：强内聚、弱耦合。

Model1: jsp + javaBean

Model2: MVC jsp + Servlet + javaBean

M:javaBean(封装数据)
V:JSP（展示数据）
C:Servlet(控制器)

三层思想：
web（表示层） service（业务层） dao（数据访问层）

EL:表达式语言
${p} == findAttribute("p");

***** EL表达式只能获取域对象中的数据

${user.name} == user.getName();

${user["name"]}

${user.city.address} 属性导航

${list[0]}

JSTL:

3类标签
通用标签：set、out、remove
条件：if choose
迭代：forEach
<c:forEach var="" begin="" end="" step="" items="" varStatus="vs">

</c:forEach>

十二、JDBC事务:

要想使用多条sql语句在一个事务中如何保证？
使用同一个Connection对象！

事务特性：
原子性：
一致性：
隔离性：
持久性：

十三、ajax
原生js
XMLHttpRequest

1个事件：
onreadystatechange
2个方法：
open()
send()
3个属性：
readyState 4
status 200
responseText 获得响应正文
$.post(url,data,funciton(result){},"json")
$.get();

十四、过滤器
Filter
过滤器链：