1.模糊查询

SELECT 字段 FROM 表 WHERE 某字段 Like 条件

SELECT \* from user WHERE login\_name LIKE '%yt%'

在服务端输出json数据时按照一定的格式输出时间字段，fastjson支持两种方式：

1.使用JSON.toJSONStringWithDateFormat方法

2.JSON.toJSONString方法增加SerializerFeature.WriteDateUseDateFormat参数

B/S结构（Browser/Server，浏览器/服务器模式）

C/S结构 服务器到浏览器

左连接

SELECT \* from user LEFT JOIN job\_hunter ON user.id=job\_hunter.user\_id;

LEFT  JOIN或LEFT OUTER JOIN       
左向外联接的结果集包括  LEFT OUTER子句中指定的左表的所有行，而不仅仅是联接列所匹配的行。如果左表的某行在右表中没有匹配行，则在相关联的结果集行中右表的所有选择列表列均为空值。         
2）RIGHT  JOIN 或 RIGHT  OUTER  JOIN       
右向外联接是左向外联接的反向联接。将返回右表的所有行。如果右表的某行在左表中没有匹配行，则将为左表返回空值。

SpringMVC常用注解

@requestMapping 用于请求url映射

@RequestBody注解实现接收http请求的json数据，将json数据转换为java对象

@ReponseBody注解实现将controller方法返回对象转化为json相应给客户

什么是SpringBoot

1.用来简化spring应用的初始搭建以及开发过程 使用特定的方式来进行配置（properties或yml文件）

2.创建独立的spring引用程序 main方法运行

3.嵌入的Tomcat 无需部署war文件

4.简化maven配置

5.自动配置spring添加对应功能starter自动化配置-spring-boot-starter-data-jpa (数据库支持)

springboot自动配置的原理

1.@EnableAutoConfiguration这个注释告诉SpringBoot“猜”你将如何想配置Spring,基于你已经添加jar依赖项。如果spring-boot-starter-web已经添加Tomcat和Spring MVC,这个注释自动将假设您正在开发一个web应用程序并添加相应的spring设置。会自动去maven中读取每个starter中的spring.factories文件 该文件里配置了所有需要被创建spring容器中的bean

2.使用@SpringbootApplication注解 可以解决根类或者配置类（我自己的说法，就是main所在类）头上注解过多的问题，一个@SpringbootApplication相当于@Configuration,@EnableAutoConfiguration和 @ComponentScan 并具有他们的默认属性值

private final static Logger logger = LoggerFactory.getLogger(UserController.class);

1、 定义成static final,logger变量不可变，读取速度快  
 2、static 修饰的变量是不管创建了new了多少个实例，也只创建一次，节省空间，如果每次都创建Logger的话比较浪费内存；final修饰表示不可更改，常量  
 3、将域定义为static,每个类中只有一个这样的域.而每一个对象对于所有的实例域却都有自己的一份拷贝.，用static修饰既节约空间，效率也好。final 是本 logger 不能再指向其他 Logger 对象