还是得循序渐进。现在，我们先接受这一点：在 JavaScript 中，一个函数可以作为另一个函数接收一个参数。我们可以先定义一个函数，然后传递，也可以在传递参数的地方直接定义函数。

```
SELECT c.id,c.class_name as '班级名称',count(1) as '班级人数',GROUP_CONCAT(s.student_name)
WHERE  s.class_id is NOT NULL
GROUP BY s.class_id
```

```
SELECT c.id,c.class_name as '班级名称',count(1) as '班级人数',GROUP_CONCAT(s.student_name)
WHERE  s.class_id is NOT NULL
GROUP BY s.class_id WITH ROLLUE;
```

```
36
37 --字符串
38 --文件类-io(common-io.jar--FileUtil)
39 --图片类工具类(裁剪，旋转，压缩，等等)
40 --验证码
41 --net抓取网页源代码和操作网页源代码(jsoup/httpclient/正则表达式)
42 --反射
43 --文档解析(office办公软件的解析，导入，导入导出)
44 --音频类-mp3-mp4的解析
45 --邮件发送
46 --手机发送，支付接口，第三方登录
47 --数字类
48 ---密码类
49 --xml解析
50 --JSON类
51 --BeanUtil转换
52
```

```
54
55 --java---人工智能--科大（讯飞拼音）
56 --通过过java调用第三方接口-可以控制硬件
57 --语音识别，文字语音
58
```

**程序清单 2-4　使用 AtomicLong 类型的变量来统计已处理请求的数量**

```
@ThreadSafe
public class CountingFactorizer implements Servlet {
    private final AtomicLong count = new AtomicLong(0);

    public long getCount() { return count.get(); }

    public void service(ServletRequest req, ServletResponse resp) {
        BigInteger i = extractFromRequest(req);
        BigInteger[] factors = factor(i);
        count.incrementAndGet();
        encodeIntoResponse(resp, factors);
    }
}
```

```
package exam.util;

import java.util.ArrayList;
import java.util.HashMap;
import java.util.Map;
import java.util.Scanner;

import javax.jms.Session;

import cn.netinnet.qdk.util.StringUtil;

public class SqlUtil {
    static Map<String, String> sessionMap = new HashMap<String, String>();
    //session 缓存的字符串
    static{
        sessionMap.put("create_user", "session.user_code");
        sessionMap.put("user_code", "session.user_code");
        sessionMap.put("institution_code", "session.agency");

    }
    /**
     * main(将普通的 sql 转成 action 中的 sql，个别字符还需要自己调整)
     * (这里描述这个方法适用条件 – 可选)
     * @param args
     *void
     * @exception
     * @since  1.0.0
      */
    public static void main(String[] args) {
        //StringBuilder sql = new StringBuilder("select ec.*, et.teacher_name,(select count(1) from e_class_student ecs where ecs.class_id=ec.class_id and ecs.institution_code='yx') as student_num, (select count(1) from e_task et where et.class_id=ec.class_id and et.status!='2' and et.institution_code='yx') as task_num , (SELECT et.teacher_name FROM e_teacher et WHERE et.teacher_code = ec.create_user ) create_teacher_name from e_class ec left join e_teacher et on et.teacher_id=ec.teacher_id where ec.institution_code='yx' and (('teacher'='teacher' and ec.teacher_id='1') or ec.create_user='netinnet' or 'teacher'='agency') and et.teacher_name like '%%' and class_name like '%%' order by ec.create_date desc;");
        System.out.println("请输入 sql 之后换行输入 end：" );

        Scanner scn = new Scanner(System.in);
```

```
        StringBuilder sql = new StringBuilder("");
        String line;
        while (!"end".equals(line = scn.nextLine())) {
            sql.append(line+"    \n");
        }

        //System.out.println(sql);


        //StringBuilder sql = new StringBuilder("SELECT * FROM e_topic et WHERE topic_id='1111'");
        System.out.println(sql);
        System.out.println("转换后的 sql： " );
        System.out.println(actionSql(sql));
    }
    /**
     * main(这里用一句话描述这个方法的作用)
     * (这里描述这个方法适用条件 – 可选)
     * @param args
     *void
     * @exception
     * @since  1.0.0
      */
    public static String actionSql(StringBuilder sql){
        String[] keyIndex={"where"};
        String tempSql = sql.toString().toLowerCase();
        sql = new StringBuilder(tempSql);
        int key=0;int pos = -1;
        while(key<keyIndex.length && (pos=sql.indexOf(keyIndex[key]))==-1){
            key++;
        };

        //if(pos == -1) return "找不到关键字";
        int fromIndex = 0;
        while ((pos=tempSql.indexOf("=", fromIndex))!=-1){
            System.out.println( " ========");
            fromIndex = pos+1;
            suffixBean leftBean = findkey(tempSql,pos,-1);
            suffixBean rightBean = findkey(tempSql,pos,1);

            String leftkey = leftBean.getSuffix();
            String rightkey = rightBean.getSuffix();
            System.out.println(leftkey);
            System.out.println(rightkey);
```

```
                if(leftkey.indexOf(".") > -1 && rightkey.indexOf(".") > -1){
                    continue;
                }
                if(leftkey.startsWith("'") && leftkey.endsWith("'") && rightkey.startsWith("'") &&
rightkey.endsWith("'")){
                    continue;
                }

                if(leftkey.startsWith("'") && leftkey.endsWith("'")){
                    suffixBean tempBean = leftBean;
                    leftBean = rightBean;
                    rightBean = tempBean;
                    leftkey = leftBean.getSuffix();
                    rightkey = rightBean.getSuffix();
                }
                String val =getName(leftkey,rightkey);
                System.out.println(val);

                tempSql =tempSql.substring(0, rightBean.getStart())+val +
tempSql.substring(rightBean.getEnd());
                fromIndex =pos+ val.length()+1;
            }
            tempSql = tempSql.replaceAll("\n", " &&\n");
            tempSql = tempSql.substring(0,tempSql.length()-3);
            return tempSql;
        }
        private static String getName(String leftkey,String rightkey) {
            String preChar ="";
            if(rightkey.startsWith("'") || rightkey.startsWith("\"")){
                preChar = rightkey.substring(0, 1);
                //leftkey = leftkey.substring(1,leftkey.length() -1);
            }

            return preChar+ Arguments(findName(leftkey)) +preChar;
        }

        public static String Arguments(String name) {
            if(sessionMap.get(name) != null){
                name = sessionMap.get(name);
            }
            return "[:"+name+"]";
        }
        private static String findName(String str){
```

```
        if(str == null || str.length() == 0) return "";
        int start = 0;
        int end = str.length();
        if(str.contains("ifnull")){
            start = str.indexOf("ifnull") + "ifnull(".length() ;
            end = str.indexOf(",")    ;
        }
        str = str.substring(start,end);
        if(str.indexOf(".") > -1){
            start = str.indexOf(".") + 1 ;
        }
        //ifnull(topic,1)

        return str.substring(start);
    }
    /*寻找=号两边的字符串*/
    public static suffixBean findkey(String tempSql, int pos, int dir) {
        char c;
        int next    = pos;
        boolean flag = false;

        next +=dir;
        int start = next;
        c = tempSql.charAt(next);
        if( c == '<' || c == '>' || c == '!'){
            next +=dir;
            start = next;
        }
        if( ('\'' == c)||    ('\"' == c)    ||    ('0' <= c && c <= '0') ||('A' <= c && c <= 'Z') ||('a' <= c
&& c <= 'z') ){
            start = next;
        }
        if(c == ' '){
            while((next = next + dir) <tempSql.length() && tempSql.charAt(next) == ' ' );
            start = next;
        }

        char startChar = ' ';
        startChar = tempSql.charAt(next);
        if(startChar !='\'' && startChar !='\"'){
            startChar = ' ';
        }
        while((next = next + dir) <tempSql.length() ){
            c = tempSql.charAt(next);
```

```
                if(c == startChar){
                    if(c == ' '){
                        next = next - dir;
                    }
                    break;
                }
            };

            int end = next;
            if (start > next ) {int temp = start; start = end; end = temp; }
            return new suffixBean(start,end+1,tempSql.substring(start,end+1));
        }

    }
    class suffixBean{
        int start;
        int end;
        String suffix;
        public suffixBean(){

        }
        public suffixBean(int start,int end,String suffix){
            this.start = start;
            this.end = end;
            this.suffix = suffix;
        }
        public int getStart() {
            return start;
        }
        public void setStart(int start) {
            this.start = start;
        }
        public int getEnd() {
            return end;
        }
        public void setEnd(int end) {
            this.end = end;
        }
        public String getSuffix() {
            return suffix;
        }
        public void setSuffix(String suffix) {
            this.suffix = suffix;
        }
```

```
    @Override
    public String toString() {
        return "suffixBean [start=" + start + ", end=" + end + ", suffix="
                + suffix + "]";
    }
}
```