

Key-Value 键值对数据在新生儿听力筛查登记系统中的应用

黎勇¹, 易松¹, 邓忠², 谭东辉²

(湘南学院附属医院信息科¹, 耳鼻咽喉头颈外科², 湖南 郴州 423000)

摘要:本文围绕新生儿听力筛查登记系统的技术实现,着重从新生儿不同报告单资料数据的保存及展示方面来阐述技术核心实现过程,针对不同的基层医院需要存储的新生儿数据格式不一致问题,在关系数据库中采用 Key-Value 形式来进行存储,而数据的查询展示则利用数据库编程技术或 JAVA 函数式编程技术实现。

关键词:Key-Value 数据存储;数据库编程;JAVA 函数式编程

中图分类号:TP311.13;R764.04

文献标识码:B

DOI:10.3969/j.issn.1006-1959.2019.04.004

文章编号:1006-1959(2019)04-0011-03

Application of Key-Value Key Data in the Neonatal Hearing Screening Registration System

LI Yong¹, YI Song¹, DENG Zhong², TANDong-hui²

(Department of Information¹, Department of Otorhinolaryngology Head and Neck Surgery², Affiliated Hospital of Xiangnan University, Chenzhou 423000, Hunan, China)

Abstract: This paper focuses on the technical realization of the newborn hearing screening registration system, focusing on the technical core implementation process from the preservation and display of different report data of newborns, and the inconsistency of the newborn data format that needs to be stored in different primary hospitals. The relational database uses the Key-Value form for storage, while the data query display is implemented using database programming techniques or JAVA functional programming techniques.

Key words: Key-Value data storage; Database programming; JAVA functional programming

先天性听力损失是新生儿最常见的出生缺陷之一,1‰~3‰的发病率居我国五类致残因素之首^[1]。就湖南省郴州地区而言,新生儿听力筛查存在如下问题:①手工信息的处理方式不但工作量大、流程复杂,而且数据的收集、整合过程难免错漏、残缺,针对大量数据的线性增长无法做到较好的存储,对数据的分析、统计更加困难。②不同地区,由于各医院的信息管理系统相互独立,患者资料难以做到实时共享,从而造成了重复检查,不但加大了患者负担,同时也浪费了医疗有限资源。利用信息化技术实现的新生儿听力筛查登记系统,不但解放了劳动量,有效的将手工作业模式转换为信息自动化管理模式,同时也确保了数据的准确性、全面性和时效性^[1]。新生儿听力筛查登记的实现其主要技术难点在于不同基层医院数据格式的统一存储、数据查询展示及数据的统计分析。现将具体技术实现介绍如下。

1 数据的存储

由于基层医院使用不同的报告单,要将报告单内容存储到关系数据库的同一个数据实体表就必须将拥有不同内容的登记表单转换为统一格式存储,一种实现方法是将整个报告单转换为 JSON 格式进行存储,若要对数据做查询分析则需要解析存储的 JSON 格式数据^[2]。考虑到系统性能及技术实现,在数据存储中采用了另一种存储方式,即在关系型数据库中采取非表格式的 Key-Value 键值对格式的数据存储方式,将不同表单数据转换为 Key-Value 键值

对形式的纵向存储^[3]。因此,在数据库中设计如表 1 结构的数据表。

表 1 数据库表结构

| 字段名称 | 字段属性 | 字段说明 |
|------------|--------------|-----------|
| id | varchar(36) | 主键 |
| templateid | varchar(36) | 关联报告单模板主键 |
| field | varchar(30) | 报告单字段 |
| value | varchar(255) | 报告单字段值 |
| rowid | varchar(36) | 同一报告单数据标识 |

在数据实体表的设计中, id 为表的主键, 通过采用注解 @GeneratedValue 的生成策略, 并指定策略的 strategy 属性为 AUTO, 自动生成最适合底层数据库的主键生成策略; templateid 为对应的报告单模板 ID, 即不同基层医院所使用的不同报告单登记模板的后台标识; field 为报告单登记模板中所用组件的命名, 如下拉框男女选择组件命名为 sex 等; value 则为报告单重组件对应的值; rowid 是通过 UUID.randomUUID() 生成的 36 位字符型长度的唯一识别码, 用于标识一个报告单的所有值。

2 数据的展示

数据的展示通过使用 easyui 前台框架的 datagrid 来展示, 实现步骤如下: ①第一步: 选择报告单模板, 通过选择的报告单模板检索报告单的字段配置信息。②第二步: 通过 <c:forEach> 标签循环读取后台返回的报告单 List 字段配置列表, 并生成 datagrid 中的 table 展示列。③第三步: 通过 POST 方式请求后台展示数据。在请求后台数据展示实现中即可通过数据库存储过程的编程实现, 同时又可通过 JAVA 函数式编程实现, 两种方式的实现核心主要体现对纵向存储的数据做行列转换及行合并操作。

基金项目: 郴州市科技局计划项目(编号: CZKJ2016036)

作者简介: 黎勇(1978.11-), 男, 湖南嘉禾人, 本科, 高级工程师, 主要从事医院信息系统集成的研究

通讯作者: 谭东辉(1968.7-), 男, 湖南桂阳人, 硕士, 教授, 主任医师, 主要从事耳鼻咽喉疾病的诊治

2.1 存储过程实现 存储过程编程主要利用动态 SQL 的拼装以及循环结构^[4,5]来实现数据的展示功能,其实现步骤如下。

第一步:定义存储过程,并确定传入、传出参数。传入参数包括了模板 id、查询条件 conditions、开始页数 page、页大小 pagesize;传出参数仅包含了符合条件的总数据条数 totalcount。

第二步:通过传入的查询条件字符 conditions 参数(格式:“field;condition;value|field;condition;value|…”),拆分重组为 SQL 查询条件语句,结合传入的开始页参数 page 及页大小 pagesize 筛选出符合条件的指定报告单模板数据。

第三步:利用 REPEAT …UNTIL …END REPEAT 循环读取所选报告单模板字段配置,利用数据库系统函数 CONCAT 及 GROUP_CONCAT 生成动态查询语句,完成数据的行列转换并输出显示。CONCAT 函数主要完成字符串的拼接,GROUP_CONCAT 函数主要作用是完成行合并。其实现代码如下所示:

```
BEGIN
    DECLARE i INT;
    SET i = (SELECT COUNT (*) FROM
tb_temp_field);
    SET @a='';

    REPEAT
    SET @field=(SELECT DISTINCT field FROM
tb_temp_field WHERE id=i);
    SET @a=CONCAT (@a,'GROUP_CONCAT (IF
(field ='',@field,'','value,''),'\ ' SEPARATOR '\ '
AS '\',@field,'\');
    SET i = i - 1;
    UNTIL i <= 0

    END REPEAT;
END;
```

第四步:利用 Hibernate 的 createStoredProc-
cedureCall 方法调用指定名称的存储过程,通过 pro-
cedureCall.getOutputs ().getCurrent () 获取 Result-
SetOutput 结果,最终通过 ResultSetOutput 获取到
datagird 的 list 数据列表,OUT 模式的 totalcount 参
数则用于分页,并展示到前台界面。

采用存储过程实现的数据展示简化了系统的
Java 编程,意味着部分业务逻辑不再应用程序中处
理,同时所有的记录检索等操作都是通过存储过程
实现,因此采用存储过程编程技术的实现减少了网
络交互,而由于不同数据库的内部编程语法不太一
致,导致了系统不具备可移植性。

2.2 JAVA 编程利用 JAVA8 的新特性实现 JAVA 函
数式编程主要利用对核心类库的集合类的 API 和
新引入的流(Stream)^[6]的改进来实现数据的展示功
能,其实现主要步骤如下:

第一步:通过 Service 层接口调用自定义函数
getDataByTemplateId(String id),该函数的主要功能
是获取传入的指定登记模板的 ID 来获取数据存储
实体表中符合条件的所有记录,其返回结果为 List
列表对象。

第二步:通过 Stream API 处理数据集合,使用
collect 来将一个数据流中的元素进行分组,由于在
数据存储表中同一报告单的一次表单提交数据是使
用的 UUID.randomUUID()生成的唯一标识,通过 col-
lect 中的数据分组将产生 Map<String,List>的结果,
其实现代码如下:

```
Map<String, List<Data>> datamap=lists.stream().collect(
Collectors.groupingBy(Data::getRowid));
```

在实现代码中,应用了定义在辅助类 Collects
中的 groupingBy 收集器,它通过接收 Data 实体的
getRowid 分类方法,并生成 Map 结果,其中
getRowid 方法返回的结果将作为 Map 的 Key,符合
条件的分类 lists 集合记录将作为 Map 的 Value。

第三步:申明空的 list 对象 List<Map<String,
Object>> datalist=new ArrayList<Map<String, Ob-
ject>>(),用于存储最终的数据结果。

第四步:申明迭代器 Iterator,用于循环读取
datamap 中的内容,申明语句如下:

```
Iterator<Entry<String, List<Data>>> it= datamap.en-
trySet().iterator();
```

第五步:使用 while 循环读取迭代器中的内容,
并将一组 list 内容通过 forEach 循环读取并 put 到新
申明的 map 对象中,最后将最终结果使用 add 加入
到第三步中的 datalist,到此完成了同一报告单的数据
分组及数据的行列转换展示,java 代码如下所示:

```
Iterator<Entry<String, List< Data>>> it=datamap.en-
trySet().iterator();
while(it.hasNext()) {
    Entry<String, List<Data>> entry=it.next();
    String rowid=entry.getKey();
    List<Data> datas=entry.getValue();
    Map<String, Object> data=new HashMap<>();
    valid.put("rowid", rowid);
    datas.forEach(v->{
        data.put(v.getField(), v.getValue());
    });
    datalist.add(data);
}
```

采用 JAVA8 编程技术实现即所有的业务逻辑都通过 JAVA 编程实现,采用函数式编程实现不仅有利于系统的扩展性、容错性,同时也做到了可移植性^[7]。

3 总结

利用 Key-Value 键值对格式有效的解决了不同报告单内容字段不一致而无法存储在关系数据库的同一数据实体表中的存储问题,同时 Key-Value 键值对格式的存储还为数据的查询及统计分析提供了便利。在数据的展示实现过程中,既可利用数据库的存储过程编程技术实现又可利用 JAVA8 的函数式编程实现。存储过程的实现简化了 JAVA 编码,利用传入的 conditions 参数,先检索符合条件的 rowid 值,通过 rowid 值快速查询有效的数据,并对数据执行行列转换及行合并的操作,利用存储过程的实现增加了数据库服务端的处理成本,大量逻辑业务的处理约束了软件的架构。利用 JAVA8 的新特性实现数据展示仅需数据库实体表,无需实现自定义函数

及存储过程,同时在系统发布过程中可以做到多 web 容器部署,从而实现负载均衡。

参考文献:

- [1]黎曼侬,王琼,党峰博,等.新生儿疾病筛查信息系统管理软件应用的效果分析[J].中国医学工程,2014(1):186-188.
- [2]徐宝磊,罗江,潘刚.基于 JSON 的数据交换技术应用研究[J].软件导刊,2017,10(16):173-175.
- [3]蒋付彬,王华军.NoSQL 数据库的应用及选型研究[J].信息与电脑(理论版),2016(3):141-142.
- [4]王志刚,江友华.MySQL 高效编程[M].北京:人民邮电出版社,2012:116-128.
- [5]梁敬彬.探讨动态 SQL 扩展的应用[J].福建电脑,2018,34(3):92-94.
- [6]Richard Warburton,著.Java8 函数式编程[M].王群峰,译.北京:人民邮电出版社,2015:15-31.
- [7]官明.函数式编程探析[J].电脑编程技巧与维护,2017(3):29-30.

收稿日期:2018-11-12;修回日期:2018-11-22

编辑/杨倩