摘要

1. 绪论

## 开发背景

随着计算机的普及以及网络的快速发展，各式各样的互联网产品应运而生，而这些互联网产品的诞生也极大地简便了人们的生活，因此，使用互联网产品的人越来越多，互联网产品逐渐成为人们生活中不可或缺的一部分。然而在使用人数人数不断增加以及全国大联网的形势下，这些互联网产品在性能方面也面临着严峻的挑战。它们必须能够在2秒内将请求信息返回给客户端，如果是使用常规的数据库，比如oracle等来存储数据，那么在海量数据的情况下，这个业务要求是无法达到的，所以更多的互联网产品会选择使用elasticsearch来存储一些在产品使用过程中变化不大的静态数据，elasticsearch能够解决查询效率的问题，但是elasticsearch提供的java接口使用起来较为复杂，而且产品的开发人员还要花一定的时间去学习使用elasticsearch，这无疑增加了开发的工作量，延长了开发周期。所以针对elasticsearch的使用情况，我们可以对elasticsearch的API进行二次封装，提供一个更简单的接口，从而简化开发。

## 研究现状及发展趋势

## 课题意义与目的

## 研究内容

## 1.5 论文形式

1. 相关技术介绍

## 2.1 elasticsearch

## 2.2 java

1. 需求分析

## 3.1 功能需求

1. 总体设计

## 4.1 结构设计

为了使该API的使用更简单，开发人员能够容易上手，同时又能够达到封装elasticsearch操作的目的，该API采用类似于jdbc的形式，通过发送sql语句向elasticsearch发送操作命令。结构设计图如下：

更新体

查询体

Sql

Sql

连接对象

连接信息

驱动程序获取对elasticsearch的连接

控 制 器

更新指挥器

查询指挥器

Elasticsearch 服务器

调 用 端

结果

结果

Sql

结果

图 4-1 API 结构图

## 4.2 功能模块设计

Elasticsearch API

添加模块

控制模块

查询模块

删除模块

更新模块

修改模块

创建模块

图 4-2 API 功能模块

ES客户端模块

驱动模块

固定sql模块

动态sql模块

Elasticsearch API 各模块功能如下：

1. 驱动模块：

解析连接信息，建立对elasticsearch的连接

1. 控制模块：

整个API的核心，通过判断输入的sql语句的类型（查询或者更新等）将sql请求转发给对应的模块来处理，最后接收相应的结果返回给用户；支持固定sql和动态sql

1. 查询模块

解析sql语句，根据解析的结果对elasticsearch执行查询，最后封装查询结果；支持的查询类型包括：聚合查询，去重查询，条件查询，排序查询，分页查询。

1. 添加模块

解析sql语句，对已存在的表插入新数据；支持批量插入

1. 删除模块

解析sql语句，删除符合某个条件的数据；支持批量删除

1. 更新模块

解析sql语句，更新某些记录的值；支持批量更新

1. 修改模块

解析sql语句，为表添加新字段或者修改已有字段的属性

1. 创建模块

读取源文件下的配置文件，根据配置文件创建对应的表

1. ES客户端模块

持有elasticsearch连接对象，封装对elasticsearch的操作，供其他功能模块调用

1. 详细设计与实现

## 5.1 驱动模块

驱动模块通过实现JDBC的Driver接口来实现驱动注册和获取连接的功能。当调用端发送包含连接信息的url时，程序首先通过正则表达式来判断该URL是否满足一定的格式，若是满足，则解析该URL，从中获取elasticsearch的主机地址，端口号以及索引名，然后调用ES客户端模块建立到elasticsearch的连接，紧接着获取elasticsearch该索引的mapping，从而获取该索引下的所有字段以及字段类型，最后封装连接对象，返回给调用端；若是不满足则抛出url不合法的异常。

判断URL格式是否合法

调用端发送获取连接的请求

抛出异常

获取主机、端口、索引名

建立到elasticsearch的连接

获取该索引下所有字段的类型

返回连接对象

## 5.2 控制模块

（1）固定sql控制模块

该模块实现JDBC的Statement接口，该接口的executeQuery方法和executeUpdate分别处理查询sql和更新sql；若是执行查询sql，则直接调用查询模块，获取查询结果，返回给调用端；若是执行更新sql，则进一步判断是哪种类型的更新操作，再调用对应的模块执行相应操作，若所有类型的更新操作都不匹配，则抛出sql不合法的异常。

否

是

调用固定sql控制器

执行查询

查询

返回查询结果

Sql以create开头

Sql以update开头

Sql以insert开头

Sql以delete开头

Sql以alter开头

创建表

更新表数据

插入新数据

删除数据

修改表配置

抛出异常

是

是

是

是

是

否

否

否

否

否

（2）动态sql控制模块

该模块实现JDBC的PreparedStatement接口；首先解析sql语句，将sql语句中的占位符分解出来，然后根据调用端设置的每个占位符代表的参数拼接sql语句，这样就将动态sql

转化为固定sql，最后调用固定sql控制模块来执行该sql语句。

调用动态sql控制器

解析动态sql语句

拼接sql语句

执行固定sql

## 5.3 查询模块

查询模块的处理过程包括：sql解析阶段、查询体构建阶段、结果集构建阶段

1. sql解析阶段

否

否

否

否

否

是

是

是

是

是

否

是

否

是

输入查询sql

设置distinct字段值、表名、分组字段、排序字段、分页信息

查询所有字段

查询特定普通列

列名为\*，聚合类型为none

对应的列名，聚合类型为none

对应列名以及聚合类型

设置列对象的别名

Where为比较表达式

Where 为null表达式

Where为 In表达式

Where为not表达式

Where为区间表达式

构建Inequality对象

构建NullExpression对象

构建InExpression对象

构建ConNotExpression对象

构建BetweenExpression对象

获取where条件左部

获取where条件右部

返回查询对象

1. 查询构建阶段
2. 结果集构建阶段

## 5.4 添加模块

## 5.5 删除模块

## 5.6 更新模块

## 5.7 修改模块

## 5.8 创建模块

## 5.9 ES客户端模块

1. 使用说明
2. 总结

## 7.1 难点分析

## 7.2 心得