**Hibernate使用Hibernate Criteria Extension（HCE）实现ORM查询简化**

KIsman：印度尼西亚雅加达比娜努沙塔拉信息大学博士，邮箱kismanhong@gmail.com

Sani M. Isa：印度尼西亚雅加达比娜努沙塔拉信息大学博士，邮箱sani.m.isa@binus.ac.id

摘要------软件开发时间是软件开发过程中的一个关键问题，hibernate作用于数据库操作层被广泛用于提高开发速度。本研究开发了Hibernate Criteria Extension（HCE）库文件，该库提供了代码简化和易于操作的API以及函数。HCE有利用点来查询关联的功能，使实体类中的映射通过自动检测来进行连接关联，该功能也可用于在数据库中有对数据限制的查询。HCE是基于hibernate配置的一个hibernate包装库，熟悉hibernate使用的开发人员可以轻松使用此库。

*关键词 - 数据库; java的;懒加载; HCE;查询;简化*

引言

当今，大多数软件应用程序都使用数据库来存储信息和数据，数据库和数据库系统是现代社会生活中不可缺少的组成部分，我们大多数人每天都会与数据库进行交互[1]。传统的数据库应用程序以文本或数字类型存储和访问信息，数据库也有其他用途，例如存储和分析地图的地理信息系统（GIS），联机分析处理（OLAP），从非常大型的数据库中提取和分析有用的信息以支持决策。

数据库是相关数据的集合，数据被记录下来并有一个隐含的意义。数据的集合通常称为数据库，被RDBMS管理[2]，本质上是一段时间内存在的信息的集合。

本文深入讨论应用程序和数据库之间的通信。软件工程师需要获取数据并将其放入数据库系统，这一过程称为设计存储。与数据库数据交互的每个操作都将通过查询进行交互，软件工程师指定的查询只能用查询指定的数据库，每个数据库供应商都有其结构化查询语言（SQL）。随着软件技术和大多数编程语言朝着面向对象的发展，一些工程师或软件机构试图简化查询过程，他们尝试将应用程序中的对象绑定到数据库，这种方法也称为对象关系映射（ORM），ORM是从对象到关系数据的翻译机制，反之亦然。ORM“方言”是为指定的SQL供应商生成SQL查询而定义的，通过使用方言，应用程序不会受到SQL特定数据库供应商的限制，程序开发人员可以选择任何数据库进行最小更改，以此可用于任何数据库供应商，这是设计的应用程序和数据库通信问题之一。

为了模拟ORM的工作原理并简化ORM的使用，本研究使用Java和Hibernate ORM框架，基于Hibernate开发了一个Hibernate包装库（称为Hibernate Criteria Extension，HCE），简称为库。作者将开发代码作为Java库（JAR），代码库是使用hibernate库及其依赖关系的库进行开发的，HCE包含Java API函数，熟悉hibernate的Java软件工程师可以轻松理解开发HCE的目的，可以简化用hibernate发起的查询。 Hibernate选择HCE核心的原因是Hibernate是一种流行的Java ORM框架，具有ORM没有的其他标准的API，当然也有一些其他Java ORM框架，包括DataNucleus、Avaje Ebean、MyBatis、Apache Cayene等。

Hibernate Criteria Extension（HCE）是在Hibernate之上的Java库。HCE使用标准API来简化hibernate的功能，可以自动检测Java对象类和普通Java对象（PO​​JO）与hibernate配置中的关系。我们只需要提及这个字段（引用列）和哪个类（引用表）就可以查询，对于关系/组合使用，我们只需要使用“.”声明字段名称。我们还可以轻松通过默认返回结果构建where子句声明，也可以通过基于传递参数的代码传递对象。使用这些HCE的这些功能使我们节省软件开发时间，减少代码量并降低软件开发成本，开发者可以让用户按时使用应用程序，完整的解释将在Hibernate Criteria Extension（HCE）部分讨论。

II 对象 - 关系数据库管理系统

目前大多数数据库系统都是基于在1970年由E.F.Codd科学出版物提出的关系模型，该关系模型旨在为数据独立性提供基础。关系模型基于关系和表格，虽然它们的性质不同，但是高度相关[3]。

根据关系模型的概念，所有数据都可以用数学表示为一组n元组的n矩阵关系，所谓元组就是一组无序的属性值，属性是一组有序的属性名称和类型名称的统称。关系模型允许通过限制，投影，笛卡尔积和关系连接的关系操作来操作数据，关系模型允许您通过定义来实现对数据的结构定义和约束，保证数据完整性。（Bauer，C.＆King，G.2007）[4]

我们假设有两个组D1，D2（D是域）在数学视图中，D1和D2的笛卡尔积是有序对集合（e1，e2），其中e1是D1的元素，e2是D2的元素。假如给定集合J = {1,2,3}和K = {c，d}，JXK的笛卡尔乘积是组合对，其中第一个元素是J的成员，第二个元素是K.组合，那么集合将是：{（1，c），（1，d），（2，c），（2，d），（3，c），（3，d）}，这个例子可以得出D1和D2是D1 X D2的子集。[3]

数据库管理系统（DBMS）用于帮助企业管理其运营的数据[5]，是数据，硬件，软件和用户的集合。

DBMS被改进为关系数据库管理系统（RDBMS），RDBMS是一种功能强大的数据库技术，用于存储和操作事务数据。当多个用户同时访问的时候，数据库需要保持一致性和数据安全性能良好，对强大功能的RDBMS也施加了限制，例如：

•复杂数据处理：RDBMS中的数据驻留在多个表中，这些表通过关联的键值相互关联。 RDBMS不会强制工程师施加高耦合的数据结构，缺乏经验的工程师可能会设计出不必要的数据复杂性或限制未来发展的系统。

破解密钥和记录：关系数据库需要共享外键来链接分布在多个表中的信息。例如，客户表中可以包括客户人口统计信息，并在表内具唯一索引号的记录。如果链接键的数据类型不同，则数据库无法完成链接，且无需设计表的开发人员进行额外的返工。同样，如果一个表没有唯一的键，数据库可能会返回不准确的结果。如果访问数据库的应用程序未编码以在编辑期间锁定记录，则用户可能会无意中损坏数据，导致记录损坏。

对象关系数据库管理系统（ORDBMS）是对关系数据库管理系统的扩展，它增加了面向对象的特性并且可以直接表示关系到数据库中的应用对象，[6]当传统SQL、DBMS无法处理复杂数据时，ORDBMS是理想的选择。

在Java应用程序中，与数据库相关函数的连接称为Java数据库连接（JDBC）。由于数据库只能解释SQL语句，不能解释Java语言语句，因此需要JDBC用于将Java语句转换为SQL语句。

III 对象关系映射

面向对象编程（OOP）是当今软件开发主流模式的一个初步开始[7]。对象关系映射是一种用来弥补面向对象程序和关系数据库之间阻抗不匹配的技术，用来消除数据的重复和维护成本高以及与其相关的错误的易感性[6]。阻抗不匹配是关系数据库中持久化对象在对象模型和关系模型之间以及对象编程语言和关系查询语言之间的一个众所周知的问题[6]，是对象和数据库模式设计之间的一致性问题，可能造成对象声明更复杂，例如，一个表及其关系由多个对象表示，应用程序将更加复杂，难以维护，使用一种或多种映射策略来处理不匹配，典型地以模式体现。策略涉及关系数据库模式与面向对象程序之间的对应关系，这种策略被用于Hibernate和TopLink等绘图工具，解决了对象 - 关系阻抗不匹配的问题。[8]

一个对象关系型应用程序是来自对象和关系范例的产物。本质上，对象关系型应用程序是使用面向对象语言编写的程序使用关系数据库进行存储和检索的应用程序。程序员必须解决对象关系应用程序生产期间的对象关系阻抗不匹配（“阻抗不匹配”）问题[9]。为了更好地理解ORM以及为什么许多软件在开发中使用orm的解决方案，我们将讨论JDBC缺点和Java ORM流行工具hibernate的缺点。

3.1 JDBC的缺点

对象关系映射（ORM）旨在覆盖JDBC的缺点，有三个主要原因对于许多应用程序来说导致直接使用JDBC不是一个好选择：

1.开发和维护SQL非常困难且耗时

2. SQL缺乏可移植性

3.编写JDBC代码非常耗时，且容易出错

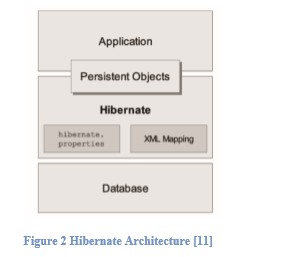
3.2. Hibernate

Hibernate是用于Java环境的对象/关系映射工具。对象关系映射（ORM）指的是利用基于SQL的模式，将数据从对象模型映射到关系数据模型的技术。Hibernate不仅负责从Java类到数据库表（从Java数据类型到SQL数据类型）的映射，还提供了数据查询和检索功能，否则将在SQL和JDBC中手动处理数据 [11]。

Hibernate是一个流行的Java ORM框架，几乎所有的Java程序员都使用它。 Hibernate尝试从95％的通用数据持久性相关编程任务中解放开发人员， 通过使用它，开发人员可以轻松维护对象类和表之间的关系，Hibernate试图删除供应商指定的SQL代码，并帮助您完成从数据表到对象图形的结果集转换的任务。

hibernate使用的特点是：hibernate需要配置映射对象到表和关系。Hibernate已经集成到许多流行的Java框架中。

hibernate的体系结构可以描述如下：



hibernate的强大功能也造成了限制，例如：（开发HCE的原因）

•在很多表关系中，让hibernate选择没有投影的数据会导致不必要的连接，会影响hibernate的性能，这就是所谓的n + 1休眠问题。

•投影有关联（例如：“classB.name”）会导致异常，Hibernate无法将“name”的值返回到基类中的属性“classB”。 （需要获得类）

投影包括hibernate中的关联仍然有复杂性，我们需要为每个连接关联创建标准或别名。

IV HIBERNATE可能改进为ORM

特别是hibernate，在持久性数据相关的编程任务中帮助我们很多，我们可以通过仅仅传递对象从数据库中获取数据将其转换为对象的集合来轻松插入数据。Java程序员可以节省应用程序开发时间，这样做程序员更乐意完成其他的编程任务。

尽管hibernate在持久性数据层中给我们带来了显着的好处，但是有可能使数据库长时间加载，本文比较关注Hibernate的改进，以下是hibernate的可能改进：

•更加简单的映射

  Hibernate的默认联合提取基于注释或XML配置，需要加入投影，调用“createAlias”或“createCriteria”方法。我们可以执行诸如“secondTable.name”之类的操作并且可以将返回值设置为Java对象类中的secondTable字段。

•更加轻松限制表的关系

当我们需要在第二个表中搜索/过滤字段时，hibernate需要像第一点那样进行连接。但是如果我们执行诸如“Restrictions.eq(“seondTable.name”, “[NAME]”)”，而不调用“createCriteria” 或者“createAlias”的方法，效率将很大程度上得到提高。

•更加简单的排序

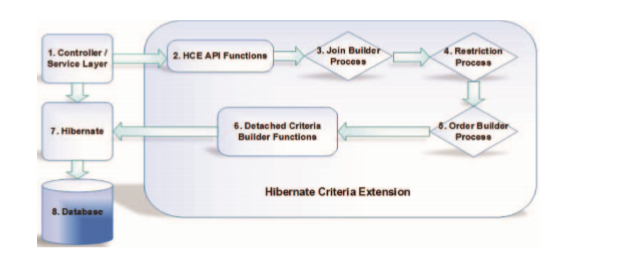
  相比之前的两点，如果我们可以执行诸如“Order.asc（”secondTable.name”）之类的操作来对数据库中的数据进行排序而不调用create criteria或alias方法，那将非常好。

` V：解决方案---利用HIBERNATE CRITERIA EXTENSION（HCE）

为了覆盖第4部分可能的改进，本研究开发了一个“hibernate包装库”作为解决方案，也叫Hibernate标准扩展（HCE）。

5.1 HCE结构

以下是HCE的结构



具体描述如下：

•应用程序控制器/服务层

（1）调用HCE API

（2）用来检索查询结果，我们也可以通过直接调用hibernate函数来传递此步骤

 （3）HCE将根据给定的预测来检测连接

（4）在连接构建器之后，HCE进行处理限制

（5）HCE还将处理给定的命令

 （6）返回分离的Criteria对象

（7）将分离的Criteria对象传递给hibernate，从数据库中检索数据并返回到控制器/服务调用方法

5.2 HCE 特点

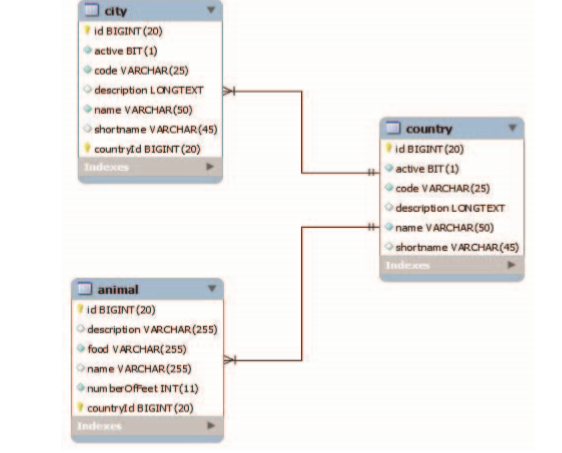
•通过传递字段名称,关联来声明hibernate投影（只选择所需的字段/列）。

•通过传递“表达式”来声明限制，表达式包含：propertyName，restrictionType，value等。

•通过传递订单来声明订单; 订单包含propertyName、orderType对关联有效。

5.3 HCE用例

HCE的用例我们将在这里进行讨论，假设有三个表：



ERD有三个数据表意味着我们需要创造三个hibernate的映射关系，animal数据表对country是多对一映射，country对city是多对一的映射关系。

HCE可以用来搜索基于hibernate映射关系的数据库，但目前只支持注解的映射关系。HCE可以通过在投影中使用点来检测连接，并将所选值设置为关联类的属性，如果设置没有预测的紧急模式，hibernate才可以使用投影。

HCE的用法如下：

|  |  |
| --- | --- |
| 说明 | HCE字符 |
| 根据字段从数据表中选取 | DetachedCriteria detachedCriteria = criteriaWithProjections(new String[] {"name"}); |
| 从数据表中选取很多列 | DetachedCriteria detachedCriteria = criteriaWithProjections(new String[]{ "id", "name", "food", "numberOfFeet", "description"}); |
| 添加数据表（有关联） | DetachedCriteria detachedCriteria = criteriaWithProjections(new String[]{"id", "name", "country.name"}); |
| 查询有限制的字段 | DetachedCriteria detachedCriteria = criteriaByProperty(new String[]{"id", "name", "description"}, Expression.eq("name", "Panda")); |
| 从数据表中选取多个字段并且一些字段有限制 | DetachedCriteria detachedCriteria = criteriaByProperty(new String[]{"id", "name", "description"}, Expression.eq("country.name", "China")); |
| 查询两个数据表并且其中一个有字段限制 | DetachedCriteria detachedCriteria = criteriaByProperty(new String[]{"id", "name", "description", "country.name” }, Expression.eq("country.name", "China")); |
| 类型限制 | DetachedCriteria detachedCriteria = criteriaByProperty(null, Expression.like("name", "Panda")); |
| 排序 | DetachedCriteria detachedCriteria = criteriaWithProjections(null, softtech.hong.hce.model.Order.asc("name") |
| 分页 | DetachedCriteria[] detachedCriterias = queryTransformer(null, projections, orders);  \*null mean that no filter is applied |
| 查询示例 | Animal animal = new Animal(); animal.setName("Panda"); DetachedCriteria detachedCriteria = queryTranslation(animal, null, null);  \*first null for projections, second one for ignore properties, in hibernate it’s called exclude properties. |
| 查询关联示例 | Animal animal = new Animal();  animal.setName( "Panda" ); Country country = new Country(); country.setName("China"); animal.setCountry(country);  DetachedCriteria detachedCriteria = queryTranslation(animal, null, null);  \*animal name and country name will be assigned as restrictions. |

相比hibernate，HCE的优点：

•使用hibernate标准的关联需要调用createCriteria或createAlias方法。HCE可以只通过使用点来检测投影中定义的关联，例如：animal.country.city.name，Hibernate需要在country和city表中创建创建Alias字段才能获取name的值。

在hibernate的critrria关联限制中，需要创建criteria或者alias方法。HCE可以通过点来实现，例如（“country.city.name”,”city.name”），通过点仍然可以实现排序。

VI. HCE 的限制

HCE的使用很大程度上覆盖了hibernate的方法，HCE不能像hibernate那样使用。下边是HCE的一些限制：

HCE仅支持hibernate注释配置，不支持xml配置，因为最后使用的是注释方式的配置。

•多对一，一对一目前不能很好的支持，建议当投影为空时，使用表达式。复杂的关联的例子：一个国家有很多城市，一个城市有很多地区。

VII 未来的研究方向

HCE通过传递Java对象的字段名称来支持投影，我们可以开发动态查询用于报告或其他系统，用户可以选择该字段并显示它的关系，HCE将自动检测表格的字段及其关系。

VIII. 结论

HCE 是对hibernate的库文件进行扩展，不是hibernate的一个静态库，我们可以使用criteria进行简单查询。正如第五部分的很多描述，理解HCE需要hibernate的很多知识，尽管HCE可以用于很多查询的例子，在很多例子中也需要hibernate或者JDBC的，例如：当查询的数据量很大的时候JDBC是很好的选择，目前HCE不支持大数据量的查询，当然也可以选hibernate。

参考：

[1] F. Of and S. Edition, Database. .

[2] U. T. Nacional, “Development of a Relational Database Management System .,” vol. 3, no. 2, pp. 33–37, 2003.

[3] P. Atzeni, Ceri, Paraboschi, and Torlone, “Database Systems.”

[4] A. Lipitsäinen, “ORM – Object Relational Mapping,” pp. 1– 30.

[5] D. Systems, “Introduction to Database Concepts.”

[6] E. Erhieyovwe, P. Oghenekaro, and N. Oluwole, “An Object Relational Mapping Technique for Java Framework,” vol. 2, no. 6, pp. 1–9, 2013.

[7] J. Barnes, “Object-relational mapping as a persistence mechanism for object-oriented applications,” p. 113, 2007.

[8] C. Ireland and D. Bowers, “Exposing the Mythௗ: ObjectRelational Impedance Mismatch is a Wicked Problem,” no. c, pp. 21–26, 2015.

[9] C. Ireland, D. Bowers, M. Newton, and K. Waugh, “Understanding object-relational mapping: A framework based approach,” vol. 2, no. 2, pp. 202–216, 2009.

[10] S. Lokhande, R. Patil, and A. Kadam, “Use of Hibernate in modern technologyௗ: Project Management,” vol. 2, pp. 222– 227, 2010.

[11] JBoss, “JBoss Enterprise Application Platform,” no. 12/2009, 2009.