ISSN: 2319-8753

International Journal of Innovative Research in Science, Engineering and Technology

*(An ISO 3297: 2007 Certified Organization)*

**Vol. 3, Issue 8, August 2014**

# Object Relational Mapping Tool for C#.NET Framework

Kuldeep Hule[1], Zakir Shaikh[2]

P.G. Student, Department of Computer Science & Engineering, Nagesh Karajagi Orchid College of Engg. & Technology Solapur University, Solapur, Maharashtra, India [1]

Assistant Professor, Department of Computer Science & Engineering, Nagesh Karajagi Orchid College of Engg. & Technology Solapur University, Solapur, Maharashtra, India [2]

**ABSTRACT:** Object relational mapping from relational databases has become topic to be dealt with in the recent decade. Object Relational Mapping is a layer which is responsible to bridge the gap between object-oriented programs and relational database. Lot of new Object Relational Mappers have been developed which includes forward engineering as well as reverse engineering approach for developing the object code. In this paper we are proposing the method of reverse engineering for developing object code file.

**KEYWORDS**: Object relational mapping, relational database, impedance mismatch, entity

## I. INTRODUCTION

Information Technology (IT) systems need persistent storage (unless, of course, the systems are really trivial). This persistent storage usually comes in the form of a relational database, such as Oracle, Microsoft Structured Query LANGAUGE (SQL) Server, IBM DB, and likewise. There are several alternatives to RDBMSs; for instance, object-oriented databases (OODBs) e.g. db4o, persistent object stores (Prevayler, XML databases etc.). However, at least until now, all these alternatives haven't really caught and the fact is that Relational Database Management Systems (RDBMSs) remain pretty much ubiquitous. Also nowadays, the object-oriented programming is already widely accepted approach for the development of business applications. Object-oriented development provides more suitable methods and facilities for modelling of the real world objects than e.g. the paradigm of structured programming.

Relational databases [10] and object-oriented programming languages are based upon distinct paradigms, with technical conceptual and cultural incompatibilities. The set of those incompatibilities is commonly referred to as the object-relational impedance mismatch problem [8]. Object-relational mapping (ORM) frameworks address the impedance problem of software implementation level [6], providing the developer with ways to declare how each technical incompatibility should be treated [6]. ORM tools brings to the same level relational resources, such as data querying and object-oriented resources, such as inheritance, polymorphism, enabling to explore synergy between those constructs [4].

Object relational mapping (ORM) in computer software is a programming technique for converting data between incompatible type systems in databases and object-oriented programming languages. This creates, in effect, "virtual object databases" which can be used from within the programming language [1].

In other words, Object Relational Mapping hides the underlying tables, SQL etc., inside the RDBMS. Due to the popularity of relational data storage with its broad installation base and the use of object-oriented concepts for software development, application objects inevitably need to be stored into (object) relational databases [7].

### 2.1. Impendence Mismatch

The impedance mismatch problem is a well-known problem in persistence objects in relational databases between both the object model and the relational model and between the object programming language and the relational query language [5]. Object oriented development is based on concept of data & their behaviors, while the relational databases

DOI: 10.15680/IJIRSET.2014.0308017

ISSN: 2319-8753

International Journal of Innovative Research in Science, Engineering and Technology

(An ISO 3297: 2007 Certified Organization)

Vol. 3, Issue 8, August 2014

are based on purely data. Thus there is always a problem of mismatch when trying to map or integrate these concepts together. This mismatch problem is what is referred to as impedance mismatch. Thus there is need to solve this problem in order to improve the interaction between object relational database technology.

### 2.2. Object Relational Mapping:

Object Relational Mapping is a layer that is responsible to bridge the gap between object-oriented programs and relational database. In fact which provides the objects with persistent services which means that ability to read data from and write data to, and delete data from data sources. Fig1 illustrates the concept of object relational mapping:

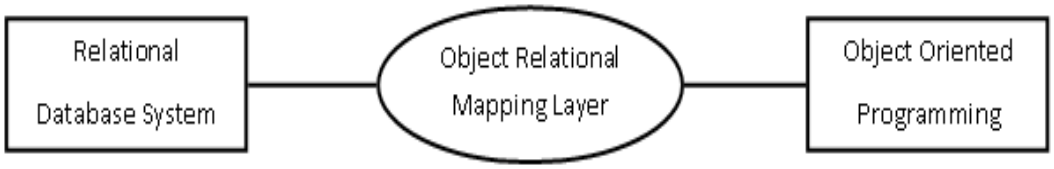


Fig. 1 Object Relational Mapping System

In other words, object relational mapping is a technology that is employed to bridge the impedance mismatch between object- oriented programs and relational database. It tries to eliminate the duplication of data and maintenance cost and susceptibility to error(s) associated with it [9]. There are a lot of problems to solve when implementing ORM framework.

## II. RELATED WORK

There is plenty of work done related with Object Relational Mapping (ORM). There may be various platforms and frameworks available for Object Relational mapping. Some of them are listed below:

### 2.1. LINQ-TO-SQL:

Language integrated query (LINQ) provides a uniform, object-oriented way to access data from heterogeneous sources, and thus simplifies the interaction between object-oriented programming and relational data. LINQ is available for programming language as part of the .NET framework 3.5, 4. LINQ maps the SQL database tables to Classes and objects and then allows searching for the information just like SQL. LINQ supports compile-time syntax checking, ability to access other data sources like XML and it's very strongly typed. The LINQ approach reduces complexity and allows you to design and debug your queries seamlessly.

### 2.2. NHibernate:

NHibernate is a port of the highly successful Hibernate ORM in Java. NHibernate is an open source (free) ORM tool that gives you a persistent, ignorant, transparent framework for persistence. NHibernateMapper tool is developed, which is able to generate NHibernate mapping file (.hbm file) based on the already available domain knowledge and existing database. These three components (application programming classes, database and mapping file) are sufficient premises for generation of complete Data Access Layer using NHibernate ORM mechanism [2].

### 2.3. Hibernate:

Hibernate is an Object-relational mapping (ORM) tool. Hibernate is a persistence framework which is used to persist data from Java environment to database. Persistence is a process of storing the data to some permanent medium and retrieving it back at any point of time even after the application that had created the data ended. Hibernate framework using the database and configuration data to provide persistence services (and persistent objects) to the application [3].
An application that uses Hibernate API must use a single configuration file. A Hibernate configuration file can be an XML based file (hibernate.cfg.xml) or it can be ordinary Java properties file (hibernate. properties) with key-value combination. This configuration file should be placed in the run-time classpath of an application [3].

### 2.4. LLBLGen Pro:-

LLBLGen Pro is an O/R mapper and code generator for the Microsoft .NET platform, created by Solutions Design bv. Computer programmers and software architects use this software to create a data-access tier and business objects tier in

DOI: 10.15680/IJIRSET.2014.0308017

ISSN: 2319-8753

International Journal of Innovative Research in Science, Engineering and Technology

*(An ISO 3297: 2007 Certified Organization)*

**Vol. 3, Issue 8, August 2014**

C# or VB.NET for several O/R mapping frameworks, like NHibernate, Entity Framework, LINQ to SQL, and its own framework. Because LLBLGen Pro supports a lot of different databases to work with, software developers can choose the best framework and database per project [11].

From above existing system following are some of the drawbacks existing system:-

1. Connection pool to database is increased.
2. Comparatively hard to fine tune generated SQL.
3. Can bloat your code because a lot of code is generated
4. It's limited when you work with a huge domain model.

## III. MATERIALS AND SYSTEM ARCHITECTURE

On the development station, we used a system with a Pentium Dual-core 2.70Ghz Processor, 2GB RAM, 32-bit System Architecture, and windows 8 Operating System. Object Relational Mapping has been implemented entirely in C# using Microsoft .NET Framework 4.0. It provides by default a wealth of frameworks, API's, standards implementations, and libraries that foster reusability and reduces development time.

The objective of any objet relational mapping framework/API is to relieve the programmers from the hard work of manual SQL statement writing and manipulation, and allow them to focus on manipulating data structure from the problem domain [1]. In this model we have aimed at providing optimum scalability and reusability.

Object-Relational Mapping, or O/R Mapping is the general term for the concept of creating mappings between tables and their fields, Object Oriented classes and their fields to be able to represent at runtime an Entity Definition Instance in the form of a table row in an Object Oriented program via a class instance and vice versa. The management logic is necessary to read an Entity Definition Instance (which is the data) from the database into an instance of an instance of a class and storing it back into the database, together with the entity definitions, is called an O/R Mapper framework [7]. In the Object Relational mapping designer, these are called Target Frameworks. ORM Designer will generate the code for ADO.NET Framework. Via this framework a developer can manipulate data in the database, using classes and their methods. We also had provided the collection, customized views of the database data. Fig 2 shows the system architecture:

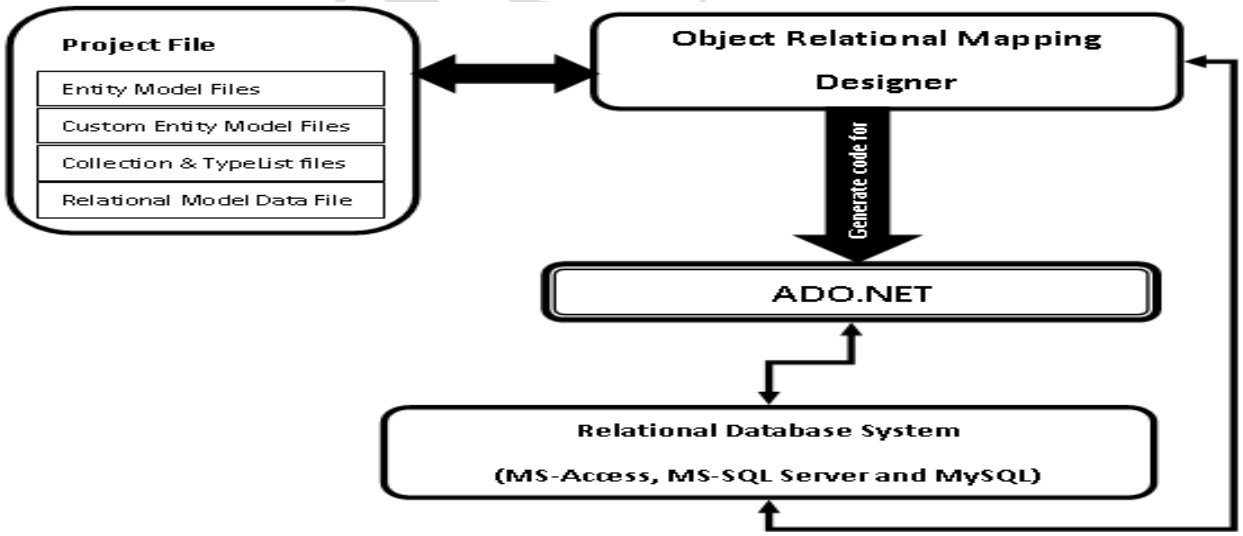


Fig. 2 System Architecture

The overall architecture for the system utilizes structures that represent tables, databases and connections to these databases. Database- First or Database-driven approach is used for this architecture which works as reverse engineered an Abstract Entity Model from a set of database schema elements.

DOI: 10.15680/IJIRSET.2014.0308017

ISSN: 2319-8753

International Journal of Innovative Research in Science, Engineering and Technology

*(An ISO 3297: 2007 Certified Organization)*

**Vol. 3, Issue 8, August 2014**

The idea behind Database-First is that the Database Schema Elements, e.g. tables, views, foreign key constraints and like, were once created using an abstract entity model and to look at the database schema elements, one can reverse engineer them to an Abstract Entity Model which is roughly the same, or represents the same, as the database schema elements are projection results of the Abstract Entity Model [11]. Fig 3 shows the schematic view of Database First model for the above architecture.

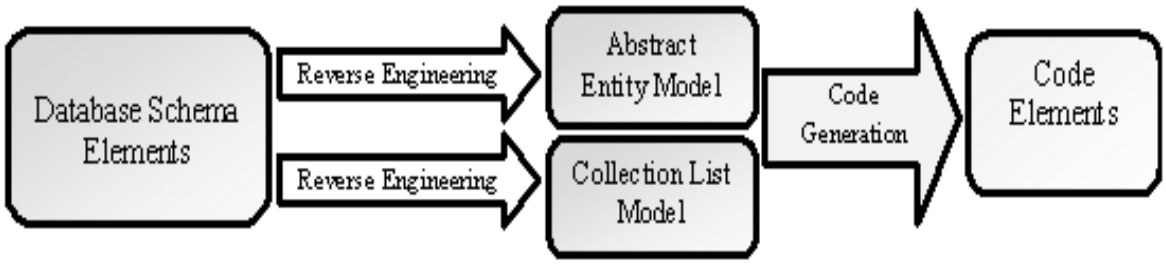


Fig.3 Database-First Approach

## IV. IMPLEMENTATION DETAILS

Object Relational Mapping has been implemented entirely in C# using Microsoft .NET Framework 4.0. As a result, C# Class Library project is created. All programming classes in this library project would be divided into 5 parts. These are Entity, CustomEntity, TypeList, and CollectionList & CustomJoinList. These models also provide the transaction facility.

Entity model which creates entities from tables which provides code for inserting data into database, modifying database data, deleting data from database & view the data from primary key field. An entity is a lightweight persistent domain object. Typically, an entity class represent a table in a relational database and each entity instance corresponds to a row in that table Fig.4 shows the entity creation Model:

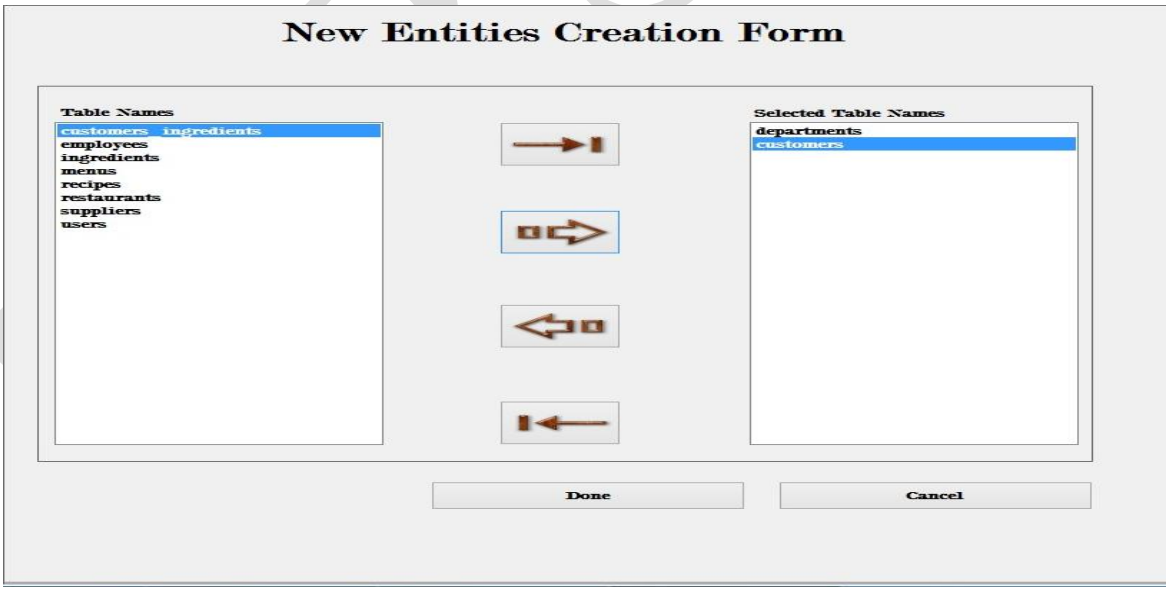


Fig. 4 Entity Creation Model

Custom Entity model creates entity from table which create entity from user choice data structure. In this model you can create customized format of Create, Read, and Update& Delete (CRUD) type of operations onto database. Fig 5 shows the Custom Entity model.

DOI: 10.15680/IJIRSET.2014.0308017

ISSN: 2319-8753

International Journal of Innovative Research in Science, Engineering and Technology

*(An ISO 3297: 2007 Certified Organization)*

**Vol. 3, Issue 8, August 2014**

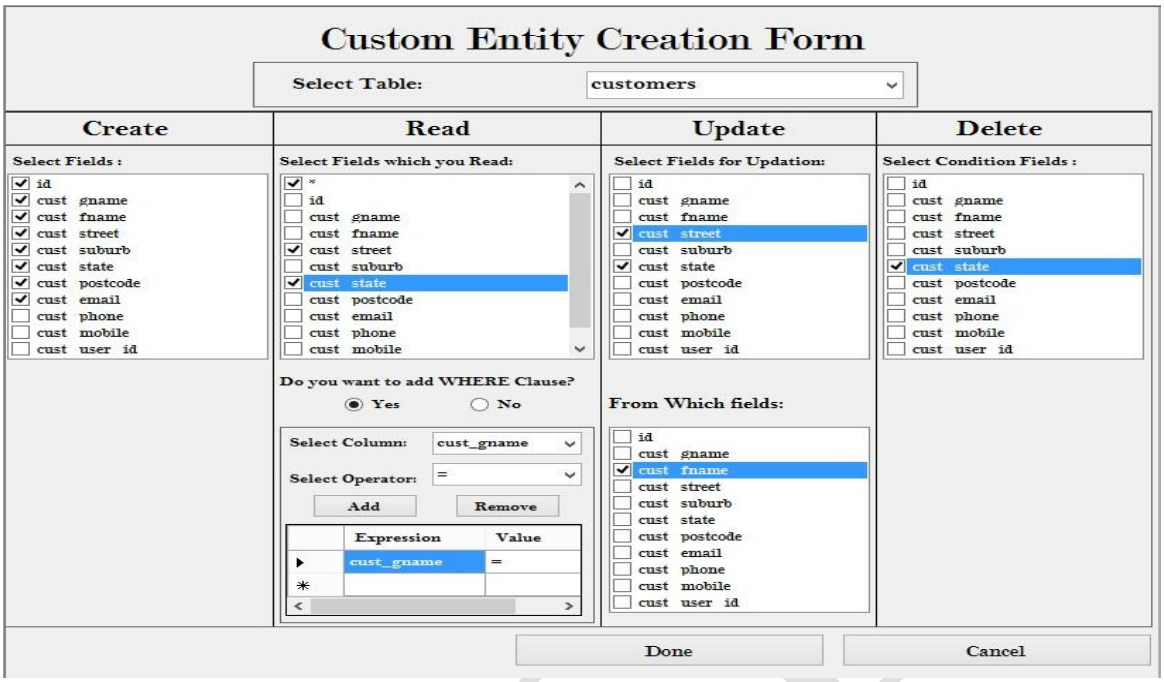


Fig. 5 CustomEntity Creation Model

TypeList model creates the select query which provides users requirement labels for field name. Fig. 6 shows the TypeList model.

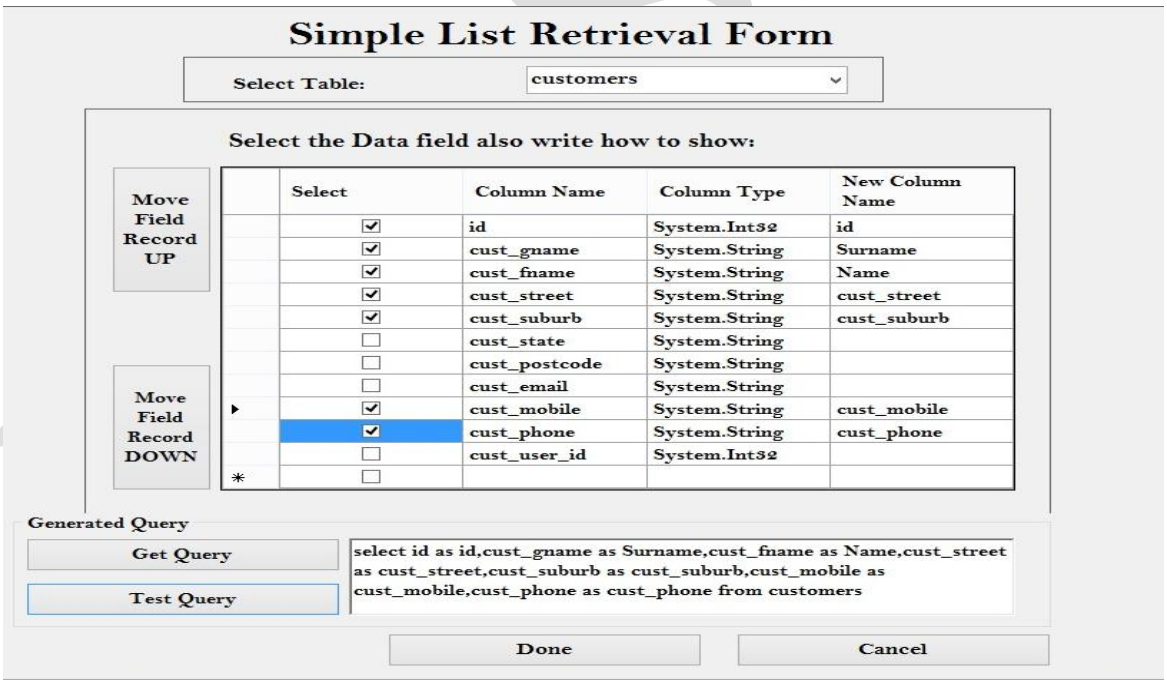


Fig. 6 TypeList Creation Model

CollectionList model creates the 2 types Database data view queries and these are Group By and Order By (sorting). Group by query uses the group by clause. The group by field contains the repetitive records or values like department number in employee. Order by model creates the query to view the data in sorted order by using order by clause. Fig 7 shows the CollectionList Model for Order by View.

ISSN: 2319-8753

International Journal of Innovative Research in Science, Engineering and Technology

*(An ISO 3297: 2007 Certified Organization)*

**Vol. 3, Issue 8, August 2014**

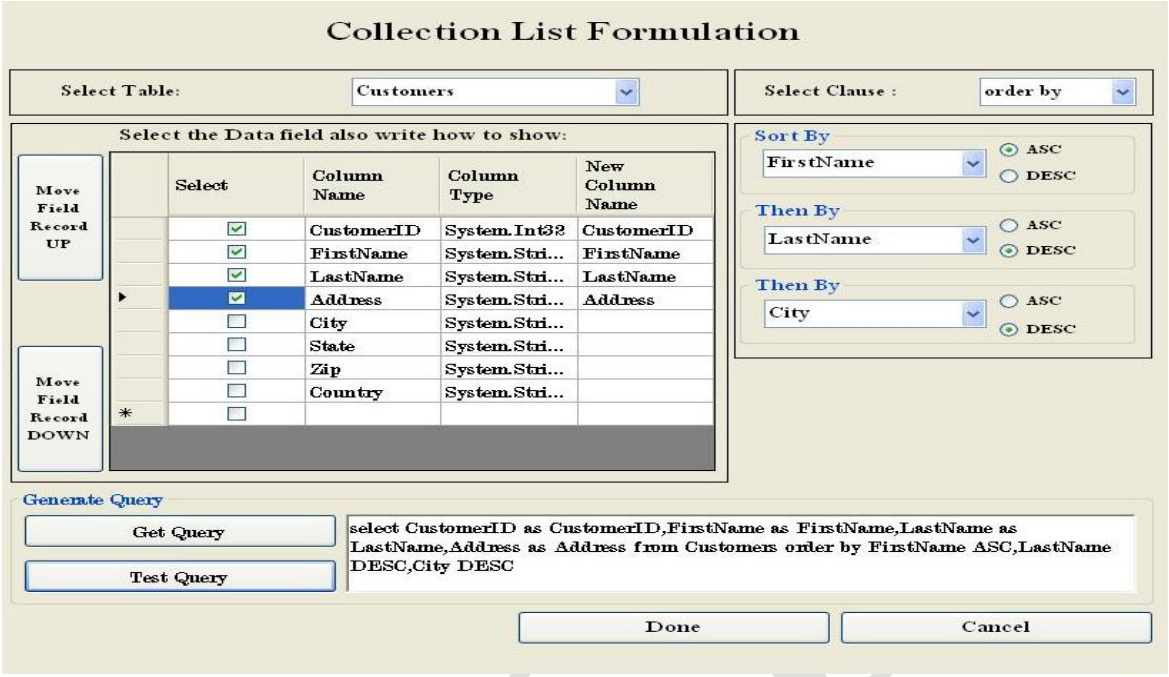


Fig.7 CollectionList Model for Order By

To retrieve the data from multiple tables on the basis of common field in those tables is possible by using Join query. Our CustomJoinList model creates the join query as per user selection criteria. Fig 8 shows the CustomJoinList model.

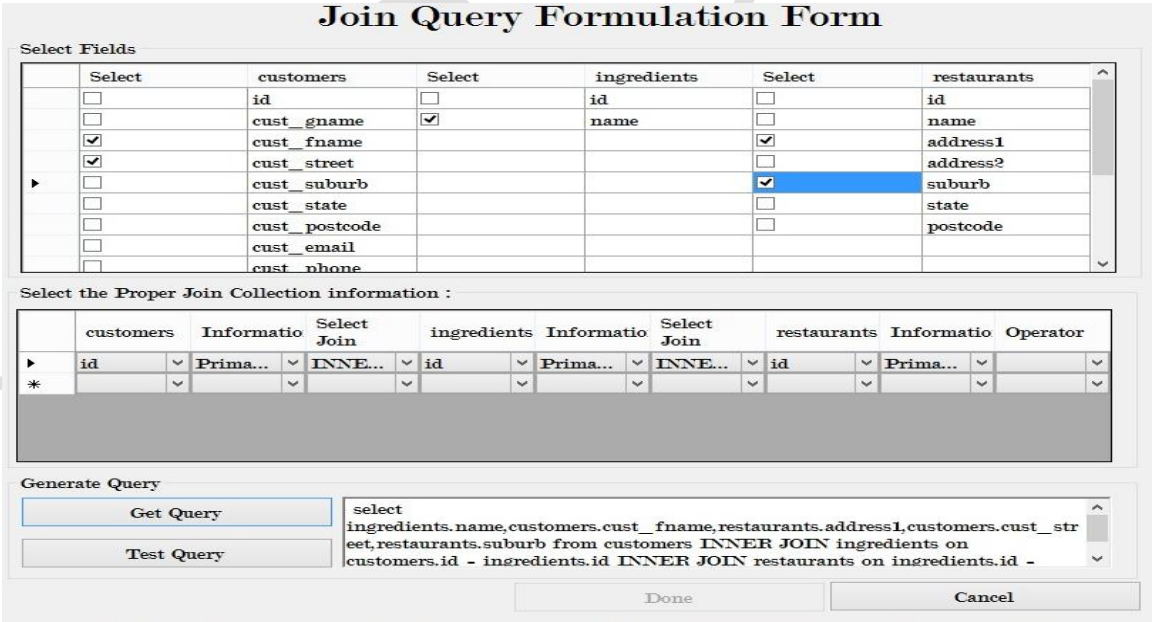


Fig.8 CustomJoinList Model

At last we have proposed some of the major advantages of our implemented ORM Framework which are:

1. It reduces the coupling between object schema and data schema increasing ability to evolve either one;
2. It implements all data-related code in one place;
3. It allows application programmers to focus on the business problems;

DOI: 10.15680/IJIRSET.2014.0308017

ISSN: 2319-8753

International Journal of Innovative Research in Science, Engineering and Technology

*(An ISO 3297: 2007 Certified Organization)*

**Vol. 3, Issue 8, August 2014**

## V. CONCLUSION

C# programming language is a modern platform that enables quick and efficient development of enterprise information systems. We have implemented the object relational mapping tool for C#.NET framework. This tool has been created by focusing on code reduction through providing the rich set queries and simple ADO.NET code functionality. Through this work the problem of impendence mismatch has been solved.

## REFERENCES

[1] Ogheneovo, Edward Erhieyovwe, Asagba, Prince Oghenekaro, Ogini, Nicholas Oluwole, "An Object Relational Mapping Technique for Java Framework"- International Journal of Engineering Science Invention, Vol. 2 Issue 6, pp.01-09, 2013.
[2] Stevica S. Cvetkovic, Milos D. Bogdanovic and Leonid V. Stoimenov, "NHibernateMapper - A Tool for Rapid Development of Data Access Layer for Interoperable GIS Solutions", magazine of ELECTRONICS, VOL. 15, NO. 1, pp. 62-66, 2011
[3] Sharayu Lokhande, Rushali Patil, Anup Kadam, "Use of Hibernate in modern technology: Project Management", International Journal of Computer Communication and Information System (IJCCIS), Vol. 2, pp. 222-227, 2010.
[4] L. DeMichiel and M. Keith, "Enterprise JavaBeansTM, version 3.0. Java Persistence API",JSR 220, 2006
[5] P. V. Zyl, D. G. Kuorie, and A. Boake," compare the performance of object databases and ORM tools", In Proceedings of SAICSIT, pp. 111-113, 2006.
[6] C. Bauer and G. King, "Hibernate in action", manning publishers Co. Boston/Massachusetts, pp. 1-25, 2005.
[7] S. Philippi, Model driven generation and testing of object-relational mappings, Journal of Systems and Software, pp. 193-207, 2005.
[8] S. Amber, "Agile database techniques: effective strategies for the agile software developer", 2003.
[9] A. Keller and C. Keene, "Persistence software: bridging object-Oriented programming and relational databases", In Proceedings of the ACM SIGMOD Intel Conference on Management of Data, pp. 540-541, 1993.
[10] Codd, E. F., A Relational Model of Data for large Shared Data Banks. Communications of the ACM, Vol.13, No.6, pp. 377-387, 1970.
[11] LLBLGen Pro team, "LLBLGen Pro document", http://www.llblgen.com/pages/overview.aspx,2014

## BIOGRAPHY

**Mr. Kuldeep Anil Hule** received B.E. degree in Computer Science and Engineering in 2012 from SOLAPUR University, Solapur, Maharashtra, India and pursuing the M.E. degree in Computer Science and Engineering in Nagesh Karajagi Orchid College of Engg. & Technology, Solapur, India. He is doing his dissertation work under the guidance of Mr. Shaikh Z. M., Assistant Professor at Nagesh Karajagi Orchid College of Engg. & Technology, Solapur, Maharashtra, India.

**Mr. Zakir Mujeeb Shaikh** received B.E. degree in Computer Science and Engineering in 2000 from Dr. Baba Saheb Ambedkar University, Aurangabad, Maharashtra, India and ME Computer from BVU Pune, Maharashtra, India in 2004. He is presently working as an Assistant Professor in Nagesh Karajagi Orchid College of Engineering and Technology, Solapur, Maharashtra, India. His research area includes Image Processing, Database and software framework.

IJIRSET