# Information Retrieval using Natural Language Interfaces

M. Sreenivasulu
Professor
Department of CSE
K.S.R.M. College of Engineering, Kadapa, Andrhra Pradesh (State) India

## ABSTRACT
In Database Management System (DBMS) is collection of interrelated data and set of programs to access and modify the data. In Relational DBMS (RDBMS), data is organized inform of tables. In order to retrieve information from the database, the end user requires the knowledge of database query language such as Structured Query Language (SQL). However, not every user is able to write SQL queries as they may not aware of structure of the database and they needs to learn SQL. So, there is a need for non-expert user to query relational databases in their natural language such as English.

In this paper, we propose s Natural language Interfaces to Relational Databases (NLIRDB) to access the data in easier manner. The end user can submit the request in natural language. The query in natural language is then translated into equivalent SQL statement and finally SQL statement will be executed to extract desired information from the database. The proposed method allows non-expert user to access information from database.

## Keywords
Database Query Generator, Morphological Analyzer, NLIRDB, Query Executor, SQL.

## 1. INTRODUCTION
Database systems are designed to manage large bodies of information. A Data Base Management System is designed to provide enterprise with centralize control of data[6][7]. Having centralized control of data has further advantages over conventional file system are as follows:

- Redundancy can be reduced.
- Inconsistency can be avoided.
- Data can be shared.
- Security restrictions can be applied.
- Integrity can be maintained.

While one important task of DBMS is to store voluminous interrelated data of some organization, another important task it has to perform is to provide a convenient and efficient way of retrieving data. Database system provides Data Manipulation Language (DML) that enables end users to access and manipulate data.

Database retrieval is the most commonly performed function on database in any organization. The end users of organizations almost concentrate on only this particular operation on the database while other intricacies of how to store efficiently is taken care of designers of the database. Unlike the designers of the system, the users are people having little or no knowledge about computers. The often expect to retrieve and manipulate the voluminous information through a few, simple and precise commands. Hence a good external interface is necessary for accessing data in the database. In this paper, we proposed a natural language interface to Relational Database Management Systems for accessing and manipulating data from database.

This paper is organized as follows. The different user interfaces to RDBMS and NLIRDB were discussed in section 2. Brief description of NLIRDB is presented in section 3 and its implementation is detailed in section 4. The results of NLIRDB were discussed in section 5. Final section presents the conclusion.

## 2. DIFFERENT USER INTERFACES TO RDBMS
A good interface is necessary for accessing and manipulating data in the database. In this paper, we considered three different user interfaces to RDBMS. They are: language based, form based and natural language interface.

### 2.1 A language based interfaces
This interface is primitive yet commonly used one in the database world. These languages are easy to learn. The users submits requests for the desired information from database in a language called "Query language".

Query languages can be categorized as being either procedural or non procedural. In procedural query languages, the users instructs the system to perform a sequence of operations on the database to compute the desired result.

Ex. Structured Query Language (SQL).

In non procedural query languages, the user describes the information desired without giving specific procedure for obtaining that information.

Ex. Query Language (QUEL), Query By Example (QBE).

### 2.2 A Form based Interfaces
One of the most popular interfaces for end user computing, particularly as front end to database, is the form based interface, where users navigate through a path of menu options to arrive at a particular form that can perform certain information based on the commands given by the user. Such interface facilities predetermined and repetitive tasks. Unanticipated or ad-hoc queries for which there are no predefined forms can not be processed. However, a large percentage of user queries are highly repetitive for most database applications.

### 2.3 Natural language interfaces
The natural language interface to a database system is designed to be the most comfortable and user friendly interface for the end user of a database system[1][2][3]. The user will enter a query in natural language (i.e. in English) to access or manipulate data from database. It will offer advantage of least training on part of user for retrieving information from database.

# 3. NATURAL LANGUAGE INTERFACE TO RELATIONAL DATABASE

Natural language interface to database enables an end user to retrieve information and data using queries/commands framed in natural language.

Natural language interfaces to a database system can be visualized as interconnection of two independent components:

- Database Query Generator
- Query Processor

## 3.1 Database Query Generator

The database query generator is responsible for translating a query/command in a natural language (i.e., English) to an intermediate or interlingual form. The intermediate form is normally a database query language as standard. The most commonly used SQL query has been used for this purpose. The intermediate SQL query is then fed to Query Processor to produce the required information from the database as output.

## 3.2 Query Processor

The Query Processor accepts the query in database query language such as SQL, and process them, printing the required information from the database.

# 4. IMPLEMENTATION

The various modules involved in development of natural language interface to relational database as shown in fig 1 and their functionalities are described in following subsections.

## 4.1 Morphological Analyzer:

Morphological Analyzer is a software component capable of detecting morphemes in a piece of text. English text is commonly pre-processed before it is indexed[4]. This morphological phase of Database Query Generator uses the dictionary to extract words from input query and to find the grammatical category of each.

The dictionary is defined as below:

Verb {"show","list","display","get","find"}

Article {"a","an","the"}

Determiner {"full","entire","all"}

Preposition {"of","for","in","to","with"}

Conjuction {"and","than"}

Participle {"being","currently"}

Operator {"more than","greater than","less than","equal to"}

Auxiliary verb {"is","are","where"}

## 4.2 Parser

A natural language parser is a program that works out the grammatical structure of sentences, for instance, which groups of words go together (as "phrases") and which words are the subject or object of a verb. Probabilistic parsers use knowledge of language gained from hand-parsed sentences to try to produce the most likely analysis of new sentences. These statistical parsers still make some mistakes, but commonly work rather well. So, the parser will check whether the given sentence is syntactically valid or not[5].

The parser uses grammar inorder to check the syntax of the given query which is mentioned as follows:

S→ VP PP

VP → <verb> NP1

PP → <prep> NP2

NP1→ <determiner | article> | <article | determiner> | € & AQP

AQP → <relation | attribute> CP | €

CP → <conjunction> AQP

NP2 → <determiner | article> | <article | determiner> | € & RDP

RDP → <value | const.rel qualifier | € > & RLP

RLP → <relation>

## 4.3 SQL Query Generator:

The SQL query generator will map the parsed query in to a SQL query i.e., a database SQL statement.

## 4.4 Query Executor:

Finally the generated SQL statement is executed and the data is retrieved from the database in the form of results.

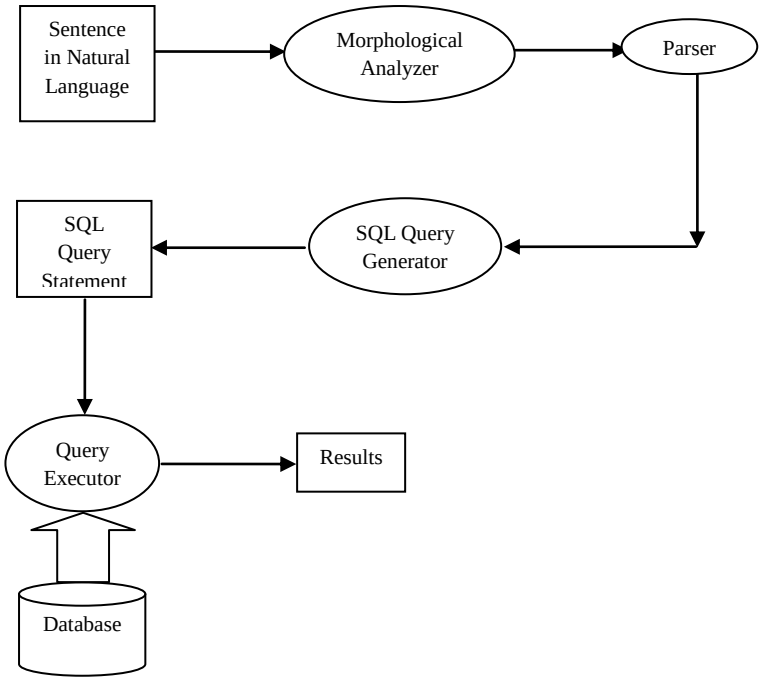


**Fig 1: Natural Language Interface To Relational Database**

## 5. RESULTS

In this paper, we considered the supplier-parts database for developing Natural Language Interface System[6][7]. The relation schemas for supplier-parts database are as follows:

Supplier (sid, sname, status, city)

Parts (pid, pname, color, weight, city)

Shipment (sid, pid, qty, city)

The NLIRDB have been implemented in java technology for platform independence[8]. The screens are designed using HTML[9], as shown in the fig 2. The validations are done using JavaScript[10], then the data is processed using JSP and sent to MYSQL database using the JDBC/ODBC connectivity[9].

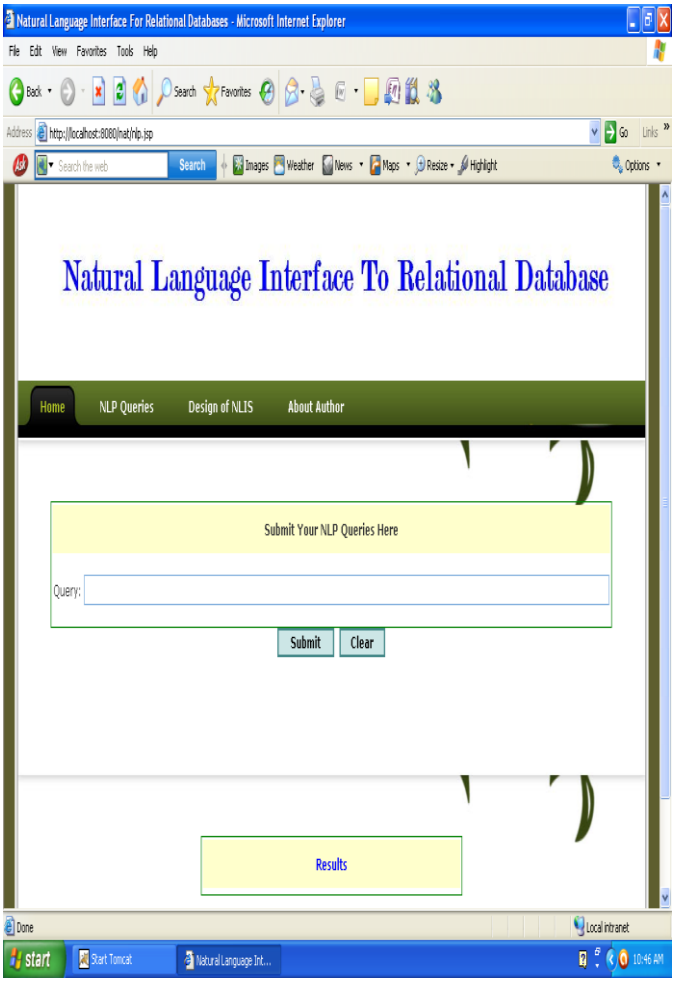


**Fig 2: NLIRDB**

The user enters a query as shown in fig 3.

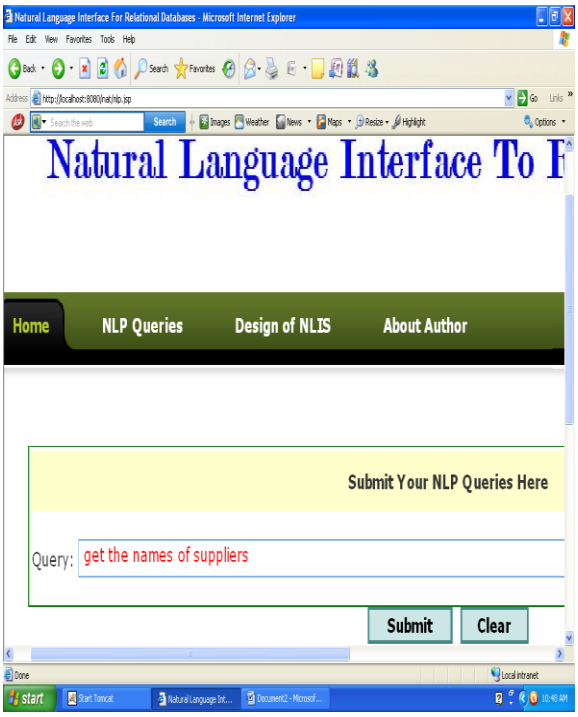


**Fig 3: User defined query**

The NLIRDB system translates the query into SQL query, execute it and display the results for entered natural language query as shown in fig 4.

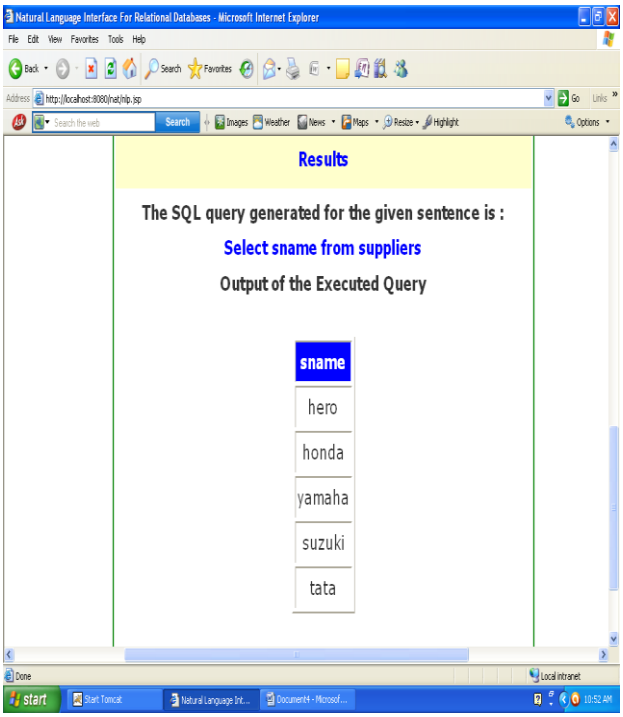


**Fig 4: Results**

## 5. CONCLUSION

Using natural language interface to RDBMS, the end user requests for desired information in natural language (i.e., in English) and can get the desired information. No training is required for retrieving the information. The NLIRDB has been tested successfully for a set of sentences. Future work is directed towards general formation of grammar which accepts a particular query in form.

## 6. REFERENCES

[1] [1]. Harjit Singh, "Providing Inferential Capability to Natural Language Database Interface", IJECSE, Vol.1, N0.3, pp. 1634 – 1639, 2012.

[2] [2]. Mrs. Neelu Nihalani, Dr. Sanjay Silakari, Dr. Mahesh Motwani, "Natural language Interface for Database: A Brief review", IJCSI International Journal of Computer Science Issues, Vol. 8, Issue 2, pp. 600-608, March 2011.

[3] [3]. Mrs. Neelu Nihalani, Dr. Sanjay Silakari, Dr. Mahesh Motwani, "An Intelligent Interface for relational databases", IJSSST, Vol. 11, No.1. pp 29-34, Jan 2010.

[4] [4]. Lappoon R. Tang, "Using machine learning approach for natural language interfaces for databases: Application of advanced techniques in inductive logic programming", Journal of Computer Science, Informatics and Electrical Engineering, vol 2, Issue 1, 2008.

[5] [5]. Alfred V. Aho, Monica S. Lam, Ravi Sethi, and Jeffrey D. Ullman, "Compilers: Principles, Techniques, and Tools", Pearson Education, 2006.

[6] [6] Silberschatz, Korth, "Data base System Concepts", McGraw hill, V edition, 2006.

[7] [7]. Raghurama Krishnan, Johannes Gehrke, "Data base Management Systems", TATA McGraw Hill, 3rd Edition, 2010.

[8] [8]. Herbert Schildt, "JAVA 2 The Complete Reference", Osborne/McGraw-Hill, 4th Edition, 2001.

[9] [9]. A.A.Puntambekar, "Web Technologies", Technical Publications, 2009.

[10] [10]. James Jawroski, "Mastering Java Script and JScript", Sybex, 1st edition, 1999.