



**INTERNATIONAL JOURNAL OF ENGINEERING SCIENCES & RESEARCH  
TECHNOLOGY**

**Research on Analysis of Hindi language Graphical user Interface**

**Ms. Nikita Bhati \*, Mr. Mahesh Singh**

\* Student, AITM., Palwal, India

[Nikita.0109@gmail.com](mailto:Nikita.0109@gmail.com)

---

**Abstract**

The interface between humans and computers is an ever critical issue due to the increased complexity of computerized systems and the wide variety of problems they solve. Controlled natural languages might prove a promising medium between humans and computers, however, they are not easy to design, and humans need time to adapt to them. Authors propose to solve these issues by using a controlled user interface which is powered by an automatically constructed application-oriented natural language. Therefore, any interface in Hindi language will be an asset to these people. This paper discusses the architecture of mapping the Hindi language query entered by the user into SQL query.

**Keywords:** Parser, Natural language, SQL, Database, Query Generator..

---

**Introduction**

We require information in our daily life. One of the major sources of information is database. Almost all applications have needed to retrieve information from database which requires knowledge of database languages like SQL. To write SQL query one need to have not only knowledge query language but also the physical structure of database. Therefore everybody is not able to write SQL queries. To override the complexity many researchers have turned out to use Natural Language (NL) i.e. Hindi, English, French, Tamil etc. instead of SQL. The idea of using NL has prompted the development of new type of processing method called Natural Language Interface to Database systems (NLIDB). It is a type of computer human interface. It is a system where user access information stored in databases by typing request in natural language. This does not require the user to know query language syntax. It makes accessing of data from database very easy for a person who has no knowledge of query language.

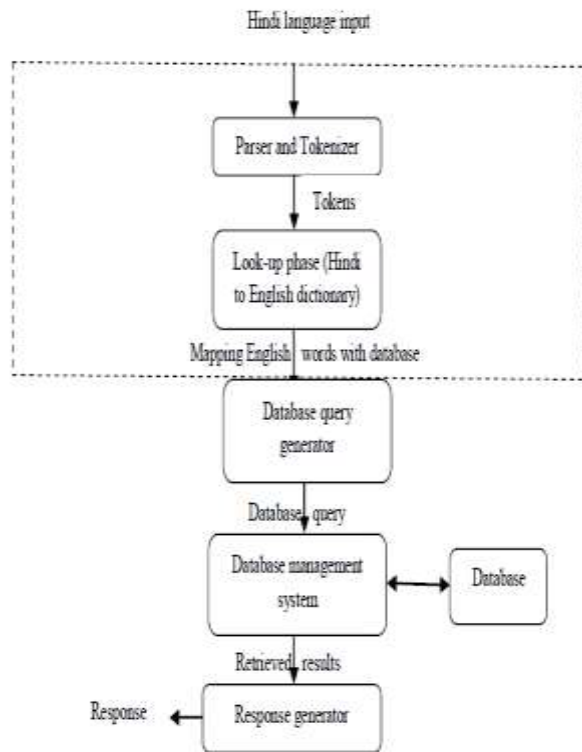
- To Parse and tokenize the input sentence using Hindi Shallow Parser.
- To find the English word corresponding to Hindi tokens from a dictionary.
- To map input Hindi query with database query.
- To generate SQL query with the help of database query generator and underlying DBMS.
- To execute the query and give the response to user.

**Materials and methods**

The design of Hindi language interface to databases includes the description of architecture of system and structure of EMPLOYEE database used in the system.

**Architecture of the system**

The architecture of Hindi language interface to databases is shown in Figure. Architecture of the system is divided five various phases. These phases are given below.



**Architecture of the proposed system**

The details of these phases are given below.

**Parsing and tokenizing input sentence using Hindi Shallow Parser**

In proposed system, Hindi Shallow Parser (developed by Language Technologies Research Centre (LTRC) at IIIT, Hyderabad) has been used to perform parsing of a sentence given in Hindi language. It involves morphological analysis, part of speech tagging, chunking, pruning, etc. The overall processing task has been broken up into many modules and each one of them performs a small logical task.

Hindi Shallow Parser uses Shakti Standard Format (SSF) for storing language analysis which has been described below.

**Shakti Standard Format**

It is a highly readable representation for storing language analysis. It is designed to be used as a common format or common representation on which all modules of a system operate. The Shakti Standard Format (SSF) representation is designed to keep both rule-based as well as statistical approaches together. Two kinds of analyses are usually done in SSF. These are constituent and relational structure level analysis. Constituent analysis is used to store simple phrase level analysis (called chunking) and the relational analysis for storing relations between the simple

phrases. SSF also involves one another concept of feature structures. Feature structures are used to store attribute-value pairs for phrasal node as well as for a word or a token. Outputs of many other kinds of analysis, such as grammatical relations, Tense, Aspect, Modality (TAM) computation, case computation, etc. , are stored using feature-structures. Though, the SSF format is fixed and it has a text representation, which makes it easy to read the output. The working of Hindi Shallow Parser has been explained below.

**Working of Hindi Shallow Parser**

The working of Hindi Shallow Parser has been discussed in this section with the help of an example. The Hindi sentence that is parsed using Hindi Shallow Parser is

सभी बच्चे किताब पढ़ते हैं।

Here, the sentence is parsed in following steps which is explained below.

**Tokenizer**

This module convert a sentence into word level tokens (consisting of words, punctuation marks, and other symbols) and return sentence marker for each sentence of input text.

<Sentence id="1">

1	सभी	unk
2	बच्चे	unk
3	किताब	unk
4	पढ़ते	unk
5	है	unk
6	.	unk

</Sentence>

**Output of tokenizer for input sentence**

A token is an instance of a sequence of characters in some particular document that are grouped together as a useful semantic unit for processing. The result of tokenizing the input sentence given

The Hindi sentence given is divided into number of tokens.

सभी, बच्चे, किताब, पढ़ते, है,

**Implementation**

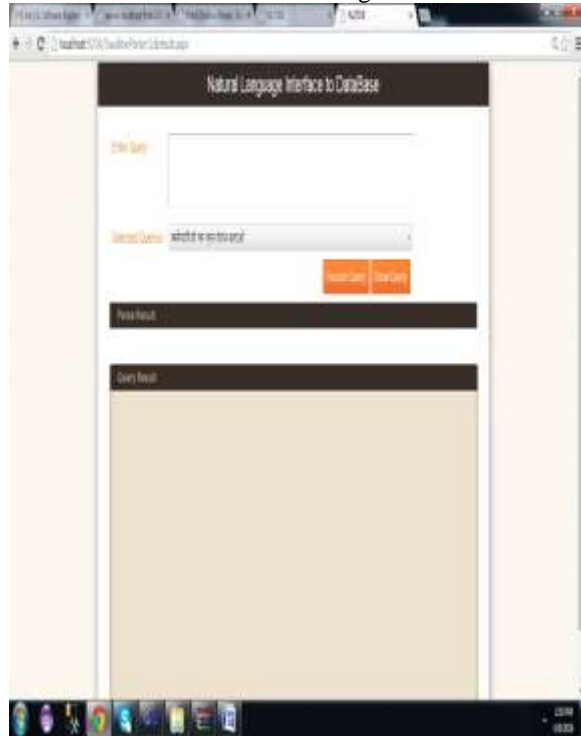
**Implementation and working of the system**

The proposed system has been developed for multiple platforms. The input Hindi sentence is parsed and tokenized using Hindi Shallow Parser. It uses

MYSQL as database and HTML and CSS for as front end. MYSQL is connected with ASP.net using mysqldata.dll. Unicode character set is used for providing input and output in Hindi. Data is also stored in Hindi in the database. The system handles three types of queries which have been illustrated below.

- Data Retrieval query.
- Update Column(s) query.
- Delete Row(s) query.

The user interface is shown in Figure.



**User interface of Natural language interface to databases**  
The interface includes a Combo Box which contains list of Hindi queries that user can ask from the system. The selected query is displayed in the text field provided above. User can also give his own query similar to type of queries in the combo box. The first Text Area contains the output of Hindi Shallow Parser while second text area is for query output. “Execute Query” button is used to execute the query and “Show SQL Query” button is used to display the corresponding SQL query generated.

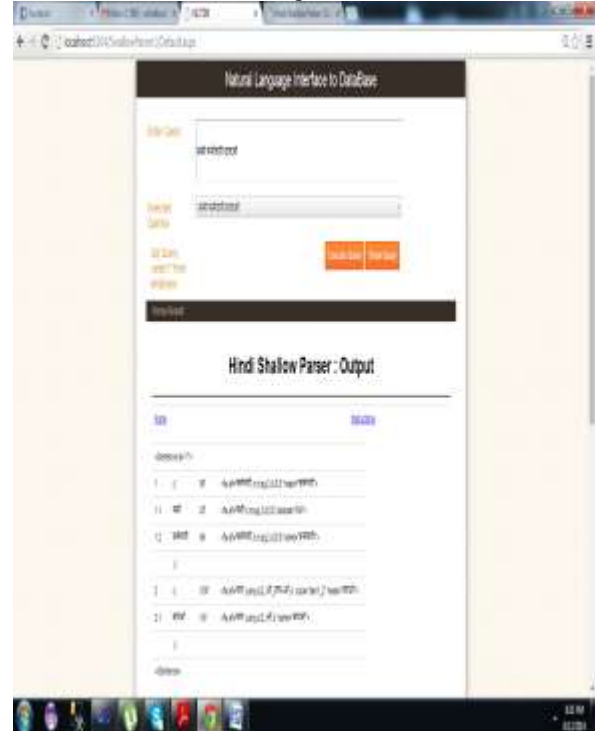
#### Data retrieval query

The proposed system can handle different form of queries which has been illustrated below.

- Query for selection of whole table.
- Query for selection of single or multiple columns.

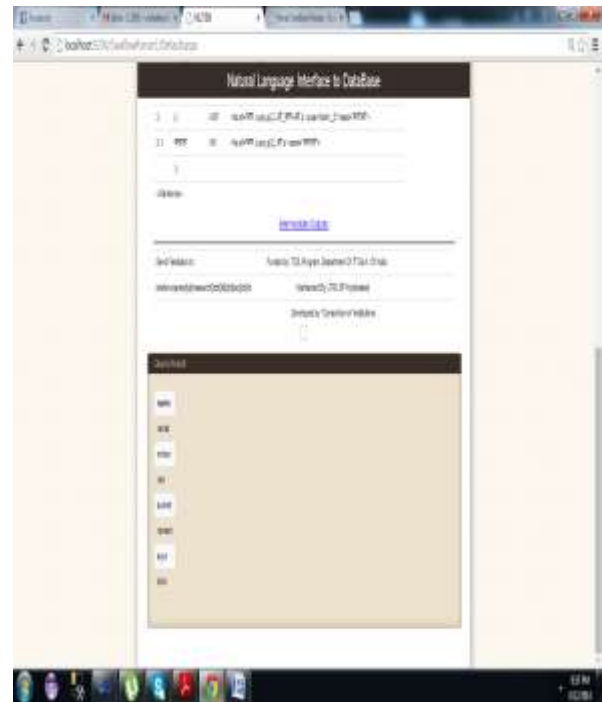
- Query for selection of certain rows from certain column, i.e., Conditional query.
- Query that doesn't include table name but table name is searched from column name.
- Query which includes joins of two tables.

The screenshot of query that involves selection of whole table is shown in Figure.





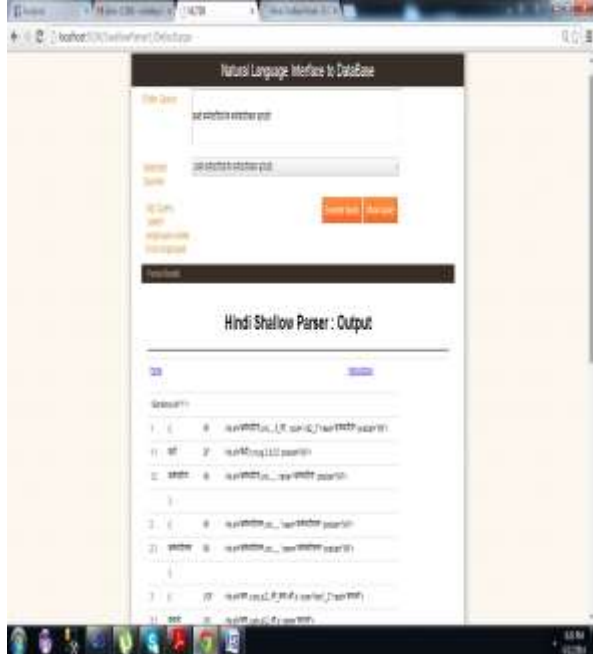
*Screenshot of query that involve selection of whole table*  
Figure, shows when “Execute Query” button is pressed, output of parser and final results are appeared in desired text areas. If user wants to see the SQL query generated, he can press “Show SQL Query” button. The SQL query generated is shown in `SELECT * FROM EMPLOYEE;`  
The screenshot of query that involves selection of single or multiple columns is shown in Figure.



*Screenshot of query that involves selection of single or multiple*

Figure, shows when “Execute Query” button is pressed, output of parser and final results are appeared in desired text areas. If user wants to see the SQL query generated, he can press “Show SQL Query” button. The SQL query generated is shown.

`SELECT NAME FROM EMPLOYEE;`  
The screenshot of query that involves condition is shown in Figure.



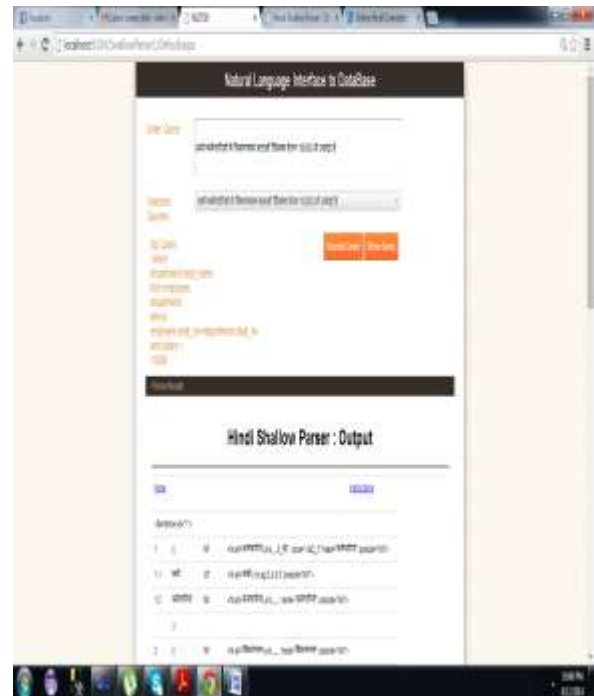


*Screenshot of query that involves Conditional*

Figure, shows the query which involves selection of employee number of that employee whose name is "दीपक". When "Execute Query" button is pressed, output of parser and final results are appeared in desired text areas. "Show SQL Query" button displays the SQL query generated. The SQL query generated is shown

`SELECT EMPNO FROM EMPLOYEE WHERE NAME= "दीपक";`

The screenshot of query that includes joining of two tables is shown in Figure.



*Screenshot of query that includes joining of two tables*

Figure, shows when "Execute Query" button is pressed, output of parser and final results are appeared in desired text areas. If user wants to see the SQL query generated, he can press "Show SQL Query" button. The SQL query generated is shown.

`SELECT DEPT_NAME FROM EMPLOYEE, DEPARTMENT WHERE EMPLOYEE.DEPTNO= DEPARTMENT.DEPTNO AND EMPLOYEE.NAME= "राम";`

**Update column (s) query**

The proposed system can handle different form of queries which has been illustrated below.

- Query that update single or multiple column(s).
- Query that update column (s) but doesn't include table name.
- Query that update column (s) by joining of two tables.



Screenshot of query that update column (s) but doesn't include

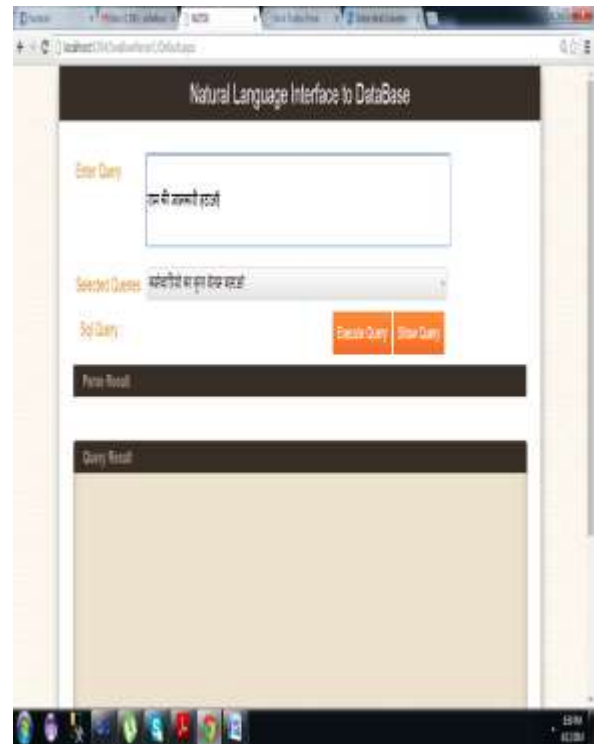
The screenshots of query that involves updating column (s) which do not include table name is shown in Figure: Screenshot of query that update column (s) but doesn't include table name in Figure, shows the input query is to change the city of employee „राम” to “जम्मू”. When “Show SQL Query” button is pressed, SQL query generated is shown  
 UPDATE EMPLOYEE SET CITY= “जम्मू” WHERE NAME= “राम”;

**Delete row (s) query**

The proposed system can handle different form of queries which has been explained below.

- Delete single or multiple rows.
- Delete row (s) when table name is not present in query.
- Delete row (s) by joining two tables.

A screenshot of query that involves deleting row(s) is shown in Figure.



Screenshot of query that involves deleting row(s)

Figure shows how delete query delete all the information about “राम”. The generated query is shown. DELETE FROM EMPLOYEE WHERE NAME= “राम”;

**Results and discussion**

The Natural Language interface to databases can handles three types of queries which has been illustrated below.

- Data Retrieval query.
- Update Column(s) query.
- Delete query.

The system has been tested on 50 Hindi input sentences. The testing results performed on data retrieval queries are shown in Table 6.1.

Table:

Table 1: Testing results performed on data retrieval queries

Hindi Sentence	SQL Query
सभी कर्मचारियों का नाम बताओ	Select Name from Employee;
उस कर्मचारी की कर्मचारीसंख्या बताओ जिसका नाम अशोक है	Select Empno from Employee where Name=“अशोक”;
उस कर्मचारी का शहर बताओ जिसका नाम राम है	Select City from Employee where Name= “राम”;

रीया की जन्मतिथि बता	Select DOB from Employee where Name = "रीया";
कर्मचारियों का वेतन दो	Select Salary from Employee;
उन कर्मचारियों का विभागनाम बताओ जिनका वेतन 50000 है	Select Dept_name from Employee, Department where Employee.Deptno= Department.Deptno and Employee.Salary= "50000";
कर्मचारियों के नाम , शहर बताओ	Select Name, City from Employee;
सिरसा के कर्मचारी का नाम बताओ	Select Name from Employee where City= "सिरसा";
उन कर्मचारी की कर्मचारीसंख्या , नाम बताओ जिनकी विभागसंख्या 7 है	Select Empno, Name from Employee, Department.Deptno and Employee.Deptno= "7";
विनोद का शहर बताओ	Select City from Employee where Name= "विनोद";
राम का विभागनाम बताओ	Department.Deptno and Employee.Name= „राम“;
उस कर्मचारी का नाम बताओ जिनकी कर्मचारीसंख्या 110 से ऊपर है	Select Name from Employee where Empno>'110';
उस कर्मचारी का नाम बताओ जिनका वेतन 60000 से ज्यादा है	Select Name from Employee where Salary>'60000';
सभी कर्मचारियों के बारे में बताओ	Select * from Employee;
हर विभाग की जानकारी दो	Select * from Department;
उस कर्मचारी का विभाग नाम बताओ जिनकी विभागसंख्या 7 है	Select Dept_name from Employee, Department where Employee.Deptno = Department.deptno and Employee.Deptno='7';
उन कर्मचारियों का नाम बताओ जिनका विभागनाम खरीद है	Select Name from Employee ,Department where Employee.Deptno= Department.Deptno and Department.Dept_name = ' खरीद ';

उस कर्मचारी का शहर बताओ जिसका नाम हिना है	Select City from Employee where Name = ' हिना ';
---	--

Table 2: Testing results performed on update column (s) queries

Hindi Sentence	SQL Query
उस कर्मचारी का नाम राम करो जिसकी कर्मचारीसंख्या 107 है	Update Employee set Name='राम' where Empno='107';
उस कर्मचारी का शहर पटियाला करो जिसका नाम अशोक है	Update Employee set City='पटियाला' where Name=' अशोक';
उस कर्मचारी का विभागनाम प्शासन करो जिसकी विभागसंख्या 1 है	Update Department set Dept_name='प्शासन' where Deptno = '1';
उन कर्मचारियों की विभागसंख्या 2 करो जिनका वेतन 43000 है	Update Employee,Department set Employee.Deptno='2' where Employee.Salary='43000';
उस कर्मचारी का शहर पटियाला करो जिसका विभागनाम खरीद है	Update Employee,Department set Employee.City = 'पटियाला' where Department.Dept_name= 'खरीद';
उस कर्मचारी का शहर शिमला करो जिसकी विभागसंख्या 3 है	Update Employee,Department set Employee.City = 'शिमला' where Department.Deptno='3';
राम का शहर जम्मू करो	Update Employee set City='जम्मू';

Table 3: Testing results performed on delete row (s) queries

Hindi Sentence	SQL Query
उस कर्मचारी की जानकारी हटाओ जिसका नाम अशोक है	Delete from Employee where Name='अशोक';
उस कर्मचारी की जानकारी हटाओ जिसका विभागनाम खरीद है	Delete from Employee ,Department where Department.Dept_name='खरीद';

## Conclusion

The proposed Natural Language interface to Databases to Employee database accepts query in Natural Language, parses and tokenizes the sentence with the help of Hindi Shallow Parser and maps the

Hindi root words with their corresponding English words with the help of dictionary maintained. After mapping, sentence is checked whether it is data retrieval, update or delete type of sentence. This is done by analyzing Hindi input sentence. After analyzing the Hindi sentence, table names, column names and conditions are searched in the database tables. The tokens are mapped with database values. After mapping, query is again analyzed to check whether it involve joining of multiple tables or not. SQL query is then generated and executed and result is displayed to user.

### References

- [1] *Androutsopoulos, G.D. Ritchie and P. Thanisch, "Natural language interfaces to databases – An Introduction", Department of Computer Science, University of Edinburgh, King's Buildings, Mayfield Road, Edinburgh EH9 3JZ, Scotland, U.K., 1995.*
- [2] *Mrs. Neelu Nihalani, Dr. Sanjay Silakari and Dr. Mahesh Motwani "Natural Language Interface for Database: A Brief Review", International Journal of Computer Science Issues, vol. 8, Issue 2, March 2011.*
- [3] *Himani Jain and Parteek Bhatia "Hindi Language Interface to Databases", Journal of Global Research in Computer Science, vol.2, no.2, pp. 107-112, April 2011.*
- [4] *F. Siasar Djahantighi, M. Norouzifard, S.H. Davarpanah and M.H. Shenassa "Using Natural Language Processing in Order to Create SQL Queries", in International Conference on computer and Communication Engineering, IEEE, Kuala Lumpur, May 2008.*
- [5] *Hu Li, Yong Shi "A Word-Net Based Natural Language Interface to Relational Database", in 2nd International Conference on computer and Automation Engineering, IEEE, pp. 514-518, 2010.*