Parallel Algorithm for Automatic Database Normalization

Sonali Bavaskar, Sayli Kapale, Yashashri Kadam, Y.V.Dongre,

Department Of Computer Engineering, V.I.I.T., Pune

yashwant.dongre@yahoo.com,sonali3092@gmail.com,yashashri153@gmail.com,sayli.kapale@gmail.com

ABSTRACT- We are going to examine the problem of database normalization in a very parallel environment. Existing sequential algorithms are much time consuming, specially the process of transforming relations into 3NF. We are going to propose parallel algorithms for automatic database normalization. In this paper we are going to develop automatic tool and implement parallel algorithm for RDBMS normalization and examine the performance comparing the existing algorithm. A set of relational tables are created with minimum data redundancy which prevents consistency and facilitates correct insertion, deletion, and modification [2].

1. **KEYWORDS- Database Normalization, Normalization Forms, Functional Dependencies, Attribute Closure, Minimal Cover.**

I.INTRODUCTION

Normalization is a process that helps database designers to design table structures for an application. To reduce redundant table data to the very minimum Normalization is used. Database normalization is the process of organizing the fields and tables of a relational database to minimize q and dependency. Normalization divides large tables into smaller which are less redundant and defines relationships between them. It is essential to remember that redundant data cannot be reduced to zero in any database management system [7]

The entire Normalization process is based upon the analysis of relation,their schemes, their primary keys and their functional dependencies.

II.NORMAL FORMS

E.F.Codd proposed three normal forms that he called first,second,and third normal form.

These forms are generally abbreviated and referred to as 1NF,2NF,3NF respectively.

In addition to these original normal forms there exist others such as the Boyce Codd Normal form, Fourth Normal Form ,Fifth Normal Form. It further extends upto Eight Normal Form

FIRST NORMAL FORM

A relation r(R) is said to be first Normal Form (1NF) if and only if every entry of the relation that is the intersection of a tuple and a column has at most a single value [8]. When a table is decomposed into two-dimensional tables with all repeating groups of data eliminated, the table said to be in it is first normal form.

To understand the application of normalization

to table data the following table structure will be taken as example:

Project No.	Project Name	Employee Number	Employee Name
P001	MySQL on Linux Using	E001	Pratik Gavali
P001	MySL on Linux Using	E002	Amit Gandhi
P001	MySQL on Linux Using	E006	Rohit Naik
P002	Using Star Office on Linux	E001	Chaitali Sonawane
P002	Using Star Office on Linux	E007	Nayan Shinde

In the above data there are a few problems:

1. The project Name in the second record is misspelled. This can be solved by removing duplicates. Do this using Normalization.

2. Data is repeated and thus occupies more space.

A table is in 1st normal form if:

1. Repeating groups are not present.
2. Key attributes are well defined.
3. All attributes are dependent on a primary key.

Following techniques to covert table in to First Normal Form:

A key that will uniquely identify each record should be assigned to the table. This key has to be unique because it should be capable of identifying any specific row from the table for extracting information for use. This key is called the primary key.

Field	Key
Project Number	Primary Key
Project Name	--
Employee Number	Primary Key
Employees Name	--
Rate Category	--
Hourly Rate	--

This table is now in First Normal Form.

Second Normal Form:

A table is said to be in its 2NF when each record in the table is in first normal form and every column should be totally dependent on its primary key [8].

A table is in 2NF form if:

1. It's in 1NF.
2. No partial dependencies are included

Following steps for covert table in to second normal form:

1. Find and remove fields that are related to the only part of the key.
2. Group the removed items in another table.
3. Assign the new table with the key that is part of a whole composite key.

In our example all the fields reveals the following:

1. Project name is only dependent on Project number.
2. Employee name, Rate category and Hourly rate are dependent on Employee number.

To covert the table into the second normal form remove and place these fields in a separate table, with the key being that part of the original key on which they are dependent [8].

So now we have three tables:

Table: EmpProj

Field	Key
Project Number	Primary key
Employee Number	Primary Key

Table: Emp

Field	Key
Employee Number	Primary Key
Employee Name	--
Rate Category	--
Hourly Rate	--

Table: Proj

Field	Key
Project Number	Primary Key
Project Name	--

Now our table is in the 2NF.

Third Normal Form

Table data is said to be in 3NF when all transitive dependencies are removed from this data.

The table is in 3rd normal form if:

1. It is in 2NF.

2. There are no transitive dependencies present.

A general case of transitive dependencies is as follows:

A, B, C are three columns in table

If C is related to B.

If B is related to A.

Then C is directly related to A.

In this Normal form, we can remove transitive dependency by splitting each relation in two separate relations. This means that data in columns A, B, C must be palced in three separate tables, which are linked using a foreign key.

In our above tables, all the fields reveal the following:

1. Employee table contains more than one non-key attribute.
2. Employee name is neither dependent on Rate Category nor Hourly Rate.
3. Hourly Rate is dependent on Rate category.

Table: EmpProj

Field	Key
Project Number	Primary Key
Employee Number	Primary Key

Table Emp

Field	Key
Rate Category	Primary Key
Hourly Rate	--

Table Proj

Field	Key
Project Number	Primary Key
Project Name	--

Now this tables are in Third Normal Form.

BCNF:

It is necessary to carry out the normalization process to next higher state that is BCNF for eliminating the anomalies in 3NF relations. A relation r(R) is in BCNF if and only if the relation is in 1NF and for every functional dependency of form X->A , we have that either A is a subset of X or X is super key of r. In other words every functional dependency is either a trivial dependency or in the case that the functional dependency is not trivial then X must be super key[9].

4NF and 5NF:

4NF is more restrictive than BCNF. It uses multivalue dependencies .If schema R is not in BCNF, then there is a non-trivial functional dependency. $\alpha \rightarrow \beta$ holding on R, where α is not a super key. $\alpha \rightarrow \beta$ implies $\alpha \rightarrow\rightarrow \beta$ by replication rule. Hence $\alpha \rightarrow\rightarrow \beta$ is non-trivial functional dependency and α is not super key, which means that R is not in 4NF. Fourth and Fifth normal forms also include composite key. To minimize the no of filed which are involved in a composite key these normal forms are used.

5NF is also known as project Join Normal Form. In this form information can be reconstructed from smaller pieces of information which can be maintained with less redundancy.

Need:

The need for parallel algorithms is increasing. The application of parallel algorithms is automatic normalization. By using parallel algorithms, we can reduce the time complexity for relational database normalization and increase the normalization speed

Advantages:

1Simplification of Data maintenance.

2. It allows data to retrieve with maximum speed.
3. It simplifies data maintenance by updating, inserting and deleting.
4. The need to restructure tables is reduced by rationalization of table data.
5. To improve the design quality of an application of table data by rationalization.

Disadvantages of sequential and Traditional Algorithms:

1. Existing algorithms are much time consuming.
2. The process of transforming relations into 3NF is also time consuming.

3. In software industry normalization is mostly carried out in manual manner.

4.There is demand for skill persons expertise in normalization for carrying it manually, so more than one person need to be involved in this process.

III.EXISTING WORK

[1]. In a paper, A Parallel Algorithm for Relational Database Normalization [1990] by Edward R. Omiecinski

He examined the problem of normalizing data in a parallel environment. When set of functional dependencies that is reduced (minimal) cover is given then generating relation schemes in 3Nf is straightforward.

He worked on reduced cover that is minimal cover for set of functional dependencies that can be produced parallely. The correctness of his algorithm is based on two important theorems.

Theorem 1: Extraneous attributes in the left-hand side of different FDs can be removed in parallel [3].

Theorem 2: Removing extraneous attributes from the right and side of FDs in different equivalence classes in parallel is equivalent to some serial execution of removing extraneous attributes, i.e., considering one FD at a time [3].

The performance of parallel algorithm is quite good compare to serial if number of FDs belongs to equivalence classes is highly skewed. Also further he translated 3NF algorithm into its parallel version and also showed that its performance is better.

[2]. In paper,Automatic Database Normalization and Primary key generation by Amir -H. Bahmani, Mahmoud Naghibzadeh, , Behnam Bahmani

For analysis of relational database, Normalization is most exercised technique. The paper presented a new complete Automatic Relational Database Normalization method. It produced the dependency matrix and also directed graph matrix and determinant key transitive dependency matrix. It then proceeded with generating 2NF, 3NF and BCNF normal form and details of methods discussed.

Two examples, one without multiple candidate keys and one with multiple candidate keys are considered and the defined algorithms are applied to produce the desired final tables.[2]

One more side product of the research was to automatically identify primary key for every final generated table. He believed that the algorithms discussed are very timely and efficiently.

[3].In paper,Parallel Algorithms for Automatic database normalization[2010] by Amir Hassan Bahmani

In this paper they have proposed parallel Algorithm for Automatic Database Normalization. The details of proposed parallel algorithm. The process is based on generation of dependency matrix, determinant key transitive dependency matrix, and directed graph matrix [1].

Also the details of method for 2NF and 3NF are discussed based on data structures. The MPI implementation results on a cluster with eight processors indicate a considerable reduction in time of the automatic database normalization process [1].

IV.PROPOSED WORK

We are going to develop an Automatic tool for Normalizing huge database in parallel environment. Also the concept of 1NF till 5NF and BCNF are discussed. Algorithm such as minimal cover and attribute closure of functional dependencies are used. We are going to find the time complexity and also comparison with sequential algorithm is done on the basis of time complexity algorithm. we are going to Normalize relation to 1NF and 2NF using Existing Algorithms and also Normalize relation up to 2NF using Parallel Algorithms.

V.CONCLUSION

In our work, we have examined the problem of database normalization in a parallel environment. Generating relation schemes in third normal form is straightforward when given a set of functional dependencies that is a reduced (minimal) cover. We showed how a reduced cover (minimal cover) for a set of functional dependencies can be produced in parallel. The performance of the parallel algorithm when compared to the serial algorithm is quite good. In addition, we translated the third normal form algorithm into its parallel version and showed that its performance is also good.

VI.REFERENCES

[1].Parallel Algorithms for Automatic database normalization[2010] by Amir Hassan Bahmani,

S.Kazem Shekofteh , Mahmoud Naghibzadeh, Behnam Bahmani , Hossein Deldari.

[2]. Automatic Database Normalization and Primary key generation by Amir -H. Bahmani, Mahmoud Naghibzadeh, , Behnam Bahmani

[3].A Parallel Algorithm for Relational Database Normalization[1990] by Edward R. Omiecinski

[4].Simple Guide to Five Normal Form in Relational Database by William Kent

[5].Database Normalization, denormalization and forces of darkness a white paper by Melissa Hollingworth

[6].Database System Concepts by Abraham Silberschatz , Henry F. Korth

[7].Database Design by Gio Wiederhold

[8].SQL,PL/SQL by Ivan Bayross

[9].Fundamentals of Relational Database by Ramon A. Mata-Toledo and Pauline K. Cushman

