

# Ameliorated Methodology for Base Design in Information System

Kamalamma. K. V., Ajeet. A. Chikkamannur

**Abstract:** The pragmatic Information retrieval technique is providing the responses to the users' query depending on their choice. The clients are struggling hard to comprehend the semantic within the reports. The data recovery within common language content isn't organized and could be semantically equivocal. The unstructured data may contain non-key attributes. The joint operation takes place between the primary key and foreign key of different tables. The foreign key of one table must be the primary key of another table is the most common in a database and heavily optimized. Join is used to connect rows in another table (or even in the same table) based on the arbitrary condition for structured data but what if for unstructured data to retrieve an information? Information containing an unstructured data and Non-key attributes needs the establishment of relation between Non-key attributes as well as Non-key joins. This paper proposes a data base design to retrieve information with Non-key attributes and representing the knowledge with decision tree. From the decision tree the semiotic is extracted i.e. path from root node to leaf node.

**Keywords:** Semantics, Non-key attribute, Decision tree, semiotics.

## I. INTRODUCTION

In Relational Database, the view resembles the table of database which consists of rows and columns. It is created by selecting attributes from the different tables present in the database. It extracts the columns based on which the required rows are selected. The customers are attempting hard to abuse the semantic learning within the reports since information recuperation depicts with normal language content which isn't for each situation all around composed and could be semantically obscure. Some of the attributes are Non-key attributes they do not uniquely identify a row in the table. Keys are used to identify any data of the table in row manner. By applying some conditions we can fetch data if we don't have keys in a database. There is a Chances of getting wrong data, if we are trying to fetch the same data more than once and if there is a duplicate entry of the data. So we have to go for many trials to get the right data.

Views can be created only for key attributes, and it is static. Proposed work considers the views with respect to domains (based on the user requirements) which makes it dynamic. After which key join is to establish a relation or join the tables. If the table contains key attribute, it will be ways to join the tables, based on certain conditions such as primary key of one table will be the foreign key of another table [11].

**Revised Manuscript Received on December 05, 2019.**

**Kamalamma K. V.** Associate Professor, Department of Computer Science and Engineering, Visvesvaraya Technological University, India.

**Dr. Ajeet A. Chikkamannur,** Professor & Head, Department of Computer Science and Engineering, Visvesvaraya Technological University, India.

(1 to many relationships). Suppose the table doesn't contain key attribute, table join is based on all possible combinations of attributes. Figure 1 describes join operation of two tables without keys. The attribute (name) of table1 has 4 values name1, name2, name3 and name4 and other 4 values in attribute(department) in table2 are d1, d2, d3 and d4. Then perform non key join between these two attributes of different tables will get 16 combinations. How to join the above tables to get 1 to 1 relationship? The primary goal of this paper is to retrieve all the information which is relevant to a user query and also accommodate the knowledge representation with decision tree.

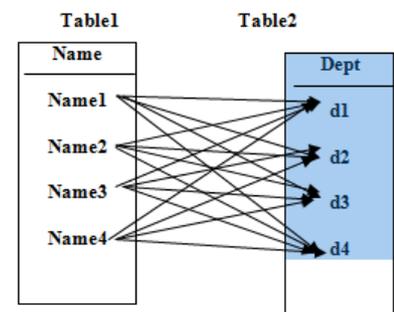


Figure 1: Join operation without key

## II. LITERATURE REVIEW

The author Weipeng [1] stated, with proof that the key dependency of the source table gives the Non-key functional dependency of the derived table. When there is a key dependency existing, there is a need for derived table to have its foreign key in order to refer the primary key from source table. However in some cases, users need not to have a primary key. In such cases, information should be fetched without the primary key.

The creator Priti Mishra and Margret H. Eich [2] express that to execute the database effectively we have to use joins. And this is the most tricky operation in the database, as no well defined interfaces exists between relations are required to present (as they are with framework and different leveled structures). The join is one of the principle social polynomial math tasks that can allow the combining of related tuples from different relations based on the different quality plans. Since it is executed repeatedly (again and again) it is costly, much research exertion has been applied to the enhancement of join handling. Having a key to each source table is certainly not a decent answer for putting away more noteworthy than 1TB of information. In spite of the fact that the join activity is exceptional to social variable based math and inquiry dialects on social databases, comparable operations exist in different data models.

The creator Harborn[3], express that the usage of join activities started before what is as of now suspected of as database framework. The capacity of connecting two or more individual records together was given by the presentation of direct document access. The pointers can be used between these documents to join the two records together in earlier defined options. By using "chain pursuing" activity the early bill of material processor encouraged and handled these joined documents.

The maker CODASYL[4], portrayed the improvement of the database task group (DBTG) database standard in the data manipulation language (DML) exercises required by the framework database systems were described. A noteworthy number of these assignments grant the seeking after of chains between records. Similarly as with the early record frameworks, notwithstanding, just predefined chains (sets) with predefined joins were permitted. To pursue chains in arranged database, use the task such as FIND\_OWNER and FIND\_NEXT. The IBM's IMS database management system gives comparable chain-pursuing abilities. The join operations are different from the DBTG compose structures operations because the IMS is simply different leveled in nature. For instance, GET\_NEXT and GET\_NEXT -WITHIN -PARENT are given. By means of the utilization of sensible connections and coherent databases, IMS gives the capacity to consolidate fragments from one physical database sections with another powerfully. To the client, it shows up as though the two fragments are combined into one. The IMS encourages the way calls like this that allows the joining of similar portion of a database from many levels into one section. Then consider linked key of IMS is the key for all the joined sections. This type of arrangement in a database is considered as a join operation.

The possibility of a class order that shows legacy connections between object classes should frequently be navigated [Atkinson et al. 1989]. It needs the join operation to traverse all the levels in the structure. Since different connections are likewise permitted among items and article classes; different sorts of join tasks are frequently given. The ORION question model gives the capacity to pursue the ISA relationship just as unpredictable trait connections [Kim 1989]. As stated by Kim, this kind of operations "are verifiable join of the classes on a class-creation pecking order established at the objective class of the inquiry."

To encourage the join operation in social databases remote keys are used frequently, the novel article identifier are utilize by the object situated in database framework. To recover the joined items quickly some uncommon lists are used , [Bertino and Kim 1989]. Some complex features requires extra sort of join operation to locate the genuine crude qualities. [Banerjee et al. 1988]. as of late, numerous specialists have examined the expansion of social database frameworks to give includes past those of the first social proposition and its related social variable based math. Starburst extensible DBMS can be used to access a pointer-based structure [ Carey et al. 1990: shekita and Carey 1990].

The reason for these pointers is to give proficient usage techniques to joins. The result of this leads a pointer structure like that gave by the previous system and progressive models. An expansion to the social model to deal with fleeting information and join preparing on it has been proposed E1-Masri 1990[5].

The possibility of a join operation has likewise been stretched out to incorporate Non-1NF information. Postgres has stretched out for the relations to incorporate strategies as information types [Stonebraker and Rowe 1986]. The join operations can be performed by A MATCH administrator, for example coordinating has been proposed similarly [Held and Carlis 1987]. A trial database framework called RAD, which permits correlations (and accordingly joins) between subjective dynamic information types. It has been proposed by [Osborn and Heaven 1986]. These assessments are given by customer portrayed frameworks. Join figurings have been parallelized for settled relations [Deshpande 1990]. Hereafter it is seen that the establishment of the association between the non-key attribute(s) of tables is redundant methodology.

Author Dr. Ajeet A chikkamannur and Dr. M Shivanand Included semiotic to keyword SECURE to provide the security to a relational database system into a lexicon of SQL. They have designed a framework the right information is given to right person [8][9][10].

### III. METHODOLOGY AND RESULTS

We propose the following methodology for

1. Non-key join.
2. Decision tree representation

#### Non-key join

- Identify the Domain on user requirement.
- Identify Schema and attribute identification
- Clustering the attributes based on user requirement and Perform join operation to establish the relation.

#### A. Domain identification:

Select the domain based on the user requirements. A domain is the area of related links, information and terms controlled by a particular ruler. Ex: Education Domain, Agriculture Domain etc... Each domain has many sub-domains. Select the domain based on the user interest.

#### B. Identify Schema and Attribute:

Identify the schema and Attributes with respect to domain. The structure behind data organization is called schema. It also represents how different table relationship enables the schemas underlying mission business rules for which the database is created. It can be derived from Generic Noun (refers to all members of a class or group they are often used when making more generalization. Then specific noun ( is a word that names a certain person, place, thing or idea) refers to an attribute. Select an attribute in such a way that all attributes (A) should belong to particular domain (D).

#### C. Cluster the attributes and Establish the Relation

After identification of schema and attribute cluster the related attribute based on the user requirements. Then perform join operation between the schemas to retrieve the information. Figure 2 describes the flow of operation for Non key join.

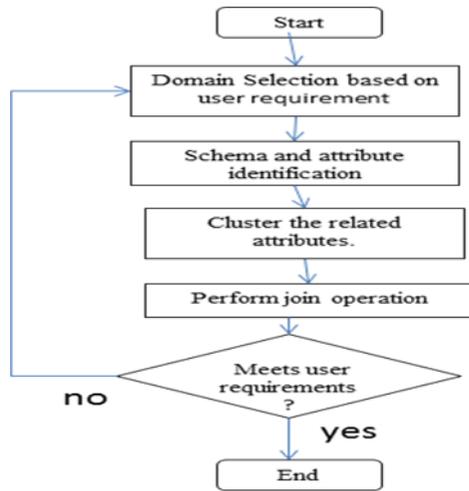


Figure2: flowchart for non-key joins operation

**Case study:** Here our main intension is to categorize the faculties with respect to the department. Initially select the college domain, then identify schema and attributes based on noun concept, (general Noun refers to schema and specific noun refers to attribute values), then clustering is done based on user requirements (consider a scenario where the Name, Subjects thought, specialization Is considered from the faculty table figure 3 and department name and subjects from the department table figure4. finally establish the correlation between the tables. Here we can join these two tables without primary and foreign keys by using the proposed non key join technique

Id	Name	Age	Specialization	Qualification	Subjects handled	Experience

Figure 3: Schema of faculty table

Id	Name	location	Subjects

Figure 4: Schema of department table

### Decision tree representation

Decision tree is a collection of nodes. The Root node to map the identification process, the branch node or decision node represents the conditions that are fulfilled, and then the leaf node represents the conclusion or decision. In the database design make a primary key as the root node and related attributes represents the branching node then to establish the link from one table to another table use non key join, and then final leaf nodes are user requirements.

### IV. CONCLUSION

The main work of this paper is on retrieval of relevant information with respect to the words specified in search, with non-key join technique. We first get the user requirements for finding the parameters that the user will be using frequently. With these parameters, join all tables. Make sure that the redundancy is eliminated. Here the traditional primary key Join methodology is not utilized to fetch any data, but the keys are used to join different tables of the database.

The Non-key attribute specified in a user query will become the root. From this, the table’s non-key attribute(s)

are becoming the link for the purpose of join. The path from root node (table) to the various links to the tables will derive the semiotics of the Information.

Further, this work is to be extended for the unstructured data like documents, images etc...

### REFERENCES

1. Performing Group-By before Join, Weipeng P. Yan Per-Ake Larson Department of Computer Science, University of Waterloo Waterloo, Ontario, Canada N2L 3G1, 1994 IEEE
2. Join Processing in Relational Databases, PRITI MISHRA and MARGARET H. EICH, ACM Computing Surveys, Vol 24, No 1, March 1992.
3. Fde S-vstems Structures and Algorithms, HARBRON, T. R. 1988. Prentice hall , englewood Cliffs, N.J.
4. Estimating block transfer and join sizes In Proceedings of SIGMOD CODASYL, CHRISTODULAKIS, S 1985, 1971 Database Task Group Report
5. Materialized View Selection in a multidimensional database, Elena Baralis, Stefano Parabhoshi, Ernest Teniente, Proceedings of the 23rd VLDB Conference Athens, Greece, 1997
6. Avoiding Cartesian Products for Multiple Joins, SHINICHI MORISHITA, Journal of the ACM, Vol. 44, No. 1, January 1997, pp. 57–85.
7. Representing Knowledge Base into Database for WAP and Web-based Expert System, Istiadi, Emma Budi Sulistiarini, International Conference on Information Systems for Business Competitiveness, April 2014, pp. 81-85
8. Dr. Ajeet A. Chikkamannur Dr. Shivanand M. Handigund “SECURE: An Ameliorated SQL Semiotic for Security” Int. J. Advanced Networking and Applications Volume: 08 Issue: 06 Pages: 3290-3293 (2017) ISSN: 0975-0290
9. Ajeet A. Chikkamannur, Dr. Handigund S. M., “An ameliorated methodology to design normalized relations” ACS/IEEE sponsored, 7th international Conference on Computer Systems and Applications (AICCSA09), Rabat, Morocco, pp 861-864, 2009.
10. Ajeet A. Chikkamannur, Dr. Handigund S. M., “Design of Normalized Relation: An Ameliorated Tool”, International Journal of Computing and Information Sciences (IJCIS), ISSN 1708-0460, Vol 9, No. 1, pp-28-30, 2011
11. Kamamma K V, Dr. Ajeet A. Chikkamannur, “ Survey on Foreign Key Identification, keyword Search and Inclusion Dependencies” , International Journal in Advent Technology, E-ISSN: 2321-9637, 2018.

### AUTHORS PROFILE



**Kamamma K. V.** received her M.Tech degree in computer Science and Engineering in 2012 from the Visvesvaraya Technological University, India. Currently pursuing the Ph.D. and the research is focused on “Design and implementation of semiotics for information system”. Her research interests are Information Retrieval, Machine Learning, Database

Mangaement System. Presently working as Associate Professor, Department of Computer Science and Engineering and teaching for undergraduate courses for 13 years.



**Dr. Ajeet A. Chikkamannur** received his Ph. D. degree in Computer & Information Science and Engineering in 2013 from the Visvesvaraya Technological University, India. on thesis titled “Design of Fourth Generation Languages with the blend of SQL and Japanese Basic English”. Currently working as Professor & Head, Department of Computer Science and

Engineering. His research interests are Object Oriented System Development, Database Management Systems, System Simulation and Modeling, Cloud Computing, Bigdata. He teaches under graduate and graduate. Published number of papers in several national, international conferences and journals.

