Volume 1, Issue 6, September 2016 ISSN: 2456-0006

International Journal of Science Technology Management and Research

Available online at: www.ijstmr.com

# Identifying Useful Data Dependency Using Agree Set form Relational Database

**Pakhale ShamLaxman**[1]
[1]*M.Tech IV Semester*
Lord Krishna College of Technology Indore

**Vijay Kumar Verma**[2]
[2]*Asst. Prof. CSE Dept.*
Lord Krishna College of Technology Indore

---

***Abstract: - Database design methodology normally starts with the first step of conceptual schema design in which users' requirements are modeled as the entity relationship (ER) diagram Data normalization is a common mechanism employed to support database designers to ensure the correctness of their design. Normalization transforms unstructured relation into separate relations, called normalized database. The main purpose of this separation is to eliminate redundant data and reduce data anomaly (i.e., data inconsistency as a result of insert, update, and delete operations). There are many different levels of normalization depending on the purpose of database designer. Most database applications are designed to be either in the third normal forms in which their dependency relations are sufficient for most organizational requirements. In this paper we proposed an efficient approach to identify data dependency using agree set concepts.***

***Keywords:- Agree Set, DBMS, Inconsistency, Dependency, Integrity***

---

## I. INTRODUCTION

Dependency discovery has attracted a lot of research interests from the communities of database design, machine learning and knowledge discovery since early 1980s. Three typical types of dependencies are often involved in the discovery, functional dependencies (FDs), Inclusion dependencies (INDs) and Conditional Functional Dependency (CFD). FDs represent value consistencies between two sets of attributes while INDs represent value reference relationships between two sets of attributes. In recent years, the discovery of conditional functional dependencies (CFDs) has also seen some work. The aim of dependency discovery is to find important dependencies holding on the data of the database. These discovered dependencies represent domain knowledge and can be used to verify database design and assess data quality.

.

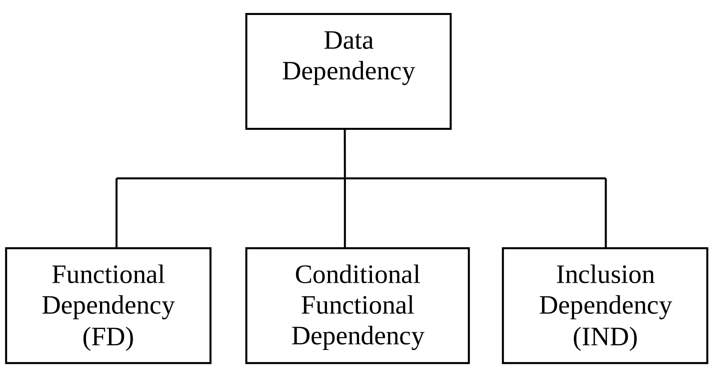


Figure 1 Types of data dependency

A functional dependency is a statement X →Y requiring that X functionally determines Y where X; Y⊆ R. The dependency is satisfied by a database instance r if for any two tuples t1; t2 ∈ r, if t1 [X] =t2 [X] then t1[Y] = t2[Y]. X is called the left-hand side (lhs) or the determinant and Y is called the right-hand side (rhs) or the dependent

A CFD f on R is a pair(R: X→Y, TP),

Where

- X and Y are sets of attribute in attr (R).
- X→Y is a standard FD referred to as the FD embedded Φ.
- Tp is a tableau with attribute in X and Y referred to as the pattern of Φ where each A in X→Y and each tuple t Tp t(A) is either a constant 'a' in dom(A) or an unmanned variable that draw values from Dom(A)

An inclusion dependency (IND) over a database schema R is a statement of the form R1[X] ⊆ R2[Y] where R1, R2 ∈ R and X, Y are sequences of attributes such that X ⊆ R1, Y ⊆R2 and |X| = |Y| . A unary inclusion dependency (UIND) is an IND such that |X| =|Y| =1. An IND R1[X] R2[Y] is of size i if |X| =|Y| =i

## II. FD DISCOVERY METHODS

FD discovery methods follow either top-down or bottom up approach. Top down method first generates candidate FDs and form an attribute lattice and the test there satisfaction. Then at lower level the satisfied FDs are used to prune candidate FDs to reduce the search space. Bottom up method compares the tuples of the relation to find agree-sets or different-sets. These sets are then used to derive FDs satisfied by the relation. Various algorithms for Functional dependency developed under each method

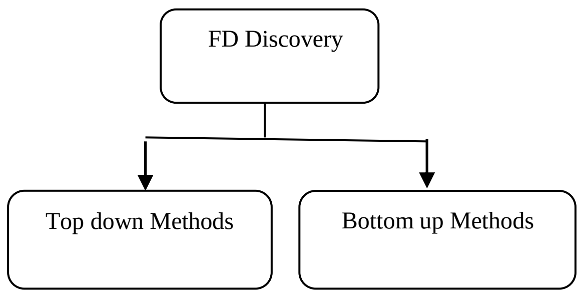


Figure 2. FD discovery methods

### 2.1 Top-down methods

start with candidate FD generation. These methods generate candidate FDs following an attribute lattice, test their satisfaction, and then use the satisfied FDs to prune candidate FDs at lower levels of the lattice to reduce the search space.

### 2.2 Bottom-Up Methods

Different from the top-down methods above, bottom-up methods compare the tuples of the relation to find agree-sets .or difference-sets. These sets are then used to derive FDs satisfied by the relation. The feature of these methods is that they do not check candidate FDs against the relation for satisfaction, but check candidate FDs against the computed agree-sets or difference-sets

## III. LITERATURE REVIEW

In 2010 "Discover Dependencies from Data - A Review". They reviewed the methods for discovering FDs, AFDs, CFDs, and INDs in relational databases and XFDs in XML databases. They show that dependency discovery problem has an exponential search space to the number of attributes involved in the data. hey showed that most data contains FDs and INDs with single or a few attributes on the lhs. With FD discovery, the direction of computation starts with FDs having fewer attributes in lhs. The discovered FDs are then used to prune other candidate FDs in the attribute lattice so that the search space of the computation is reduced. The most commonly proposed and cited method in the literature is the partition method and the negative cover method [1].

In 2011 Wenfei Fan, Floris Geerts & Jianzhong Li , Ming Xiong proposed "Discovering Conditional Functional Dependencies". They provide three methods for CFD discovery. The first, referred to as CFD Miner, is based on techniques for mining closed item sets. The other two algorithms are developed for discovering general CFDs. One algorithm, referred to as CTANE, is a level wise algorithm that extends TANE, a well-known algorithm for mining FDs. The other, referred to as Fast CFD, is based on the depth-first approach used in Fast FD, a method for discovering FDs [2].

In 2012 Jixue Liu, Jiuyong Li, Chengfei Liu, & Yongfeng Chen proposed "Discover Dependencies from Data a Review". They proposed reviews for functional dependency, conditional functional dependency, approximate functional dependency, and inclusion dependency discovery in relational databases and a method for discovering XML functional dependencies. They reviewed the methods for discovering FDs, AFDs, CFDs, and INDs in relational databases and XFDs in XML databases [3].

In 2013 Sujoy Dutta & Dr. Laxman Sahoo proposed "Mining Full Functional Dependency to Answer Null Queries and Reduce Imprecise Information Based on Fuzzy Object Oriented Databases". They proposed the concept of fuzzy functional dependency is extended to full functional dependency on similarity based fuzzy object oriented data model. They also add a data mining algorithm to discover all functional dependencies among attributes. Their major objective is to reduce imprecise information over databases [4].

In 2014 P. Andrew, J. Anish kumar & S. Balamurugan, proposed "Investigations on Methods Developed for Effective Discovery of Functional Dependencies". They give the details about various methods to discover functional dependencies from data. They also discussed Effective pruning for the discovery of conditional functional dependencies [5].

In 2015 Thorsten Papenbrock & Jens Ehrlich proposed "Functional Dependency Discovery: An Experimental Evaluation of Seven Algorithms". They describe, evaluate, and compare the seven most cited and most important algorithms. They classify the algorithms into three different categories, explaining their commonalities. The descriptions provide additional details. Their evaluation of careful re-implementations of all algorithms spans a broad test space including synthetic and real-world data [6].

## IV. PROPOSED APPROACH

There are six components are used in single key based cipher model
Consider a simple database with five attribute
Employee Number (Emp_N), Department No (D_N), Year, department Name (D_Name) and Manager No (Mgr_N)
Maximal equivalence class are { {1,2},{1,6},{2,7},{3,4,5}} the concur set for the pair of tuples (1,2) is concur set con(1, 2)={A} Similarly, we have con(1,6) = con(2,7) = con(3,4) ={B,D,E}, con(3,5) = {E}, con(4,5) ={C,E} so the concur set of r con( r )={A, BDE , E CE}

Table 1 Simple employee Database

| S N | Emp_N (A) | D_N (B) | Year ( C) | D_Name ( D) | Mgr_N (E) |
|---|---|---|---|---|---|
| 1 | 1 | 1 | 1985 | Production | 5 |
| 2 | 1 | 5 | 1994 | Marketing | 12 |
| 3 | 2 | 2 | 1992 | Sales | 2 |
| 4 | 3 | 2 | 1998 | Sales | 2 |

| | | | | | |
|---|---|---|---|---|---|
| 5 | 4 | 3 | 1998 | Purchase | 2 |
| 6 | 5 | 1 | 1975 | Production | 5 |
| 7 | 6 | 5 | 1988 | Marketing | 12 |

Table 3 Maximal sets from the agree sets

| RHS | Cmax (RHS,r) | Size1 | | Size2 | |
|---|---|---|---|---|---|
| | | Candidate | Traversal | Candidate | Traversal |
| A | {AC,ABD} | A,B,C,D | A | BC,BD, CD | BC,CD |
| B | {BCDE,ABD,ABCD} | A,B,C,D, E | B,D | AC,AE, CE | AC,AE |
| C | {BCDE,AC,ABCD} | A,B,C,D, E | C | AB,AD, AE,DB, BE,DE | AB,AD, AE |
| D | {BCDE,ABD,ABCD} | A,B,C,D, E | B,D | AC,AE, CE | AC,AE |
| E | {BCDE} | B,C,D,E | B,C,D,E | - | |

So numbers of functional dependencies are

BC→ A, CD→A, D→B, AC→B,
AE →B, AB→C
AD→ C, AE→ C, B→D, AC→ D, AE→D,
B→E, C→E, D →E

Total dependencies 14 are generated in these dependencies some of the dependencies are redundant dependencies. Like B derives D and D derive B (B→D, D→B). So they are equal equivalent (B↔D). From the generated dependencies AB→C AD→C, B→E, D→E are redundant dependencies. Our approach is to remove these redundant dependencies and generate correct minimal dependencies.
Now the final functional dependencies are

| | | |
|---|---|---|
| **BC→A** | **AB→C** | **B→E** |
| **CB→A** | **AE→C** | **C→E** |
| **AC→B** | | |
| **AE→B** | | |

So there are only 8 functional dependencies are generated

## V. PROPOSED ALGORITHM

Discovering minimal functional dependencies Input: a relation r
Output: minimal functional dependencies for relation r
(1) AGREE SET: computes agree sets from r
(2)CMAX SET: derives complements of maximal sets from agree sets

(3) LEFT HAND SIDE: computes lhs of functional dependencies from complements of maximal sets

(4) DELETE REDUNDANT CANDIDATES AND DEPENDENCIES: find equivalence and remove them also replace deleted attribute by their equivalent

(5)FD OUTPUT: outputs functional dependencies

## VI. COMPARATIVE ANALYSIS

From the table it is clear that the number of dependency is less as compared to the existing method

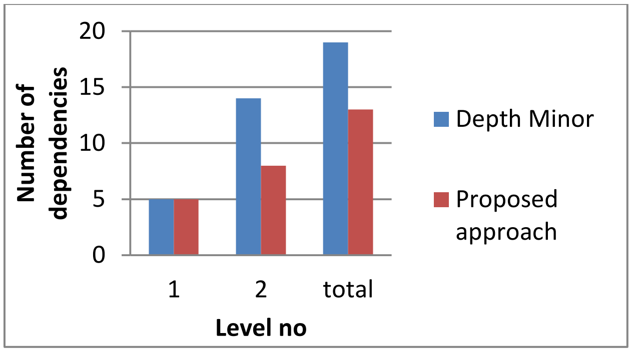


Figure 3 Comparison using no of dependencies

## CONCLUSION AND FUTURE WORKS

Form the comparison graph it is clear that the we modified the Depth minor and remove redundant dependency and generate less candidates form the relational database. The proposed methods use simple calculation and closure set to identify the required dependency. In future we try to find conditional dependency and inclusion dependency

## REFERENCES

[1] Jixue Liu1 Jiuyong Li1 Chengfei Liu2 Yongfeng Chen3 “Discover Dependencies from Data - A Review” School of Management, Xian University of Architecture November 8, 2010

[2] Wenfei Fan , Floris Geerts & Jianzhong Li “Discovering Conditional Functional Dependencies IEEE Transactions On Knowledge And Data Engineering, Vol.23, No. 5, May 2011.

[3] Jixue Liu, Jiuyong Li, Chengfei Liu, & Yongfeng Chen “Discover Dependencies from Data—A Review” IEEE Transactions On Knowledge And Data Engineering, Vol. 24, No. 2, February 2012 251.

[4] Sujoy Dutta & Dr. Laxman Sahoo “Mining Full Functional Dependency to Answer Null Queries and Reduce Imprecise Information Based on Fuzzy Object Oriented Databases” International Journal of Computer Science & Engineering Technology (IJCSET) ISSN : 2229-3345 Vol. 4 No. 03 Mar 2013.

[5] P.Andrew,J.Anishkumar &S. Balamurugan “Investigations on Methods Developed for Effective Discovery of Functional Dependencies” International Journal of Innovative Research in Computer and Communication Engineering (An ISO 3297: 2007 Certified Organization) Vol. 3, Issue 2, February 2015.

[6]Thorsten Papenbrock2 Jens Ehrlich “Functional Dependency Discovery: An Experimental Evaluation of Seven Algorithms” Proceedings of the VLDB Endowment, Vol. 8, No. 10 Copyright 2015 VLDB Endowment 21508097/ 15/06.