Volume 3, Issue 11, November 2018 ISSN (online): 2456-0006
International Journal of Science Technology Management and Research
Available online at: www.ijstmr.com

A Recent Overview of Various Data Dependency Methods

Garima Sarkar
M Tech(Software Engineering) 4th Sem.
Jawaharlal Institute of Technology, Borawan
Khargone M. P. India
garimasrkr09@gmail.com

Sachin Mahajan
Assistant Professor Dept. C.S.E.
Jawaharlal Institute of Technology, Borawan
Khargone M. P. India
sachinmahajan248@gmail.com

Abstract: ***The schemas alone are not sufficient to describe a database. Data semantics in a relational database are expressed by integrity constraints and the most important of these constraints are functional dependencies. FDs are relationships. (constraints) between the attributes of a database relation. FD states that the value of some attributes are uniquely determined by the values of some other attributes. Discovering FDs can also help a database designer to decompose a relational schema into several relations through the normalization process to get rid or eliminate some of the problems of unsatisfactory database design. Database normalization is the process of designing a database satisfying a set of integrity constraints efficiently and in order to avoid inconsistencies when manipulating the database. In this paper we give an overview of Data dependency methods and compare them based on the techniques they used to functional dependency .***

Keywords: Attributes, Redundant, Dependency, Integrity, Normalization

I. INTRODUCTION

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. The next step of logical design focuses on the translation of conceptual schemas into relations or database tables. Conceptual schema and logical designs are two important steps regarding correctness and integrity of the database model. 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.
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.

II. TERMINOLOGY

A *functional dependency* (FD) is a relationship between two attributes, typically between the PK and other non-key attributes within a table. For any relation R, attribute Y is functionally dependent on attribute X (usually the PK), if for every valid instance of X, that value of X uniquely determines the value of Y. This relationship is indicated by the representation below:
X⟶Y
SIN ⟶ Name, Address, Birth date
For the second example, SIN and Course determine the date completed (Date Completed). This must also work for a composite PK
SIN, Course ⟶ Date Completed
Some of the terminologies used in DF are

Armstrong's Axioms

Following three inference axioms for FDs defined on sets of attributes X, Y, and Z known as Armstrong's Axioms

(1) F1. (Reflexivity) If $Y \subseteq X$, then $X \rightarrow Y$.
(2) F2. (Augmentation) If $X \rightarrow Y$, then $XZ \rightarrow YZ$.
(3) F3. (Transitivity) If $X \rightarrow Y$ and $Y \rightarrow Z$, then $X \rightarrow Z$.

Closure

Let $X,Y \subseteq U$ and F be a set of FDs. The closure of X ($X \neq \Phi$) F, denoted X^+, is defined as {Y|X → Y can be deduced from F using Armstrong's Axioms}. The closure of F, denoted F+, is the set of all FDs that can be deduced from F using Armstrong's Axioms.

Definition indicates FD X → Y holds if and only if Y ∈ X^+. For X, Y⊆ U, we use $(XY)^+$ to denote the closure of X UY. Let F = {A → C, BD → AC}. By Definition 2, A^+ = {A, C} and $(BD)^+$ = {A, B, C, D}.

Cardinality of the partition

Let X ⊆ U and let t_1---------- t_n be all the tuples in a relation r(U). The partition over X, denoted π_X, is a set of the groups such that t_i and t_j, 1 ≤ i, j ≤ n, i≠ j, are in the same group if and only if $t_i[X] = t_j[X]$. The number of the groups in a partition is called the cardinality of the partition, denoted $|\pi_X|$. .

TID	A	B	C	D	E
T1	0	0	0	2	0
T2	0	1	0	2	0
T3	0	2	0	2	2
T4	0	3	1	2	0
T5	4	1	1	1	4
T6	4	3	1	1	2
T7	0	0	1	2	0

Table 1 Simple relational database

In 1 table, by Definition, π_A = {{t1, t2, t3, t4, t7}, {t5, t6}}, and π_{CE} = {{t1, t2}, {t3}, {t4, t7}, {t5}, {t6}}. By Definition 3, $|\pi_A|$ = 2 and $|\pi_{CE}|$=5.

Equivalent attributes

Let X, Y ⊆ U. If X → Y and Y → X, then X and Y are said to be equivalent sets of attributes, denoted by the equivalence X ↔ Y. The following theorem shows that we can check whether or not the equivalence X ↔ Y is satisfied in a relation U by comparing the closures of X and Y. Let U = {A, B, C, D} and F = {BD → AC, CD → B}. By Definition, $(BD)^+$ = {A, B, C, D} and $(CD)^+$ = {A, B, C, D}. Since CD ⊆ (BD) + and BD ⊆ (CD) +, we have BD ↔ CD.

Nontrivial closure

Let F be a set of FDs and X^+ be the closure of X . The nontrivial closure of X, denoted X*, is defined as X* = X^+ − {X}. For X,Y ⊆ U, we use (XY)* to denote the nontrivial closure of the set of attributes X ∪ Y, similarly to how we use $(XY)^+$ to denote the closure of attributes X ∪ Y. We have $(XY)^+$ = (XY)* ∪ XY

III. OTHER FUNCTIONAL DEPENDENCY

Fully Functional Dependence (FFD)

The term full functional dependency is used to describe the minimum set of attributes in the determinant of an FD. The rules for full functional dependency are that if the set of attributes Y are to be fully dependent on the set of attributes X, the following must hold:

Y is functionally dependent on X and

Y is not functionally dependent on any subset of X

Consider the following FD from our previous example relation

{Person-ID, Project, Project-Budget} → {Time-spent-by-person-on-project}

This FD is true, as it states that for each value of Person-ID, Project and Project-Budget, there is a unique value for the time spent. However, the attribute Project-Budget is not required in the determinant. According to these rules, the above FD is not fully dependent, whereas, by removing the attribute Project-Budget, we can acquire an FD which is Fully Functionally Dependent. We say that the attribute Project-Budget is redundant. It may be removed without loosing information.

Partial functional dependency

A functional dependency **X→Y** is called a **partial functional dependency** if some attribute A∈**X** can be removed from **X** and the dependency will still hold.

Transitive functional dependency

Transitive dependencies occur when there is an indirect relationship that causes a functional dependency. For example, ”A→ C” is a transitive dependency when it is true only because both “A → B” and “B → C” are true.

Multi-Value Dependency (MVD)

Multivalued dependencies occur when the presence of one or more rows in a table implies the presence of one or more other rows in that same table. For example, imagine a car company that manufactures many models of car, but always makes both red and blue colors of each model. If you have a table that contains the model name, color and year of each car the company manufactures, there is a multivalued dependency in that table. If there is a row for a certain model name and year in blue, there must also be a similar row corresponding to the red version of that same car

Join Dependency

n dependency theory, a join dependency is a constraint on the set of legal relations over a database scheme. A table T is subject to a join dependency if T can always be recreated by joining multiple tables each having a subset of the attributes of T. If one of the tables in the join has all the attributes of the table T, the join dependency is called trivial.

The join dependency plays an important role in the Fifth normal form, also known as project-join normal form, because it can be proven that if you decompose a scheme R in tables R_1 to R_n the decomposition will be a lossless-join decomposition if you restrict the legal relations on R to a join dependency on R called *(R_1, R2,R3……R_n).

IV. LITERATURE REVIEW

In 2010 Y. V. Sreevani and T. Venkat Narayana explores inconsistencies in the existing databases by finding the functional dependencies extracting the required information or knowledge based on rough sets. They also discusses attribute reduction through core and reducts which helps in avoiding superfluous data. Here a method is suggested to solve this problem of data inconsistency based medical domain with a analysis. They examine and to eliminate all unnecessary knowledge from the knowledge base by preserving only that part of the knowledge which is really useful. They paper gives some insight into rough sets which can be used to know data dependencies and extraction of knowledge. The ideas envisaged and depicted here are useful in the domain which deal huge collection of databases to analysis and take rational decisions in the areas such as banking, stock markets, medical diagnosis etc.

In 2011 Tennyson X. Chen and Martin D. Meyer, first discuss why practitioners need to further improve their databases after they apply the traditional normalization methods, because of the existence of functional entanglement, a phenomenon we defined. They outline two methods for identifying functional entanglements in a normalized database as the first step toward data quality improvement. They analyze several practical methods for preventing common data anomalies by eliminating and restricting the effects of functional entanglements. They reveal shortcomings of the traditional database normalization methods with respect to the prevention of common data anomalies, and offer practitioners useful techniques for improving data quality. Theyr provides practitioners with some important principles that promote improved database design and implementation by going above and beyond the traditional normalization techniques.

In 2012 Mohd Kamir Yusof and Atiqah Azlan described details about current techniques in reducing inconsistent data. The theory and implementation of each technique have been shows and explained. The advantages and disadvantages for each technique also were summarized also describe . They give more understanding about techniques can be used in reducing inconsistent data. In future work, the researcher will aim to implement fuzzy multi attributes decision making for reducing inconsistent data from heterogeneous databases. Fuzzy multi attributes was chosen because this technique is better compared to other techniques for reducing inconsistent data.

In 2013 G.Sunitha and Dr.A.Jaya resolve this issue by doing normalization of multiple databases automatically. Proposed approach provides automatic normalization of databases up to 3NF.The unique feature of the research work is automatic normalization and thereby saving time and reducing mind work Proposed research work performed 1st normal form, 2nd normal form and 3rd normal form. These phases are used to normalize the database tables automatically. The status of the research work was completed with the 1st normal form. Going forward the normalization of 2nd normal form and 3rd normal form will be done. The future enhancements that can be done in system are as follows normalization can be extended up to fifth normal form.

In 2014 Ziawasch Abedjan and Patrick Schulze present a new algorithm DFD for discovering all functional dependencies in a dataset following a depth first traversal strategy of the attribute lattice that combines aggressive pruning and efficient result verification. Proposed approach is able to scale far beyond existing algorithms for up to 7.5 million tuples, and is up to three orders of magnitude faster than existing approaches on smaller datasets. DFD benefits from aggressive pruning through a random-walk depth-first traversal. They used large real world datasets, shows that proposed approach outperform . In general, a random walk approach for choosing Lhs candidates works well. They imagine that incorporating some heuristics might further improve the runtime, despite considering the number of distinct values of a column combination as a heuristic had no influence.

In 2015 Thorsten Papenbrock and Jens Ehrlich classify the algorithms into three different categories, explaining their commonalities. Then they describe all algorithms with their main ideas. The descriptions provide additional details where the original papers were ambiguous or incomplete. The evaluation of careful re-implementations of all algorithms spans a broad test space including synthetic and real-world data. They show that all functional dependency algorithms optimize for certain data characteristics and provide hints on when to choose which algorithm. In summary, they show that all current approaches scale surprisingly poorly, showing potential for future research.

In 2016 kshay Kulkarni and Sachin Batule, present TANE, a proficient algorithm for finding functional dependencies from larger databases. TANE is based on partitioning the sets of rows with respect to their attribute values which makes testing the validity of functional dependency fast even for big databases. The results have shown that the algorithm is faster in use. It is observed that for benchmark databases the running times have improved. Functional dependencies are important metadata which can used to gain knowledge. Discovery of functional dependency can help in removing inconsistent and redundant data. Proposed algorithm, TANE, for the discovery of functional and approximate dependencies from relations. The approach is based on deriving dependencies from partitions and searching for it in level-wise manner

In 2017 A.B Amure and M.A. Amosu focuses on solving redundancy problem of Functional Dependencies in a relational schema using the closure of Functional dependency and minimal cover algorithm. Proposed algorithm designed using the three Armstrong's axioms which are reflexivity, augmentation and transitivity and implemented to generate closure of functional dependencies (FDs) called F+. Other rules used are decomposition, union and pseudo-transitivity, which are products of the three basic axioms.The attribute closure

algorithm will also generate the attribute closure X+ under any given set of FDs. Minimal cover (G+) algorithm is then designed and implemented to eliminate redundant FDs. This save storage and optimize the database by eliminating unneeded attributes in FDs

V. COMPARISION

We compare some of the method based on the techniques used to discover functional dependency.

Algorithm name	Techniques used
TANE	partition refinement and apriori-generation
FUN	Cardinality value and free sets
FD Mine	level-wise bottom-up and stripped partitions
Dep-Miner	difference sets and agree set

CONCUSION

In this paper we review some of the recent methods for functional dependency. We give the basic concepts of functional dependency and terminology. Each and every method have some limitation and advantages. Each method used different techniques .The main objective is discover meaningful functional dependency for a given relational database.

REFERENCE

1 Y.V.Sreevani1, Prof. T. Venkat Narayana Rao2 "Identification and Evaluation of Functional Dependency Analysis using Rough sets for Knowledge Discovery" (IJACSA) International Journal of Advanced Computer Science and Applications, Vol. 1, No. 5, November 2010.
2 Tennyson X. Chen, Martin D. Meyer, Nanthini Ganapathi, Shuangquan (Sean) Liu, and Jonathan M. Cirella Improving Data Quality in Relational Databases: Overcoming Functional Entanglements ©2011 Research Triangle Institute. RTI International is a trade name of Research Triangle Institute.
3 Thierno Diallo & JeanMarc Petit "Discovering Editing Rules For Data Cleaning" This article was presented at the 9th International Workshop on Quality in Databases (QDB) 2012.
4 Mohd Kamir Yusof and Atiqah Azlan Comparative Study of Techniques in Reducing Inconsistent Data International Journal of Database Theory and Application Vol. 5, No. 1, March, 2012
5 G.Sunitha, Dr.A.Jaya A Knowledge Based Approach For Automatic Database Normalization International Journal of Advanced Research in Computer Engineering & Technology (IJARCET) Volume 2, No 5, May 2013
6 Ziawasch Abedjan and Patrick Schulze DFD: Efficient Functional Dependency Discovery Andrew, J.Anishkumar & S.Balamurugan "Investigations on Methods Developed for Effective Discovery of Functional Dependencies International Journal of
7 Thorsten Papenbrock and Jens Ehrlich Functional Dependency Discovery: An Experimental Evaluation of Seven Algorithms Proceedings of the VLDB Endowment, Vol. 8, No. 10 Copyright 2015 VLDB Endowment 2150-8097/15/06.
8 Akshay Kulkarni, Sachin Batule, Functional Dependencies Discovery in RDBMS Volume 6, Issue 4, April 2016 ISSN: 2277 128X International Journal of Advanced Research in Computer Science and Software Engineering.
9. A.B Amure1, M.A. Amosu Functional Dependency: Design and Implementation of a Minimal Cover Algorithm IOSR Journal of Computer Engineering (IOSR-JCE) e-ISSN: 2278-0661,p-ISSN: 2278-8727, Volume 19, Issue 5, Ver. I (Sep.- Oct. 2017), PP 77-81 www.iosrjournals.org