Mathematical Formulation and Implementation of Query Inversion Techniques in RDBMS for Tracking Data Provenance

Anika Tabassum
Department of Computer Science & Engineering
East West University
Dhaka, Bangladesh
09anika136@gmail.com

Anannya Islam Nady
Department of Computer Science & Engineering
East West University
Dhaka, Bangladesh
anannyanody@gmail.com

Mohammad Rezwanul Huq
Department of Computer Science & Engineering
East West University
Dhaka, Bangladesh
mrhuq@ewubd.edu

***Abstract*— Nowadays the massive amount of data is produced from different sources and lots of applications are processing these data to discover insights. Sometimes we may get unexpected results from these applications and it is not feasible to trace back to the data origin manually to find the source of errors. To avoid this problem, data must be accompanied by the context of how they are processed and analyzed. Especially, data-intensive applications like e-Science always require transparency and therefore, we need to understand how data has been processed and transformed. In this paper, we propose mathematical formulation and implementation of query inversion techniques to trace the provenance of data in a relational database management system (RDBMS). We build mathematical formulations of inverse queries for most of the relational algebra operations and show the formula for join operations in this paper. We, then, implement these formulas of inversion techniques and the experiment shows that our proposed inverse queries can successfully trace back to original data i.e. finding data provenance.**

Keywords— inversion queries; data provenance; database;

I. INTRODUCTION

Data provenance describes the data source and its transformation to the present state. It describes the history of present digital objects and the creation process of data. Sometimes Data Provenance called "lineage" or pedigree" [1]. In the present era, studies on data provenance have gained considerable importance because of the complexity to find the relation between processed data and source data.

Furthermore, data can be physically moved from one place to another after executing the query, a program or a manual transformation. Also, data can be moved virtually as a result of high-level descriptions of the relationships that are specified between data source [4]. Therefore, data provenance is required for tracking the source data.

Lots of labeled and unlabeled data are producing from different sources day by day. Among those, most of the data are stored in the database after some transformation. So, we may lose the original data. Furthermore, many data-intensive applications are developed, and some scientific model simplifies raw data product to produce new data products [1]. Based on the produced output data, it is important to have the ability to trace back the source data because sometimes we may need to know where the derived data came from so that in case of any unexpected tuples, we will be able to detect those ambiguous tuples using data provenance techniques. Therefore, the main motivation of our work is to develop the inverse query to find the data provenance on a relational database management system.

In this paper, we present several mathematical formulations of inverse queries to join operations. Then, we implement these formulas in structured query language (SQL) for different types of queries executing over data stored in an RDBMS. We develop a couple of algorithms of inversion mechanism which work dynamically over an RDBMS. We present an algorithm for creating an output table automatically on a database to store the derived tuples of user queries. From the tuples of that output tables, we can track the original data using our inversion techniques. We present the results obtained for three types of queries. Those are single table queries, multiple table queries, and aggregate function queries. From these results, researchers or interested users can understand the source of data, i.e. data provenance.

II. RELATED WORK

At present, some research works took place related to data provenance where all the research work tried to find the history of data for various purposes. A telecom service provider can be greatly benefited through Big Data analytics. In [2], authors listed some major aspects influencing the operational performance of telecom contact center. Also, they discussed some challenges and types of big data analytics and data provenance. Therefore, the data provenance can positively affect the operational performance by tracing back to input data that contributed to producing the output.

In recent years, we have witnessed important progress on formal models for provenance. Lots of data management operations involve computations that observe at how and where a tuple was produced. In [3], they adopted the most general formalism for tuple-based provenance. Here, the

978-1-5386-8052-0/19/$31.00 ©2019 IEEE

query language is developed for provenance that can express all types of queries. Also, they discussed some reasons why provenance storage and querying support is beneficial to an RDBMS query system.

Transparency and confidence are some prominent issues in cloud-based systems today. To solve these problems, cloud providers should have a high level of assurance and accountability in order to maintain trust between them and users. That trust can be achieved through data provenance. Because it provides historical data from its original resources and gains trust between them and users [6]. In [6], the authors discussed the overview of data provenance in cloud computing and some approaches in provenance logging system and some challenges in the cloud. Also, they proposed a trusted model to provide secure access to data provenance via a secured communication channel.

According to [7], the authors proposed a technique to record and query semantic provenance data. Also, it presents a Semantic Provenance Annotation Model. In another research [9], authors proposed a data provenance scheme to record the historical backdrop of the responsibility and accountability so that this technique can ensure the security of data provenance and accomplish client protection conservation. At present, trustworthy data collection, data mining, and fusion are vital for the Internet of Things (IoT) applications [10]. IoT application requires accuracy, security and precise data collection. Therefore, in this research, authors introduced a provenance-based trust the board arrangement which helps in setting up a trust relationship among deployed gadgets in the IoT.

III. Data Provenance

Data Provenance refers to records of the inputs, entities, systems, and processes that influence data of interest describing the origin and history of data and adds value to data by describing how it was obtained. Provenance word came from the French term 'provenir', the meaning of the word is 'to come from'. Provenance information describes the origin and the history of data in its lifecycle [5]. It also describes the relations between origin and output by explaining from where output data came, describing in detail how an output record was produced.

Provenance has been used to denote the source of data. Any kind of analysis of data can be viewed as applying a transformation to a collection of existing data items to create a collection of the new data item. The SQL query is a transformation which executes some operations on the input table and returns an output table.

IV. Proposed method

Generally, data can be moved from one database to another. In this case, we may lose some important information during the database transformation process. But the problem that this paper address is that most of the time people do not know the origin of data. To retrieve specific data from the database, we use the SQL query. In this paper, our main goal is to develop the inverse technique dynamically to find the provenance of data (source of data) so that people can track the source data in the easiest and fastest way.

In this paper, first, we provide mathematical formulations of query inversion mechanism to find data provenance. Query inversion is basically a process of tracking back the origin data. Afterward, we implement the formula in SQL and propose an algorithm for creating an output table and another algorithm for inserting the results into the output table so that we can find provenance in an automated manner.

V. Mathematical formulation of inverse queries

A. Mathematical Formulation of Inverse Operation on traditional data for JOIN operation:

Formula:

$$\prod_{k=1}^{N} y_k[\tau.(n_e)] = \prod_{i=1}^{n}.\prod_{j=1}^{n} V\begin{cases} x_1[i] * x_2[j] & if\ x_1[i] = x_2[j] \\ (0,0) & otherwise \end{cases}$$

Query Inversion formula:

Input sequence 1:

$\prod_{i=1}^{n} x_1'[i] = T'(y^1[\tau.(n_e)])$

Input sequence 2:

$\prod_{i=1}^{n} x_2'[j] = T'(y^2[\tau.(n_e)])$

Here, $x_1[i]$ and $x_2[j]$ are two input sequence, $y_k[\tau.(n_e)]$ is the output sequence where τ is the trigger rate, n_e is the execution time, T' is the reverse transformation and n is particular point in time in the output sequence. N is the total number of transformation, k represents the particular output and its value goes to 1, 2, 3….N.

B. Mathematical Formulation of Inverse Operation on streaming data for JOIN operation:

Formula:

$$\prod_{k=1}^{N} y_k[\tau.(n_e)] = \prod_{i=\tau.n_e-n_w+1}^{\tau.n_e}.\prod_{j=\tau.n_e-n_w+1}^{\tau.n_e} V\begin{cases} x_1[i] * x_2[j] & if\ x_1[i] = x_2[j] \\ (0,0) & otherwise \end{cases}$$

Query Inversion formula:

Input sequence 1:

$\prod_{i=\tau.n_e-n_w+1}^{\tau.n_e} x_1'[i] = T'(y^1[\tau.(n_e)])$

Input sequence 2:

$\prod_{i=\tau.n_e-n_w+1}^{\tau.n_e} x_2'[j] = T'(y^2[\tau.(n_e)])$

Where N is the total number of transformation, k represents the particular output and its value goes to 1, 2, 3….N. τ indicates the trigger of the normal equation, n_e is the number of execution time and n_w is used to indicate the window size. V represents the pair of join. x[i] and x[j] is the two input sequence. When all the value of x[i] and x[j] are same the then we will get the output.

VI. Algorithms to Dynamically Generate Outputs

A. *Algorithm for creating an output table*

Input: User Query
Output: Output Table

```
1. $OutputTableName: 'Output_'.
   Time();
2. $FinalQuery = "Create Table".
   $OutputTableName. "(" ;
3. For i IN 1 to total number of
   attributes
4. IF i is equal to total number
   of attributes
5. DO $FinalQuery ← $FinalQuery.
   "a_". "$i". "varchar(100)";
6. ELSE
7. DO $FinalQuery ← $FinalQuery.
   "a_". "$i". "varchar(100)," ;
8. END For
9. Set $FinalQuery to
   $FinalQuery.")";
```

B. *Algorithm for Inserting derived tuples into output tuples*

Input: Derived tuples of the user query
Output: Insertion into the output table

```
1. For i IN 1 to number of tuples
2.   $FinalQuery = "Insert Into" ;
3.   $FinalQuery ← $FinalQuery.
   $OutputTableName. "values (";
4.   $row <- all the tuples of
   output table;
5.   For i IN 1 to total number of
   attributes
6.     IF i is equal to total
   number of attributes
7.       DO $FinalQuery <-
   $FinalQuery. $row[$i];
8.     ELSE
9.       DO $FinalQuery <-
   $FinalQuery. $row[$i] . "," ;
10.  END FOR
11.  END FOR
```

VII. Query Inversion Flowchart

We have designed a prototype to find the source data from the user query. Here the user can execute any query to find data from the database and then can track data provenance through inverse queries. The following flowcharts show the mechanism for different types of queries.

A. *Single Table Queries:*

Fig. 1 shows the flowchart for inversion of the most simple single table queries. User will provide the source table name and then based on the row id of the output table, we find the input data, contributing to producing the output.

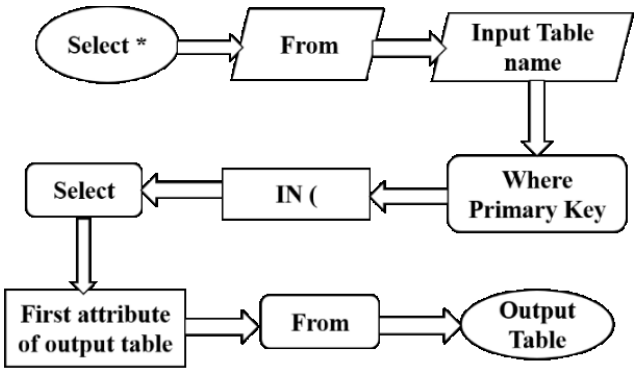

Fig. 1. Inversion flowchart for single table queries

B. *Multiple Table Queries:*

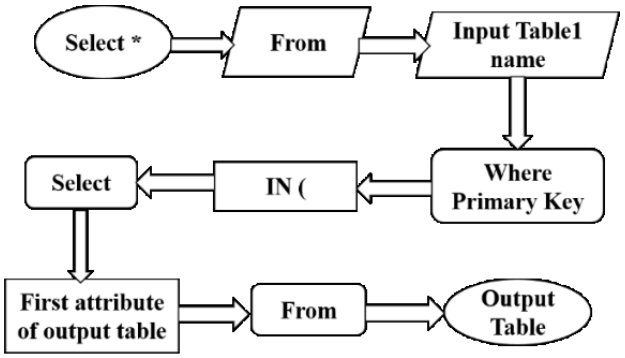

Fig. 2. Inversion flowchart for input sequence 1

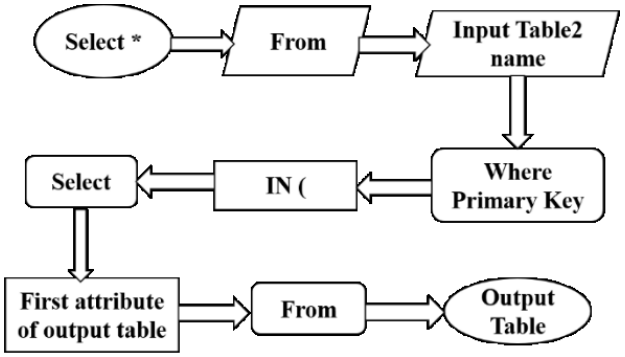

Fig. 3. Inversion flowchart for input sequence 2

Fig. 2 and 3 show the flowchart for the inversion of multi-table queries. Fig. 3 explains the process of obtaining input data contributing to produce output data from the first table (input sequence 1) which is the same as Fig. 1. Fig. 3 shows the same as Fig. 2 for the second table (input sequence 2).

C. *Aggregate Queries:*

Fig. 4 shows the inversion flowchart for inversion of aggregate queries where the user has to provide the source table name and the group by attribute and then the proposed inverse query would able to obtain input data, that contributed to producing the output.

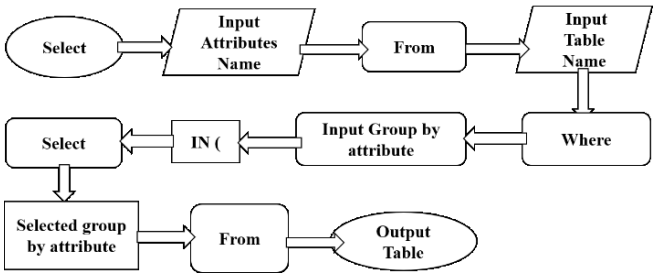

Fig. 4. Inversion flowchart for Aggregate Queries

VIII. Inverse Queries Applied over RDBMS

In this section, we will show inverse query using which we can find the provenance of data over an RDBMS. We may consider two sample relations INSTRUCTOR and TEACHES which have shown in Table I and Table II.

TABLE I. INSTRUCTOR TABLE

ID		Name	Dept_name	Salary
22222		Einstein	Physics	95000
12121		Wu	Finance	90000
32343		Ei Said	History	60000
45565		Katz	Comp. Sci	75000
98345		Kim	Elec. Eng	80000
76766		Crick	Biology	72000
10101		Srinivasan	Comp. Sci	65000
58583		Califieri	History	62000
83821		Brandit	Comp. Sci	92000
15151		Mozart	Music	40000
33456		Gold	Physics	87000
76543		Singh	Finance	80000

TABLE II. TEACHES TABLE

ID	Course_id	Section_id	Semester	Year
10101	CS-101	1	Fall	2009
10101	CS-315	1	Spring	2010
10101	CS-347	1	Fall	2009
12121	FIN-201	1	Spring	2010
15151	MU-199	1	Spring	2010
22222	PHY-101	1	Fall	2009
32343	HIS-351	1	Spring	2010
45565	CS-101	1	Spring	2010
45565	CS-319	1	Spring	2010
76766	BIO-101	1	Summer	2009
76766	BIO-301	1	Summer	2010
83821	CS-190	1	Spring	2009
83821	CS-190	2	Spring	2009
83821	CS-319	2	Spring	2010
98345	EE-181	1	Spring	2009

A. *Single Table Query:*

User Query - Q1:

```
SELECT name FROM Instructor;
```

TABLE III. OUTPUT OF Q1

ID	Name
22222	Einstein
12121	Wu
32343	Ei Said
45565	Katz
98345	Kim
76766	Crick
10101	Srinivasan
58583	Califieri
83821	Brandit
15151	Mozart
33456	Gold
76543	Singh

Inverse Query:

```
Select * from instructor where ID in
(Select ID from Output_1);
```

TABLE IV. DATA PROVENANCE OF Q1

ID	Name	Dept_name	Salary
22222	Einstein	Physics	95000
12121	Wu	Finance	90000
32343	Ei Said	History	60000
45565	Katz	Comp. Sci	75000
98345	Kim	Elec. Eng	80000
76766	Crick	Biology	72000
10101	Srinivasan	Comp. Sci	65000
58583	Califieri	History	62000
83821	Brandit	Comp. Sci	92000
15151	Mozart	Music	40000
33456	Gold	Physics	87000
76543	Singh	Finance	80000

B. *Multiple Table Query:*

User Query – Q2:

```
Select name, dept_name,
course_id, sec_id, semester
FROM Instructor NATURAL JOIN Teaches
where salary > 70000;
```

TABLE V. OUTPUT OF Q2

id	name	dept_ name	course _id	sec_ id	semeste r
12121	Wu	Financ e	FIN-201	1	Spring
22222	Einstein	Physic s	PHY-101	1	Fall
45565	Katz	Comp. Sci	CS-101	1	Spring
45565	Katz	Comp. Sci	CS-319	1	Spring
76766	Crick	Biolog y	BIO-101	1	Summer
76766	Crick	Biolog y	BIO-301	1	Summer
83821	Brandt	Comp. Sci	CS-190	1	Spring
83821	Brandt	Comp. Sci	CS-190	2	Spring
83821	Brandt	Comp. Sci	CS-319	2	Spring
98345	Kim	Elec. Eng	EE-181	1	Spring

Inverse Query:

```
query1= Select * from Instructor where
ID in (Select ID from Output_2);
```

TABLE VI. DATA PROVENANCE OF Q2 FOR INPUT TABLE 1

ID	name	dept_name	salary
22222	Einstein	Physics	95000
12121	Wu	Finance	90000
45565	Katz	Comp. Sci	75000
98345	Kim	Elec. Eng	80000
76766	Crick	Biology	72000
83821	Brandit	Comp. Sci	92000

```
query2 = Select * from Teaches where ID
in (Select ID from Output_2);
```

TABLE VII. DATA PROVENANCE OF Q2 FOR INPUT TABLE 2

ID	course_id	sect_id	semester	year
12121	FIN-201	1	Spring	2010
22222	PHY-101	1	Fall	2009
45565	CS-101	1	Spring	2010
45565	CS-319	1	Spring	2010
76766	BIO-101	1	Summer	2009
76766	BIO-301	1	Summer	2010
83821	CS-190	1	Spring	2009
83821	CS-190	2	Spring	2009
83821	CS-319	2	Spring	2010
98345	EE-181	1	Spring	2009

C. *Aggregate Function Query:*

User Query – Q3:

```
Select dept_name, sum(salary)
from Instructor Group By dept_name;
```

TABLE VIII. OUTPUT OF Q3

Dept_name	Sum(salary)
Biology	72000
Comp. Sci	232000
Elec. Eng	80000
Finance	170000
History	122000
Music	40000
Physics	182000

Inverse Query:

```
Select dept_name, salary from
Instructor where dept_name IN
(Select dept_name from Output_3);
```

TABLE IX. DATA PROVENANCE OF Q3

Dept_name	Salary
Physics	95000
Finance	90000
History	60000
Comp. Sci	75000
Elec. Eng	80000
Biology	72000
Comp. Sci	65000
History	62000
Comp. Sci	92000
Music	40000
Physics	87000
Finance	80000

IX. CONCLUSION AND FUTURE WORK

Our inversion technique is working for single table query, multiple table query, and an aggregate function. As it will not work for the nested query, in future our main target is to improve our algorithm so that it can also satisfy nested queries. Also, we will apply our query inversion mechanism on streaming data.

In our paper, we proposed query inversion techniques to track the origin of data. First, we discussed different types of the formulation of transformation properties in traditional data. Then, we applied our developed inversion mechanism over a relational database. We proposed a generalized system that works for all single table query, multiple table query, and aggregate function query. We applied our mechanism on some relational operations such as selection operation, projection operation, aggregate functions (max, min, count, avg, sum, etc.) and different types of join operations. Our experimental analysis has shown much promise to find data provenance especially for data-intensive applications with ease.

REFERENCES

[1] Md. S. Uddin, D.V. Alexandrov, A. Rahman," Query Inversion to Find Data Provenance", 2018.

[2] Jose, B. Ramanan, T. R. Kumar, S. M.," Big data provenance and analytics in telecom contact centers", In *Region 10 Conference, TENCON 2017-2017 IEEE* (pp. 1573-1578). IEEE, November 2017.

[3] Karvounarakis, Grigoris, Z. G. Ives, and V. Tannen ,"Querying data provenance", In *Proceedings of the 2010 ACM SIGMOD International Conference on Management of data* (pp. 951-962). ACM, June 2010.

[4] Tan, W. Chiew, "Research problems in data provenance", *IEEE Data Eng. Bull.*, *27*(4), 45-52, 2004.

[5] Cheney, J., Chiticariu, L., & Tan, W. C. " Provenance in databases: Why, how, and where", *Foundations and Trends® in Databases*, *1*(4), 379-474, 2009.

[6] Saad, M. I. Mohd, K. A. Jalil, & M. Manaf, "Data provenance trusted model in cloud computing", In *Research and Innovation in Information Systems (ICRIIS), 2013 International Conference on* (pp. 257-262). IEEE, November 2013.

[7] Xu, Guoyan, & Z. Wang, "Data provenance architecture based on semantic web services", In *Service-Oriented System Engineering (SOSE), 2010 Fifth IEEE International Symposium on* (pp. 91-94). IEEE, June 2010.

[8] Simmhan, Yogesh L., B. Plale, & D. Gannon," A survey of data provenance techniques", *Computer Science Department, Indiana University, Bloomington IN*, *47405*, 69, 2005.

[9] Alharbi, Khalid, & X. Lin, " Pdp: A privacy-preserving data provenance scheme",In *2012 32nd International Conference on Distributed Computing Systems Workshops* (pp. 500-505), IEEE, June 2012.

[10] Elkhodr, Mahmoud, B. Alsinglawi and M. Alshehri, "Data Provenance in the Internet of Things", In *2018 32nd International Conference on Advanced Information Networking and Applications Workshops (WAINA).* IEEE, May 2018.