ISSN: 2278 – 7798
International Journal of Science, Engineering and Technology Research (IJSETR)
Volume 5, Issue 4, April 2016

Database Schema Matching Approach in a Homogenous Distributed Database

[1]Er AnchitSajal Dhar,[2]Dr Wilson Jeberson,ErAmita Banerjee

[1] Department of Computer Science & Information Technology, SHIATS ALLAHABAD
[2]Department of Computer Science & Information Technology, SHIATS ALLAHABAD
[3] Department of Computer Science & Information Technology, SHIATS ALLAHABAD

Abstract. Schema matching is one of the critical step and a basic problem in many database domains such as database integrity and semantic query processing. In the current and normal scenario the matching procedure is generally done manually which is quite time consuming and has numerous issues. In this paper we suggest an automated approach which can distinguish differences between two databases on schema-level, structure level and on constraint level in a homogenous distributed database based on SQL Server

Keywords: Schema matching, database integrity, homogenous distributed database.

I. Introduction

The rapid growth of the networking and database technology has had a major impact on the information processing for the requirement of organization. Information has become the most critical resource in many organization and therefore it an efficient access to the information as well as sharing. As a result many effort has been reported on interconnecting the increase number of database scattered across several site. In order to reconcile the requirement tool are being developed for efficient interconnecting different database schema as well as for administrating the distributed environment

Maintaining the integrity of the distributed database is critical for various application using it. In addition the databases must be consistent during the updates in the schema which should be reflected to all the entities participating in the network. A fundamental approach in schema matching is *Match* which takes input as two schemas and that have a corresponding relationship.

Database schema changes can affect the database itself and anything which uses the database. Currently schema is matching is generally done manually. This task is quite tedious, time taking and is quite prone to error. In this paper we discuss a system named as DB-Compare which is used to automatically search and find the divergence which may be present due to network failure, incorrect configurations or due to human error.When presented with a pair of "client" schemas that need to be matched (and their corresponding database instances), DB-Compare matches them using probabilistic methods, an attempt is made to match every attribute of one client schema with every attribute of the other client schema, resulting in individual scores.

II. Schema Matching Issues

A schema consists of a set of related *elements*, such as tables, columns, classes, Stored Procedure, Stored Functions, Triggers views and attributes. The result of a Match operation is a *mapping*. A mapping consists of a set of *mapping elements*, each of which indicates that certain elements of schema S1 are related to certain elements of schema S2. For example, a mapping

between purchase order schemas POand *Porder*could include a mapping element that relates element.

PO	Porder
Lines Item Line Qty Uom	Items Item Item number Quantity Unit of Measure

Figure 1 Two Schema to be matched

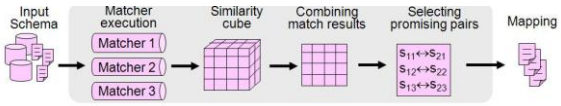
Schema matching isinherently subjective. Schemas may not completely capture the semantics of the data they describe, and there may be several plausible mappings between two schemas (making the concept of a single best mapping ill-defined). This subjectivity makes it valuable to have user input to guide the match and essential to have user validation of the result. This guidance may come via an initial mapping, Thus, the goal of schema matching is: Given two input schemas in any data model and, optionally, auxiliary information and an input-mapping, compute a mapping between schema elements of the two input schemas that passes user validation.

III. Related Works

As the existing system there are several approaches for the database comparison using Bayesian approach and many other technique but they are using having some limitation which gives us the scope to develop a new approach for the comparison of database

IV. Methodology

Methodology: All currently promoted matching systems use a combination of different matching techniques for improving the quality of the matching results. In our work we restrict ourselves to the most common system architecture of parallel combination.



System Architecture

Now the schemas are ready to be superimposed, giving rise to some intermediate integrated schema(s). The inter-mediate results are analysed and, if necessary, restructured in order to achieve several desirable qualities. A global conceptual schema may be tested against the following qualitative criteria: All of the above issues and activities are strongly influenced by the data model used to express conceptual schemas. The relationship between the comparison and conforming activity and the choice of data model is apparent in all the methodologies that perform these activities "by layers" These layers correspond to the different semantic constructs supported by the model; The comparison activity focuses on primitive objects first (e.g., entities in the entity-relationship model); then it deals with those modelling constructs that represent associations among primitive objects (e.g., relationships in the entity-relationship model). Note that relational-model-based methodologies do not show up in the result because the relation is their only schema construct. A few qualitative observations can be made concerning the relative merit of different models.

- Completeness and Correctness. The integrated schema must contain all concepts present in any component schema correctly. The integrated schema must be a representation of the union of the application domains associated with the schemas.

A simpler data model, that is, one with fewer data-modelling constructs, properties, and constraints has an advantage in conforming and merging activities. This stems from various factors:

1. Minimality. If the same concept is represented in more than one component schema, it must be represented only once in the integrated schema.
2. Understandability. The integrated schema should be easy to understand for the designer and the end user. This implies that among the several possible representations of results of integration allowed by a data model, the one that is (qualitatively) the most understandable should be chosen. the possibility of type conflicts is smaller;

Proposed Algorithms:

1. **Algorithms used for table comparison**
   1. While database1.schema is not empty() and database2.schema is not empty()
   2. Fetch the tables from database1.information_schema view where the object type is 'u' in local temporary tables
   3. Fetch the tables from database2.information_schema view where the object type is 'u' in local temporary tables
   4. Compare the objects of table type of database1.schema and database2.schema using subquery, inner join and not in clause
   5. Fetch the result in global temporary tables for further reference
   6. End

Algorithms used for columns comparison in common tables

1. While database1.schema is not empty() and database2.schema is not empty()
2. Fetch the tables from database1.information_schema view where the object type is 'u' in local temporary tables
3. Fetch the tables from database2.information_schema view where the object type is 'u' in local temporary tables
4. Compare the objects of column type of database1.schema and database2.schema using subquery, inner join and not in clause
5. Fetch the result in global temporary tables for further reference
6. End

Algorithms used for datatype comparison in common tables

1. While database1.schema is not empty() and database2.schema is not empty()
2. Fetch the tables from database1.information_schema view where the object type is 'u' in local temporary tables
3. Fetch the tables from database2.information_schema view where the object type is 'u' in local temporary tables
4. Compare the objects of datatype of common table and common columns of database1.schema and database2.schema using subquery, inner join and not in clause
5. Fetch the result in global temporary tables for further reference
6. End

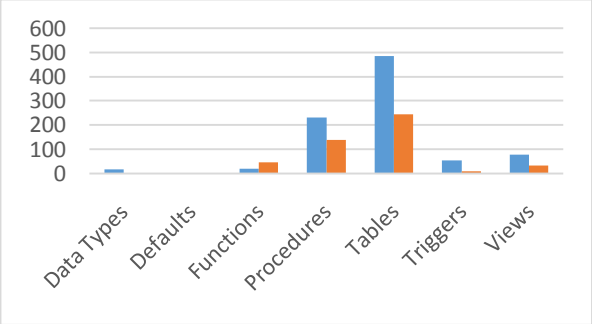
Algorithms used for Column Length comparison in common tables

1. While database1.schema is not empty() and database2.schema is not empty()
2. Fetch the tables from database1.information_schema view where the object type is 'u' in local temporary tables
3. Fetch the tables from database2.information_schema view where the object type is 'u' in local temporary tables
4. Compare the objects of datatype (column length) of common table and common columns of database1.schema and database2.schema using subquery, inner join and not in clause
5. Fetch the result in global temporary tables for further reference

6. End

V. Results & Discussion

The tool was tested on fifteen location on real database where the process of reverse engineering was going to be implemented and the result achieved where up to the mark.



The figure is showing the differences between the testing databases and the linked server database .The graph is displaying the number of equality as well as number of difference in the schema structure of location databases.

VI. Conclusion

The tool developed for the database comparison is working efficiently in homogenous database. The result generated were analysed and were correct and up to the mark. The tool developed is network based and it fetches the schema pattern to the local system therefore analysis and comparison does not affect the database running on the location. Since the database integrity is very much important therefore this approach is helpful for the database administrator and architectures to analyse the schema structure of their databases**.**

VII. Future Scope

This approach will also enables to converge the database objects so that they are consistent at different databases in the future moreover it can be implemented in hetrogenous database environment

References

1. ***ADoan, P. Domingos, A. Halevy****: Reconciling Schemas of Disparate Data Sources: A Machine-Learning Approach. SIGMOD 2001, pp. 509-520.*
2. **Bozovic, N.; Vassalos, V.,***"Two-phase schema matching in real world relational databases," in Data Engineering Workshop, 2008. ICDEW 2008. IEEE 24th International Conference on , vol., no., pp.290-296, 7-12 April 2008 doi: 10.1109/ICDEW.2008.4498334*
3. **Castelli, D.**, *"A strategy for reducing the effort for database schema maintenance," in Software Maintenance and Reengineering, 1998. Proceedings of the Second Euromicro Conference on , vol., no., pp.29-35, 8-11 Mar 1998 doi: 10.1109/CSMR.1998.665729*
4. **Casanova, M.A.; Breitman, K.K.; Brauner, D.F.; Marins, A.L.A.**, *"Database Conceptual Schema Matching," in Computer , vol.40, no.10, pp.102-104, Oct. 2007 doi: 10.1109/MC.2007.342*
5. ***Clifton.C, E. Hausman, A. Rosenthal:*** *Experience with a Combined Approach to Attribute-Matching Across Heterogeneous Databases. Proc. 7th IFIP Conf. On DB Semantics, 1997.*
6. **Desai, B.C.; Pollock, R.**, *"On schema integration in a*

heterogeneous distributed database management system," in Computer Software and Applications Conference, 1989. COMPSAC 89., Proceedings of the 13th Annual International , vol., no., pp.218-224, 20-22 Sep 1989 doi: 10.1109/CMPSAC.1989.65088

7. ***Rahm E., P.A. Bernstein**: On Matching Schemas Automatically. MSR Tech. Report MSR-TR-2001-17, 2001*
8. **Geller, J.; Perl, Y.; Neuhold, E.**, *"Structural schema integration in heterogeneous multi-database systems using the Dual Model," in Interoperability in Multidatabase Systems, 1991. IMS '91. Proceedings., First International Workshop on , vol., no., pp.200-203, 7-9 Apr 1991 doi: 10.1109/IMS.1991.153706*

9. **Han-Chieh Wei; Elmasri, R.,***"PMTV: a schema versioning approach for bi-temporal databases," in Temporal Representation and Reasoning, 2000. TIME 2000. Proceedings. Seventh International Workshop on, vol., no., pp.143-151, 2000 doi: 10.1109/TIME.2000.856595*
10. ***J.A. Wald, P.G. Sorenson:** Explaining Ambiguity in a Formal Query Language. ACM TODS 15(2), 1990, 125-161.*

11. **Khedri, N.; Khosravi, R.**, *"Handling Database Schema Variability in Software Product Lines," in Software Engineering Conference (APSEC), 2013 20th Asia-Pacific , vol.1, no., pp.331-338, 2-5 Dec. 2013 doi: 10.1109/APSEC.2013.52*

12. **Kapfhammer, G.M.; McMinn, P.; Wright, C.J.**, *"Search-Based Testing of Relational Schema Integrity Constraints Across Multiple Database Management Systems," in Software Testing, Verification and Validation (ICST), 2013 IEEE Sixth International Conference on , vol., no., pp.31-40, 18-22 March 2013 doi: 10.1109/ICST.2013.47*

13. ***L. Palopoli, G. Terracina, D. Ursino**: The System DIKE: Towards the Semi-Automatic Synthesis of Cooperative Information Systems and Data Warehouses. ADBIS-DASFAA 2000, Matfyzpress, 108-117.*

14. **Lukovic, I.; Mogin, P.**, *"An approach to relational database schema integration," in Systems, Man, and Cybernetics, 1996., IEEE International Conference on , vol.4, no., pp.3210-3215 vol.4, 14-17 Oct 1996 doi: 10.1109/ICSMC.1996.561500*
15. **Maule, A.; Emmerich, W.; Rosenblum, D.S.,** *"Impact analysis of database schema changes," in Software Engineering, 2008. ICSE '08. ACM/IEEE 30th International Conference on , vol., no., pp.451-460, 10-18 May 2008*

doi: 10.1145/1368088.1368150

16. **Massey, K.D.; Kerschberg, L.; Michaels, G.**, *"VANILLA: a dynamic data schema for a generic scientific database," in Scientific and Statistical Database Management, 1997. Proceedings., Ninth International Conference on , vol., no., pp.104-107, 11-13 Aug 1997*
doi: 10.1109/SSDM.1997.621163
17. **Meurice, L.; Cleve, A.,** *"DAHLIA: A visual analyzer of database schema evolution," in Software Maintenance, Reengineering and Reverse Engineering (CSMR-WCRE), 2014 Software Evolution Week - IEEE Conference on , vol., no., pp.464-468, 3-6 Feb. 2014*
doi: 10.1109/CSMR-WCRE.2014.6747219

 ErAnchitSajalDharis post graduate student pursuing M.Tech in Computer Science and Engineering from Sam Higginbottom Institute of Agricultural, Technology and Sciences in the Department of Computer Science and Information Technology.His areas of interest are Data Structures, Algorithm Design and Analysis and Database Programming.

 Dr. W. Jeberson is working as an Professor and Head of the Department of Computer Science and Information Technology , Sam Higginbottom Institute of Agriculture, Technology and Sciences. He has a vast experience of more than 8 years in teaching and 3 years in industry. He has attended many Internation Conferences and has immense knowledge in the field of Computer Science. His area of specialization are Software Engineering and e- governance.

 ErAmita Banerjee is under Graduate Studentpursuing B.Tech in Computer Science and Engineering from Sam Higginbottom Institute of Agricultural, Technology and Sciences in the Department of Computer Science and Information Technology.Her areas of interest are Data Structures, Web Designing and Database Programming.