

A Survey on Database Systems Handling Computable and Real-World Dependencies

Beena J Stuvart¹, Preeja V²P.G. Student, Department of CSE, SCTCE, Trivandrum, Kerala, India¹Asst. Professor, Department of CSE, SCTCE, Trivandrum, Kerala, India²

ABSTRACT: Databases are integral to many scientific application domains in which the data is complex and may involve real-world activities that are external to the database. Consider two values, x and y , in the database, where $y = F(x)$. To preserve the consistency of data, F needs to be executed to recalculate the value of y and update its value in the database, whenever x changes. It is possible only when F is executed by the DBMS using SQL function. But it is more challenging where F is a human action such as taking manual measurements, conducting a wet-lab experiment or taking the reading of instrument. In this case, when x changes, y remains invalid with the current value of x until the human action involved in the derivation is performed and its output result is reflected into the database. As a result, an update operation in the database may render dependent data items invalid or inconsistent until the real-world activities involved in deriving these items are re-performed and the output results are reflected back into the database. These real-world activities may take long time to perform, and hence it takes long time delays to updates in the database. The long delays between the updates and the need for the intermediate results makes supporting real-world activities in the database engine a challenging task. The survey focuses on various database systems handling dependency involving human action.

KEYWORDS: Database, Dependency, real-world activity.

I. INTRODUCTION

Data dependency plays an important role in database systems. A dependency occurs in a database when information stored in the table uniquely identifies other information in the same table. The functional dependencies [1],[2] are used to model dependencies in the database. A relation satisfies functional dependency $X \rightarrow Y$ if, whenever two tuples agree on X they also agree on Y . The database dependency focused on two concerns such as solving the inference problem and pointing out the historical dependencies ie the efficient data representation, avoiding update anomalies and removing redundancy.

The survey concerns with two types of dependencies such as computable dependency and real-world dependency. A computable dependency involves user defined functions created inside the database and executed directly by the database. The dependency that involves real-world activities [3] and human cooperation are called real-world dependency. The real-world activities cannot be coded inside the database. The concepts of real-world activity are important because the real-world activity functions are close to the normal function. The real-world activities are mapped to a nondeterministic function inside the database because it may not generate the same output when experiments are conducted multiple times.

Databases are integral to many application domains such as experimentation in scientific domains [4] such as physics, biology etc. In these applications the data processing is complex and it involves activities that require human intervention. The real-world activities that involve human action are external to the database systems such as taking manual measurements, collecting instrumental reading etc. To preserve the consistency of the data in the database, the conventional database system use simple functions or procedures that can be coded inside the database and executed automatically by the database. For example deriving age from the date of birth, it requires a simple function and can be

International Journal of Innovative Research in Science, Engineering and Technology

(An ISO 3297: 2007 Certified Organization)

Vol. 4, Issue 7, July 2015

executed directly by the database. In distinction, when human activities are involved in the derivation of the database items, it cannot be coded inside the database systems. As a result, some instants of the database values that are dependent with these real-world activities are invalid. Thus a part of the basic database system is uncertain or invalid until the activities are re-executed and the changes in the output values are reflected back in the database. When querying the database at this moment, the quality of the yielded query result becomes confusing and the result may contain invalid data. Thus handling real-world dependency inside the database is a challenging task.

II. MANAGING DEPENDENCY IN DATABASE SYSTEMS

The computable and real-world dependencies managed in different databases are different. Most of the database systems support computable dependency, because computable dependencies involve derivation functions that can be executed by the database system. Real-world dependency is supported by some database system, because it involves sequence of human action. To prevent inconsistency problem among data items and to normalize database schema, functional dependencies are used. But the functional dependencies cannot solve the inconsistency problem occur due to real-world activities involving human action, because it cannot be executed directly inside the database systems. Thus managing the real-world dependency inside the database is difficult. The dependency issues in a collection of databases in different scenarios as well as in different circumstances are described in the following section.

2.1 Immortal database

Immortal database is a multi-version transaction [5] time database system. Both current and previous database states are accessed by a mechanism of creating new version of data for every transaction. Immortal database is distinguished from other database systems based on its timestamping and managing versions. To agree with transaction serialization order, record versioning is used. To propagate the timestamp to all updates of the database, a lazy timestamping mechanism is used. The versions of the records are linked via version chaining. When a record is updated, then each update creates a new version.

Initially immortal database stores older versions of a recording the same page as the current record version. A pointer is needed to point to the previous version on the same page. All versions are accessed from the most recent record through version chaining, until the disk page is full. The older versions are moved to another page, when the page becomes full. To store the slot number of the earlier version in the page, version pointer field is used.

The computable dependencies among the data items are only handled by multi-version database systems. Hence, the provided flexibility degree and isolation level are based on the traditional concept of transactions. For example, if two values are dependent through an external activity, then a transaction updating one of the values would create a newer version of the database where both viewed as consistent and up-to-date values. It is correct based on the classic approach of transaction but semantically incorrect because the external dependency is not considered. Immortal database have some limitations such as the maintenance of the auxiliary metadata delegated to end-users may cause error, the simple queries become more complex when metadata is added. The main drawback is curation operation [11] which will be performed manually.

2.2 Active database

Integrating production rules [6] facility into the database system provides a mechanism for a number of advanced database features such as derived data maintenance, triggers, version control, integrity constraint etc. A database system with rule processing capabilities provides a useful platform for efficient knowledge-base and expert systems. An active database is a database system with production rules and it consist of an event-driven architecture which can responds to condition both inside and outside the database. Active databases [7],[8] can be used for computation of derived data. Derived data is maintained in the database to interact and encapsulate base data which record real-world facts.

To respond automatically to the actions taking place either inside or outside the database, triggers are used. The outside actions are mapped to operations that the database system can capture. A change inside the database may

International Journal of Innovative Research in Science, Engineering and Technology

(An ISO 3297: 2007 Certified Organization)

Vol. 4, Issue 7, July 2015

trigger the execution of an event outside the database to change the derived data. The system does not keep track of all invalid data items, until the activity is performed.

2.3 Workflow management

In working environments, production involves certain procedures to be executed several times. The individual tasks performed and their relationships performed in these procedures are described in a workflow [9]. To store task description and to implement workflow functionalities, the workflow management system uses DBMS. Workflow management systems (WFMS) [10], [11] are used to define, monitor and execute workflows. The basic functionality of workflow management system is workflow specification, workflow execution, workflow evolution and workflow auditing. The workflow execution handles the dependency.

The process consists of the execution of applications outside the WFMS. The WFMS must follow the logic of the workflow and at any point interact with the appropriate external system to transfer the control of execution as needed. The data may need translation during execution. The Invocation may be automatic or not, depending on the specification. Even for a completely automated workflow, users retain ultimate control during its execution, monitoring and even influencing its operation. Status monitoring is also possible. At any point users may ask for the execution status of the entire workflow or parts of it. Status information may be provided by the WFMS or any external system used in the workflow. Since the dependencies are modelled at the workflow management level i.e., outside the DB, directly querying the database without consulting the workflow system will not reflect such dependencies, and hence may reveal potentially-invalid values to end-users without notification.

2.4 HandsOn database

HandsOn DB [12] is a prototype database engine. It is used for managing dependencies that involve real-world activities while preserving the consistency of the derived data. HandsOn DB enables users to reflect the updates immediately into the database. It provides a facility to instant availability of the data, while the DBMS keeps track of the derived data by marking them as potentially invalid and reflecting their status in the query results. HandsOn DB introduces mechanism for modelling dependency, tracking out-dated data, Data querying, curation and manipulation mechanism.

HandOn DB system provides the user to register events or activities and specify the dependency between the data items into the database. Both column-based or cell based dependencies are allowed in HandsOn database system. In column based dependency, every cell in the column follows the same dependency. In cell-based, different cells in a given column may follow different dependencies. Once the dependencies are defined in the database, they are automatically carried out by the DBMS. In order to properly preserving the consistency of the data, both real-world and computable dependencies need to be defined in the database. One of the main functionality of HandsOn database system is tracking out-dated data. A major benefit for defining real-world activities and dependencies inside the database is that the DBMS will keep track of the possibly old data items that are waiting on external activities to be performed. HandOn DB provides systematic mechanisms for invalidating and revalidating the data inside the database. Curation operators are used for this purpose. It manipulates user-defined dependencies among the database items. The curation operator helps to identify the proper order of revalidation.

III. COMPARISION

The table 1 shows the comparison of various database such as immortal database, active database, workflow management and HandsOn database handling dependency based certain features such as type of dependency, access type, status monitoring and curation operation. Some databases handle both types of dependencies such as computable and real-world dependencies. Status monitoring is only possible in HandsOn database.

International Journal of Innovative Research in Science, Engineering and Technology

(An ISO 3297: 2007 Certified Organization)

Vol. 4, Issue 7, July 2015

Features	Immortal database	Active database	Workflow management	HandsOn database
Type of dependency	Focus on computable dependency	Focus on computable dependency and real-world dependency	Focus on computable dependency and real-world dependency	Focus on computable dependency and real-world dependency
Access type	Access to both current and previous database states	Base or derived dates are accessible.	Current database state is accessible	Database items are accessed with their status
Status monitoring	Not possible	Not possible	possible	Possible
Curation operation	Curation operation is performed manually	Does not support curation operation	Curation operation is no longer needed	Curation operation is integrated inside the database

Table1: Comparison of databases handling dependency

IV. CONCLUSION

Integrating real-world activities into the database engine is a challenging task because it affects the consistency of the data. When real world activities are involved in a database operation, then all dependent and derived values are invalid until that action is performed and the result is reflected back into the database. So handling dependency inside the database is an important task. The different databases use different approaches for handling real-world dependencies. The one possible approach is to store metadata information such as version number or timestamp along with the data values. However this approach has several limitations such as when metadata information is integrated in the query engine, the simple query become more complex. Another possible approach is to delay the updates to the database until all derived values are computed, that is keeping partial results outside the database. It is feasible in some scenarios involving simple dependencies but it does not scale up with complex dependencies. Another approach is in the context of workflow management, where the external activities, dependencies and the database updates are modelled as process of workflow. It handles dependency in efficient manner but sometimes it reveals potentially-invalid values to the end-users. HandOn DB addresses all the limitation and it enables users to reflect the updates immediately to the database. HandsOn DB provides mechanism for defining the real-world activities, tracking out-dated data etc. and provides data values with their status. Thus, HandOn DB is best for scientific data management.

REFERENCES

- [1] D. Maier, "Theory of relational databases," in Comp. Sci. Press, 1983.
- [2] J. Ullman, "Principles of database and knowledge-based systems," vol. 1, 1988.
- [3] M. Y. Eltabakh, W. G. Aref, A. K. Elmagarmid, Y. N. Silva, and M. Ouzzani, "Supporting real-world activities in database management Systems," in ICDE, 2010, pp. 808–811
- [4] M. Y. Eltabakh, M. Ouzzani, W. G. Aref, A. K. Elmagarmid, Y. Laura-Silva, M. U. Arshad, D. E. Salt, and I. Baxter, "Managing

International Journal of Innovative Research in Science, Engineering and Technology

(An ISO 3297: 2007 Certified Organization)

Vol. 4, Issue 7, July 2015

- biologicaldata using bdbms,” in ICDE, 2008, pp. 1600–1603.
- [5] M. Mokbel D. Lomet, R. Barga and G. Shegalov. “Transaction time support inside a database engine”. in ICDE, page 35–46., 2006.
- [6] J.WidomA.Aiken and J. Hellerstein. “Behavior of database production rules: Termination, confluence, and observable determinism,”.In SIGMOD,, pages 59–68, 1992.
- [7] J. Widom and S. Ceri, Eds., Active Database Systems: Triggers and Rules for Advanced Database Processing. Morgan Kaufman
- [8] U. Dayal, “Active database management systems,” SIGMOD Rec.,vol. 18, no. 3, pp. 150–169, 1989.
- [9] A. Ailamaki and et. al.” Scientific workflow management by database management”. in SSDBM, pages 150–159, 1998.
- [10] S. Shankar, A. Kini, D. J. DeWitt, and J. Naughton, “ Integrating databases and workflow systems,” SIGMOD Rec., vol. 34, no. 3, pp. 5–11, 2005.
- [11] P. Buneman, A. Chapman, and J. Cheney, “Provenance management in curated databases,” in SIGMOD, 2006, pp. 539–550.
- [12] Ahmed Elmagarmid Mohamed Eltabakh, WalidAref and MouradOuzzani. Handsondb: Managing data dependencies involving human actions,IEEE, 2013.