LONG-TERM PRESERVATION OF DATABASES THE MEANINGFUL WAY

Janet Delve
University of Portsmouth
School of Creative Technologies
Eldon Building, Winston Churchill Avenue, Portsmouth, PO12DJ, UK
+44 2392 845524
Janet.Delve@port.ac.uk

Rainer Schmidt
AIT Austrian Institute of Technology GmbH
Donau-City-Straße 1
1220 Vienna, Austria
+43(0) 50550-4273
rainer.schmidt@ait.ac.at

Kuldar Aas
National Archives of Estonia
J. Liivi 4
Tartu, 50409, Estonia
+372 7387 543
kuldar.aas@ra.ee

ABSTRACT

Long-term preservation of databases has been discussed in some detail over recent years, for example as part of the PLANETS project, and we have seen the rise of standards like SIARD and ADDML to address this issue. However, these tools / standards are not particularly geared towards the reuse of preserved data, addressing as they do the use case of accessing a single database snapshot covering just one instance in time, and then allowing pre-defined or custom queries to be carried out on this.

This paper will show how the EC-funded E-ARK project (http://www.eark-project.com/) is addressing wider use cases of database archiving and access. A gap analysis carried out in the early phases of the project has identified the fact that archives are not able to carry out Big Data querying / data mining across a variety of archived databases carrying related entities etc., as opposed to querying single databases as mentioned above.

Part of the E-ARK project approach is to address wider use cases by using a combination of state-of-the-art techniques taken from data warehousing, Online Analytical Processing (OLAP), data mining and semantic annotation. Overall this approach means that:
• during the pre-ingest or ingest workflow denormalized representations will be created of the original relational database;
• the database content will be semantically enriched according to available centrally controlled vocabularies;
• the enriched representations will be stored next to the original database;
• when users are interested in a special topic which might be covered in multiple database snapshots, they are allowed to create semantic queries which identify appropriate OLAP cubes and can use additional data mining techniques to combine and make sense of the data in these.

Whilst the work is still ongoing, the paper will shed some light on the details of this approach, and present a conceptual technological solution.

Categories and Subject Descriptors

Ingesting and preserving databases and special records, Perspectives on past and present projects, Re use of public information, Role of standards, Strategy and approach, The cloud, mobile, social, Big Data, Transforming archives through information technologies

General Terms

Database Archiving, Database Preservation, Online Analytical Processing (OLAP), Big Data, Data Warehousing, Online Transaction Processing (OLTP), SIARD, SIARD-DK, ADDML, normalization, denormalization.

Keywords

Reuse, Database Preservation, Data Mining, Data Warehousing, OLAP, OLTP, E-ARK project, Digital Preservation (DP), Denormalization.

1. INTRODUCTION

In the final panel discussion at the Goportis "Getting Ready for Digital Preservation" meeting in Hamburg on October 20th 2011, Seamus Ross asserted that the biggest challenge then facing the DP community was database archiving. In conversation at the 2014 iPRES conference in Melbourne, Professor Ross mentioned to Dr Delve that he saw databases as being one of the greatest inventions of the 20th century, being used as they are in so many different walks of life [Ross, personal communication]. Ross's views coincide with the approach to digital archiving taken by the 3 year E-ARK project, which was set up in February 2014 to develop a pan-European digital archiving system to cater for both records and databases. E-ARK included databases because they are such key digital "workhorses": driving applications large and small; mainframe and web-based; long-standing and recent; and because the data they contain *and* their robust functionality need to be retained for posterity. This paper begins by charting a brief history of database development, focusing on the relational database, and delineating how it changed from being mainly transaction-oriented (via Online Transaction Processing - OLTP) to now also being analysis-driven (as seen by the advent of analytical databases, data warehousing, data cubes, multidimensional databases, dimensional modeling, Online Analytical Processing - OLAP etc.). Following this overview comes a brief review of how some of the national archives in E-ARK currently carry out database archiving. We then discuss some of the innovations in E-ARK where we are using data warehousing / OLAP techniques as part of the OAIS process for archiving databases. We conclude by outlining further work.

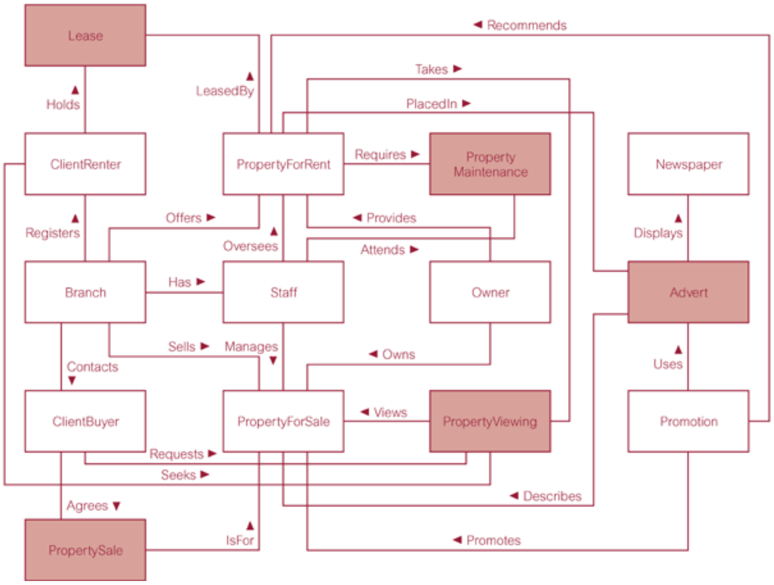

Figure 1. Entity-Relational Model © Connolly and Begg

2. BACKGROUND TO DATABASE DEVELOPMENT

2.1 Database definitions

Connolly and Begg [2] define a database as a "shared collection of logically related data (and a description of this data), designed to meet the information needs of an organization". It is the fact that database data is *logically related* that gives databases their all important *structure*. Date's database contribution is seminal but his definition of a database system as "basically just a *computerized record-keeping system"* [3] belies the mathematical complexity that underlies these digital organizers. Connolly and Begg outline the development of the database through the early hierarchical databases, and they focus on the rigorous process of normalization which lies behind the relational database model. When computer science started as a discipline, there was a concerted effort to formalize it as a serious academic discipline, with the aim that it would have the same gravitas as other subjects, for example, mathematics. It was in this sprit that the mathematician and Oxford graduate Edgar F. (Ted) Codd, then working for IBM, devised the relational system which was the basis for the relational database. The database he outlined in 1972 [1] was founded on actual symbolic relations, with a solid mathematical foundation. In this way, the data models, and eventually the tables and associated Structured Querying Language (SQL) of the database would be well-formed and reliable, producing data that can be used and analyzed with confidence in a wide variety of situations. There are also special mathematically-based relational languages a) to describe the formation of queries (relational algebra), which is particularly useful for query optimization, and b) to ensure queries are logically formed (relational calculus).

2.2 Database Normalization

Relational data modeling in a nutshell is centered on entities which are chosen due to their importance to the modeler. However, once chosen, there are strict rules called normal forms, which ensure that each entity has at least one unique key, and that every attribute describing that entity is related to the key, the whole key (keys can be multi-part) and nothing but the key (so it should not depend on a non-key attribute). This then results in a data model comprising many discrete entities that have to be connected by joins across the keys (called foreign keys). For large and complex models, this can result in an unwieldy tangle of connections which are costly to query in terms of creating joins, then ensuring that queries are performed in the best order. However, one reason normalization has been so popular is that it caters well for transaction processing in many domains such as banking, insurance, retail etc.

2.3 OLTP

By isolating each entity, it is then possible to minimize the disruption to the database when inserting, updating and deleting data, as these operations only affect the entity concerned. In this manner, data relating to one entity is not included in another entity, and there are no duplicate entries. In terms of database processing, this is extremely efficient, and is the main reason the relational database has remained popular for so long. However, the *quid pro quo* is that all the individual entities have to be linked via *joins* and these can be processor-intense, especially if poorly constructed (which is unfortunately all too easy to do and is why relational algebra exists to indicate the best sequence in which to join and query the database tables). These problems become particularly important when considering massive centralized databases, perhaps belonging to an international company, with many millions of transactions.

2.4 Centralized database problems

Two different problems arose with large centralized databases. First, local users wanted more control, and did not want to wait for the central database to process transactions, so they would take a copy of just the material relating to their business area or location, and would put it into their own mini database, and use it for their own queries. The difficulty with this is that from one central database, a myriad of mini local databases would spring up, each being customized in different ways. Trying to analyze the data from a global perspective then became unsatisfactory, as each part of the organization relied on their own mini database for current information, and these were often contradictory. The second problem was that with large volumes of data in huge databases, users would frequently want to carry out analysis across time to measure past performance and to try to predict future action. Obtaining the relevant data from each normalized database turned out to be extremely time consuming, and broad analytical questions such as “how do house prices compare across five states / counties over the last five years?” proved just too time-consuming to answer.

2.5 Analytical databases

W.H. (Bill) Inmon provided the solution to both problems [5] by introducing the data warehouse which is subject –oriented (as opposed to application - oriented) and takes in snapshots of databases over time. It also integrates this data with other relevant information, and crucially, the data in the warehouse is not updated, it is only ever added to, so the *raison d’etre* for normalization (separating entities to minimize update disturbance) is no longer there. Data warehouses thus have a much more flexible architecture, and it is this feature that makes them useful when considering archiving databases, especially those with many complex joins. Data warehouses are often modeled on a star schema [6], which is created by taking the relational model and recasting it with a central fact table. For a retail example, the fact table would contain the items that are numerical, so the number of products sold etc. This fact table is surrounded by dimension tables that *describe* the products. Most importantly, these dimension tables do not have to be normalized, so they can be much more intuitive and user-friendly (and archivist-friendly) and can contain rich detail, especially regarding time. If required, the dimension tables can be a mixture of normalized and unnormalized data, then known as a starflake schema. OLAP is a complementary technique to data warehousing, where these dimensions are used to query the data which is visualized as a cube (see below). For an example of a humanities data warehouse which is based on census data, see [4]. This architecture has now become mainstream, and “ordinary” relational databases can now be set up to be analytical as opposed to transactional, and extra features appeared in the standard query language SQL in the SQL99 version to cater for some of the new types of analytical queries.

This background discussion has thus served as an introduction to the problems of saving accurate relational database models with all their complicated joins, and has also introduced data warehousing techniques which can help to access data across several archived databases. However, before looking at new techniques being used by E-ARK, we need to inspect the current practices regarding database archiving.

3. CURRENT PRACTICE IN DATABASE ARCHIVING

The area of database archiving has been an active issue for already more than four decades. However, the principles of the approaches developed back in 1970s and 1980s have remained more or less the same and could be summarized as the following three step process:

- take a temporal snapshot of the original database
- migrate the snapshot into open formats while changing as little as possible of the original data structures
- when access is required, the database snapshot is reconstructed, based on the data in open formats, in a modern database management system (DBMS).

Of course, such an approach is highly practical and solves the database preservation issue for most interested bodies, including also government and scientific archives. E-ARK especially benefits from the experience of consortium member the Danish National Archive (DNA), which archives all its data in the form of databases, and uses its own version of SIARD: SIARD-DK. As far as the authenticity of the data is concerned, we can also state that the approach of keeping the data models in active use and preservation as close to each other as possible is probably the best possible next to emulation, which in the case of database and system preservation is nowadays regarded as being too expensive for practical purposes for most memory institutions.

The main problems of this approach are related to access and re use. The preservation of near to original data models in different snapshots requires users to go through rather a lot of steps before getting to the actual data they need:

- locate relevant database snapshot(s)
- load it into a database management system
- execute relevant queries, which you might also need to construct yourself after consulting the specific data model

What the users actually need does of course differ in great detail. We can, as an example, look at the main three user groups of public archives: citizens seeking specific information around their rights; government employees needing information for their work; scientists and researchers carrying out large-scale analysis of both historic and current data.

For the first two user groups the most usual need is to find information about a specific entity at a given point in time or over a time period. As an example, this might be about the details and ownership of a piece of property either in 1986 or between 1970 – 2000. The main difficulty in carrying out such queries according to the current logic in database preservation is that instead of using the most obvious search phrase in the archival catalogue, the address of the building, the user needs to look for “the database snapshot which includes details about properties in the 1980s” or, in the case of a longer time period, “for all database snapshots from the 1970s to the 2000s which include details about properties”. In the long term we also have to take into account that the scope of data gathered into single databases can change quite drastically, as an example due to shifting data gathering and management mandates between different government institutions. Taken all the above, we can state that the level of content and technical knowledge needed from the users to carry out even the simplest queries on top of archived databases is just too high, especially when compared to the ease of use of current government service portals.

For researchers the most usual use case is to look into a specific topic, locate ALL data relevant to this topic and analyze it all in common. The problem for such use cases is that the amount of database snapshots to go through is growing too large. As an example, when a researcher wants to carry out analysis on building ownership over three decades, and the archives operate a logic of archiving snapshots every five years, the need is to go through, learn to understand the data models of and execute relevant queries on six different database snapshots. And of course, when the researcher needs information from N different databases then the amount of snapshots to go through would probably be N times 6.

As a first summary we can therefore state, that the main problem to reusing archived databases is that it relies too much on the same “original data model, temporal snapshots” logic which is not sufficiently simple and useful for any of the relevant user groups of archives. Therefore the main aims of the discussions below are to explain a bit what some of the most current technologies in data warehousing and Big Data are able to do in terms of generating user friendly representations of archived data for a variety of use cases.

4. “BIG DATA” TECHNIQUES as used in E-ARK

4.1 Scalable Data Analysis

The E-ARK project is developing a reference implementation for a scalable e-Archiving service. This platform will, besides scalable storage and repository services, also implement advanced search and data analysis strategies. The current prototype setup is based on a scalable architecture involving technologies like Apache Hadoop [http://hadoop.apache.org] for scalable storage and computation, the Apache PIG [http://pig.apache.org] data analytics platform, and the Lily [http://http://www.lilyproject.org] content repository. Being implemented atop data-intensive technologies, the e-Archiving service is capable of storing and efficiently processing large volumes of archived data on multiple computer nodes. Combined with content extraction and information retrieval tools (including for example Apache Tika and SolR), the platform is used to generate content-based and searchable data-sets, enabling users to query information beyond the metadata level. While there are a range of challenges which need to be addressed regarding the processing of complex objects like images and documents at scale [8], a major and so far hardly-addressed challenge is the development of search and analysis strategies across archived relational databases.

4.2 Database Representations

The currently developed E-ARK SIP, AIP and DIP specifications (Submission, Archival and Dissemination Information Packages from the OAIS model) provide built-in support for handling relational databases. This includes archiving databases at multiple layers which can include the primary object, serialized and semantically enriched representations (e.g. based on XML schema), as well as representations that are prepared for later analysis steps. Likewise, E-ARK is supporting access to archived databases at different levels. This includes (a) access based on generic databases that can be loaded and accessed through an Relational Database Management System (RDBMS), (b) access based on aggregated and pre-processed data sets using OLAP-based methods such as denormalization, and (c) access to single records which result from queries which have been executed across multiple archived databases.

4.3 Loading Archived Databases

Archival formats for preserving relational databases have been developed to archive relational data sets independently of the database management system which was used to create, store, and access the data. Tools like CHRONOS and SIARD are capable of exporting database content to disk using an open archival format, thereby preserving the original structure and functional elements at different levels [7]. Communication with the RDBMS is typically handled using standardized APIs (like JDBC) and drivers enabling an application to connect to SQL database and other tabular data sources. Database archiving tools also enable users to load archived data back into a live database system and/or enable users to directly query exported data. The Database Preservation Toolkit [http://keeps.github.io/db-preservation-toolkit/] which is extended in the context of the E-ARK project, supports conversion of live or backed-up databases into preservation formats, the conversion between database export formats, as well as loading preservation formats back into live systems.

4.4 Extracting and Aggregating Data

The serialization and de-serialization of single relational databases and the associated archival formats are essential for developing and implementing the archival workflow and information package specifications developed in E-ARK. A major interest in the context of the E-ARK project, however, is search and access of database records beyond single archived databases. Data warehouses, typically used in the business domain, provide concepts to integrate data from multiple sources into a single platform, for the purpose of data analytics and reporting. Data warehousing makes use of tools enabling Extract-Transform-Load (ETL) processes to derive, extract, and aggregate data originating e.g. from RDBMSs or flat files. The load phase, adds and updates the data sets within the data warehouse, which are modeled according to a well defined structure. Data warehouses typically have a low transaction rate and aggregate historical data, which can be contrasted to the processing transactional data sets. For online analytical processing (OLAP), data is typically organized along the abstraction of data cubes. These enable the user to analyze data and create reports along dimensions like time, location, and other units, required to generate for example Web analytics, or sales statistics. As analytical queries are complex and resource demanding, OLAP systems often need to organize the data sets in a read efficient manner or pre-aggregate them in order to be able to generate timely results.

4.5 A NOSQL-based Approach

While there is ongoing work within the E-ARK project to make use of existing OLAP systems for performing analytical queries across archived databases, the work carried out in context of the scalable e-Archiving service is focused on providing a generic method for searching archived databases which can be implemented on top of the Apache Hadoop software stack. The goal is to support the analysis of a large number of records based on a simple and non-relational model. The approach is supported by the design of E-ARK information packages enabling the preservation of databases at multiple layers, as mentioned before. In this respect, work is dealing with a strategy to generate denormalized versions of archived databases in order to generate flat and non-relational data representations, which can be loaded into a distributed NOSQL store like HBase. Search and analysis will be supported based on full text indexing as well as the

Apache PIG data analytics platform. Current experiments aim at supporting OLAP-like data aggregation and preprocessing based on automatically detected dimensions. Here, dimensions like for example in the form of country/town/zip-code/street are detected automatically by analyzing the relations within a database. This information is then exploited to physically organize the data along the dimensions, enabling efficient range queries and aggregation, and avoiding expensive scans over the entire data store.

5. CONCLUSIONS AND FURTHER WORK

It is central to E-ARK to archive both records and databases, and work to define the definitive E-ARK SIPs, AIPs and DIPs that fully cater for both data types continues, alongside the developments outlined above.

6. ACKNOWLEDGMENTS

This work is supported by the European Commission under FP7 CIP PSP grant agreement number 620998 - E-ARK.

7. REFERENCES

[1] Codd, E.F. 1972. “Further Normalization of the data base relational model” in *Data Base Systems*, (Rustin R., ed.). Prentice Hall.

[2] Connolly, T.M. and Begg, C.E. 2014. *Database Systems: A Practical Approach to Design, Implementation and Management* (6th edition). Addison-Wesley. Harlow, England.

[3] Date, C.J. 2003. *An Introduction to Database Systems* (7th edition). Addison-Wesley Longman. Reading Massachusetts. 5.

[4] Healey, R. and Delve, J. 2007. 'Integrating GIS and Data Warehousing in a Web Environment: A Case Study of the US 1880 census. *International Journal of Geographical Information Science (IJGIS)*. Volume 21. Issue 6. Taylor and Francis. 603-624.

[5] Inmon. 2005. *Building the Data Warehouse*. Wiley. Foster City. U.S.A.

[6] Kimball, R and Ross, M. 2013.*The Data Warehouse Toolkit* (3rd edition). Wiley. Foster City. U.S.A.

[7] Lindley, A. 2013. "Database Preservation Evaluation Report - SIARD vs. CHRONOS," Preservation of Digital Objects (IPRES). 10th International Conference; ed. José Borbinha, Michael Nelson, Steve Knight, 2-6 Sept 2013.

[8] Schmidt, R., Rella, M., Schlarb, S. 2014. "ToMaR -- A Data Generator for Large Volumes of Content," Cluster, Cloud and Grid Computing (CCGrid). 14th IEEE/ACM International Symposium. 26-29 May 2014. 937, 942,