



Preservation Handbook

Relational Databases

Author	Hamish James
Version	1
Date	April 2005
Change History	

Definition

Relational Databases

A database is a collection of discrete data items that are organized according to a specific *data model*, which is typically implemented through the use of a *DataBase Management System* (DBMS) software application. The *hierarchical* and the *network* models were early data models. Many databases are now based on the *relational* data model. This is a formal mathematical model developed by Edgar Codd in the 1970s. *Object-orientated* data models have also been developed, and there are a number of hybrid object-relational database products.

Databases based on the relational model (the model is not fully implemented by any database management software) store data in tables consisting of records (rows) and fields (columns). Primary keys (unique values consisting of the contents of one or more fields in a record) are used to ensure the uniqueness of each row in a table. Rows in different tables can be linked to each other through the use of foreign keys (primary keys for one table, stored in another table).

Database Management Systems

A DBMS is the software that manages the data in a database and provides functionality to add, store and retrieve data held in a database.

- A Relational DataBase Management System (RDBMS) is designed to support a database that uses the relational data model
- An Object Orientated DataBase Management System (OODBMS) is designed to support data organised as objects that also contain the methods that may be used to access and manipulate the data.

A desktop DBMS typically contains user-friendly components for searching and querying the database. Most DBMSs incorporate tools and scripting languages that allow for advanced customization of the way a database is accessed.

Additional Information

- Search Oracle
< <http://searchoracle.techtarget.com/> > Last checked 03/04/2005
- Definition of a database
< <http://webopedia.internet.com/TERM/d/database.html> > Last checked 03/04/2005
- Database Management System
< <http://www.techweb.com/encyclopedia/defineterm?term=dbms> > Last checked 03/04/2005
- Date, C. J., *An Introduction to Database Systems*. Addison-Wesley. Reading, Mass. 1990 (5th ed)



Technical Environment

Relational database management systems run under standard operating systems (DOS, Windows, Mac OS, Unix, Linux etc.) and on standard desktop or server computers.

Common Formats

It is useful to distinguish between desktop RDBMS's, primarily designed to run on desktop computers and support at most a few users, and server RDBMS's, which are designed to support large numbers of concurrent users and transactions, accessing the database from a variety of clients. Common server based RDBMSs include SQL Server, MySQL, SyBase, DB2, PostgreSQL and Oracle. Common desktop RDBMSs include Access, FileMaker Pro and Foxpro. Also of note is McKOI, an example of a Java based RDBMS.

RDBMSs manage a variety of content types, including conventional data types (text, numbers, dates) as well as variable length text and binary data (for images, document objects). In addition to the main data in a database, a database, or its associated RDBMS, may also store report definitions, user forms, query and view definitions, programming language code modules, user and permission information, activity logs and other features. These various forms of information may be held in a single file (e.g. Microsoft Access *.mdb) or in a variety of specialised files (e.g. Microsoft Foxpro: *.dbf; *.ndx; *.cdx; *.prg etc.).

The following table gives a simple overview of different data storage formats used by various RDBMS packages. It does not include separate file formats that may be used for storing query definitions, code and other aspects of a RDBMS.

Format	Related File Extensions	Notes
Microsoft Access	.mdb	Best characterised as a desktop RDBMS. Single file can contain data, database structure, forms, queries, reports and Visual Basic for Applications code. Frequent version changes and a lack of backward compatibility for some features make this complex proprietary format unsuitable for preservation
Xbase (dBASE, Foxpro and other software)	.dbf (data file) .cdx (index file) .idx (index file) .ndx (index file) .dbt (memo fields)	Used with desktop RDBMSs. Originally developed for dBASEIII, this format is also used by Microsoft Foxpro and other RDBMS packages. The format is also supported through data exchange standards such as ODBC.
Claris FileMakerPro	.fp3 .fp4 .fp5 (ver. 5, 6) .fp6 (ver. 7)	Best characterised as a desktop RDBMS. Similar to Microsoft Access; single file contains all aspects of the database. Not suitable for preservation.
Microsoft SQL Server	.mdf (primary data) .ndf (secondary data)	A server RDBMS. SQL Server does not enforce file extensions, the default extensions are listed here. Each database comprises a primary data file and a transaction log, and may also have one or more secondary data files that are referenced from the primary data file. SQL Server is a database server package that has undergone upgrades that are not backwardly compatible in the past. Not recommended for preservation.



MySQL	.frm (table definition file) .myd (data file) .myi (index file)	A server RDBMS. MySQL supports several different storage engines, which affect the internal structure of these files. Logical tables (tablespaces) may consist of more than one physical data file.
Oracle	.dbf (data file) .ctl (control file)	A server RDBMS. Oracle provides database views, that give details of the logical database to physical file mappings.
PostgreSQL	none	A server RDBMS. Tables files are named by their database object ID (ver 7.2 on)

Structured Query Language (SQL)

Most RDBMSs support a version of Structured Query Language (SQL). SQL is defined in a series of ANSI standards (ANSI SQL92, 98, 2000), but most database software does not fully comply with these standards, failing to implement some features and also including non-standard extensions (for example, Microsoft's Transact-SQL, used by SQL Server). Despite this, SQL can be used to define the structure of a database and to query its contents fairly independently of the particular software package being used. This makes SQL an important standard when planning the preservation of a database.

Additional Information

- Xbase: What is Xbase
< <http://www.e-bachmann.dk/docs/xbase.htm> >, last checked 13/10/2004
- SQL Server 2000 Survival Guide
< http://www.akadia.com/services/sqlsrv_programming.html>, last checked 13/10/2004
- Oracle Database Documentation Library
< <https://cwisdb.cc.kuleuven.ac.be/ora10doc/index.htm> >, last checked 13/10/2004
- SQL Databases
< <http://cbbrowne.com/info/rdbmssql.html> >, last checked 14/10/2004



Ingest Checklist

Level 1 (Essential)

- Purpose of database described
- Content of database described
- Content of each table described
- Content of each field described
- Data type of each field recorded
- Primary key for each table recorded
- Foreign keys for each table recorded
- All coding schemes fully described
- No access restrictions that prevent viewing and export of database content

Level 2 (Preferred)

- Purpose and cardinality of relationships (primary key, foreign key) described
- Fields use appropriate data types for their content
- Date data types include 4 digit century (or intended year range of 2 digit centuries is clear)
- Decimal currency data types used only for decimal money systems
- Standardisation rules used for field content are fully described

Level 3 (Best Practice)

- Contextual information provided in user documentation
- SQL setup scripts included
- ER diagram or other appropriate visual model of database provided

Inform Depositor

- Forms, reports and all simple queries will not be preserved
- Non SQL92 or SQL99 compliant queries will not be preserved



Preservation

Significant Characteristics

The key significant properties are the values (content) and type (data type) of each field, column headings (field names) and definitions of relationships. Additional significant characteristics, possibly including the order of rows in a table or meaning of codes used in fields should be determined by consultation with the depositor.

The AHDS does not treat report definitions, user forms, any source or executable code, or user permission and security information as significant.

Technique

This technique is designed for databases that are completed and will not be updated, or will be updated infrequently. It is not appropriate for active, transaction orientated, databases

1. Export each table to an ASCII or (preferred) UNICODE delimited text file. Use the tab or pipe character as the field delimiter, *unless these characters occur in the data*, in which case another delimiter character should be selected. Do not surround textual values in quotes. Memo fields should normally be kept with the rest of the table content, unless they are very long in which case they can be stored as separate text files.

When exporting memo fields, check for end-of-line codes and remove these or represented them in a different way (e.g. by including a <EOF> tag in the text) to avoid erroneous end-of-record markers in the delimited text.

2. Binary data fields (images, audio, documents etc.) should be extracted and stored separately according to the guidance for their own digital resource type. These files should be linked to the main exported table by way of an additional field that contains the name of the file created from the binary data field.
3. If not adequately defined in the depositor-supplied documentation, the database schema should be defined in AHDS created documentation. Ideally, the schema should be presented as ANSI/ISO SQL Data Definition Language statements (SQL92 or SQL99), but E-R or UML diagrams with lists detailing tables, fields and relationships are acceptable.
4. Export the SQL definition of queries as appropriate to an ASCII or (preferred) UNICODE text file. Record the SQL standard
5. If necessary, preserve the original order of rows in each table by adding an extra field filled with sequential numbers.

Validation of Exported Data

- Check number of columns and number of rows
- Check number of characters in memo fields
- Check text fields for unknown characters (i.e. "□")
- Check precision and format of numbers
- Check precision and format of dates



Problems and Issues

Conversion Between Database Data File Formats

Conversion between different database data file formats can be difficult because of differences in the data types and data type domains supported by various RDBMSs. In general, it is best to export data directly from the original RDBMS used by the depositor to a software neutral format, such as delimited text. In the case of server orientated RDBMSs, the physical data files can be difficult to use if they are simply copied from their original location, and the depositor should be strongly encouraged to export the data before depositing it with the AHDS.

Additional Information

- Data Structure Differences between Access and SQL Server
< <http://www.databasejournal.com/features/mssql/article.php/1490561> >
Last checked 03/04/2005