

Flat and relational structures for a terrestrial vertebrate database

P C Withers

Department of Zoology, University of Western Australia, Crawley WA 6009
email: philip.withers@uwa.edu.au

Abstract

A cooperative effort among a number of governmental agencies, industry including private environmental consultancies, and other interested parties such as university academics and the Royal Society of Western Australia, could enable the establishment of a regional terrestrial vertebrate database for Western Australia. The form of such a fauna database is not obvious because of conflicts between ease of data entry for those individuals collecting the raw data, optimal strategies for storing and accessing a large amount of data by database managers, and expeditious and aesthetic accessing of the data by end-users. A spreadsheet is a powerful tool for data entry, manipulation and summary, and is widely available and used by biologists. Such a simple flat database structure (e.g. a pre-formatted Microsoft Excel spreadsheet) has advantages in ease of data entry, widespread availability of software, and minimal training requirements for data-entry operators. Limitations of spreadsheets include redundancy of repeated cells, the opportunity for mistakes in redundant data entry, and the potential for many empty cells. For data storage and access, a more complex relational database with pre-designed relational tables (e.g. Microsoft Access database) has advantages of optimal data structuring and sophisticated search capabilities compared to spreadsheets. I suggest that the optimal strategy for a regional terrestrial vertebrate database is a combination of a flat spreadsheet structure for data entry and submission, then conversion to a relational database for storage, management, and end-user access.

Keywords: flat database, spreadsheet, relational database, terrestrial vertebrates, regional fauna

Introduction

Currently, many fauna surveys are undertaken each year in Western Australia for a variety of reasons, including government-sponsored regional surveys (e.g. Department of Conservation and Land Management, Western Australian Museum), private industry surveys (e.g. as part of Environmental Impact Assessment submissions for the Environmental Protection Authority), and as independent scientific studies (e.g. university-based research). As more fauna survey data are collected every year, it is becoming increasingly obvious and imperative that some structure needs to be established to centralise, store, manage and provide future access to these data. Hence the increasing local interest in the establishment of a terrestrial fauna database (e.g. Regional Fauna Database Workshop organised by the Royal Society of Western Australia, 5th April 2002 and this issue of the Journal of the Royal Society of Western Australia).

What I address here is the possible structure for such a regional terrestrial vertebrate database. This requires a very general overview of various structures of databases from a functional viewpoint to bridge the gap from simple data entry via spreadsheets to the structure of complex "relational" databases. Most biologists are familiar with "flat" spreadsheets, and routinely use such programs (e.g. Microsoft Excel[®]) for data entry, structuring, and statistical and graphical

analyses. A relational database (e.g. Microsoft Access[®]) can more economically and efficiently arrange large amounts of data, but at the cost of increased complexity in design and management. Few biologists need to use a relational database, and therefore generally do not have the required knowledge or skills.

I describe here in general terms how a flat spreadsheet works and why it may be the method of choice for entry of data, and how relational databases are much more powerful and complex and why they may be the method of choice for storage of a centralised database for fauna data. The general principles of spreadsheet and relational database programs are similar regardless of the computer platform (i.e. PC, MAC, Linux) and the particular spreadsheet or relational database program. I will use Microsoft Excel[®] as an example of a spreadsheet program, and Microsoft Access[®] as an example of a relational database program, as these are generally available to biologists. Microsoft's user's guides for Excel (Anon 1994a) and Access (Anon 1994b) provide more detail on use of these programs, and examples.

"Flat" Spreadsheet Databases

Most biologists are familiar with and use "flat" spreadsheets (e.g. programs such as Excel, Lotus 123, Quattro Pro, StarOffice) for routine data entry, manipulation and analyses, and as a simple database. A spreadsheet is a "flat" database (Fig 1) because its data are arranged as a 2-dimensional table, with columns (A, B, C, etc) and rows (1, 2, 3, etc). Each cell of the table is uniquely identified by its column and row (e.g. C11), as well as the worksheet name (e.g. trapdata) and file name (e.g. Trapping Data.xls). Each column can be given a



Figure 1. General 2-dimensional structure of a spreadsheet, showing columns (A, B, etc), rows (1, 2, etc), the workbook name, active sheet tab, tab scroll button, and vertical and horizontal scroll bars, illustrated by an example of pit-trapping data in a Microsoft Excel spreadsheet.

header name (in row 1), and that column is called a field; each row is a record entry. Each file, or “workbook”, can contain a number of different tables (“worksheets”). Information can be linked between worksheets (and even between different workbooks). For example, a list of species names can be used as a drop-down list (see below) for data entry, thus facilitating entry of species names and avoiding errors. Nevertheless, the essential structure of a worksheet table is a simple 2-dimensional grid of values. A spreadsheet approach for summary of fauna sampling data from pit-trapping (e.g. Fig 1) could include fields such as dates for trapping, a pit identification number, identity of specimens captured, sex of individual, mass, etc. Additional information for use in a wider regional context could include a study site identification, GPS locality, and vegetation, soil, geographical and geological information.

In a flat spreadsheet, each record (row of information) usually needs to contain a cell entry for each field. Any information that is the same between rows, such as locality, GPS reading, or pit trap grid identity, needs to be entered individually in each row. This repetition is redundant, it requires computer memory for storage, and even more seriously it is prone to data entry errors since misspelling of names or incorrect entry of values leads to confusion. For example, if a data-entry operator occasionally misspells the species name *Diplodactylus assimilis* as *Diplodactylus assimilus*, then this species would appear as two separate species in any sorted list, pivot table, query or report. Such an error might be fairly obvious to a biologically-experienced database user, but many input errors would not be so obvious. For example,

entering a UTM coordinate of 11 4 23 818 E as 11 4 32 818 E is not an obvious error, and might never be identified as an error once it was miss-entered. Such data miss-entry can be avoided by copying to fill a series of redundant cells, or using a drop-down list for entry of fixed values in a specific field. For example, a list of species names in one worksheet (or even in another workbook) can be used to define a list, which can then be used to select names from the dropdown list, avoiding any possibility of typing errors.

Another limitation of a flat spreadsheet is that often some fields don't relate to all records. For example, a field for entry of tail break occurrence might be useful for some lizards but is not applicable to other lizards, frogs, birds or mammals. A field for observation of lactation might be relevant for female mammals but not males or other vertebrates. To minimise empty cells, different worksheets might be used for different types of animals (e.g. separate worksheet for amphibians, reptiles, birds and mammals).

Simple forms are available in Excel for data entry, but these forms (Fig 2) are not customisable and are relative primitive compared to more sophisticated forms that can be designed in a relational database (such as Access) for data entry. A spreadsheet data form is a dialog box that provides a convenient form for entry or display of one complete row of information, or record, at one time. A form can be used to add a new row of data, delete a row of data, move to previous or subsequent records, and set criteria for data entry, but a form does not provide access to dropdown lists. Quite sophisticated data entry coding and programming with macros can be accomplished in spreadsheets. A macro is a series of commands that can be run in a spreadsheet whenever a repetitive task needs to be performed.

Using a simple flat spreadsheet approach for fauna survey results would involve considerable redundancy and many empty cells, but hopefully potential for miss-entry of data could be minimised by use of drop-down lists and careful checking by the data-entry operator. Provision of appropriate drop-down lists and templates for organisation of data would facilitate preparation of spreadsheets in a standardised format. However, storage and retrieval of information from a flat spreadsheet is not optimal; even a simple relational database is better for this. Nevertheless, I suggest that despite its limitations (redundancy, empty cells), a spreadsheet would be the best format for routine data entry by individuals, and transfer to a centralised database manager.

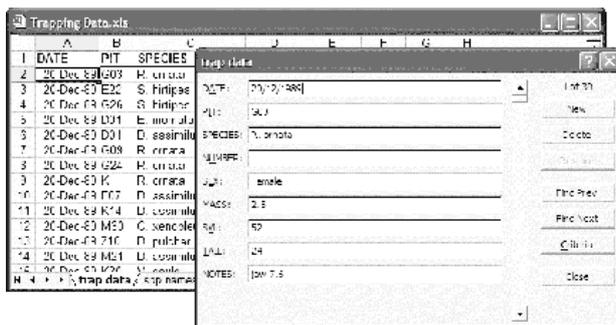


Figure 2. Example of a form used for data entry in a Microsoft Excel spreadsheet.

Relational Databases

There are a variety of more complex designs for databases than spreadsheets, which allow relationships between entities (Kroenke & Dolan 1988; Robinson 1989; Date 1990; Gault 1994; Viescas 1993). The hierarchical database model provides limited relationality, the network database model provides more relationality, and the relational database model provides the most powerful description of relationships between entities. In a hierarchical database model, a set of records can have links to other

record types, but these links can only be “one-to-many” (e.g. a pit trapping grid may contain a number of different pit traps, but each pit trap is only found in one particular grid). A network model can have “many-to-many” relationships (e.g. a pit trap may capture a number of different species, and each species can be captured in a number of different pits). A relational model replaces the structural complexity of hierarchical and network models by using flat files (that have been normalised) and providing operations for manipulating these flat files.

A relational database has a number of different tables that store related information. The arrangement of data amongst tables should minimise redundancy, likelihood of data entry errors, and numbers of empty cells. It does not necessarily reflect the structure of the data as collected, or as used (e.g. as reports). A relation is a 2-dimensional table that differs from a general flat file in that;

- each column has a distinct name;
- all data items in a single column are of the same type;
- all rows are distinct — there can be no identical, duplicate rows; and
- the order of rows and columns doesn't affect the information content of the file.

Three kinds of relationships can occur in a relational database, and it is important that the correct relationship is used. A one-to-many relationship is common. Here, a record key in a table can match more than one record in another table, but a record in the second table can only match a single record in the first table. For example, a pit trap grid (e.g. grid A) might have a number of separate pit traps (pits A1 to A36), but an individual pit trap (e.g. A1) occurs only in a single grid. A many-to-many relationship is also fairly common in relational databases. Here, a record in one table can have more than one matching record in a different table, and a record in the second table can have more than one match in the first table. A one-to-one relationship is less common; here a



Figure 3. Example of links between relations (normalized tables).

record in one table has only one matching record in another table, which only has one matching record in the first table. Thus, a relational database can contain a number of tables linked by various relationships.

For a regional terrestrial vertebrate database, a number of data tables would be required, and relationships between them could be one-to-many, many-to-many and even one-to-one if necessary. A simple example of relationships for a pit-trapping database has a number of one-to-many relationships (Fig 3). This simple example illustrates the basic differences between using a flat spreadsheet and a relational database, and the relative advantages of the relational approach. An Access database could contain different tables for general field trip information, trapping data, and pit trap data. Establishment of relationships between the various tables allows efficient storage of information and retrieval of information by queries or as reports.

Entry of data in a relational database program, such as Access, can be facilitated by the use of forms (Fig 4). This is a major advantage of Access over forms in

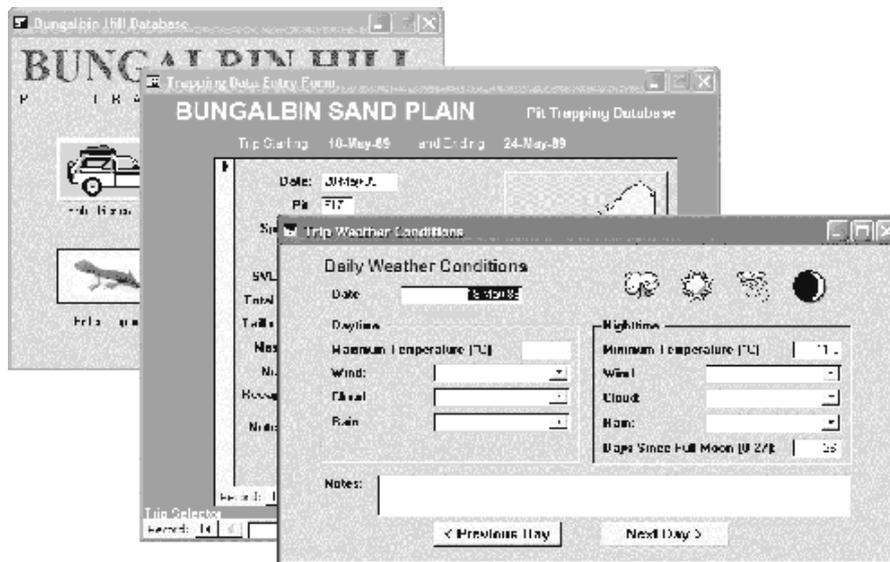


Figure 4. Examples of forms used for data entry in a relational database program (Microsoft Access).

Excel, where data entry via a form is possible but the form is much less flexible than in Access. Forms in Access can be quite complex, with fields in particular arrangements with graphics to optimise the data entry process.

Data can be readily retrieved from a relational database using queries and reports. A query is a question about data stored in a relational database. A select query (Fig 5) allows viewing and analysis of data from one or more tables. Other types of query include crosstab, action, union, pass-through and data-defining queries. There is considerable flexibility in the design of particular queries and they can be quite powerful. Reports are information selected, retrieved and organised to fit specific requirements. They allow presentation of subsets of the data in a meaningful organisation, often for presentation. Forms allow graphical viewing of all of the information for a record at a time, queries allow selection, analysis and viewing of specific sets of data, and reports organise and print data for formal summary and presentation.

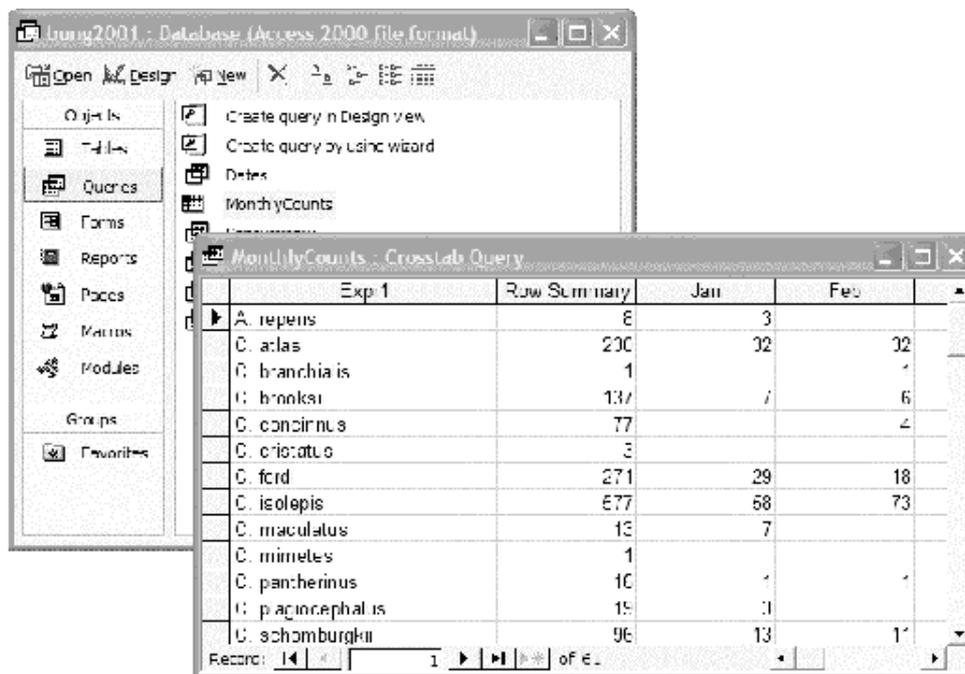
Clearly, relational databases can be a very powerful tool for data entry, manipulation and summary but they are more complex and less widely used by scientists than flat spreadsheets. They require considerable skill in designing the relations for optimal functioning, and can be difficult to modify during the inevitable long-term development that would take place for a large, sophisticated database. Nevertheless, I suggest that any large regional faunal database would have to be organised as a relational database, even though the best format for routine data entry by individuals would be a spreadsheet format despite its limitations (redundancy, empty cells).

Strategy for a Terrestrial Vertebrate Database

I suggest that any large regional terrestrial vertebrate database should be organised as a relational database. Then there would be three technical challenges;

1. Providing suitable flat spreadsheets with standardised drop-down lists, templates and structures, for routine data entry (e.g. using Microsoft Excel). This simple and widely available spreadsheet technology would be a suitable method for submission of data to be deposited automatically into a centralised database. From the perspective of a contributor to the database, data entry must be simple enough that it will not deter individuals un- or semi-skilled in database operation from contributing data. A simple spreadsheet data entry system could even be required for CALM licensing returns, thus making all regional data potentially available to a centralised database.
2. Designing an optimal relational database for storage of information and subsequent access to information through queries and reports (e.g. using Microsoft Access); this more complex relational approach would require centralisation, staff with the appropriate database skills, and access to sophisticated technology for internet accessibility.
3. As seamlessly as possible facilitate transfer of information from spreadsheets to the relational database; this may well be the biggest technical challenge of the three.

From the perspective of a paying end-user, a high quality database and access service would have to be



Exp 1	Row Summary	Jan	Feb
A. repens	8	3	
C. atlas	200	32	32
C. branchialis	1		
C. bronksii	137	7	6
C. concinnus	77		4
C. cristatus	3		
C. fcid	271	29	18
C. isolepis	677	58	73
C. maculatus	13	7	
C. mimetes	1		
C. pantherinus	10		
C. pacincephalus	15	1	
C. schomburgkii	96	13	1

Figure 5. Example of use of queries in Microsoft Access, to extract information from a database. Here, a query for monthly counts provides a summary of the total numbers of each species captured in each month, during the multi-year trapping program.

provided, to justify a cost for database access. Data retrieval by queries must be simple enough that it will not deter use, and provide a sufficiently attractive service to justify the costs that would be required to maintain the regional database.

Establishment of a terrestrial vertebrate database raises the issue of data verification. It is essential that some consideration be given to the extent of data verification (especially species identification) from the perspective of data contributors, database managers and end-users. Even checking of data entry files by data contributors for errors would be time-consuming and tedious, but necessary. The highest level of verification for species identification would be voucher specimens lodged in the Western Australian Museum for confirmation of identity by recognised experts in the field. An intermediate level of verification could be consistency of identification with species range distributions. Some level of data verification would be essential to providing a justification for a cost to end-users for accessing the database.

Acknowledgements: I am grateful to Dr AR Roberts for designing the Access database for pit-trapping data at Bungalbin, which was used for examples in this paper. I also thank Dr GG Thompson for his enthusiastic organisation of the workshop, and for reading this manuscript.

References

- Anon 1994a Microsoft Excel: User's Guide. Microsoft, Redmond.
- Anon 1994b Microsoft Access: User's Guide. Microsoft, Redmond.
- Date C J 1990 An Introduction to Database Systems. Addison-Wesley, Reading.
- Gault F D 1984 Database management systems for science and technology. In, Database Management in Science and Technology: A CODATA Sourcebook on the Use of Computers in Data Activities (eds J R Rumble & V E Hampel), pp 39-73. North-Holland, Amsterdam.
- Kroenke D & Dolan K. 1988 Database Processing: Fundamentals, Design, and Implementation. Science Research Associates, Chicago.
- Robinson H 1989 Database Analysis and Design. Chartwell-Bratt, Sweden.
- Viescas J L 1993 Running Microsoft Access. Microsoft Press, Redmond.