**Databases: The Table**

# 1 Introduction

Introduce different scenarios as to how we come to use a database

- in industry, data collected from manufacturing process in databases and interested in the production process and improving, e.g., yield.

- a clinical trial where data is gathered on observational units for a variety of different purposes

  Clinical trials study how well a new drug or treatment works, and in order for the Food and Drug Administration (FDA) to approve the drug, there must be convincing evidence that the treatment is safe and effective. therefore it is critical that accurate, reliable, and secure data are kept on the patients involved.

  Clinical trials involve many people, including doctors and nurses at multiple remote locations who monitor the health of the patient, lab workers who process lab tests, social workers and health care professionals who maintain contact with the patients, a researcher team, including doctors and statisticians who follow the progress of the trial, analyze the data, and report results, data managers and programmers who collect and clean data, and managers who oversee the trial. These team members must share ideas, files, information and knowledge on a real time basis

  Clinical trials involve large numbers of patients over long periods of time. Several kinds of information need to be kept on a patient, including personal data such as name and address, lab results, and who is the attending physician. After an initial interview and once a participant agrees to join the study, a baseline visit gives information against which to measure future changes. The participant receives the

test drug, a comparison drug, or a placebo, and visits the physician's clinic on numerous occasions for check ups and additional lab work to assess the effects of the treatment and the health of the patient. Typically, patients return to the clinic at regular time intervals, but patients may miss appointments, drop out of the study, and otherwise have varying numbers of checkups. Clinical studies also have enrollment windows during which patients can join the study, and as the study progresses, patients may drop out before completion of the study. Live data – monitor the results for ethical stop of trial when treatment has been shown to be far superior to another.

- information is gathered for tracking inventory and sales in Wal-Mart. Different groups decide to "mine" it for relationships to see if they can improve the Supply Chain Network (SCN), marketing strategies, etc.

- you are starting a study with different types of data (images, numbers, files, etc.) and a large quantity of it (e.g. from a collection source such as a computer network). Rather than using some ad hoc solution to manage the data without knowing precisely how you will use it, you choose to keep your options open and to use a general database system to manage the data. S-Net is an example.

Cover topics such as

- imposed on users because of corporate/institutional approaches to gathering and managing data

- meta-data

- synchronization

- client-server computing

- security

- performance (specialized)
- connections to data frames and statistical data "models"
- live data

Other advantages of a database:

Synchronized access to data

Propagation and standards enforced when updates, deletions, and additions made

Centralized data for backups

Often times we are forced to use a database because that is how it the data are made available to us.

## 2 The Basic Relational Component: The Table

The basic conceptual unit in a relational database is the two-dimensional table. A simple example appears in Figure 1, where the table contains laboratory results and test dates for three patients in a hypothetical clinical trial. The data form a rectangular arrangement of values similar to a data frame, where a row represents a case, record, or experimental unit, and a column represents a variable, characteristic, or attribure of the cases. In this example, the three columns correspond to a patient identification number, the date of the patient's lab test, and the result of the test, and each of the eight rows a specific lab test for a particular patient. We see that patient #101 received tests on four occasions, patient #102 was given three tests, and the third patient has been tested only once.

The terminology used in database management differs from a statistician's vocabulary. A data frame or table is called a relation. Rows in tables are commonly called tuples, rather than cases, and columns are known as attributes. The degree of a table corresponds to its number of columns, and the cardinality of a table refers to the number of rows. Statisticians

| ID | Test Date | Lab Results |
|---|---|---|
| 101 | 2000-01-20 | 3.7 |
| 101 | 2000-03-15 | NULL |
| 101 | 2000-09-21 | 10.1 |
| 101 | 2001-09-01 | 12.9 |
| 102 | 2000-10-20 | 6.5 |
| 102 | 2000-12-07 | 7.3 |
| 102 | 2001-03-13 | 12.2 |
| 103 | 2000-02-16 | 10.1 |

Figure 1: Lab results for 3 patients in a hypothetical clinical trial. Reported here are the patient identification number (ID), the date of the test, and the results. The results from patient #101s test on March 15, 2000 are missing.

| Object | Statistics | Database |
|---|---|---|
| Table | Data frame | Relation |
| Row | Case | Tuple |
| Column | Variable | Attribute |
| Row ID | Row name | Key |
| Row count | size | cardinality |
| Column count | dimension | degree |

Figure 2: Correspondence of statistics descriptors to database terms for a two-dimensional table.

usually refer to these as the dimension and the sample size or population size, respectively. Table 2 summarizes these various table descriptors.

### 2.1 Entity

An entity is an abstraction of the database table. It denotes the general object of interest. In the example found in Figure 1, the entity is a lab test. An instance of the entity is a single, particular occurrence, such as the lab test that patient #102 received on the 7th of December 2000. A natural follow on to the idea that a case is a single, particular occurrence of the entity, is that the rows in a table are unique. To uniquely identify each row in the table, we use what is called a key, which is simply an attribute, or

a combination of attributes. In our clinical trial (Figure 1), the key for the table is a composite key made from the patient identfication number and test date. (We assume here that patients do not have more than one lab test on the same day). When we look over the rows in the table, we see that the test dates are unique, yet we do not use the single attribute test date for the key to this table because although we have not observed two patients with the same test date so far, the design of the study allows patients to receive lab tests on the same day.

In the S language, the row name of a data frame serves as a key. Although, it does not have the flexibility of being defined in terms of a composite set of variables, the values of the row name play a similar role to the key in a database. Most importantly, row names provide convenient means for indexing data frames and identifying cases in plots.

## 2.2 Meta Information

Relational databases allow us to define data types for columns and to impose integrity constraints on the values in the columns. These standards can be enforced when updates are propagated and when new data are added to the database. As statisticians, we know that our analysis of the data is only as good as the data. If the data are riddled with errors and missing values then our findings may be compromised. The database management system helps maintain standards in data entry. In addition to checking that data being entered match the specified type, the database management system offers additional qualifiers for attributes. For example, the values of a variable may be restricted to a particular range or to a set of specified values; default values may be specified or values may not be allowed to be left empty (NULL); and duplicate records can be kept out of the database.

### 2.2.1 Data Types

As with data frames, all values in one column of a database table must have the same data type, but the columns may be of different types from each other. In Table 1, the patient ID is a 4 byte integer; the date of the lab test has type DATE, i.e. year-month-date; and the lab results are 4 byte floating point representations. Databases offer a great variety of data types ranging from the typical exact and approximate number representations, such as integer and floating point, to booleans, character strings, and various time formats. Table **??** contains a list of general data types. (Some may not be supported by all relational databases.) Also, application specific vendors may provide specialized data types, such as the MONEY type in financial databases, and the BLOB type (a binary large object) for images. In comparison, R offers the same four basic data types integer, numeric, logical and character vectors, but it does not have the variety in size, e.g. it stores integers in 8-byte format only.

The categorical variable represents an important kind of information; it is qualitative in nature and takes on a finite number of numeric or character values. Categorical variables need to be treated specially in many statistical procedures, such as analysis of variance and logistic regression. R represents this type as a "factor" and the computational procedure for say an ANOVA automatically handles factors appropriately. The comparable column in a database table would be either an integer or character data type where the values are restricted to a predefined, finite set.

Time data provide another example of specialized data types that need to be addressed, i.e. in time series analysis. Both databases and R have three basic types of time: a date, a time interval, and a time stamp. The time stamp refers to system time. Time stamps are critical to database integrity, for the system time keeps multiple users of the database from updating the same record concurrently. Dates and time stamps in R are stored in one of two basic classes: POSIXct, which represents as a numeric vector

| Data Type | Explanation |
|---|---|
| integer | 4 bytes |
| small integer | 1 byte |
| big integer | 8 bytes |
| numeric | numeric(p,s) p = precision, s = scale |
| decimal | same as numeric except that s is a minimum value |
| real | single-precision floating point |
| double precision | double-precision floating point |
| float | float(p) p = precision |
| character | char(x) x = number of characters |
| character varying | varchar(x) x = maximium number of characters |
| bit | bit(x) x = number of bits |
| bit varying | bit(x) x = maximum number of bits |
| date | year, month, and day values |
| time | hour, minute, and second values |
| timestamp | year, month, day, hour, minute, and second values |
| year-month interval | duration in years, months, or both |
| day-time interval | duration in days, hours, minutes, and/or seconds |

Figure 3: A list of general data types for databases. They may not be supported by all relational databases. Note that the time and timestamp types may include a time zone offset.
**??**

the (signed) number of seconds since the beginning of 1970; and POSIXlt, which is a named list of vectors each representing a part of the time such as the year, month, week, day, hour, minute, and second. POSIXct is more convenient for including in data frames and using in statistical procedures, whereas POSIXlt is useful when indexing particular days, hours, etc. and displaying time in graphics. Time intervals can be computed by subtraction of two date objects of the POSIXct class. As with databases, the POSIXlt and POSIXct objects may include a time zone attribute, if not specified, the time is interpreted as the current time zone.

These S time classes are handy for they give a default character format for displaying time, i.e. Fri Aug 20 11:11:00 1999, and they provide an easy means to change this format. Database management systems similarly provide functions to manipulate and display dates and times, but the implementation varies. In addition, some include checks for compatibility between begin and end dates, arithmetic on dates, allowing a date of eternity, i.e. 9999-12-31 23:59:59.999999; and date extraction functions to pull out components from a date such as the hour or day.

### 2.3 Missing Values

Statisticians take great care when handling missing data: they impute, infer, or otherwise fill in these values when possible; they check for bias introduced by missing values; measure the impact of the missing data; and on occasion resort to examining original records in search of lost data. Researchers have developed statistical procedures and mathematical theory to back-up these procedures for imputing missing values. In practice, statisticians need software to provide consistent and meaningful ways to deal with missing values. In R, vectors may contain the special value of NA to denote Not Available. Its counterpart in the database table is NULL.

The use of NULL is discouraged in many guides on databases because unexpected results may be obtained when operating on columns that con-

tain NULL values. For example, logical operations on a field that contains a NULL will not result in TRUE or FALSE but in NULL, which may inadvertently lead to data loss with an improperly worded logical expression.

It is important to know how NULL values are handled when they are passed from a database table to a host program. In databases, arithmetic operations on columns that contain NULL values will result in NULL, but aggregate functions such as the average function discard NULLs and compute the average of the known values. S handles NAs in a similar fashion, with three important differences. First, care has been taken to include meaningful ways of handling NAs that reflect the nature of the particular statistical procedure. For example, the default procedure in a cross tabulation that yields counts of cases for each factor level excludes the NA as a factor level. Second, many procedures allow the user to easily change the default handling of NAs. For example, in the simple mean function, the default procedure includes NA so the presence of one NA in a vector will result in an NA for the mean, but the user may specify via a parameter that the NAs be excluded in the calculation. Finally, in an arithmetic computation, R distinguishes between operations that results in overflow (+Inf), underflow (-Inf), or a computational error (NaN). Most database management systems represent all of these by NULL.

## 2.4 Transactional Data

Typically the data in a database continuously evolves as transactions occur, new tuples get inserted, old records deleted, and others updated as new information becomes available. The data are live, meaning that actions on the database tables need to be regularly re-run in order to get the latest results. Further, the changes made by one user are visible to other users because of the centralized storage of the data. This concept of continuously changing data differs dramatically from R's functional programming model. R does not easily support concurrent access to data. Instead, it supports

persistence of data; data objects are saved from one session to the next, and the statistician "picks up" where he left off in the previous session.

## 2.5 Summary: Data frames vs. Database Tables

We summarize here the basic features of database tables and how they compare to data frames in S.

- The database table is similar in form to the data frame, where rows represent cases and columns represent variables. The columns may be of different data types. All data in one column must be of the same type.
- The database provides built-in type information and validation of the fields in the table. The database offers a great variety of data types and built-in checks for valid data entries.
- Tables have unique row identifiers called keys. Keys may be composite, i.e. made up of more than one attribute. The S language uses row names to uniquely identify a row in a data frame.
- The general purpose missing value in a database is the NULL. Care must be taken with logical, arithmetic, and aggregate operations on attributes that contain NULL values as unexpected results may occur. Unlike with S, many databases do not distinguish NA from overflow, underflow, and other computational errors.
- The database table contains live, transactional data; we get updated results when we re-run the same query. The S model supports persistence of data for the individual user from one session to the next.