Relational Database Design: Rules of Thumb

Thomas H. Grayson
23 January 2002

Database Design Rules of Thumb

Keep data items atomic (e.g., first and last names are separate). Concatenating columns together later on-the-fly is generally easy, but separating them is not.

- *What is an example of where parsing subfields from a column may go awry?*
- *When might you want to include the combined fields in a column anyway?*

Define the primary key first. Use a descriptive name (PARCEL_ID, not ID)

In fact, use descriptive names that give a new user a decent chance of guessing what they mean for *all* your columns! (E.g., use PARCEL_COUNT rather than PACT)

Use a single column for the primary key whenever possible; multi-column primary keys are appropriate for many-to-many relationships

Use lookup tables rather than storing long values

Use numeric keys whenever possible *(What about ZIP codes?)*

Avoid intelligent keys (exception: lookup tables)

Avoid using multiple columns to represent a one-to-many relationship (e.g., columns such as CHILD1, CHILD2 in a table called PARENT rather than putting the children in a separate table.

For readability, use the primary key name for foreign keys unless the same foreign key is used multiple times in the same table (e.g., state of work and state of residence for a person might both be foreign keys thatreference a table of states)

Do not include two columns whose values are linked together (e.g., county name and county ID) unless one of the columns is the primary key of the table

Avoid allowing NULL values in columns that have a discrete range of possible values (e.g., integers between 1 and 10, inclusive)
(not applicable to DBF files, which do not support NULLs)

Avoid using multiple tables with similar structures that represent minor variants on the same entity (e.g., putting Boston parcels and Cambridge parcels in separate tables). *Why is this rule often hard to practice with GIS?*

Plan ahead for transferring data to a different database. For example, you may want to move data from Oracle to DBF, or Microsoft Access to Oracle.

- Avoid column names with characters with other than UPPER CASE letters (A-Z), digits (0-9), and the underscore (_). Other characters may not be accepted by a database. Some database systems may be case sensitive with regard to column names, while others are not.
- Keep your column names relatively short. Different databases support different numbers of characters in column names (e.g., 30 for Oracle, 64 for Microsoft Access, 10 for DBF). Try to make column names differ in the first few characters rather than at the end to avoid column name duplication if the names are truncated during the conversion process (e.g., use COL1 and COL2, not LONG_COLUMN_NAME_1 and LONG_COLUMN_NAME_2).

> *Note that keeping column names short may be at odds with keeping your column names meaningful for neophytes. Be aware that you are making a tradeoff!*

Remember that these are rules of thumb, *not* absolute laws! Bend the rules if you must but have a justification for your decision. The limitations of a GIS software package often provide a good reason.