John Paul Welsh Database Systems Lab 2 March 1, 2015

Superkey, Candidate Key, and Primary Key

A superkey is a column or group of columns that can uniquely identify a row in a table. It may include more columns than its necessary. A candidate key is a minimum-length superkey that can uniquely identify a row in a table with as few columns as possible. There can be "ties" between multiple candidate keys, in terms of number of columns. A primary key is the candidate key that you choose to be the most important. Breaking a tie between candidate keys, it will be used from then on as the go-to way to get at a unique row.

Data Types

Example data types that you can have in a database are:

- integer (a positive or negative whole number)
- boolean (true or false)
- char (a set number of characters)
- text (a variable number of characters)

For an example database, consider one designed to keep track of information about a hypothetical musical group. A sample table in this database could be called "members," and its columns (and the types of those columns) would be the following:

- lastname (text)
- firstname (text)
- age (integer)
- homestate (char, length 4)
- vocalrange (text)

The first five of these columns would be non-nullable, because these fields are all information that any member of the band would have. The sixth column, "vocalcategory," can be nullable because some members of the band may not be vocalists, and therefore would not have a category to speak of (such as alto or soprano).

The Relational Rules

The first relational rule states that a table should not have two values in one cell. An example of a situation that breaks this rule is if, under a column called "children," there is more than one child's name listed in the same cell for a given row. This rule is important because a database user ought to rely on getting the same number of results from a query no matter the context, and having one result per cell is the simplest and most reliable way to do so.

The second rule states that one should make queries based on what the data is that you're looking for, and not where in the table it is located. This is because the data in tables are sets, and sets have no intrinsic ordering, so asking for information about the third row in a table may not give the same answer each time.

The third rule states that no two rows should be exactly the same in a given table. This is important because it allows us to make correct reasoning about the table's data. Having two identical rows would lead it to chance what the answer to a query would be, and this destroys the "correctness" of the relational model.