15.12: THREE KINDS OF KEYS

Contributed by Chuck Severance
Clinical Associate Professor (School of Information) at University of Michigan

Now that we have started building a data model putting our data into multiple linked tables and linking the rows in those tables using *keys*, we need to look at some terminology around keys. There are generally three kinds of keys used in a database model.

- A *logical key* is a key that the "real world" might use to look up a row. In our example data model, the `name` field is a logical key. It is the screen name for the user and we indeed look up a user's row several times in the program using the `name` field. You will often find that it makes sense to add a `UNIQUE` constraint to a logical key. Since the logical key is how we look up a row from the outside world, it makes little sense to allow multiple rows with the same value in the table.
- A *primary key* is usually a number that is assigned automatically by the database. It generally has no meaning outside the program and is only used to link rows from different tables together. When we want to look up a row in a table, usually searching for the row using the primary key is the fastest way to find the row. Since primary keys are integer numbers, they take up very little storage and can be compared or sorted very quickly. In our data model, the `id` field is an example of a primary key.
- A *foreign key* is usually a number that points to the primary key of an associated row in a different table. An example of a foreign key in our data model is the `from_id` .

We are using a naming convention of always calling the primary key field name `id` and appending the suffix `_id` to any field name that is a foreign key.