Interactive - 10 - add a index on the name table

We have already added indexes but we need to make this process explicit.

The most common indexes are based on special kind of tree called a b-tree.

It is a very wide tree that allows us to get to the data stored on disk quickly. This is not the only kind of index.

There are indexes for other things that are built into PostgreSQL.

Let's create an index on the real_name field.

The general form for the create index looks like:

```
create [unique] index <index-name> on <table-name> (
    <colunm-name-1>,
    <col-2>,
    ...
);
```

So...

```
CREATE INDEX name_list_idx1 on name_list ( real_name );
```

Now when we search for a name it will use the index to help to find the names in the table. If we have a million names in many blocks of data it will only take a few blocks of index search to get to the row with the name we are interested in.

The disadvantage is that we have to have the exact name to search on. Do you remember our lower-case name, `bob true` from the 2nd homework. If we look for that we will not find it at all because we fixed the name to be `Bob True`.

```
select * from name_list where real_name = 'bob true';
```

Indexes are important because this is the primary way that we can influence the performance of a database. If we add the word "unique" then it also grantees that a column of data is unique.

Note that we can build indexes on more than one column. If that is the case then a for a select to use the multi-column index will require that all of the columns in the index are in the where clause.

Indexes are also used when we sort data to get it back in order.

Book: Page 108 to 111