Interactive - 02 - Insert data into name_list

Insert some data into the table.

```
insert into <name> ( <column-name>, <column-name>, .... ) values
    ( <colun-data1>, <colun-data2>, ... ),
    ( <colun-row2>, <colun-row2>, ... ),
    ( <colun-row2>, <colun-row2>, ... )
;
```

so our example, we will use all the columns so we can leave the list of columns out and just put the data in the table declared order.

```
INSERT INTO name_list values
    ( 'bob true',            22, 'WY' ),
    ( 'Jane True',           20, 'WY' ),
    ( 'Tom Ace',             31, 'NJ' ),
    ( 'Steve Pen',           33, 'NJ' ),
    ( 'Laura Jean Alkinoos', 34, 'PA' )
;
```

and a single row (note that the column order is chained). It is better to list the columns so that if the table is changed the data can still be loaded. It is possible to add new columns to a table and break old code - so it is better to have:

```
insert into name_list ( real_name, age, state ) values
    ( 'bob true',            22, 'WY' ),
...
...
...
...
    ( 'Laura Jean Alkinoos', 34, 'PA' )
;
```

Also note that it is intentional at this point that `bob true` is all lower case. We will fix that in a little bit.

```
INSERT INTO name_list ( real_name, state, age )
    values  ( 'Philip Schlump', 'WY', 62 );
```

Each set of data that we inset is separated with a comma, `,`. When we insert text we use single quotes, `'` to mark that it is text. To insert a quote mark we put in 2 of them in a row. So, To insert `that's` we would use in our insert statement, `'that''s`‘.

Numbers are respresented as a set of digits. With fractional(floating point) numbers you put in a decimal point - but note that our column age is not a float - so just whole years.

If all has worked correctly then you should have 6 rows in the table.

```
select count(1) as x
from name_list
;
```

6 rows is good. Click to grade the homework.

Book: Pages 7 to 9