Q1. Where are we instantiating instances of the `Property` class?

The initialize method instances the property class

Q2. Where are we defining the SQL that enables us to save the ruby `Property` object into the database?

The save method defines the sql that allows us to save Ruby into the database. The method uses INSERT INTO to achieve this.

Q3. In `console.rb`, which lines modify the database?

There are only a few times the database is modified in the console.rb file. At the top of the file the delete_all method is called to clear all rows from the database. At the bottom of the file two properties are Inserted into the database using the save method.

Q4. Why do we not define the id of a `Property` object at the point we instantiate it ('new it up')?

The database handles the auto incrementation of the property id's. Our ruby program does not have write access to the id's because incorrect modification of the id's could cause problems within the database.

Q5. Where and how do we assign the id (that is generated by the database) to the ruby `Property` object?

To assign the id generated by the database we need to the database to return the row of the property that we just created. This happens within the save method of the program.

Q6. Why do we put a guard (an `if` clause) on the `@id` attribute in the constructor?

We check to see if the id is valid

Q7. Why are some of the CRUD actions represented by instance methods, and others by class methods?

Some CRUD operations such as update and select_all require us to create new properties which requires the class to do so. The rest only need to access the properties of the property class

Q8. What type of data structure is returned by calls to `db.exec_prepared()`? In the `save` method, how do we access the id from the returned data structure?

A hash is returned by exec_prepared, we access this by using the id as key.

Q9. Why do we use prepared statements when performing database operations?

Prepared statements protect against sql injection attacks by sanitizing the input string and banning certain characters such as the semi colon.