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

"In console.rb"

```
property1 = Property.new({
  'address' => '89 Bouverie Road, WESTON UNDER WETHERLEY, CV33 3XG',
  'value' => 120000,
  'bedrooms' => 3,
  'build' => 'detached'
  })

property2 = Property.new({
  'address' => '15 Hart Road, NORTHFLEET,DA10 1ZX',
  'value' => 90000,
  'bedrooms' => 1,
  'build' => 'flat'
  })
```

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

"In property.rb, in the instance method 'save'"

```
sql = "INSERT INTO properties
    (
      address,
      value,
      bedrooms,
      build
    )
    VALUES
    (
      $1,$2,$3,$4
    )
```

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

```
Property.delete_all()
...
```

**# Q4. Why do we not define the id of a Property object at the point we instantiate it (‘new it up’)?**

Because it's not our responsibility?

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

First in properties.sql. Then, in property.rb, in the initializer.

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

mmmm.. because not everything has an id in the options?

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

The methods that return all instances are class methods as they don t need an instance to be present.

**# 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.
By using square brackets?

**# Q9. Why do we use prepared statements when performing database operations?**
To protect our code against SQL injection attacks
They allow us to use apostrophes in our text values
They are faster for mass updates/ saves