Student Roster DB From Schema

It's time to start building the databases from all those schema we designed, and we'll use Ruby to do it.

Objectives

Create the student database in Ruby

Revisit the Student Roster Schema Challenge and write a ruby program that creates the database from this schema.

First, include the SQLite3 gem for Ruby. Install it by running:

```
gem install sqlite3
```

Next, create a file called `setup.rb` that you will only run once to create your database. Use the code below as a template to get you started.

```ruby
require 'sqlite3'


# If you want to overwrite your database you will need
# to delete it before running this file
$db = SQLite3::Database.new "students.db"


module StudentDB
 def self.setup
  $db.execute(
   <<-SQL
    CREATE TABLE students (
     id INTEGER PRIMARY KEY AUTOINCREMENT,
     first_name VARCHAR(64) NOT NULL,
     last_name VARCHAR(64) NOT NULL,
      # ADD THE ADDITIONAL ATTRIBUTES HERE!

      created_at DATETIME NOT NULL,
      updated_at DATETIME NOT NULL
    );
  SQL
 )
end

def self.seed
 # Add a few records to your database when you start
 $db.execute(
  <<-SQL
    INSERT INTO students
     (first_name, last_name, created_at, updated_at)
    VALUES
     ('Brick','Thornton',DATETIME('now'), DATETIME('now'));

    # CREATE TWO MORE STUDENTS WHO ARE AT LEAST AS COOL AS THIS ONE.

  SQL
 )
end
end
```

Verify your database is working properly. You can do this by loading your `setup.rb` file in irb and run the `StudentDB.setup` and `StudentDB.seed`methods. Then open up the Sqlite console with the `students.db` database and make sure you can run `SELECT` on these records.

Create the Student Class

There are some clear parallels with the Ruby objects and the Database tables.

SQL land	**Ruby land**
A table named `students`	A class named `Student`
A row from `students`	An instance of `Student`
SELECT * FROM students	Student.all

SELECT * FROM students WHERE first*name = 'bob'*
OR
*SELECT * FROM students WHERE first*name = ? , '
bob'
	Student.where('first_name = ?', 'bob')

SELECT * FROM students WHERE id = 10
OR
SELECT * FROM students WHERE id = ? , 10
	Student.where('id = ?', 10)

INSERT INTO students (field1, field2, ...)
VALUES(value1, value2, ...)
	student = Student.new(data)
	student.save

DELETE FROM students WHERE id = 40	???

NOTE: In the SQL above, two versions of `WHERE` clauses are given. In the second version, a SQL placeholder `?` is used. This format helps avoid SQL injection attacks You can see more examples of placeholders in WHERE clauses and placeholders in INSERT clauses.

Add methods to the Student Class to execute SQL commands

Now write Ruby code that allows you to complete the following tasks:

1. Add a student
2. Delete a student
3. Show a list of all students
4. Show a list of students with a particular `first_name`
5. Show a list of students with any particular attribute

Use the table above as a guide for your methods. What can you infer about each method's declaration and structure?

Extra Credit:

- List which students have a birthday this month
- List students by birthday