CS 2316 Exam 3 Study Guide

1. **Short Answer**

Given the following table schema for a dive log:

```
create table dive (
  datetime text, -- the day and start time of the dive
  duration text, -- duration of the dive from descent to ascent
  location text, -- the location and dive site name
  buddy text,   -- dive buddy
);
```

(a) (5 points) Which field, or fields, would make the best primary key? Why?

Solution:
```
datetime should be unique - can't do two dives at once
```

Given the following schema:

```
create table dog (
    dog_id integer primary key autoincrement,
    name text not null check(name != ''),
    sex text not null check(sex = 'M' or sex = 'F'),
    owner_id references owner(owner_id)
);

create table owner (
    owner_id integer primary key autoincrement,
    name text not null check(name != '')
);
```

(b) (5 points) Write a `select` statement that will return all the male dogs.

Solution:
```
select * from dog where sex = 'M';
```

(c) (5 points) Write a `select` statement that will return all the dogs owned by 'Tom'.

Solution:
```
select * from dog join owner using (owner_id) where owner.name = 'Tom';
```

(d) (5 points) Write a `select` statement that will return all the dogs whose names start with 'Butter'.

Solution:
```
select * from dog where name like 'Butter%';
```

2. (10 points) Write the create statements for a database of employees and departments. For employees store name, hire date, salary, and department. For department store the name, description, and the department head, who is an employee. Add any fields you need for keys and be sure to include foreign keys as necessary.

```
create table employee (
  employee_id integer primary key,
  name text,
  hiredate text,
  salary float,
  department_id references department(department_id)
);

create table department (
  department_id integer primary key,
  name text,
  description text,
  head references employee(employee_id)
);
```

Given:

```
<!doctype html>
<html>
  <head>
    <title>Hello, World!</title>
  </head>
  <body>
    <a name="top" />
    <h1>Level One Heading (H1)</h1>
    <p>This is paragraph text under the first heading.</p>
    <p>Jump to <a href="#table">simple table</a></p>
    <h2>Level Two Heading (H2)</h2>
    <p>Chris Simpkins, Director General of the Royal British Legion</p>
    <img src="http://goo.gl/RXfE7U" />
    <p>Got this picture from <a href="http://goo.gl/uoaBmP">this page</a>.</p>
    <p>Send email to <a href="mailto:bob@aol.com">Bob</a></p>
    <h2>List One: (unordered list):</h2>
    <ul>
      <li>Item 1</li>
      <li>Item 2</li>
    </ul>
    <h2>List Two: (ordered list):</h2>
    <ol>
      <li>Item 1</li>
      <li>Item 2</li>
    </ol>
    <a name="table"/>
    <h2>Simple Table:</h2>
    <table>
      <tr>
        <th>Col1</th><th>Col2</th><th>Col3</th>
      </tr>
      <tr>
        <td>1,1</td><td>1,2</td><td>1,3</td>
      </tr>
      <tr>
        <td>2,1</td><td>2,2</td><td>2,3</td>
      </tr>
    </table>
    <p>Back to <a href="#top">top</a></p>
  </body>
</html>
```

3. How would List One be rendered in a browser?

   - Item 1
   - Item 2

4. How would List Two be rendered in a browser?

   1. Item 1
   2. Item 2

5. How would Simple Table be rendered in a browser?

Col1	Col2	Col3
1,1	1,2	1,3
2,1	2,2	2,3

Assume the text of the HTML code above is stored in the string variable `hello`, newline characters have been removed, and the `re` module has been imported as `re`.

6. Write a regular expression matcher that finds all the level two headings and returns their text (without the `<h2></h2>` tags) in a list of strings.

```
re.findall(r'<h2>(.+?)</h2>', hello, re.S)
```

7. Write a regular expression that extracts the URL of the JPEG image of Chris Simpkins, Director General of the Royal British Legion.

```
re.findall(r'Chris Simpkins.+?<img src="(.+?)"', hello, re.S)[0]
```

8. Using the `requests` module, write a snippet of code that downloads the JPEG image from the URL extracted above and stores the image in a file on disk named `chris-simpkins.jpg`.

```
cs_url = re.findall(r'Chris Simpkins.+?<img src="(.+?)"', hello, re.S)[0]
resp = requests.get(cs_url)
open('chris-simpkins.jpg', 'wb').write(resp.content)
```

```
{
  "dinner": {"items": {"spaghetti": "$8.00"}, "hours": "3-10"},
  "lunch": {"items": {"hamburger": "$5.00"}, "hours": "11-3"},
  "breakfast": {"items": {"breakfast burritos": "$6.00", "pancakes": "$4.00"},
      "hours": "7-11"}
}
```

Assume the JSON data above is stored in a file called `menu.json`.

9. Write Python code that uses the `json` module to load the JSON data stored in `menu.json`, convert it to a Python `dict` object, and store it in a variable called `menu`.

```
menu = json.loads(open('menu.json', 'r').read())
```

10. Write Python code to add a new menu item, ”sptzle”, with a price of $9.00 to the `items` (dict) value associated with the `dinner` key of the dictionary stored in `menu` above.

```
menu['dinner']['items']['sptzle'] = '$9.00'
```

11. Write Python code that converts the (now modified) Python dictionary stored in the `menu` variable to a JSON string and writes that JSON string to the `menu.json` file, overwriting its previous contents.

```
open('menu.json', 'w').write(json.dumps(menu))
```