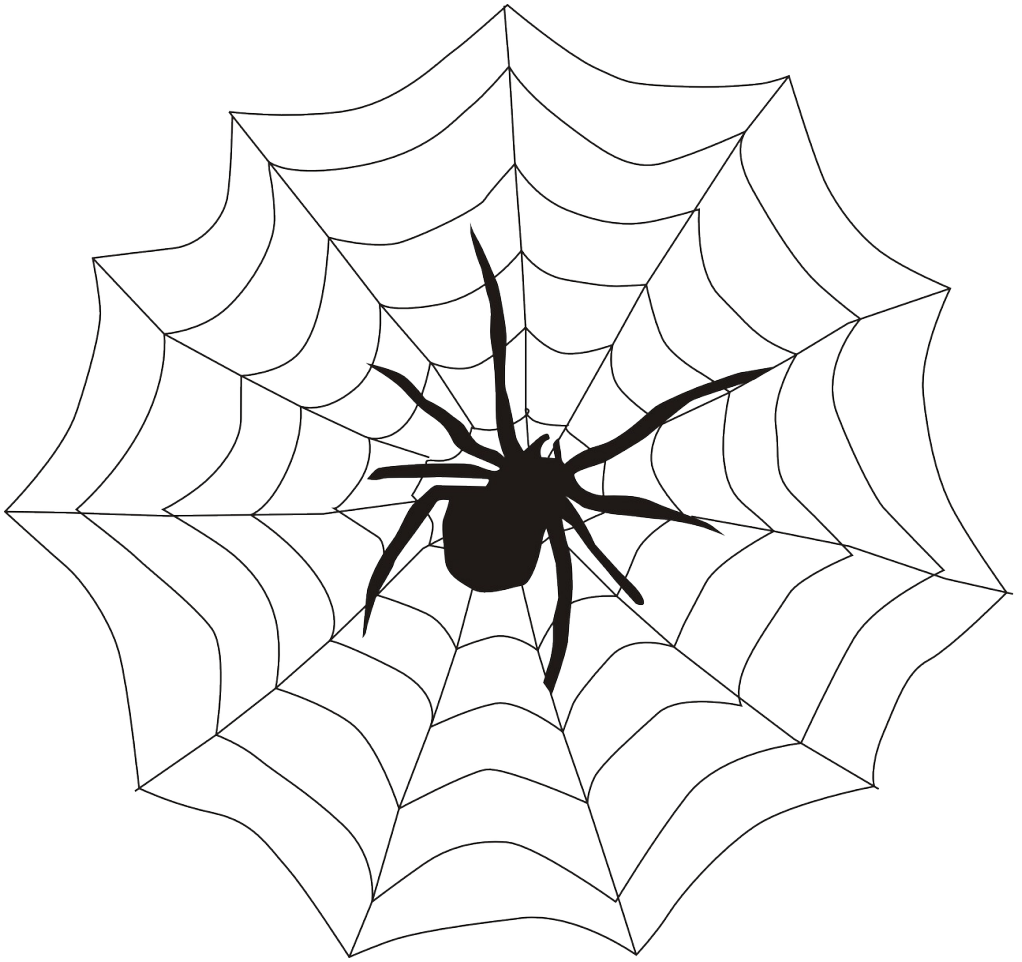
HackerFrogs Afterschool
Intro to SQL /w SQL Murder Mystery

Class:
Web App Hacking

Workshop Number:
AS-WEB-05

Document Version:
1.75

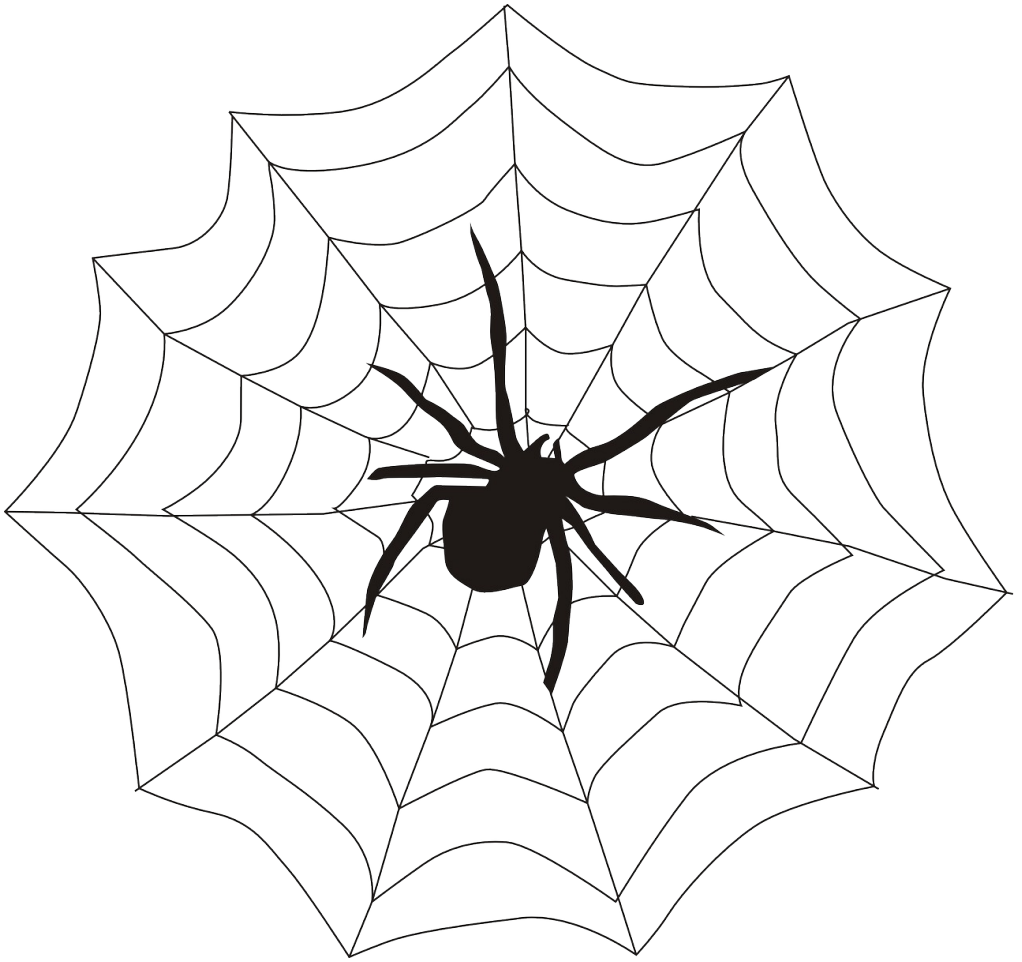
Special Requirements:
None



What We Learned In The Previous Workshop

This is the fifth workshop for intro to web app hacking.

Let's take a few moments to review the concepts we learned in the previous workshop.



OS Command Injection

Operating System (OS) Command Injection is a web app vulnerability where arbitrary OS commands can be performed on the webserver through a web interface.

OS Command Injection

OS Command Injection can often lead to complete compromise of the webserver, and if so, the server can be used as a foothold to attack other machines on the network.

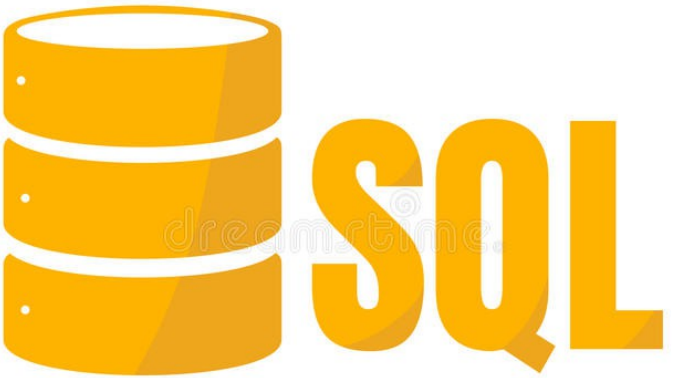
Now On To Our Topic!

Let's move on to the topic of this workshop:

The SQL database language

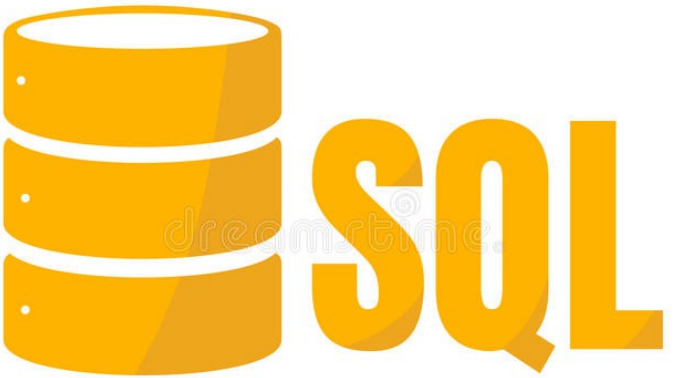
What is SQL?

Structured Query Language (SQL) is a programming language used for managing data held in relational databases.



What is SQL?

Simply put, SQL allows users to access and manage data contained in relational databases, which are common components in most web applications.



Why Learn About SQL?

As far as web app hacking is concerned, insecure implementation of SQL in web apps can lead to a very serious vulnerability called SQL Injection.

Why Learn About SQL?

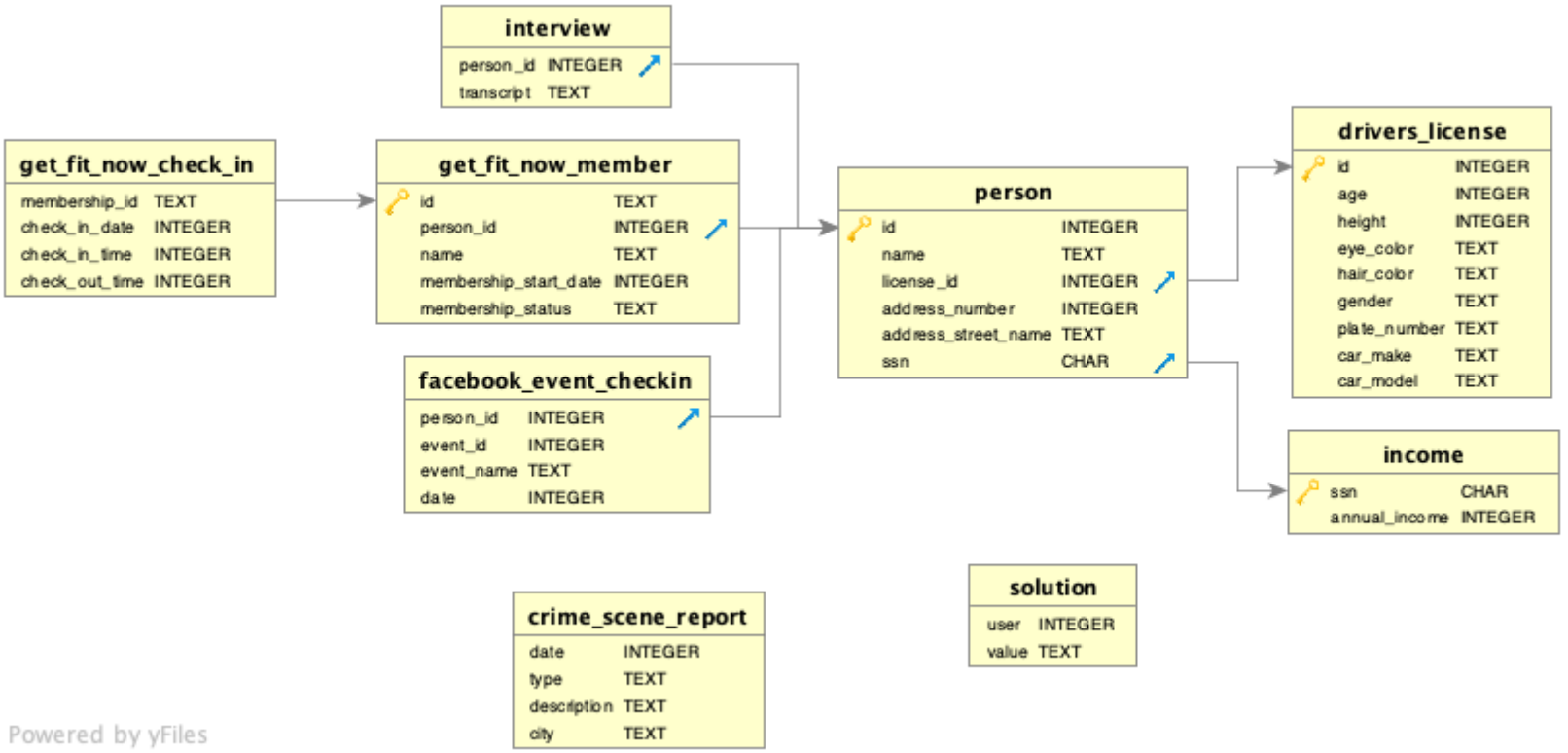
We will learn about SQL Injection in a later session, but we should first learn how to use SQL in its intended manner before we learn how to abuse it.

Let's Learn SQL Through a Game

In a web browser, navigate to the following URL:

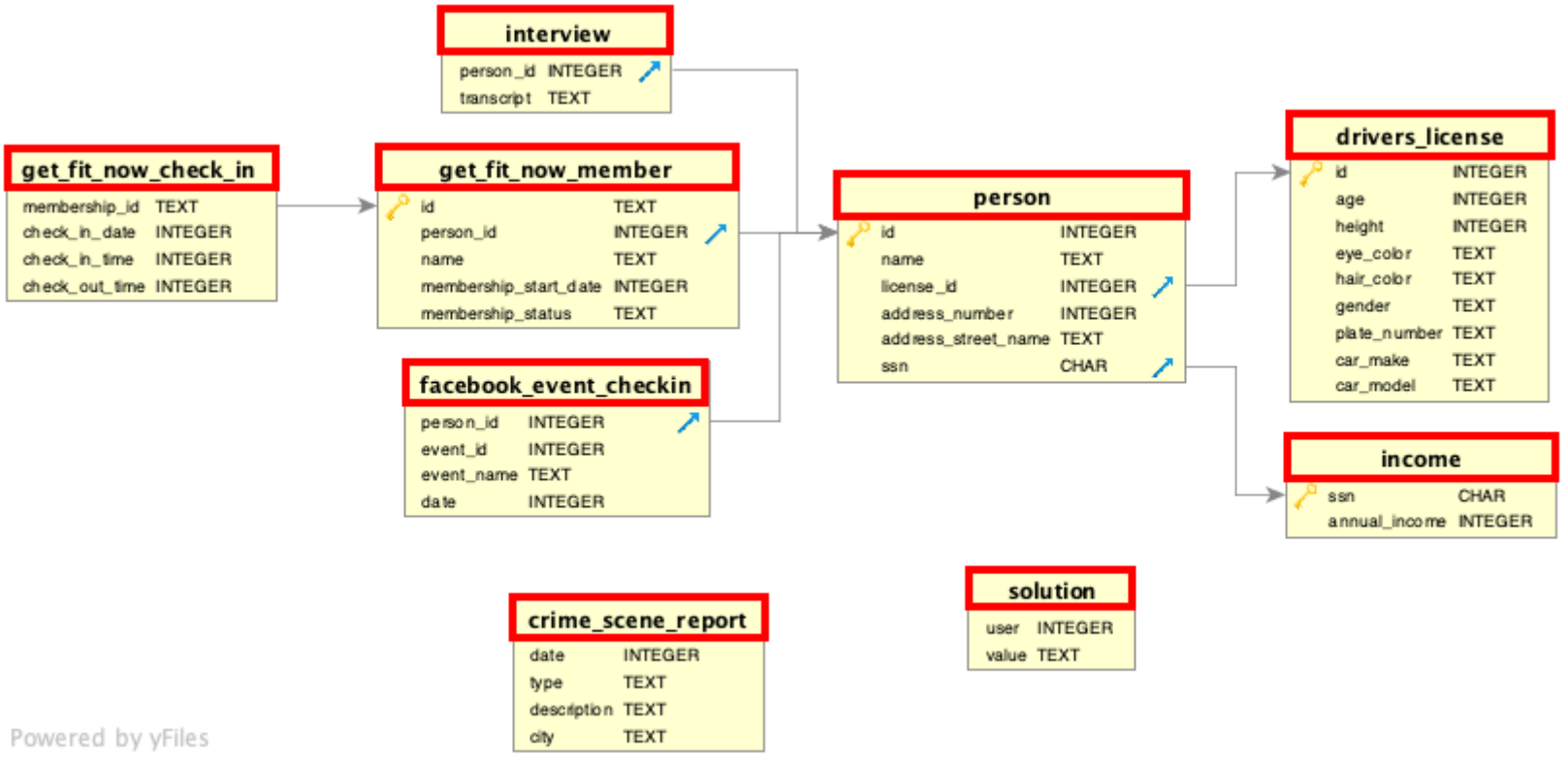
https://mystery.knightlab.com/walkthrough.html

SQL Database Structure



The largest unit in a SQL system is the **database**,

SQL Database Structure



The largest unit in a SQL system is the **database**, which contain one or more **tables**.

SQL Database Structure

id	name	license_id	address_number	address_street_name	ssn
10000	Christoper Peteuil	993845	624	Bankhall Ave	747714076
10007	Kourtney Calderwood	861794	2791	Gustavus Blvd	477972044
10010	Muoi Cary	385336	741	Northwestern Dr	828638512

Tables themselves can be subdivided into two elements:

SQL Database Structure

id	name	license_id	address_number	address_street_name	ssn
10000	Christoper Peteuil	993845	624	Bankhall Ave	747714076
10007	Kourtney Calderwood	861794	2791	Gustavus Blvd	477972044
10010	Muoi Cary	385336	741	Northwestern Dr	828638512

Columns, whose names identify the data that should be entered into them...

SQL Database Structure

id	name	license_id	address_number	address_street_name	ssn
10000	Christoper Peteuil	993845	624	Bankhall Ave	747714076
10007	Kourtney Calderwood	861794	2791	Gustavus Blvd	477972044
10010	Muoi Cary	385336	741	Northwestern Dr	828638512

And **rows**, which contain data entries that correspond to the columns they fall under.

Basic SQL Statements

```
1 SELECT *
2 FROM person;
```

This statement is used to retrieve all the rows from a specified table. It uses two SQL keywords: SELECT and FROM

The SELECT Command

```
1 SELECT *
2 FROM person;
```

The **SELECT** command is the most common SQL command, which returns SQL entries depending on what additional parameters are provided.

The FROM Command

```
1 SELECT *
2 FROM person;
```

The **FROM** command specifies which table to retrieve data from.

The * Wilcard Character

```
1 SELECT *
2 FROM person;
```

The last important detail about this statement is the * wildcard character, which indicates all columns

Basic Table Selection Practice

```
1 SELECT *
2 FROM person;
```

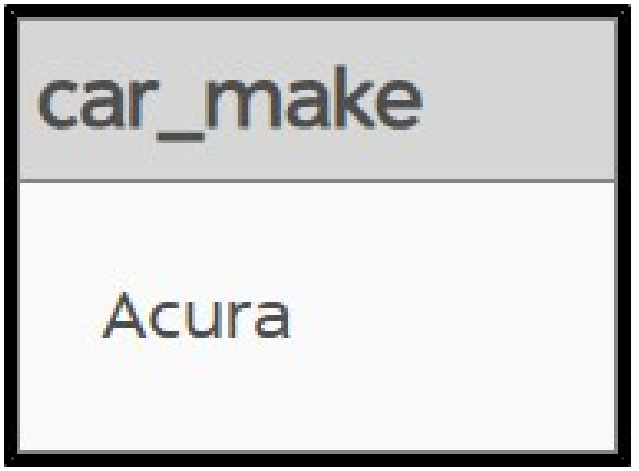
Retrieve all the columns from the `interview` table: What is the first number in the person_id column?

Basic Table Selection Practice

person_id	transcript
28508	‘I deny it!’ said the March Hare.

The answer is 28508. Now retrieve all the columns from the `drivers_license` table: What is the first entry in the `car_make` column?

Basic Table Selection Practice



The answer is Acura. Great! Now let's more on more advanced statements

Selecting Rows with Specific Values

```sql
SELECT *
FROM person
WHERE name = 'Kourtney Calderwood';
```

Most of the time when we interact with databases, we want specific information, and we can search for that with the **WHERE** filter and **=** character

The WHERE Filter

```
1 SELECT * FROM person WHERE name = 'Kinsey Erickson'
```

id	name	license_id	address_number	address_street_name	ssn
89906	Kinsey Erickson	510019	309	Northwestern Dr	635287661

The **WHERE** filter allows very specific information to be returned by a query

The = Operator

```
1 SELECT * FROM person WHERE name = 'kinsey Erickson'
```

No data returned

The = operator is used with the WHERE filter to return exact matches. If the contents of a row differ by even a single character, the match will not return. (The 'k' in the above example is not capitalized, so a valid match is not found)

WHERE Statement Practice

Look in the `person` table

```
name = 'Hana Beverage'
```

Let's practice getting specific rows using these criteria. What street does Hana live on?

WHERE Statement Practice

SELECT * FROM person WHERE name = 'Hana Beverage';

address_street_name
Ayling Circle

Now look at same person table, but this time

`Id = 10914`

What is this person's name?

WHERE Statement Practice

SELECT * FROM person WHERE Id = 10914;

name
Cinderella Coachman

Now look at same person table, but this time `address_street_name = 'Deercliff Circle'`

What is this person's SSN?

WHERE Statement Practice

SELECT * FROM person WHERE address_street_name = 'Deercliff Circle'

Great! Let's move on to more advanced SQL statements

Searching for Partial Matches

```
1 SELECT *
2 FROM person
3 WHERE name LIKE '%more%';
```

Sometimes we want to search the database, but we only have vague details, such as part of a name of the last few digits of a number

Searching for Partial Matches

```
1 SELECT *
2 FROM person
3 WHERE name LIKE '%more%';
```

These type of queries require the WHERE filter with the LIKE keyword and % wildcard character

The LIKE Keyword

```
1 SELECT *
2 FROM person
3 WHERE name LIKE '%more%';
```

The LIKE keyword is used instead of the = operator when we want to return partial matches (as opposed to exact matches)

The % Wilcard Character

```
1 SELECT *
2 FROM person
3 WHERE name LIKE '%more%';
```

The % wildcard character is used with the LIKE keyword to return anything on that side of the query term

The % Wilcard Character

```
1 SELECT *
2 FROM person
3 WHERE name LIKE '%more%';
```

In the above example, the % wildcard characters will match any values to the left and right of the “more” string

The AND Clause

```
1 SELECT * FROM person WHERE name = 'John Dile'
2 AND address_street_name = 'Icehouse Ave'
```

id	name	license_id	address_number	address_street_name	ssn
13915	John Dile	909334	3783	Icehouse Ave	957131634

The **AND** clause is used with the **WHERE** filter to further restrict the results returned, only returning rows that match both parameters on either side of the **AND** clause.

The AND Clause

```
SELECT *
FROM person
WHERE address_street_name LIKE 'West%Circle'
AND name LIKE 'Alva%'
```

name	license_id	address_number	address_street_name
Alvaro Kistler	377111	3588	West Peregrine Circle

Using the AND clause is very useful when we have a couple of pieces of info to search for, but we don't exact details

The AND Clause

```
SELECT *
FROM person
WHERE address_street_name LIKE 'West%Circle'
AND name LIKE 'Alva%'
```

name	license_id	address_number	address_street_name
Alvaro Kistler	377111	3588	West Peregrine Circle

In the above example, we know that the street is West something Circle, and the person's name is Alva something

Advanced Search Practice

Let's practice our AND and LIKE keyword searches. We know the name of the person is Jenny something, and they live on Rising something. What's the person's full name?

Advanced Search Practice

```
SELECT *
FROM person
WHERE address_street_name LIKE 'Rising%'
AND name LIKE 'Jenny%'
```

name	license_id	address_number	address_street_name
Jenny Forbess	735368	1165	Risinghill Dr

Next, we know the **last** name of the person is Living something, and they live on something Blvd. What's the person's full name?

Advanced Search Practice

```
SELECT *
FROM person
WHERE address_street_name LIKE 'Rising%'
AND name LIKE 'Jenny%'
```

name	license_id	address_number	address_street_name
Jenny Forbess	735368	1165	Risinghill Dr

Next, we know the **last** name of the person is Living something, and they live on something Blvd. What's the person's full name?

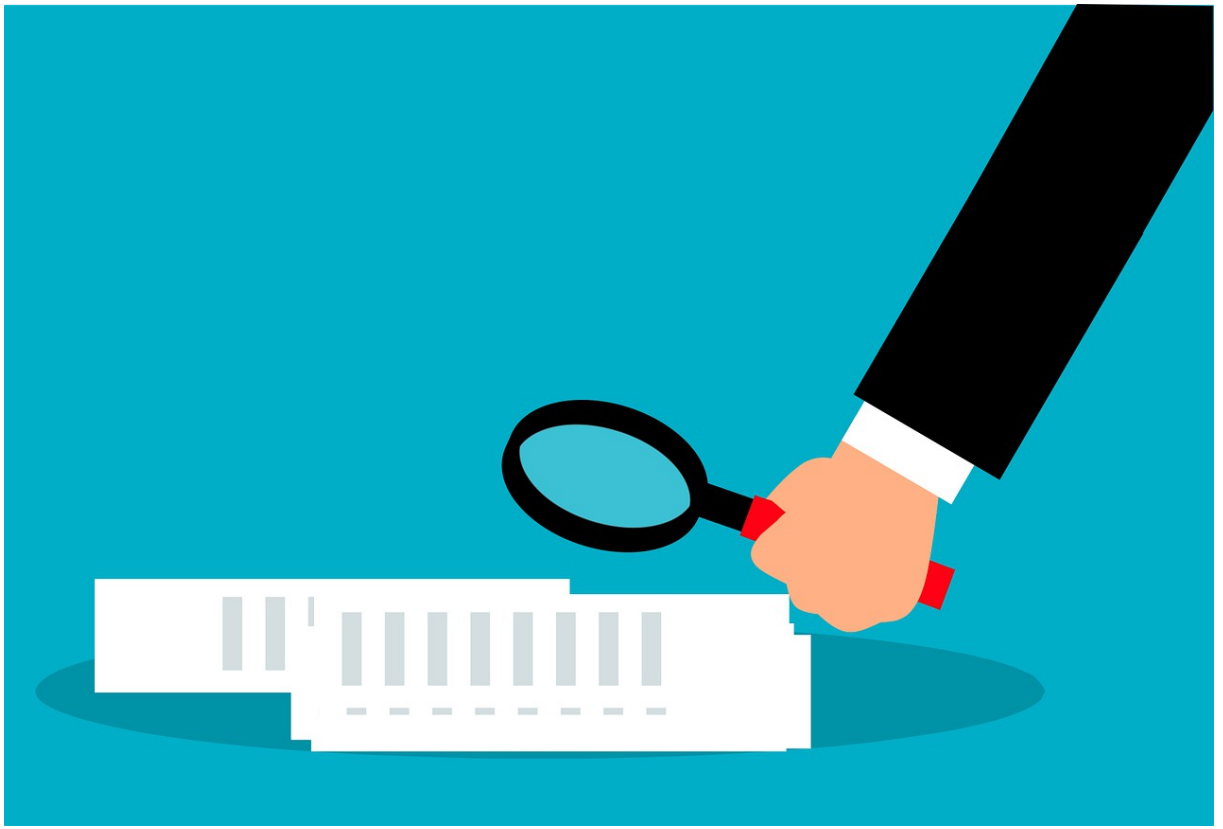
Advanced Search Practice

```
SELECT *
FROM person
WHERE address_street_name LIKE '%Blvd'
AND name LIKE '%Living%'
```

name	license_id	address_number	address_street_name
Hector Livingstone	972988	1217	E Mc Eldowney Blvd

Great! Now let's practice a challenge over at PicoCTF...

Summary



Let's review the SQL concepts we learned in this workshop:

Structured Query Language (SQL)

Structured Query Language (SQL) is a programming language used for managing data held in relational databases.

The SELECT Command

```
1 SELECT count(*)
2 FROM person;
```

The **SELECT** command is common to all SQL queries looking to return specific info from the database.

What's Next?

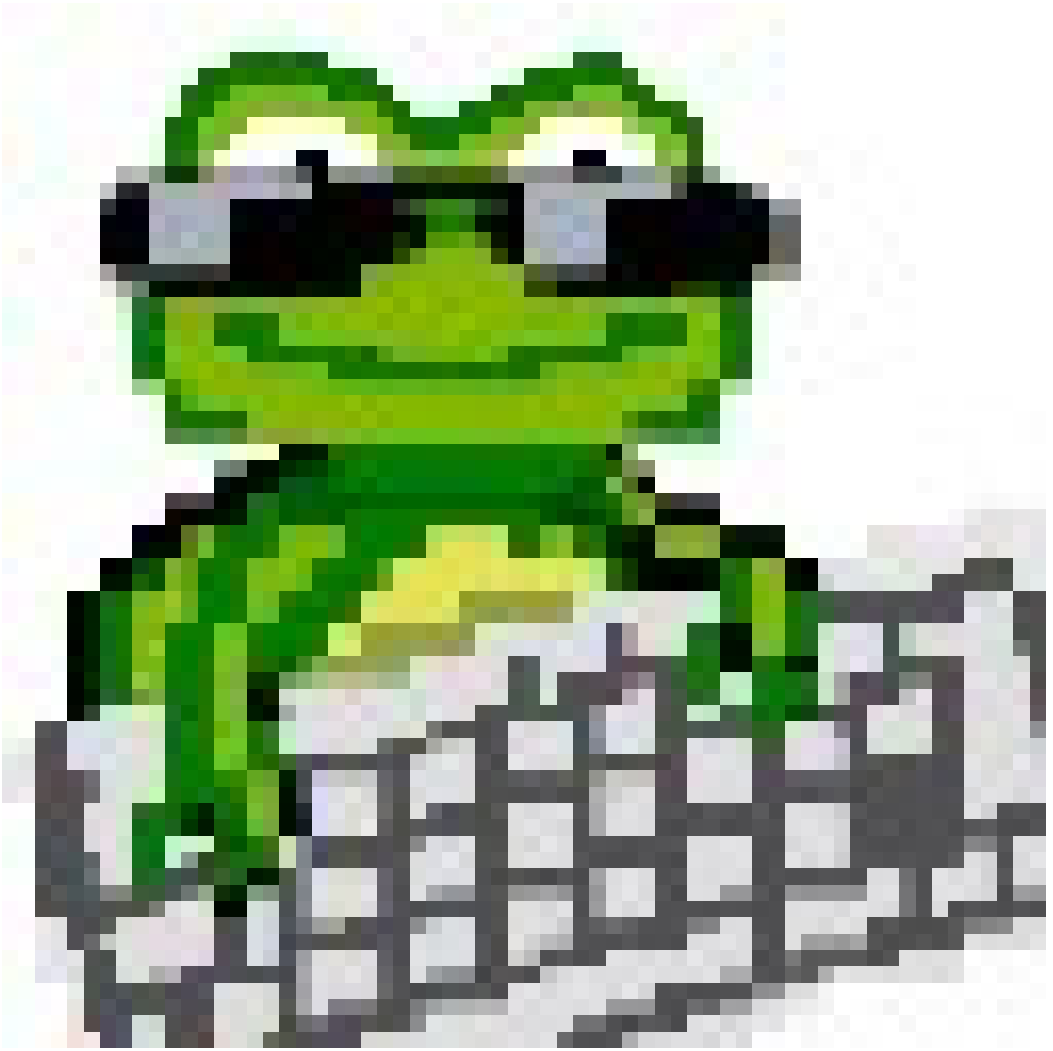
In the next HackerFrogs Afterschool web exploitation workshop, we'll learn about SQL injection vulnerabilities with the TryHackMe platform.

Extra Credit

Looking for more study material on this workshop's topics?

See this video's description for links to supplemental documents and exercises!

Until Next Time, HackerFrogs!



Searching for Specific Row Values

```
1 SELECT *
2 FROM person
3 WHERE name = 'Kourtney Calderwood';
```

Another important skill for searching in databases is filtering rows based on row contents. This requires two SQL features, the **WHERE** filter and the **=** operator