

# Introduction to Website Database Vulnerabilities

Mr. Lee and Mrs. Weiss



# Introduction to Website Database Vulnerabilities

**Database Vulnerabilities = Hackable!**

**How do you define “to hack a database”?**

To hack a database is to “trick” the databases to do things that the programmers did not intend to allow to be done.



# Introduction to Website Database Vulnerabilities

## DISCLAIMER:

Remember that you are testing a vulnerability in a permitted, contained environment. Attempting any of these techniques on another website that is not wholly owned by you is considered a **cyber crime** and most likely falls under **federal jurisdiction**.

In other words: ***DON'T TRY THIS AT HOME!***



# Introduction to Website Database Vulnerabilities

## Types of Vulnerabilities

**Error based:** If the application displays database errors to the user, an attacker may learn key information.

**Union based:** append the data that the attacker wants to a table that is already displayed in the page.

**Blind(brute force):** E.g. you could send the statement "if the first name of the username is an *a*, wait 10 seconds. If the application takes 10 seconds to perform the query, the username starts with an *a*. Obviously brute force is time intensive, so this approach should be considered a last resort.



# Introduction to Website Database Vulnerabilities

## **YOUR TASK:** **Test Error Based Vulnerability**

Log onto the following website: <http://34.197.52.233/DVWA-master/index.php>  
This is a sample web application.

Fill in the table with your answers first, and then supporting screenshots when applicable. Read carefully!



# Follow-up:

Presentation of some commands, and the results, visually.

We focused on the general syntax of SQL by using the “SELECT” command, since that is the command students saw on the worksheet. For example, show how variable type and upper/lowercase do not make a difference.





Computer programming

# Unit: Intro to SQL: Querying and managing data

## CONTENTS

[About](#)[Projects completed](#)[SQL basics](#)[More advanced SQL queries](#)[Relational queries in SQL](#)[Modifying databases with SQL](#)[Further learning in SQL](#)

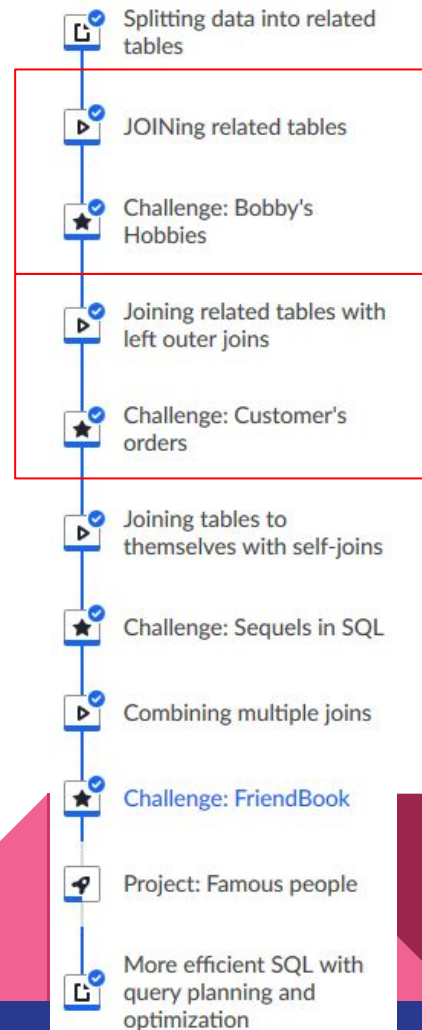
Khan offers 4 sub-units on SQL.

You don't have to cover them all.

# Topics/days

Each lesson is a video followed by an assignment.

We recommend assigning up to 2 “challenges” a day.





# SQL Server not needed for small tasks

Computing > Computer programming > Intro to SQL: Querying and managing data > Relational queries in SQL

Splitting data into related tables

JOINING related tables

Challenge: Bobby's Hobbies

Joining related tables with left outer joins

Challenge: Customer's orders

Joining tables to themselves with self-joins

Challenge: Sequels in SQL

Not secure | khanacademy.org/computing/computer-programming/sql/relational-queries-in-sql/pc/challenge-friendbook

Apps APCS Khan Academy College Board Hunter Scratch - Imagine... Optimum Webmail WhatsApp CS4All Snap! Build Your O... private Other bookmarks

table of friend connections between the people. In this first step, use a JOIN to display a table showing people's names with their hobbies.

Bobby McBoblyface coding  
Lucy Bobble dancing  
...

SELECT ... JOIN ...;

```

11
12 CREATE table hobbies (
13     id INTEGER PRIMARY KEY AUTOINCREMENT,
14     person_id INTEGER,
15     name TEXT);
16
17 INSERT INTO hobbies (person_id, name) VALUES (1, "drawing");
18 INSERT INTO hobbies (person_id, name) VALUES (1, "coding");
19 INSERT INTO hobbies (person_id, name) VALUES (2, "dancing");
20 INSERT INTO hobbies (person_id, name) VALUES (2, "coding");
21 INSERT INTO hobbies (person_id, name) VALUES (3, "skating");
22 INSERT INTO hobbies (person_id, name) VALUES (3, "rowing");
23 INSERT INTO hobbies (person_id, name) VALUES (3, "drawing");
24 INSERT INTO hobbies (person_id, name) VALUES (4, "coding");
25 INSERT INTO hobbies (person_id, name) VALUES (4, "dilly
    -dallying");
26 INSERT INTO hobbies (person_id, name) VALUES (4, "neowing");
27
28 CREATE table friends (
29     id INTEGER PRIMARY KEY AUTOINCREMENT,
30     person1_id INTEGER,
31     person2_id INTEGER);
32
33 INSERT INTO friends (person1_id, person2_id)
34     VALUES (1, 4);
35 INSERT INTO friends (person1_id, person2_id)
36     VALUES (2, 3);
37
38

```

DATABASE SCHEMA

persons		5 rows	
id (PK)	INTEGER		
fullname	TEXT		
age	INTEGER		

hobbies		10 rows	
id (PK)	INTEGER		
person_id	INTEGER		
name	TEXT		

friends		2 rows	
id (PK)	INTEGER		
person1_id	INTEGER		
person2_id	INTEGER		

Undo Start over