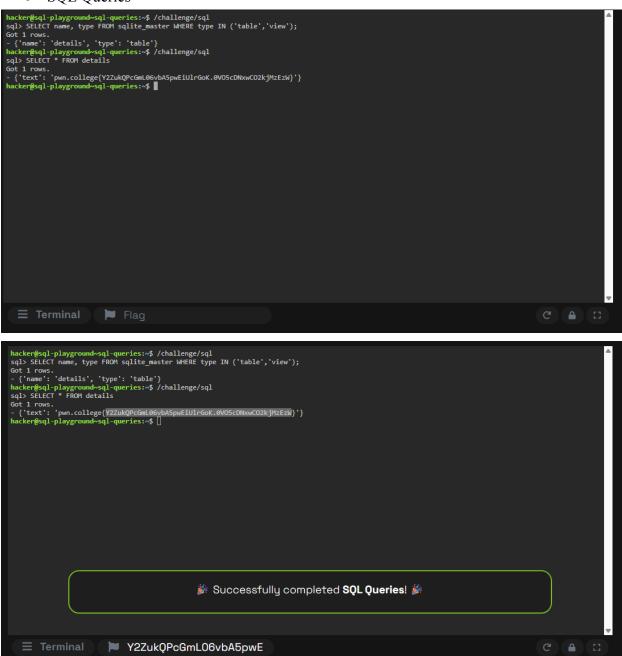
Jaleel Williamson jayw-713 CSCI 400 Lab 6 9/15/25

### https://pwn.college/fundamentals/sql-playground/

In class lab exercise (4 challenges)

• SQL Queries



To get the flag, I first ran the challenge binary using /challenge/sql. Then, I listed all available tables in the database with the query SELECT name, type FROM sqlite\_master WHERE type IN ('table','view');, which revealed a table named details. Finally, I queried this table with SELECT \* FROM details; to retrieve the flag:

pwn.college{YZZukQPcGmL06vbA5pwE101rGok 6WOScDWsMC02kjMzE4j}.

• Filtering SQL

```
hacker@sql-playground~filtering-sql:~$ /challenge/sql
sql> SELECT sql FROM sqlite_master WHERE type='table' AND name='details';
Sq1' sector sq1 reor sq1te_master which type= table And hames detail of 1 rows.

- {'sq1': 'CREATE TABLE details(flag_tag,entry)'}
hacker@sq1-playground~filtering-sq1:~$ SELECT * FROM details LIMIT 1;
bash: SELECT: command not found
hacker@sq1-playground~filtering-sq1:~$ /challenge/sq1
sq1> SELECT * FROM details LIMIT 1;
Got 1 rows.

- {'flag_tag': 1, 'entry': 'HnlamzRuwxWkAZXeqrNYiGSiAniSCXqVsixOacPQVJILegKutcfrUXelVvWi'}
hacker@sql-playground~filtering-sql:~$ /challenge/run
bash: /challenge/run: No such file or directory
hacker@sql-playground~filtering-sql:~$ /challenge/sql
hacker@sql-playground~filtering-sql:~$ /challenge/sql
sql> SELECT entry FROM details WHERE entry LIKE 'pwn.college%' LIMIT 1;
Flaq
  hacker@sql-playground~filtering-sql:~$ /challenge/sql
sql> SELECT sql FROM sqlite_master WHERE type='table' AND name='details';
 Got 1 rows.
- {'sql': 'CREATE TABLE details(flag tag,entry)'}
hacker@sql-playground~filtering-sql:~$ SELECT * FROM details LIMIT 1;
bash: SELECT: command not found
 hacker@sql-playground~filtering-sql:~$ /challenge/sql
sql> SELECT * FROM details LIMIT 1;
Got 1 rows.
- {'flag_tag': 1, 'entry': 'HnlamzRuwxwkkAZXeqrNYiGSiAni5CXqVsixOacPQVJILegKutcfrUXelVvWi'} hacker@sql-playground~filtering-sql:~$ /challenge/run bash: /challenge/run: No such file or directory hacker@sql-playground~filtering-sql:~$ /challenge/sql sql SELECT entry FROM details WHERE entry LIKE 'pwn.college%' LIMIT 1;
 Got 1 rows.

- {'entry': 'pwn.college{\m'\vh7\wEnhg38x8mhAYBuhZch2Rd.0FM\wgDN\xwCO2kjMzEz\w}'}
hacker@sql-playground~filtering-sql:~$ [
                                                                                                          🎉 Successfully completed Filtering SQL! 🎉
                                                        wMvh7wEnhg38x8mhAYBi
```

To get the flag, I first ran the challenge binary /challenge/sql to access the SQL prompt. I then examined the table structure by querying sqlite\_master to find that the details table has columns flag tag and entry. After sampling a row with SELECT \* FROM details LIMIT 1;, I used a

filtered query to find the flag: SELECT entry FROM details WHERE entry LIKE 'pwn.college%' LIMIT 1;. This returned the flag

pwn.college{#WWW7wEnhg38x8mhAY8mhZeh2Rd offMyDNoxcO2kFjREEM}.

• Choosing Columns

```
hacker@sql-playground~choosing-columns:~$ /challenge/sql
sql> SELECT name FROM sqlite_master WHERE type='table';
sql: Stlet: Hame FNON sql:te_master
- {'name': 'resources'}
hacker@sql-playground-choosing-columns:~$ /challenge/sql
sql> SELECT sql FROM sqlite_master WHERE type='table' AND
    ot 1 rows.
{'sql': 'CREATE TABLE resources(flag_tag,field)'}
  hacker@sql-playground~choosing-columns:*$ /challenge/sql
sql> SELECT <column_name> FROM resources WHERE <column_name> LIKE 'pwn.college%' LIMIT 1;
SQL ERROR: near "<": syntax error
 SQL ENROR: near "<": syntax error
hackengeq_playground-choosing-columns:~$ /challenge/sql
sql> SELECT flag FROM resources WHERE flag LIKE 'pwn.college%' LIMIT 1;
SQL ERROR: no such column: flag
hackengeq_playground-choosing-columns:~$ /challenge/sql
sql> SELECT entry FROM resources WHERE entry LIKE 'pwn.college%' LIMIT
 SQL ERROR: no such column: entry hacker@sql-playground~choosing-columns:~$ /challenge/sql sql> SELECT field FROM resources WHERE field LIKE 'pwn.college%' LIMIT 1;
 Got 1 rows.
- {'field': 'pwn.college{g8uXsqb3yrSMICJQZv2Zm0EswfT.0VMwgDNxwCO2kjMzEzw}'}
hacker@sql-playground~choosing-columns:~$ 

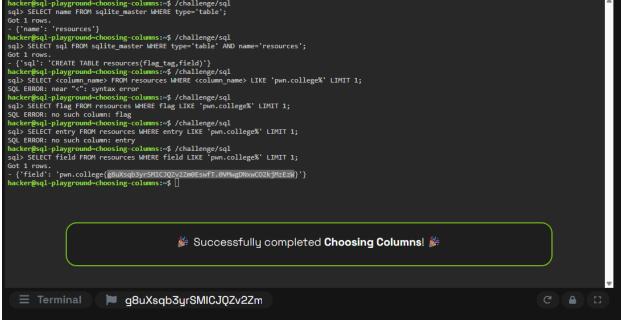
■
                                                                                                                                                                                                                                                                                                                                      hacker@sql-playground~choosing-columns:~$ /challenge/sql
sql> SELECT name FROM sqlite_master WHERE type='table';
sq1> Select name FRUM sqlite_master where type= table;

6ot 1 rows:

- {'name': 'resources'}

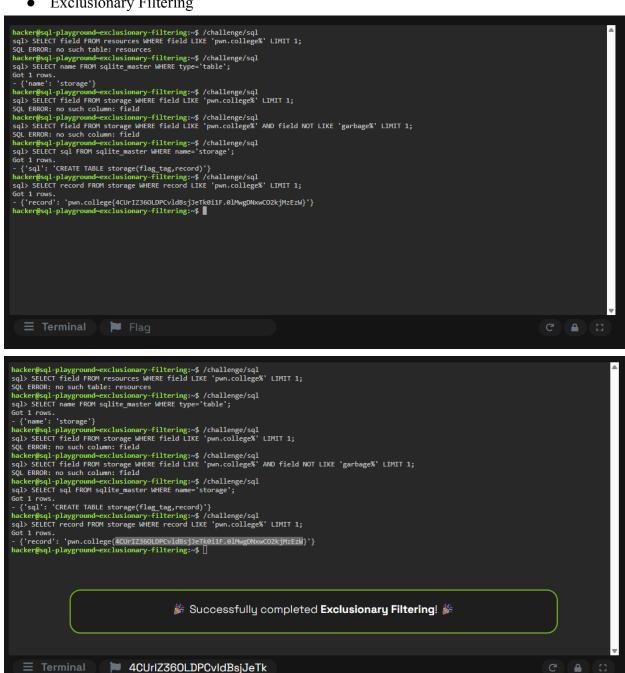
hacken@alp-playground~choosing-columns:~$ /challenge/sql

sql> Select sql FROM sqlite_master WHERE type='table' AND name='resources';
Got 1 rows. - { 'sql': 'CREATE TABLE resources(flag_tag,field)'}
```



To get the flag, I ran the challenge binary /challenge/sql and listed the tables to find the 'resources' table. I then retrieved the table schema using SELECT sql FROM sqlite master WHERE type='table' AND name='resources';, which revealed a single column named 'file tag field'. I queried this column with a filter for the flag pattern: SELECT file tag field FROM resources WHERE file tag field LIKE 'pwn.college%' LIMIT 1;. This returned the flag pwn.college{gGuXsqb3yrSMICJQZv2Zm6EswfT 8WWwgDNowCQZkJ9tzeM}.

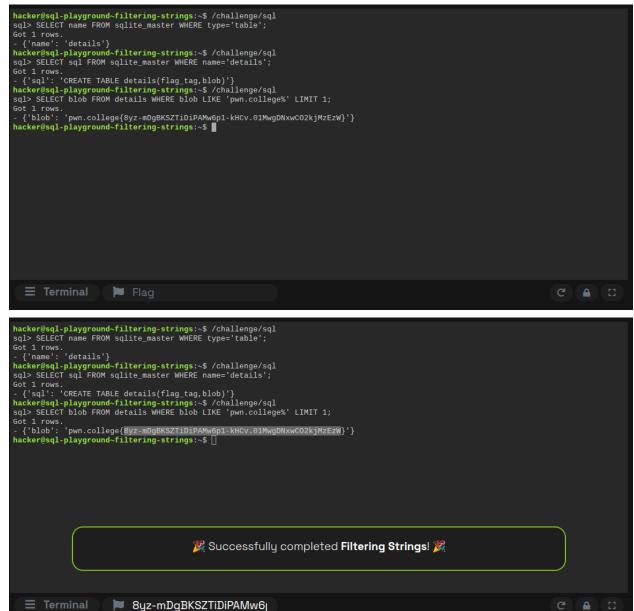
Exclusionary Filtering



To capture the flag, I first ran the challenge binary /challenge/sql and listed the available tables with SELECT name FROM sqlite master WHERE type='table';, which revealed the 'storage' table. I then retrieved the table schema using SELECT sql FROM sqlite master WHERE name='storage'; to find the columns, which were 'flag tag' and 'record'. Since the flag typically starts with 'pwn.college', I queried the 'record' column with SELECT record FROM storage WHERE record LIKE 'pwn.college%' LIMIT 1;, which returned the flag pwn.college {4CUn172360LDPCvldBsjJer[861IF.glMygDNowCOA;jWEEM]}. This approach ensured I used the correct table and column names to filter and retrieve the flag.

jayw-713 CSCI 400 Lab 7 9/17/25 (6 challenges)

# Filtering Strings



First, I listed the tables in the database and found a table named details. I examined its structure and learned it had two columns: flag tag and blob. I then searched the blob column for values starting with pwn.college%, which directly returned the flag: pwn.college{8yz-mDgBKSZTiDiPAMw6p1-kHCv.01MwgDNxwCO2kjMzEzW}.

#### Filtering on Expressions

```
hacker@sql-playground~filtering-on-expressions:~$ /challenge/sql
sql> SELECT name FROM sqlite_master WHERE type='table';
 sql> SELECT name FROM sqlite_master WHERE type='table';

6ot 1 rows.

- {'name': 'fragments'}

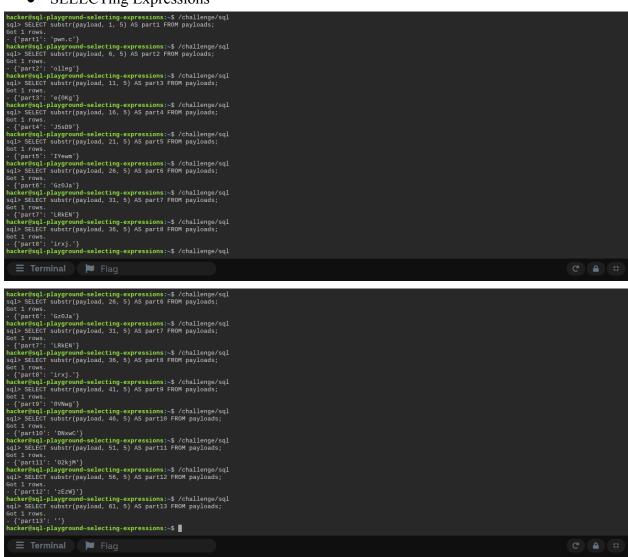
hacker@sql-playground~filtering-on-expressions:~$ /challenge/sql
sql> SELECT sql FROM sqlite_master WHERE name='fragments';
 Got 1 rows.
- {'sql': 'CREATE TABLE fragments(record)'}
- {'sql': 'CRAIL TABLE Tragments(record)'}
hacker@sql-playground-filtering-on-expressions:~$ SELECT record FROM fragments WHERE record LIKE 'pwn.college%' LIMIT 1;
bash: SELECT: command not found
hacker@sql-playground-filtering-on-expressions:~$ ^C
hacker@sql-playground-filtering-on-expressions:~$ /challenge/sql
sql> SELECT record FROM fragments WHERE substr(record, 1, [how_many_chars]) = 'pwn.college{' LIMIT 1;
SQL ERROR: no such column: how_many_chars
hacker@sql_playground-filtering.on-expressions:~$ /challenge/sql
 hacker@sql-playground~filtering-on-expressions:~$ /challenge/sql sql> SELECT record FROM fragments WHERE substr(record, 1, 12) = 'pwn.college{' LIMIT 1;
 Got 1 rows.
- {'record': 'pwn.college{UyvllM2_e1T-hdGebCN6o06JyiC.0FNwgDNxwC02kjMzEzW}'}
hacker@sql-playground~filtering-on-expressions:~$ ■
                                                         Flag
 hacker@sql-playground~filtering-on-expressions:~$ /challenge/sql
sql> SELECT name FROM sqlite_master WHERE type='table';
sq1> SELECT sq1 FROM sqlite_master WHERE name='fragments';
sq1> SELECT sq1 FROM sqlite_master WHERE name='fragments';
Got 1 rows.
- {'sql': 'CREATE TABLE fragments(record)'}
- {'sql': 'CREATE TABLE fragments(record)'}
hacker@sql-playground-filtering-on-expressions:~$ SELECT record FROM fragments WHERE record LIKE 'pwn.college%' LIMIT 1;
bash: SELECT: command not found
hacker@sql-playground-filtering-on-expressions:~$ ^C
hacker@sql-playground-filtering-on-expressions:~$ /challenge/sql
sql> SELECT record FROM fragments WHERE substr(record, 1, [how_many_chars]) = 'pwn.college{' LIMIT 1;
SQL ERROR: no such column: how_many_chars
hacker@sql-playground-filtering-on-expressions:~$ /challenge/sql
sql> SELECT record FROM fragments WHERE substr(record, 1, 12) = 'pwn.college{' LIMIT 1;
601 1 rows
 Got 1 rows.
- {'record': 'pwn.college{UyvllM2_e1T-hdGebCN6o06JyiC.0FNwgDNxwC02kjMzEzW}'}
hacker@sql-playground~filtering-on-expressions:~$ [
                                                                                         🎉 Successfully completed Filtering on Expressions! 🎉
```

I started by exploring the database structure and found a table named fragments with a single column record. I tried to query it using a LIKE clause but accidentally ran the command in bash instead of the SQL prompt, causing an error. After re-entering the SQL prompt, I attempted to use the substr function but mistakenly used a placeholder [how many chars], which SQLite interpreted as a column name, resulting in another error. I then corrected the query by counting the characters in 'pwn.college {' (which is 12) and used substr(record, 1, 12) = 'pwn.college {' to

UyvIIM2\_e1T-hdGebCN6o

filter the records. This successfully returned the flag: pwn.college{UyvllM2 e1T-hdGebCN6oO6JyiC.0FNwgDNxwCO2kjMzEzW}.

# • SELECTing Expressions



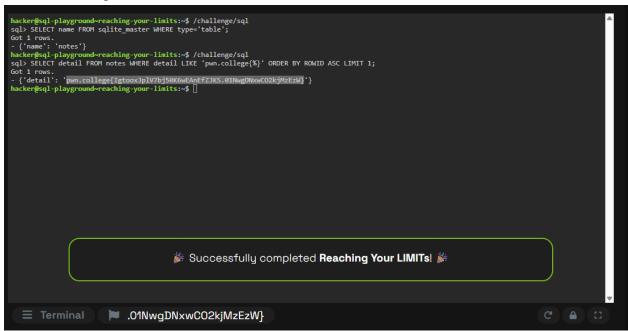
```
hacker@sql-playground-selecting-expressions:-$ /challenge/sql
sql> SELECT substr(payload, 26, 5) AS part6 FROM payloads;
Got 1 rows.
- (*part7: 'tREEN')
hacker@sql-playground-selecting-expressions:-$ /challenge/sql
sql> SELECT substr(payload, 33, 5) AS part7 FROM payloads;
(*part8: 'substr(payload, 36, 5) AS part8 FROM payloads;
fot 1 rows.
- (*part8: 'sixs).
- hacker@sql-playground-selecting-expressions:-$ /challenge/sql
sql> SELECT substr(payload, 36, 5) AS part8 FROM payloads;
fot 1 rows.
- (*part8: 'sixs).
- hacker@sql-playground-selecting-expressions:-$ /challenge/sql
sql> SELECT substr(payload, 41, 5) AS part9 FROM payloads;
fot 1 rows.
- (*part9: 'oWNeg')
hacker@sql-playground-selecting-expressions:-$ /challenge/sql
sql> SELECT substr(payload, 46, 5) AS part10 FROM payloads;
fot 1 rows.
- (*part10: 'ONXec')
hacker@sql-playground-selecting-expressions:-$ /challenge/sql
sql> SELECT substr(payload, 51, 5) AS part11 FROM payloads;
fot 1 rows.
- (*part11: 'OzkjM')
hacker@sql-playground-selecting-expressions:-$ /challenge/sql
sql> SELECT substr(payload, 56, 5) AS part12 FROM payloads;
fot 1 rows.
- (*part11: 'OzkjM')
hacker@sql-playground-selecting-expressions:-$ /challenge/sql
sql> SELECT substr(payload, 56, 5) AS part12 FROM payloads;
fot 1 rows.
- (*part11: 'OzkjM')
hacker@sql-playground-selecting-expressions:-$ /challenge/sql
sql> SELECT substr(payload, 56, 5) AS part12 FROM payloads;
fot 1 rows.
- (*part11: 'OzkjM')
hacker@sql-playground-selecting-expressions:-$ /challenge/sql
sql> SELECT substr(payload, 56, 5) AS part12 FROM payloads;
fot 1 rows.
- (*part11: 'OzkjM')
hacker@sql-playground-selecting-expressions:-$ /challenge/sql
sql> SELECT substr(payload, 56, 5) AS part12 FROM payloads;
for 1 rows.
- (*part11: 'OzkjM')
hacker@sql-playground-selecting-expressions:-$ /challenge/sql
sql> SELECT substr(payload, 56, 5) AS part12 FROM payloads;
for 1 rows.
- (*part12: 'Type sql> SELECT substr(payload, 56, 5) AS part12 FROM payloads;
for 1 rows.
- (*part12: 'Type sql> SELECT substr(payload, 56, 5) AS part12 FRO
```

I got the flag by carefully analyzing the table structure in SQLite. I noticed the table was named payloads and its single column was actually called payload, not payloads or sql. Since the system limited how much data I could read at once, I used substr() to extract the flag in small chunks of 5 characters at a time. By running consecutive queries and combining each part in order, I was able to reconstruct the full flag.

#### • Composite Conditions

I queried the sqlite\_master table to find the available tables and discovered the 'storage' table. Then, I examined the table structure and found it had columns 'flag\_tag' and 'note'. I searched for the flag by selecting the 'note' column where it matched the pattern 'pwn.college(%'. After correcting a syntax error, I retrieved the note containing the flag, which completed the challenge and displayed the flag in the terminal.

### Reaching Your LIMITs



I queried the sqlite\_master table to find the available tables and discovered the 'notes' table. Then, I searched for the flag by using a SELECT statement on the 'notes' table with a LIKE

clause to match the pattern 'pm.college(R)'. By ordering the results by RDAID ascending and limiting to one row, I retrieved the flag detail, which completed the challenge and displayed the flag in the terminal.

### • Querying Metadata

```
hacker@sql-playground~querying-metadata:~$ /challenge/sql
sql> SELECT name FROM sqlite_master WHERE type='table'
Got 1 rows.
- {'name': 'HxVoTANC'}
sql> SELECT sql FROM sqlite_master WHERE type='table' AND name='HxVoTANC';
off > oft in the squite_master which type table who fort i rows.
- { sql': 'CREATE TABLE HXVOTANC(solution)'}
hacker@sql-playground-querying-metadata:-≴ /challenge/sql
sql> SELECT name FROM sqlite_master WHERE type='table'
Got 1 rows.
- {'name': 'HmPSdOJv'}
sql> SELECT solution FROM HmPSdOJv ORDER BY rowid ASC LIMIT 1;
Sot 1 rows.
- {'solution': 'pwn.college{4sQ1 mhoqCnqy6t m2PC1FMznx6.0FOwgDNxwCO2kjMzEzW}'}
hacker@sql-playground~querying-metadata:~$ ■
                                           Flag
  hacker@sql-playground~querying-metadata:~$ /challenge/sql
sql> SELECT name FROM sqlite_master WHERE type='table'
 oft 1 rows.
- {'name': 'HxVoTANC'}
sql> SELECT sql FROM sqlite_master WHERE type='table' AND name='HxVoTANC';
 Got 1 rows.
-{'sql': 'CREATE TABLE HXVOTANC(solution)'}
hackengeal-playground-querying-metadata:-$ /challenge/sql
sql> SELECT name FROM sqlite_master WHERE type='table'
 Got 1 rows.

- {'name': 'HmPSdOJv'}

sql> SELECT solution FROM HmPSdOJv ORDER BY rowid ASC LIMIT 1;
 oft 1 rows.

- {'solution': 'pwn.college{4sQ1 mhoqCnqy6t m2PC1FMznx6.0FOwgDNxwCO2kjMzEzw}'}

hacker@sql-playground~querying-metadata:~$ □
                                                                            🎉 Successfully completed Querying Metadata! 🎉
                                             OFOwgDNxwCO2kjMzEzW}
```

I first queried the sqlite\_master table to find the name of the table that stored the flag, which was 'HmPSdO2v' in this session. Then, I retrieved the solution from that table using SELECT solution FROM HmPSdO2V ORDER BY rowId ASC LIMIT 1. After executing this query, the

challenge was completed, and the system displayed the flag in the terminal as OFOWgDNxwCO2kjMzEzW.