

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

```
hacker@sql-playground-sql-queries:~$ /challenge/sql
sql> SELECT name, type FROM sqlite_master WHERE type IN ('table','view');
Got 1 rows.
- {'name': 'details', 'type': 'table'}
hacker@sql-playground-sql-queries:~$ /challenge/sql
sql> SELECT * FROM details
Got 1 rows.
- {'text': 'pwn.college{Y2ZukQPcGmL06vbA5pwEiUlrGoK.0V05cDNbwCO2k}MzEzW}'}
hacker@sql-playground-sql-queries:~$
```

Terminal Flag

```
hacker@sql-playground-sql-queries:~$ /challenge/sql
sql> SELECT name, type FROM sqlite_master WHERE type IN ('table','view');
Got 1 rows.
- {'name': 'details', 'type': 'table'}
hacker@sql-playground-sql-queries:~$ /challenge/sql
sql> SELECT * FROM details
Got 1 rows.
- {'text': 'pwn.college{Y2ZukQPcGmL06vbA5pwEiUlrGoK.0V05cDNbwCO2k}MzEzW}'}
hacker@sql-playground-sql-queries:~$
```

Successfully completed SQL Queries!

Terminal Y2ZukQPcGmL06vbA5pwE

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';
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': 'HnlamzRuwxdkAZXeqrNYiGSiAniSCXqVsixOacPQVJIlegKutcfuUXelVvWi'}
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{wMvh7wEnhg38x8mhAYBuhZch2Rd.0FMwgDNbwCO2kjMzEzW}'}
hacker@sql-playground~filtering-sql:~$
```

```
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': 'HnlamzRuwxdkAZXeqrNYiGSiAniSCXqVsixOacPQVJIlegKutcfuUXelVvWi'}
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{wMvh7wEnhg38x8mhAYBuhZch2Rd.0FMwgDNbwCO2kjMzEzW}'}
hacker@sql-playground~filtering-sql:~$
```

🚩 Successfully completed **Filtering SQL!** 🚩

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';
Got 1 rows.
- {'name': 'resources'}
hacker@sql-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)'}
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
hacker@sql-playground~choosing-columns:~$ /challenge/sql
sql> SELECT flag FROM resources WHERE flag LIKE 'pwn.college%' LIMIT 1;
SQL ERROR: no such column: flag
hacker@sql-playground~choosing-columns:~$ /challenge/sql
sql> SELECT entry FROM resources WHERE entry LIKE 'pwn.college%' LIMIT 1;
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.0VMwgDNbxwCO2kjMzEzW}'}
hacker@sql-playground~choosing-columns:~$
```

```
hacker@sql-playground~choosing-columns:~$ /challenge/sql
sql> SELECT name FROM sqlite_master WHERE type='table';
Got 1 rows.
- {'name': 'resources'}
hacker@sql-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)'}
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
hacker@sql-playground~choosing-columns:~$ /challenge/sql
sql> SELECT flag FROM resources WHERE flag LIKE 'pwn.college%' LIMIT 1;
SQL ERROR: no such column: flag
hacker@sql-playground~choosing-columns:~$ /challenge/sql
sql> SELECT entry FROM resources WHERE entry LIKE 'pwn.college%' LIMIT 1;
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.0VMwgDNbxwCO2kjMzEzW}'}
hacker@sql-playground~choosing-columns:~$
```

🎉 Successfully completed **Choosing Columns!** 🎉

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

```
hacker@sql-playground-exclusionary-filtering:~$ /challenge/sql
sql> SELECT field FROM resources WHERE field LIKE 'pwn.college%' LIMIT 1;
SQL ERROR: no such table: resources
hacker@sql-playground-exclusionary-filtering:~$ /challenge/sql
sql> SELECT name FROM sqlite_master WHERE type='table';
Got 1 rows.
- {'name': 'storage'}
hacker@sql-playground-exclusionary-filtering:~$ /challenge/sql
sql> SELECT field FROM storage WHERE field LIKE 'pwn.college%' LIMIT 1;
SQL ERROR: no such column: field
hacker@sql-playground-exclusionary-filtering:~$ /challenge/sql
sql> SELECT field FROM storage WHERE field LIKE 'pwn.college%' AND field NOT LIKE 'garbage%' LIMIT 1;
SQL ERROR: no such column: field
hacker@sql-playground-exclusionary-filtering:~$ /challenge/sql
sql> SELECT sql FROM sqlite_master WHERE name='storage';
Got 1 rows.
- {'sql': 'CREATE TABLE storage(flag_tag,record)'}
hacker@sql-playground-exclusionary-filtering:~$ /challenge/sql
sql> SELECT record FROM storage WHERE record LIKE 'pwn.college%' LIMIT 1;
Got 1 rows.
- {'record': 'pwn.college{4CUrIZ36OLDPCvldBsJJeTk0i1F.0lMwgDNowC02kjMzEzW}'}
hacker@sql-playground-exclusionary-filtering:~$
```

```
hacker@sql-playground-exclusionary-filtering:~$ /challenge/sql
sql> SELECT field FROM resources WHERE field LIKE 'pwn.college%' LIMIT 1;
SQL ERROR: no such table: resources
hacker@sql-playground-exclusionary-filtering:~$ /challenge/sql
sql> SELECT name FROM sqlite_master WHERE type='table';
Got 1 rows.
- {'name': 'storage'}
hacker@sql-playground-exclusionary-filtering:~$ /challenge/sql
sql> SELECT field FROM storage WHERE field LIKE 'pwn.college%' LIMIT 1;
SQL ERROR: no such column: field
hacker@sql-playground-exclusionary-filtering:~$ /challenge/sql
sql> SELECT field FROM storage WHERE field LIKE 'pwn.college%' AND field NOT LIKE 'garbage%' LIMIT 1;
SQL ERROR: no such column: field
hacker@sql-playground-exclusionary-filtering:~$ /challenge/sql
sql> SELECT sql FROM sqlite_master WHERE name='storage';
Got 1 rows.
- {'sql': 'CREATE TABLE storage(flag_tag,record)'}
hacker@sql-playground-exclusionary-filtering:~$ /challenge/sql
sql> SELECT record FROM storage WHERE record LIKE 'pwn.college%' LIMIT 1;
Got 1 rows.
- {'record': 'pwn.college{4CUrIZ36OLDPCvldBsJJeTk0i1F.0lMwgDNowC02kjMzEzW}'}
hacker@sql-playground-exclusionary-filtering:~$
```

🎉 Successfully completed **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{4CUUn172360LDPCvldBsjJer[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

```
hacker@sql-playground~filtering-strings:~$ /challenge/sql
sql> SELECT name FROM sqlite_master WHERE type='table';
Got 1 rows.
- {'name': 'details'}
hacker@sql-playground~filtering-strings:~$ /challenge/sql
sql> SELECT sql FROM sqlite_master WHERE name='details';
Got 1 rows.
- {'sql': 'CREATE TABLE details(flag_tag,blob)'}
hacker@sql-playground~filtering-strings:~$ /challenge/sql
sql> SELECT blob FROM details WHERE blob LIKE 'pwn.college%' LIMIT 1;
Got 1 rows.
- {'blob': 'pwn.college{8yz-mDgBKSZTiDiPAMw6p1-kHCv.01MwgDNxwC02kjMzEzW}'}
hacker@sql-playground~filtering-strings:~$
```

```
hacker@sql-playground~filtering-strings:~$ /challenge/sql
sql> SELECT name FROM sqlite_master WHERE type='table';
Got 1 rows.
- {'name': 'details'}
hacker@sql-playground~filtering-strings:~$ /challenge/sql
sql> SELECT sql FROM sqlite_master WHERE name='details';
Got 1 rows.
- {'sql': 'CREATE TABLE details(flag_tag,blob)'}
hacker@sql-playground~filtering-strings:~$ /challenge/sql
sql> SELECT blob FROM details WHERE blob LIKE 'pwn.college%' LIMIT 1;
Got 1 rows.
- {'blob': 'pwn.college{8yz-mDgBKSZTiDiPAMw6p1-kHCv.01MwgDNxwC02kjMzEzW}'}
hacker@sql-playground~filtering-strings:~$
```

🎉 Successfully completed **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';
Got 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)'}
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;
Got 1 rows.
- {'record': 'pwn.college{UyvllM2_e1T-hdGebCN6o06JyiC.0FNwgDNxwCO2kjMzEzW}'}
hacker@sql-playground~filtering-on-expressions:~$
```

```
hacker@sql-playground~filtering-on-expressions:~$ /challenge/sql
sql> SELECT name FROM sqlite_master WHERE type='table';
Got 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)'}
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;
Got 1 rows.
- {'record': 'pwn.college{UyvllM2_e1T-hdGebCN6o06JyiC.0FNwgDNxwCO2kjMzEzW}'}
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

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, 1, 5) AS part1 FROM payloads;
Got 1 rows.
- {'part1': 'pwn.c'}
hacker@sql-playground-selecting-expressions:~$ /challenge/sql
sql> SELECT substr(payload, 6, 5) AS part2 FROM payloads;
Got 1 rows.
- {'part2': 'olleg'}
hacker@sql-playground-selecting-expressions:~$ /challenge/sql
sql> SELECT substr(payload, 11, 5) AS part3 FROM payloads;
Got 1 rows.
- {'part3': 'e{0Kg'}
hacker@sql-playground-selecting-expressions:~$ /challenge/sql
sql> SELECT substr(payload, 16, 5) AS part4 FROM payloads;
Got 1 rows.
- {'part4': 'J5sD9'}
hacker@sql-playground-selecting-expressions:~$ /challenge/sql
sql> SELECT substr(payload, 21, 5) AS part5 FROM payloads;
Got 1 rows.
- {'part5': 'IVewm'}
hacker@sql-playground-selecting-expressions:~$ /challenge/sql
sql> SELECT substr(payload, 26, 5) AS part6 FROM payloads;
Got 1 rows.
- {'part6': 'Gz0Ja'}
hacker@sql-playground-selecting-expressions:~$ /challenge/sql
sql> SELECT substr(payload, 31, 5) AS part7 FROM payloads;
Got 1 rows.
- {'part7': 'LRkEN'}
hacker@sql-playground-selecting-expressions:~$ /challenge/sql
sql> SELECT substr(payload, 36, 5) AS part8 FROM payloads;
Got 1 rows.
- {'part8': 'irxj.'}
hacker@sql-playground-selecting-expressions:~$ /challenge/sql
```

```
hacker@sql-playground-selecting-expressions:~$ /challenge/sql
sql> SELECT substr(payload, 26, 5) AS part6 FROM payloads;
Got 1 rows.
- {'part6': 'Gz0Ja'}
hacker@sql-playground-selecting-expressions:~$ /challenge/sql
sql> SELECT substr(payload, 31, 5) AS part7 FROM payloads;
Got 1 rows.
- {'part7': 'LRkEN'}
hacker@sql-playground-selecting-expressions:~$ /challenge/sql
sql> SELECT substr(payload, 36, 5) AS part8 FROM payloads;
Got 1 rows.
- {'part8': 'irxj.'}
hacker@sql-playground-selecting-expressions:~$ /challenge/sql
sql> SELECT substr(payload, 41, 5) AS part9 FROM payloads;
Got 1 rows.
- {'part9': '0VNwg'}
hacker@sql-playground-selecting-expressions:~$ /challenge/sql
sql> SELECT substr(payload, 46, 5) AS part10 FROM payloads;
Got 1 rows.
- {'part10': 'DNxwC'}
hacker@sql-playground-selecting-expressions:~$ /challenge/sql
sql> SELECT substr(payload, 51, 5) AS part11 FROM payloads;
Got 1 rows.
- {'part11': '02kJM'}
hacker@sql-playground-selecting-expressions:~$ /challenge/sql
sql> SELECT substr(payload, 56, 5) AS part12 FROM payloads;
Got 1 rows.
- {'part12': 'zEzW'}
hacker@sql-playground-selecting-expressions:~$ /challenge/sql
sql> SELECT substr(payload, 61, 5) AS part13 FROM payloads;
Got 1 rows.
- {'part13': ''}
hacker@sql-playground-selecting-expressions:~$ █
```

```
hacker@sql-playground-selecting-expressions:~$ /challenge/sql
sql> SELECT substr(payload, 26, 5) AS part6 FROM payloads;
Got 1 rows.
- {'part6': 'Gz0Ja'}
hacker@sql-playground-selecting-expressions:~$ /challenge/sql
sql> SELECT substr(payload, 31, 5) AS part7 FROM payloads;
Got 1 rows.
- {'part7': 'LRkEN'}
hacker@sql-playground-selecting-expressions:~$ /challenge/sql
sql> SELECT substr(payload, 36, 5) AS part8 FROM payloads;
Got 1 rows.
- {'part8': 'lrxj.'}
hacker@sql-playground-selecting-expressions:~$ /challenge/sql
sql> SELECT substr(payload, 41, 5) AS part9 FROM payloads;
Got 1 rows.
- {'part9': '0VNwg'}
hacker@sql-playground-selecting-expressions:~$ /challenge/sql
sql> SELECT substr(payload, 46, 5) AS part10 FROM payloads;
Got 1 rows.
- {'part10': 'DNxwC'}
hacker@sql-playground-selecting-expressions:~$ /challenge/sql
sql> SELECT substr(payload, 51, 5) AS part11 FROM payloads;
Got 1 rows.
- {'part11': '02kjM'}
hacker@sql-playground-selecting-expressions:~$ /challenge/sql
sql> SELECT substr(payload, 56, 5) AS part12 FROM payloads;
Got 1 rows.
- {'part12': 'zEz'}
hacker@sql-playground-selecting-expressions:~$ /challenge/sql
sql> SELECT substr(payload, 61, 5) AS part13 FROM payloads;
Got 1 rows.
- {'part13': ''}
hacker@sql-playground-selecting-expressions:~$
```

Successfully completed SELECTing Expressions! 🎉

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

```
hacker@sql-playground-composite-conditions:~$ /challenge/sql
sql> SELECT name FROM sqlite_master WHERE type='table';
Got 1 rows.
- {'name': 'storage'}
hacker@sql-playground-composite-conditions:~$ /challenge/sql
sql> SELECT sql FROM sqlite_master WHERE type='table' AND name='storage';
Got 1 rows.
- {'sql': 'CREATE TABLE storage(flag_tag,note)'}
hacker@sql-playground-composite-conditions:~$ /challenge/sql
sql> SELECT length((SELECT note FROM storage WHERE note LIKE 'pwn.college%' LIMIT 1));
Got 1 rows.
- {'length((SELECT note FROM storage WHERE note LIKE 'pwn.college%' LIMIT 1))': 60}
hacker@sql-playground-composite-conditions:~$ /challenge/sql
sql> SELECT FROM storage WHERE note LIKE 'pwn.college%' LIMIT 1;
SQL ERROR: near "FROM": syntax error
hacker@sql-playground-composite-conditions:~$ /challenge/sql
sql> SELECT note FROM storage WHERE note LIKE 'pwn.college%' LIMIT 1;
Got 1 rows.
- {'note': 'pwn.college{In6RmBQl0Gny5ugpwuilsF3jjgb.0lNwgDNbowC02kjMzEzW}'}
hacker@sql-playground-composite-conditions:~$
```



```
hacker@sql-playground-composite-conditions:~$ /challenge/sql
sql> SELECT name FROM sqlite_master WHERE type='table';
Got 1 rows.
- {'name': 'storage'}
hacker@sql-playground-composite-conditions:~$ /challenge/sql
sql> SELECT sql FROM sqlite_master WHERE type='table' AND name='storage';
Got 1 rows.
- {'sql': 'CREATE TABLE storage(flag_tag,note)'}
hacker@sql-playground-composite-conditions:~$ /challenge/sql
sql> SELECT length((SELECT note FROM storage WHERE note LIKE 'pwn.college{% ' LIMIT 1));
Got 1 rows.
- {'length((SELECT note FROM storage WHERE note LIKE 'pwn.college{% ' LIMIT 1))': 60}
hacker@sql-playground-composite-conditions:~$ /challenge/sql
sql> SELECT FROM storage WHERE note LIKE 'pwn.college{% ' LIMIT 1;
SQL ERROR: near "FROM": syntax error
hacker@sql-playground-composite-conditions:~$ /challenge/sql
sql> SELECT note FROM storage WHERE note LIKE 'pwn.college{% 'LIMIT 1;
Got 1 rows.
- {'note': 'pwn.college{In6RmBQl0Gny5ugpwuiwsF3jjgb.01NwgDNbwCO2k-jMzEzW}'}
hacker@sql-playground-composite-conditions:~$
```

🎉 Successfully completed **Composite Conditions!** 🎉

Terminal .01NwgDNbwCO2k-jMzEzW

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

```
hacker@sql-playground-reaching-your-limits:~$ /challenge/sql
sql> SELECT name FROM sqlite_master WHERE type='table';
Got 1 rows.
- {'name': 'notes'}
hacker@sql-playground-reaching-your-limits:~$ /challenge/sql
sql> SELECT detail FROM notes WHERE detail LIKE 'pwn.college{% ' ORDER BY ROWID ASC LIMIT 1;
Got 1 rows.
- {'detail': 'pwn.college{Igt0oxJp1V7bJ50K6wEAnEfZjKS.01NwgDNbwCO2k-jMzEzW}'}
hacker@sql-playground-reaching-your-limits:~$
```

🎉 Successfully completed **Reaching Your LIMITs!** 🎉

Terminal .01NwgDNbwCO2k-jMzEzW

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';
Got 1 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': 'HmPSdO3v'}
sql> SELECT solution FROM HmPSdO3v ORDER BY rowid ASC LIMIT 1;
Got 1 rows.
- {'solution': 'pwn.college{4sQ1_mhoqCnqy6t_m2PC1FMznx6.0F0wgDNxwCO2kjmZEzW}'}
hacker@sql-playground-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';
Got 1 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': 'HmPSdO3v'}
sql> SELECT solution FROM HmPSdO3v ORDER BY rowid ASC LIMIT 1;
Got 1 rows.
- {'solution': 'pwn.college{4sQ1_mhoqCnqy6t_m2PC1FMznx6.0F0wgDNxwCO2kjmZEzW}'}
hacker@sql-playground-querying-metadata:~$
```

🎉 Successfully completed **Querying Metadata!** 🎉

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.