

ATTACK DEFENSE LABS COURSES  
PENTESTER ACADEMY TOOL BOX PENTESTING  
JOINT WORLD-CLASS TRAINERS TRAINING HACKER  
TOOL BOX PATV HACKER  
HACKER PENTESTING  
PATV RED TEAM LABS ATTACK DEFENSE LABS  
TRAINING COURSES ACCESS POINT PENTESTER  
TEAM LABS PENTESTER TOOL BOX PENTESTING  
ACCESS POINT WORLD-CLASS TRAINERS TRAINING  
WORLD-CLASS TRAINERS  
ATTACK DEFENSE LABS TRAINING COURSES PATV ACCESS  
PENTESTER ACADEMY TOOL BOX PENTESTING  
ATTACK DEFENSE LABS TRAINING COURSES PENTESTER ACADEMY  
COURSES PENTESTER ACADEMY TOOL BOX PENTESTING  
TOOL BOX HACKER PENTESTING  
PATV RED TEAM LABS ATTACK DEFENSE LABS  
COURSES PENTESTER ACADEMY  
PENTESTER ACADEMY ATTACK DEFENSE LABS  
ATTACK DEFENSE LABS TRAINING COURSES  
WORLD-CLASS TRAINERS  
RED TEAM TRAINING COURSES  
PENTESTER ACADEMY TOOL BOX PENTESTING

# ATTACK DEFENSE

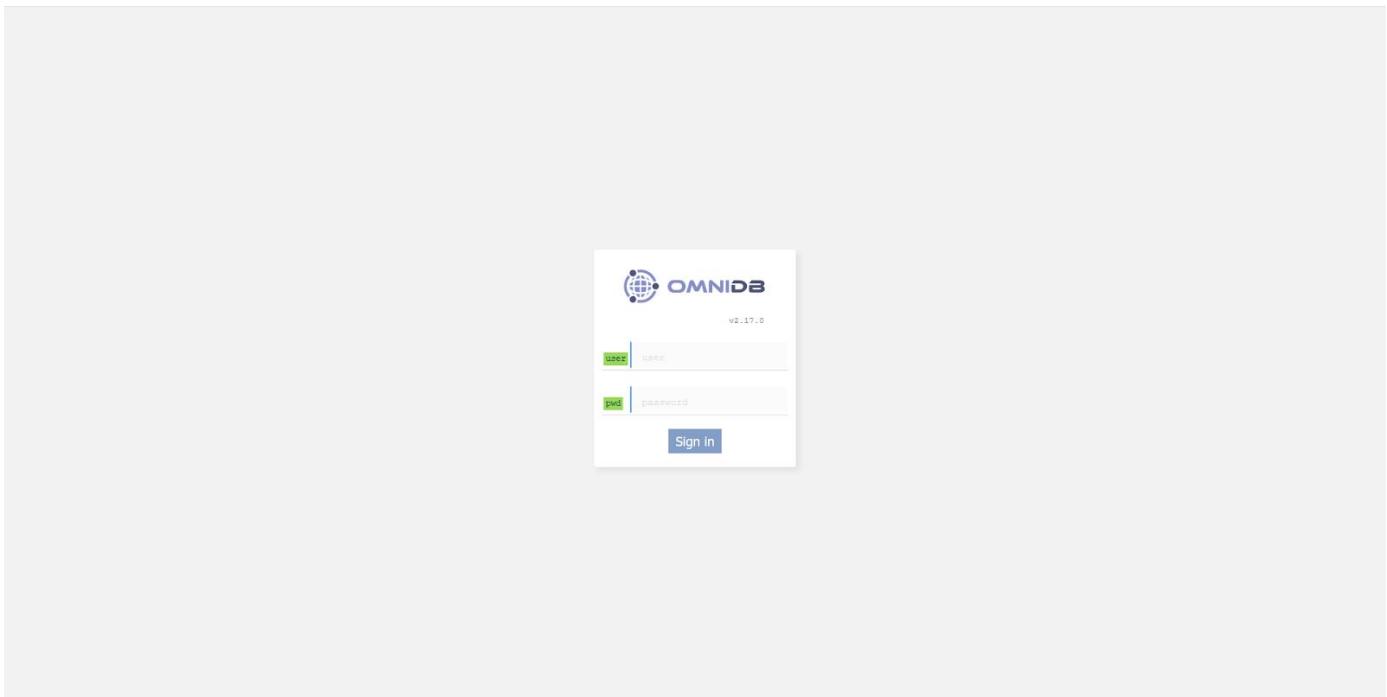
by PentesterAcademy

Name	SQL Basics
URL	<a href="https://attackdefense.com/challengedetails?cid=1801">https://attackdefense.com/challengedetails?cid=1801</a>
Type	Webapp Pentesting Basics

**Important Note:** This document illustrates all the important steps required to complete this lab. This is by no means a comprehensive step-by-step solution for this exercise. This is only provided as a reference to various commands needed to complete this exercise and for your further research on this topic. Also, note that the IP addresses and domain names might be different in your lab.

In this exercise, we will take a look at basic SQL queries this includes usage of SELECT, CREATE, INSERT, UPDATE, UNION and JOIN statements.

#### Home Page:



## Enter the Credentials:

**Username:** admin

**Password:** admin

The screenshot shows the Omnidb MySQL interface. On the left, the database structure is displayed under 'MySQL 5.5.56-log'. It includes the following databases: information\_schema, mysql, performance\_schema, test, and wpdatabase. Under wpdatabase, there are 13 tables: wp\_commentmeta, wp\_comments, wp\_links, wp\_options, wp\_posts, wp\_term\_relationships, wp\_term\_taxonomy, wp\_terms, wp\_usermeta, wp\_users, wp\_commentsmeta, wp\_commentmeta, and wp\_commentsmeta. A 'Properties' tab is visible at the bottom left, and a 'Data' tab is selected at the bottom center.

**Step 1:** Expand the Databases section on the Left Panel and click on any of the table from the wpdatabase.

This screenshot shows the same interface as above, but with the 'wp\_commentmeta' table highlighted in the 'Tables (13)' section of the left sidebar. The right panel shows the 'Console' tab with a single line of text '1'.

Properties	DDL
Property	Value
Table Schema	wpdatabase
Table Name	wp_commentmeta
Table Type	BASE TABLE
Engine	InnoDB
Version	10
Row Format	Compact
Table Rows	0
Average Row Length	0
Data Length	16384
Max Data Length	0
Index Length	32768

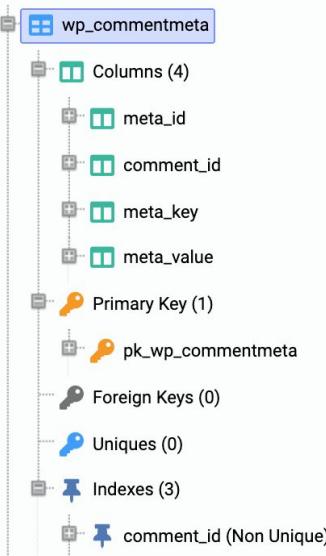
Information regarding the table is displayed in the Properties tab.

**Step 2:** Click on the DDL Tab.

Properties	DDL
	<pre> 1 CREATE TABLE `wp_commentmeta` ( 2   `meta_id` bigint(20) unsigned NOT NULL AUTO 3   `comment_id` bigint(20) unsigned NOT NULL D 4   `meta_key` varchar(255) COLLATE utf8mb4_uni 5   `meta_value` longtext COLLATE utf8mb4_unico 6   PRIMARY KEY (`meta_id`), 7   KEY `comment_id` (`comment_id`), 8   KEY `meta_key` (`meta_key`(191)) 9 ) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLA </pre>

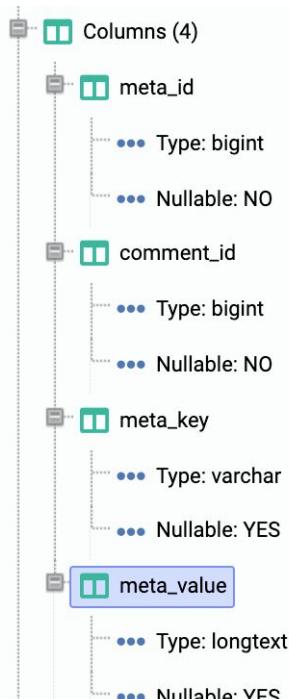
The table definition is mentioned on the DDL tab.

**Step 3:** Expand "wp\_commentmeta" table and the child tree.



The columns, primary key, foreign key and Indexes information can be accessed from the left section.

#### **Step 4:** Expand the columns.



The datatype and properties of the columns are displayed.

## Getting started with basic SQL Queries

**Task 1:** Identifying the databases present on the system.

**Query:** show databases;

The screenshot shows a MySQL command-line interface. At the top, there are tabs for 'Console' (selected), 'Query', and a '+' button. Below the tabs, a query window contains the command: '1 show databases;'. Underneath the query window is a large empty space. At the bottom, there is a summary bar with a play button, a refresh icon, and a list icon. The text reads: 'Number of records: 5', 'Start time: 05/06/2020 11:56:36', and 'Duration: 5.478 ms'. Below this is a 'Data' tab with a table titled 'Database'. The table has five rows, each containing a number from 1 to 5 and a database name: 'information\_schema', 'mysql', 'performance\_schema', 'test', and 'wpdatabase' respectively.

	Database
1	information_schema
2	mysql
3	performance_schema
4	test
5	wpdatabase

**Task 2:** Selecting a database.

**Query:** use wpdatabase;



```
Console Query +
1 use wpdatabase;
```

▶ ⏷ Start time: 05/06/2020 12:07:55 Duration: 1.815 ms

Data Done

**Task 3:** Listing tables in a database.

**Query:** show tables;

```
Console Query +
1 show tables;
```

▶ ⏷ Number of records: 13  
Start time: 05/06/2020 12:13:49 Duration: 0.988 ms

Data	
	Tables_in_wpdatabase
1	wp_commentmeta
2	wp_comments
3	wp_links
4	wp_options
5	wp_postmeta
6	wp_posts
7	wp_strong_views
8	wp_term_relationships
9	wp_term_taxonomy
10	wp_termmeta
11	wp_terms
12	wp_usermeta
13	wp_users

#### Task 4: Retrieving Data from the table.

**Query:** select \* from wp\_users;

The screenshot shows a MySQL database interface with the following details:

- Console Tab:** Shows the query: `select * from wp_users;`
- Execution Results:**
  - Number of records: 1
  - Start time: 05/06/2020 14:51:53 Duration: 1.773 ms
- Data Tab:** Displays the structure and data of the wp\_users table.

	ID	user_login	user_pass	user_nicename	user_email	user_url	user_registered	user_activation_key	user_status	display_name
1	1	admin	\$P\$Bf7EcG6x5hwZcW6DuxJ...	admin	admin@admin.xyz		2020-02-21 11:49:01		0	admin

**Task 5:** Retrieving data from only certain columns of a table.

**Query:** select user\_login,user\_pass from wp\_users;

The screenshot shows a MySQL Workbench interface. The top bar has tabs for 'Console' (selected), 'Query', and a '+' button. The main area contains a SQL query:

```
1 select user_login,user_pass from wp_users;
```

Below the query results, there are three status indicators: a play button, a refresh icon, and a three-dot menu icon. The text next to them reads: 'Number of records: 1', 'Start time: 05/06/2020 14:54:27', and 'Duration: 0.755 ms'. A 'Data' tab is visible at the bottom left.

	user_login	user_pass
1	admin	\$P\$Bf7EcG6x5hwZcW6DuxJ...

**Task 6:** Retrieving data from table by mentioning the database name.

**Query:** select \* from wpdatabase.wp\_posts;

The screenshot shows a MySQL Workbench interface. The top bar has tabs for 'Console' (selected), 'Query', and a '+' button. The main area contains a single line of SQL code:

```
1 select * from wpdatabase.wp_posts;
```

At the bottom, there are several status icons (play, stop, refresh, etc.) followed by the output of the query:

Number of records: 4  
Start time: 05/06/2020 15:40:56 Duration: 3.284 ms

CSV

Data												
	ID	post_author	post_date	post_date_gmt	post_content	post_title	post_excerpt	post_status	comment_status	ping_status	post_password	
1	1	1	2020-02-21 11:49:01	2020-02-21 11:49:01	<!-- wp:paragraph --><p>Welc...	Hello world!		publish	open	open		
2	2	1	2020-02-21 11:49:01	2020-02-21 11:49:01	<!-- wp:paragraph --><p>This ...	Sample Page		publish	closed	open		
3	3	1	2020-02-21 11:49:01	2020-02-21 11:49:01	<!-- wp:heading --><h2>Who ...	Privacy Policy		draft	closed	open		
4	4	1	2020-02-21 11:49:08	0000-00-00 00:00:00		Auto Draft		auto-draft	open	open		

**Task 7:** Sorting data on the basis of post title.

**Query:** select \* from wpdatabase.wp\_posts order by post\_title;



The screenshot shows the MySQL Workbench interface with a query editor and results viewer. The query editor contains the SQL command: `select * from wpdatabase.wp_posts order by post_title;`. The results viewer displays the sorted data from the previous table, showing four records ordered by post\_title.

Data												
	ID	post_author	post_date	post_date_gmt	post_content	post_title	post_excerpt	post_status	comment_status	ping_status	post_password	
1	4	1	2020-02-21 11:49:08	0000-00-00 00:00:00		Auto Draft		auto-draft	open	open		
2	1	1	2020-02-21 11:49:01	2020-02-21 11:49:01	<!-- wp:paragraph --><p>Welc...	Hello world!		publish	open	open		
3	3	1	2020-02-21 11:49:01	2020-02-21 11:49:01	<!-- wp:heading --><h2>Who ...	Privacy Policy		draft	closed	open		
4	2	1	2020-02-21 11:49:01	2020-02-21 11:49:01	<!-- wp:paragraph --><p>This ...	Sample Page		publish	closed	open		

Number of records: 4  
Start time: 05/06/2020 15:51:00 Duration: 2.436 ms

CSV  

**Task 8:** Retrieving first record from the table.

**Query:** select \* from wpdatabase.wp\_posts order by post\_title limit 1;

Console    Query    +

```
1 select * from wpdatabase.wp_posts order by post_title limit 1;
```

Number of records: 1  
Start time: 05/06/2020 15:53:30 Duration: 3.023 ms

CSV    ↗

Data													
	ID	post_author	post_date	post_date_gmt	post_content	post_title	post_excerpt	post_status	comment_status	ping_status	post_password	post_name	to_ping
1	4	1	2020-02-21 11:49:08	2020-02-21 11:49:08		Auto Draft		auto-draft	open	open			

**Task 9:** Retrieving a particular row from the table.

**Query:** select \* from wpdatabase.wp\_posts order by post\_title limit 2,1;

Console    Query    +

```
1 select * from wpdatabase.wp_posts order by post_title limit 2,1;
```

Number of records: 1  
Start time: 05/06/2020 15:56:21 Duration: 2.973 ms

CSV    ↗

Data													
	ID	post_author	post_date	post_date_gmt	post_content	post_title	post_excerpt	post_status	comment_status	ping_status	post_password	post_name	to_ping
1	3	1	2020-02-21 11:49:01	2020-02-21 11:49:01	<!-- wp:heading --><h2>Who ...	Privacy Policy		draft	closed	open			

**Task 10:** Create a new table "new\_users" with name, email and id as columns. Use the id column as a primary key.

**Query:** create table new\_users (

name varchar(100),  
email varchar(100),

```
    id varchar(100),  
    primary key(id)  
)
```

```
1 create table new_users (  
2     name varchar(100),  
3     email varchar(100),  
4     id varchar(100),  
5     primary key(id)  
6 )
```

Start time: 05/06/2020 16:04:12 Duration: 3.67 ms

CSV Data Done

### Task 11: Inserting rows into the table.

**Query:** insert into wpdatabase.new\_users values ("John","john@attackdefense.com","pa-01234");

```
1 insert into wpdatabase.new_users values ("John","john@attackdefense.com","pa-01234");
```

Start time: 05/06/2020 16:07:28 Duration: 3.569 ms

CSV Data Done

### Task 12: Updating record in the table.

**Query:** update wpdatabase.new\_users set name="james" where id="pa-01234";

```
1 update wpdatabase.new_users set name="james" where id="pa-01234";
```

Start time: 05/06/2020 16:09:07 Duration: 8.21 ms

CSV 

Data Done

**Task 13:** Combining records from two tables using UNION Statement, UNION statement requires equal number of columns to be selected from each table.

Checking columns in wp\_users table

**Query:** desc wpdatabase.wp\_users;

```
1 desc wpdatabase.wp_users;
```

Number of records: 10  
Start time: 05/06/2020 16:14:00 Duration: 2.784 ms

CSV 

	Field	Type	Null	Key	Default	Extra
1	ID	bigint(20) unsigned	NO	PRI		auto_increment
2	user_login	varchar(60)	NO	MUL		
3	user_pass	varchar(255)	NO			
4	user_nicename	varchar(50)	NO	MUL		
5	user_email	varchar(100)	NO	MUL		
6	user_url	varchar(100)	NO			
7	user_registered	datetime	NO		0000-00-00 00:00:00	
8	user_activation_key	varchar(255)	NO			
9	user_status	int(11)	NO		0	
10	display_name	varchar(250)	NO			

Checking columns in new\_users table (created in Task 10).

**Query:** desc wpdatabase.new\_users;

The screenshot shows the MySQL Workbench interface. At the top, there are tabs for 'Console', 'Query', and a '+' button. The 'Query' tab is selected. Below the tabs, a single line of code is shown: '1 desc wpdatabase.new\_users;'. Underneath the code, there is a large empty white area where the results will be displayed. At the bottom of the interface, there are several status indicators: a play button, a refresh icon, a number of records (3), a start time (05/06/2020 16:14:42), a duration (2.479 ms), and CSV export options.

Field	Type	Null	Key	Default	Extra
1 name	varchar(100)	YES			
2 email	varchar(100)	YES			
3 id	varchar(100)	NO	PRI		

Union query:

**Query:** select user\_login,user\_email from wpdatabase.wp\_users union select name,email from wpdatabase.new\_users;

The screenshot shows the MySQL Workbench interface. At the top, there are tabs for 'Console', 'Query', and a '+' button. The 'Query' tab is selected. Below the tabs, the query is shown: '1 select user\_login,user\_email from wpdatabase.wp\_users union select name,email from wpdatabase.new\_users;'. Underneath the code, there is a large empty white area where the results will be displayed. At the bottom of the interface, there are several status indicators: a play button, a refresh icon, a number of records (2), a start time (05/06/2020 16:20:41), a duration (2.192 ms), and CSV export options.

Data		
	user_login	user_email
1	admin	admin@admin.xyz
2	james	john@attackdefense.com

The data from the tables wp\_users and new\_users was presented in a single table.

**Task 14:** Insert more data into the new\_users table and use the JOIN statement to retrieve the users who have common name in new\_users and wp\_users table.

Inserting user named admin to new\_users table.

**Query:** insert into wpdatabase.new\_users values("admin","admin@admin.xyz","pa-02333");



The screenshot shows a MySQL database interface with a 'Query' tab selected. The query entered is:

```
1 insert into wpdatabase.new_users values("admin","admin@admin.xyz","pa-02333");
2
```

Below the query, it says 'Start time: 05/06/2020 16:29:41 Duration: 2.193 ms'. At the bottom, there are 'Data' and 'CSV' buttons, and the status 'Done'.

Similarly insert user "jimmy" and "david".

**Query:** insert into wpdatabase.new\_users  
values("jimmy","jimmy@attackdefense.com","pa-02323");



```
Console Query +  
1 insert into wpdatabase.new_users values("jimmy", "jimmy@attackdefense.com", "pa-02323");  
2
```

Start time: 05/06/2020 16:29:53 Duration: 6.719 ms

Data Done CSV

**Query:** insert into wpdatabase.new\_users  
values("david", "[david@attackdefense.com](mailto:david@attackdefense.com)", "pa-02663");

```
Console Query +  
1 insert into wpdatabase.new_users values("david", "david@attackdefense.com", "pa-02663");
```

Start time: 05/06/2020 16:30:05 Duration: 1.841 ms

Data Done CSV

Viewing records in table new\_users:

**Query:** select \* from wpdatabase.new\_users;

Console Query +

```
1 select * from wpdatabase.new_users
```

Number of records: 4  
Start time: 05/06/2020 16:30:26 Duration: 0.737 ms

CSV ↴ ↵

	name	email	id
1	james	john@attackdefense.com	pa-01234
2	jimmy	jimmy@attackdefense.com	pa-02323
3	admin	admin@admin.xyz	pa-02333
4	david	david@attackdefense.com	pa-02663

Using a JOIN statement to combine rows from new\_users and wp\_users table.

**Query:** select \*

```
from wpdatabase.wp_users
JOIN wpdatabase.new_users on
wpdatabase.new_users.name=wpdatabase.wp_users.user_login
```

Console Query +

```
1 select *
2 from wpdatabase.wp_users
3 JOIN wpdatabase.new_users on wpdatabase.new_users.name=wpdatabase.wp_users.user_login
```

Number of records: 1  
Start time: 05/06/2020 16:39:42 Duration: 2.545 ms

CSV ↴ ↵

	ID	user_login	user_pass	user_nicename	user_email	user_url	user_registered	user_activation_key	user_status	display_name	name	er
1	1	admin	\$P\$BF7EcG6x5hwZcW6DuxJ...	admin	admin@admin.xyz		2020-02-21 11:49:01		0	admin	admin	admin@o

The columns from both the tables are returned in the data.

**Task 15:** Using the join statement, display the value in user\_login and user\_pass field from the table wp\_users and display the id from new\_users table.

**Query:** select

```
wpdatabase.wp_users.user_login,wpdatabase.wp_users.user_pass,wpdatabase.new_users.id  
from wpdatabase.wp_users JOIN wpdatabase.new_users on  
wpdatabase.new_users.name=wpdatabase.wp_users.user_login
```

A screenshot of a MySQL database interface. The top bar shows tabs for 'Console' (selected), 'Query', and '+'. A blue bar indicates 'reconnecting...'. The main area contains a query window with the following code:

```
1 select wpdatabase.wp_users.user_login,wpdatabase.wp_users.user_pass,wpdatabase.new_users.id  
2 from wpdatabase.wp_users  
3 JOIN wpdatabase.new_users on wpdatabase.new_users.name=wpdatabase.wp_users.user_login
```

Below the query window, a message states 'Number of records: 1 Start time: 05/06/2020 16:45:07 Duration: 2.098 ms'. The bottom section is titled 'Data' and displays a single row in a table:

	user_login	user_pass	id
1	admin	\$P\$BF7EcG6x5hwZcW6DuxJ...	pa-02333

Buttons for CSV export and copy/paste are visible on the right.

**Task 16:** Use an alias "name" and "password" for the column user\_login,user\_pass respectively in the table wp\_users.

**Query:** select user\_login as name, user\_pass as password from wpdatabase.wp\_users;

A screenshot of a MySQL database interface. The top bar shows tabs for 'Console' (selected), 'Query', and '+'. A blue bar indicates 'reconnecting...'. The main area contains a query window with the following code:

```
1 select user_login as name, user_pass as password from wpdatabase.wp_users;
```

Below the query window, a message states 'Number of records: 1 Start time: 05/06/2020 16:52:39 Duration: 2.16 ms'. The bottom section is titled 'Data' and displays a single row in a table:

	name	password
1	admin	\$P\$BF7EcG6x5hwZcW6DuxJ...

Buttons for CSV export and copy/paste are visible on the right.

Data		
	name	password
1	admin	\$P\$Bf7EcG6x5hwZcW6DuxJ...

**Task 17:** Identify the distinct authors who have published a post. The post\_author column in wp\_posts table consists the id of authors.

**Query:** select distinct(post\_author) from wpdatabase.wp\_posts;



The screenshot shows a MySQL query interface. The 'Query' tab is selected, displaying the SQL command: 'select distinct(post\_author) from wpdatabase.wp\_posts;'. Below the query, the results are shown in a table titled 'Data'. The table has one row with a value of 1 in the 'post\_author' column. The status bar at the bottom indicates 'Number of records: 1', 'Start time: 05/06/2020 16:58:20', and 'Duration: 0.554 ms'.

Data	
	post_author
1	1

Only 1 author has published a post.

**Task 18:** Count the posts stored in the wp\_posts table. Use the COUNT aggregate function.

**Query:** select count(\*) from wpdatabase.wp\_posts;



```
>_ Console Query +  
1 select count(*) from wpdatabase.wp_posts;  
  
Number of records: 1  
Start time: 05/06/2020 17:00:31 Duration: 3.199 ms  
  
Data  
count(*)  
1 4
```

CSV ↴ ↵

**Task 19:** Show all the data of the posts which have the status "publish".

**Query:** select \* from wpdatabase.wp\_posts where post\_status='publish';



```
>_ Console Query +  
1 select * from wpdatabase.wp_posts where post_status='publish';  
  
Number of records: 2  
Start time: 05/06/2020 17:14:54 Duration: 3.392 ms  
  
Data  
ID post_author post_date post_date_gmt post_content post_title post_excerpt post_status comment_status ping_status  
1 1 1 2020-02-21 11:49:01 2020-02-21 11:49:01 <!- wp:paragraph --> <p>Hello world!</p> Hello world!  
2 2 1 2020-02-21 11:49:01 2020-02-21 11:49:01 <!- wp:paragraph --> <p>This is a sample page.</p> Sample Page
```

CSV ↴ ↵

**Task 20:** Show all the data of the posts which have "wp:paragraph" in it's content. Use the Like clause

**Query:** select \* from wpdatabase.wp\_posts where post\_content like '%wp:paragraph%';

The screenshot shows a MySQL query interface with a blue header bar. The 'Query' tab is selected. Below the header, there is a code editor containing the SQL query: 'select \* from wpdatabase.wp\_posts where post\_content like '%wp:paragraph%' ;'. At the bottom of the interface, there is a summary: 'Number of records: 3', 'Start time: 05/06/2020 17:19:14 Duration: 3.623 ms', and a 'Data' tab. The 'Data' tab displays a table with 3 rows of data. The columns are: ID, post\_author, post\_date, post\_date\_gmt, post\_content, post\_title, post\_excerpt, post\_status, comment\_status, and ping\_status. The data is as follows:

	ID	post_author	post_date	post_date_gmt	post_content	post_title	post_excerpt	post_status	comment_status	ping_status
1	1	1	2020-02-21 11:49:01	2020-02-21 11:49:01	<!-- wp:paragraph --> <p>Welc...	Hello world!		publish	open	open
2	2	1	2020-02-21 11:49:01	2020-02-21 11:49:01	<!-- wp:paragraph --> <p>This ...	Sample Page		publish	closed	open
3	3	1	2020-02-21 11:49:01	2020-02-21 11:49:01	<!-- wp:heading --><h2>Who ...	Privacy Policy		draft	closed	open

**Task 21:** Show all the data for the posts which have post\_status set to "publish" and comment\_status set to "closed".

**Query:** select \* from wpdatabase.wp\_posts where post\_status="publish" and comment\_status="closed"

The screenshot shows a MySQL query interface with a blue header bar. The 'Query' tab is selected. Below the header, there is a code editor containing the SQL query: 'select \* from wpdatabase.wp\_posts where post\_status="publish" and comment\_status="closed"' . At the bottom of the interface, there is a summary: 'Number of records: 1', 'Start time: 05/06/2020 17:22:09 Duration: 1.814 ms', and a 'Data' tab. The 'Data' tab displays a table with 1 row of data. The columns are: ID, post\_author, post\_date, post\_date\_gmt, post\_content, post\_title, post\_excerpt, post\_status, comment\_status, and ping\_status. The data is as follows:

	ID	post_author	post_date	post_date_gmt	post_content	post_title	post_excerpt	post_status	comment_status	ping_status
1	1	1	2020-02-21 11:49:01	2020-02-21 11:49:01	<!-- wp:paragraph --> <p>Hello world!	Hello world!		publish	closed	

The Data present in the table can also be viewed by right clicking the table name from the left panel and selecting the Query data option.

The screenshot shows the MySQL Workbench interface. On the left, the object browser displays various database tables under the 'wp\_posts' schema. A context menu is open over the 'wp\_posts' table, with 'Query Data' highlighted. The main pane shows a query results grid with one record and some performance metrics. The bottom pane shows the DDL for creating the 'wp\_posts' table.

Number of records: 1  
Start time: 05/06/2020 17:22:09 Duration: 1.814 ms

ID	post_author	post_date	post_date_gmt	post_content
1	2	1	2020-02-21 11:49:01	2020-02-21 11:49:01

Properties DDL

```
1 CREATE TABLE `wp_posts` (
2   `ID` bigint(20) unsigned NOT NULL,
3   `post_author` bigint(20) unsigned NOT NULL,
4   `post_date` datetime NOT NULL DEFAULT CURRENT_TIMESTAMP,
5   `post_date_gmt` datetime NOT NULL DEFAULT CURRENT_TIMESTAMP,
6   `post_content` longtext COLLATE utf8mb4_unicode_ci
7 ) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4;
```

The screenshot shows the OmniDB interface. On the left, a tree view displays database schema information for the 'wpdatabase' database, specifically for the 'wp\_posts' table, which has 23 columns. The main area shows a query window with the following SQL code:

```

1 SELECT t.ID
2      , t.post_author
3      , t.post_date
4      , t.post_date_gmt
5      , t.post_content
6      , t.post_title
7      , t.post_excerpt
8      , t.post_status
9      , t.comment_status
10     , t.ping_status
11     , t.post_password

```

Below the query, it says "Number of records: 4" and "Start time: 05/06/2020 17:24:06 Duration: 6.933 ms". The results are displayed in a table titled "Data":

	ID	post_author	post_date	post_date_gmt	post_content	post_title	post_excerpt	post_status	comment_status	ping_status
1	1	1	2020-02-21 11:49:01	2020-02-21 11:49:01	<!- wp:paragraph --> <p>Welc...	Hello world!		publish	open	open
2	2	1	2020-02-21 11:49:01	2020-02-21 11:49:01	<!- wp:paragraph --> <p>This ...	Sample Page		publish	closed	open
3	3	1	2020-02-21 11:49:01	2020-02-21 11:49:01	<!- wp:heading --><h2>Who ...	Privacy Policy		draft	closed	open
4	4	1	2020-02-21 11:49:08	0000-00-00 00:00:00		Auto Draft		auto-draft	open	open

CSV and copy/paste buttons are available at the bottom right.

Similarly, OmniDB also provides option to EDIT table ,INSERT, Update records:

### EDIT Table:

The screenshot shows the OmniDB interface. The schema tree on the left is identical to the previous one. The main area shows a query window with the following SQL code:

```

select * from wpdatabase.wp_posts t
1 order by t.ID

```

Below the query, it says "Query 10 rows" and "Number of records: 4 Response time: 0.282 seconds". The results are displayed in a table:

	ID (bigint)	post_author (bigint)	post_date (datetime)	post_date_gmt (datetime)	post_content (longtext)	post_title (text)	post_excerpt (text)	post_status (text)
1	<del>x</del>	1	1	2020-02-21 11:49:01	2020-02-21 11:49:01	<!- wp:paragraph --> <p>Welc...	Hello world!	publish
2	<del>x</del>	2	1	2020-02-21 11:49:01	2020-02-21 11:49:01	<!- wp:paragraph --> <p>This ...	Sample Page	publish
3	<del>x</del>	3	1	2020-02-21 11:49:01	2020-02-21 11:49:01	<!- wp:heading --><h2>Who ...	Privacy Policy	draft
4	<del>x</del>	4	1	2020-02-21 11:49:08	0000-00-00 00:00:00		Auto Draft	auto-draft
5	+							

## Insert record:

The screenshot shows the MySQL Workbench interface. On the left, the object browser displays the database structure for the 'wpdatabase' database, specifically the 'wp\_posts' table which has 23 columns. On the right, the 'Console' tab contains the following SQL code:

```
1 INSERT INTO wpdatabase.wp_posts (
2     ID
3     , post_author
4     , post_date
5     , post_date_gmt
6     , post_content
7     , post_title
8     , post_excerpt
9     , post_status
10    , comment_status
11    , ping_status
```

The 'Data' tab at the bottom is selected.

## Update record:

The screenshot shows the MySQL Workbench interface. On the left, the object browser displays the database structure for the 'wpdatabase' database, specifically the 'wp\_posts' table which has 23 columns. On the right, the 'Console' tab contains the following SQL code:

```
1 UPDATE wpdatabase.wp_posts
2 SET ID = ? -- bigint PRIMARY KEY
3     , post_author = ? -- bigint
4     , post_date = ? -- datetime
5     , post_date_gmt = ? -- datetime
6     , post_content = ? -- longtext
7     , post_title = ? -- text
8     , post_excerpt = ? -- text
9     , post_status = ? -- varchar
10    , comment_status = ? -- varchar
11    , ping_status = ? -- varchar
```

The 'Data' tab at the bottom is selected.

## Getting Started with PHPMyAdmin

**Login Page:**



Welcome to phpMyAdmin

**Log in** (SSL)

Username:

Password:

**Go**

The login credentials are mentioned in the challenge description.

Username: root

Password: Blank or <No Password>

**Main Dashboard:**

The screenshot shows the phpMyAdmin interface with the following details:

- Left Panel:** Shows a tree view of databases: New, information\_schema, mysql, performance\_schema, test, and wpdatabase.
- General settings:**
  - Change password
  - Server connection collation: utf8mb4\_unicode\_ci
- Appearance settings:**
  - Theme: pmahomme
  - Font size: 82%
  - More settings
- Database server:**
  - Server: 127.0.0.1 via TCP/IP
  - Server type: MySQL
  - Server connection: SSL is not being used
  - Server version: 5.5.56-log - MySQL Community Server (GPL)
  - Protocol version: 10
  - User: root@localhost
  - Server charset: UTF-8 Unicode (utf8)
- Web server:**
  - nginx/1.10.3
  - Database client version: libmysql - mysqld 5.0.12-dev - 20150407-\$Id: b5c5906d452ec590732a93b051f3827e02749b83\$
  - PHP extension: mysqli curl mbstring
  - PHP version: 7.0.33-12+ubuntu16.04.1+deb.sury.org+1
- phpMyAdmin:**
  - Version information: 4.8.1
  - Git revision: 15ca375 from master branch, committed on Dec 14, 2013 at 06:50 PM by Fernando
  - Documentation
  - Official Homepage
  - Contribute
  - Get support
  - List of changes
  - License

The databases are listed on the Left Panel. Expand the wpdatabase.

The screenshot shows the phpMyAdmin interface with the following details:

- Left Panel:** Shows a tree view of databases: New, information\_schema, mysql, performance\_schema, test, and wpdatabase. The wpdatabase node is expanded, showing its sub-tables: new, new\_users, wp\_commentmeta, wp\_comments, wp\_links, wp\_options, wp\_postmeta, wp\_posts, wp\_strong\_views, wp\_termmeta, wp\_terms, wp\_term\_relationships, wp\_term\_taxonomy, wp\_usermeta, and wp\_users.
- General settings:**
  - Change password
  - Server connection collation: utf8mb4\_unicode\_ci
- Appearance settings:**
  - Theme: pmahomme
  - Font size: 82%
  - More settings

All the tables present inside the database are displayed. Expand the wp\_users table.

The screenshot shows the phpMyAdmin interface with the 'wp\_users' table selected. On the left, a tree view displays the table structure, including 'Columns' (New, display\_name, ID, user\_activation\_key, user\_email, user\_login, user\_nicename, user\_pass, user\_registered, user\_status, user\_url) and 'Indexes' (New, PRIMARY, user\_email, user\_login\_key, user\_nicename). A message box at the bottom right states: 'The phpMyAdmin configuration storage is not completely configured. Or alternately go to 'Operations' tab of any database to set it up there.' A 'Console' button is also visible at the bottom right of the message box.

The information regarding the columns and the indexes are displayed on the left panel.

### Clicking on the Databases Tab.

The screenshot shows the 'Databases' tab in the phpMyAdmin interface. The top navigation bar includes tabs for 'Databases', 'SQL', 'Status', 'User accounts', 'Export', 'Import', 'Settings', 'Binary log', 'Replication', 'Variables', and 'More'. The 'Databases' tab is active. Below the tabs, a 'Create database' button and a 'Database name' input field containing 'latin1\_swedish\_ci' are shown. A 'Create' button is next to the input field. The main area displays a table of databases:

Database	Collation	Master replication	Action
information_schema	utf8_general_ci	✓ Replicated	<input type="button" value="Check privileges"/>
mysql	latin1_swedish_ci	✓ Replicated	<input type="button" value="Check privileges"/>
performance_schema	utf8_general_ci	✓ Replicated	<input type="button" value="Check privileges"/>
test	latin1_swedish_ci	✓ Replicated	<input type="button" value="Check privileges"/>
wpdatabase	latin1_swedish_ci	✓ Replicated	<input type="button" value="Check privileges"/>
Total: 5	latin1_swedish_ci		

At the bottom, there are buttons for 'Check all' and 'With selected: Drop'.

All the Databases are listed on the web page.

### Click on the wpdatabase Table

The screenshot shows the MySQL Workbench interface for a database named 'wpdatabase'. The main window displays a table of 14 tables within the database. The columns in the table include 'Table', 'Action' (with icons for Browse, Structure, Search, Insert, Empty, Drop), 'Rows', 'Type', 'Collation', 'Size', and 'Overhead'. The 'wp\_users' table is highlighted in the list. Below the table, there are buttons for 'Check all' and 'With selected'. At the bottom, there are links for 'Print' and 'Data dictionary', and a 'Create table' button with fields for 'Name:' and 'Number of columns: 4'. A 'Go' button is also present.

Table	Action	Rows	Type	Collation	Size	Overhead
new_users	Browse Structure Search Insert Empty Drop	4	InnoDB	latin1_swedish_ci	16 Kib	-
wp_commentmeta	Browse Structure Search Insert Empty Drop	0	InnoDB	utf8mb4_unicode_ci	48 Kib	-
wp_comments	Browse Structure Search Insert Empty Drop	1	InnoDB	utf8mb4_unicode_ci	96 Kib	-
wp_links	Browse Structure Search Insert Empty Drop	0	InnoDB	utf8mb4_unicode_ci	32 Kib	-
wp_options	Browse Structure Search Insert Empty Drop	146	InnoDB	utf8mb4_unicode_ci	144 Kib	-
wp_postmeta	Browse Structure Search Insert Empty Drop	2	InnoDB	utf8mb4_unicode_ci	48 Kib	-
wp_posts	Browse Structure Search Insert Empty Drop	4	InnoDB	utf8mb4_unicode_ci	80 Kib	-
wp_strong_views	Browse Structure Search Insert Empty Drop	0	InnoDB	utf8mb4_unicode_ci	16 Kib	-
wp_termmeta	Browse Structure Search Insert Empty Drop	0	InnoDB	utf8mb4_unicode_ci	48 Kib	-
wp_terms	Browse Structure Search Insert Empty Drop	1	InnoDB	utf8mb4_unicode_ci	48 Kib	-
wp_term_relationships	Browse Structure Search Insert Empty Drop	1	InnoDB	utf8mb4_unicode_ci	32 Kib	-
wp_term_taxonomy	Browse Structure Search Insert Empty Drop	1	InnoDB	utf8mb4_unicode_ci	48 Kib	-
wp_usermeta	Browse Structure Search Insert Empty Drop	17	InnoDB	utf8mb4_unicode_ci	48 Kib	-
wp_users	Browse Structure Search Insert Empty Drop	1	InnoDB	utf8mb4_unicode_ci	64 Kib	-
14 tables	Sum	178	InnoDB	latin1_swedish_ci	768 Kib	0 B

The tables inside the database wpdatabase are displayed.

### Click on the Table wp\_users

Showing rows 0 - 0 (1 total, Query took 0.0002 seconds.)

```
SELECT * FROM `wp_users`
```

Profiling [Edit inline] [ Edit ] [ Explain SQL ] [ Create PHP code ] [ Refresh ]

ID	user_login	user_pass	user_nicename	user_email	user_url	user_registered	user_activation_key	user_s
1	admin	\$P\$Bf7EcG6x5hwZcW6DuxJ5bTvhbir3jl.	admin	admin@admin.xyz		2020-02-21 11:49:01		

Query results operations

Print Copy to clipboard Export Display chart Create view

The record of the tables are displayed on the web page.

**Click on the Structure tab to see the structural information of the tables.**

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
1	ID	bignum(20)		UNSIGNED	No	None	AUTO_INCREMENT		<input type="button" value="Change"/> <input type="button" value="Drop"/> More
2	user_login	varchar(60)	utf8mb4_unicode_ci		No				<input type="button" value="Change"/> <input type="button" value="Drop"/> More
3	user_pass	varchar(255)	utf8mb4_unicode_ci		No				<input type="button" value="Change"/> <input type="button" value="Drop"/> More
4	user_nicename	varchar(50)	utf8mb4_unicode_ci		No				<input type="button" value="Change"/> <input type="button" value="Drop"/> More
5	user_email	varchar(100)	utf8mb4_unicode_ci		No				<input type="button" value="Change"/> <input type="button" value="Drop"/> More
6	user_url	varchar(100)	utf8mb4_unicode_ci		No				<input type="button" value="Change"/> <input type="button" value="Drop"/> More
7	user_registered	datetime			No	0000-00-00 00:00:00			<input type="button" value="Change"/> <input type="button" value="Drop"/> More
8	user_activation_key	varchar(255)	utf8mb4_unicode_ci		No				<input type="button" value="Change"/> <input type="button" value="Drop"/> More
9	user_status	int(11)			No	0			<input type="button" value="Change"/> <input type="button" value="Drop"/> More
10	display_name	varchar(250)	utf8mb4_unicode_ci		No				<input type="button" value="Change"/> <input type="button" value="Drop"/> More

Table structure Relation view

Add 1 column(s) after display\_name Go

Indexes

Action	Keyname	Type	Unique	Packed	Column	Cardinality	Collation	Null	Comment
<input type="button" value="Edit"/> <input type="button" value="Drop"/>	PRIMARY	BTREE	Yes	No	ID	1	A	No	
<input type="button" value="Edit"/> <input type="button" value="Drop"/>	user_login_key	BTREE	No	No	user_login	1	A	No	
<input type="button" value="Edit"/> <input type="button" value="Drop"/>	user_nicename	BTREE	No	No	user_nicename	1	A	No	
<input type="button" value="Edit"/> <input type="button" value="Drop"/>	user_email	BTREE	No	No	user_email	1	A	No	

Create an index on 1 columns Go

**Click on the SQL tab.**

The screenshot shows the MySQL Workbench interface. The top navigation bar indicates the connection is to 'Server: 127.0.0.1' and the database is 'wpdatabase'. The current table is 'wp\_users'. Below the navigation bar, there are several tabs: 'Browse', 'Structure', 'SQL' (which is selected), 'Search', 'Insert', 'Export', 'Import', 'Privileges', 'Operations', and 'Triggers'. The main area has a title 'Run SQL query/queries on table wpdatabase.wp\_users: [?](#)' and contains the following SQL query:

```
1 SELECT * FROM `wp_users` WHERE 1
```

Below the query, there are several buttons: 'SELECT \*', 'SELECT', 'INSERT', 'UPDATE', 'DELETE', 'Clear', and 'Format'. There is also a link 'Get auto-saved query' and a checkbox 'Bind parameters [?](#)'. At the bottom of the SQL pane, there are options for 'Delimiter' (set to ';'), checkboxes for 'Show this query here again' (checked), 'Retain query box', 'Rollback when finished', and 'Enable foreign key checks' (checked), and a 'Go' button.

To the right of the SQL pane, there is a 'Columns' section listing the fields for the wp\_users table:

ID
user_login
user_pass
user_nicename
user_email
user_url
user_registered
user_activation_key
user_status
display_name

The SQL queries can be executed on the SQL tab. The same queries which were executed on the OmniDB SQL console can be executed here. The various buttons provided on the page "select", "insert", "update" can be used to change the statement.

## Click on the Search Tab

The screenshot shows the MySQL Workbench interface with the following details:

- Server: 127.0.0.1 » Database: wpdatabase » Table: wp\_users
- Tab bar: Browse, Structure, SQL, Search, Insert, Export, Import, Privileges, Operations, Triggers.
- Search tab selected.
- Query builder: "Do a 'query by example' (wildcard: "%")".
- Table structure:

Column	Type	Collation	Operator	Value
ID	bigint(20)		=	<input type="text"/>
user_login	varchar(60)	utf8mb4_unicode_ci	LIKE	<input type="text"/>
user_pass	varchar(255)	utf8mb4_unicode_ci	LIKE	<input type="text"/>
user_nicename	varchar(50)	utf8mb4_unicode_ci	LIKE	<input type="text"/>
user_email	varchar(100)	utf8mb4_unicode_ci	LIKE	<input type="text"/>
user_url	varchar(100)	utf8mb4_unicode_ci	LIKE	<input type="text"/>
user_registered	datetime		=	<input type="text"/> <input type="button" value="Go"/>
user_activation_key	varchar(255)	utf8mb4_unicode_ci	LIKE	<input type="text"/>
user_status	int(11)		=	<input type="text"/>
display_name	varchar(250)	utf8mb4_unicode_ci	LIKE	<input type="text"/>

- Buttons: + Options, Go.

The search tab provides an easy way to search the table instead of writing SELECT statements with where clause. The values can be directly set to value with various operators.

## Click on the Insert Tab:

← Server: 127.0.0.1 » Database: wpdatabase » Table: wp\_users

Browse Structure SQL Search Insert Export Import Privileges Operations Triggers

Column	Type	Function	Null	Value
ID	bigint(20) unsigned			
user_login	varchar(60)			
user_pass	varchar(255)			
user_nicename	varchar(50)			
user_email	varchar(100)			
user_url	varchar(100)			
user_registered	datetime			0000-00-00 00:00:00 
user_activation_key	varchar(255)			
user_status	int(11)			0
display_name	varchar(250)			

Ignore Go

Column	Type	Function	Null	Value
ID	bigint(20) unsigned			
user_login	varchar(60)			
user_pass	varchar(255)			
Console name	varchar(50)			

The Insert tab functionality can be used to insert records into the table.

**Click on the Export Tab:**

The screenshot shows the phpMyAdmin interface with the following details:

- Server: 127.0.0.1
- Database: wpdatabase
- Table: wp\_users
- Export Tab is selected (highlighted in blue).
- Other tabs visible include Browse, Structure, SQL, Search, Insert, Import, Privileges, Operations, and Triggers.
- Main content area: "Exporting rows from \"wp\_users\" table".
- Export method section:
  - Quick - display only the minimal options (selected)
  - Custom - display all possible options
- Format section: SQL dropdown menu set to SQL.
- Action button: Go.

Specific Table can be exported from the export table.

Click on the Import section

The screenshot shows the phpMyAdmin interface for a MySQL database named 'wpdatabase'. The current table is 'wp\_users'. The top navigation bar includes links for Browse, Structure, SQL, Search, Insert, Export, Import, Privileges, Operations, and Triggers. The 'Import' tab is selected. Below the tabs, the title 'Importing into the table "wp\_users"' is displayed. The 'File to import:' section contains instructions about compressed files and a file selection input field ('Choose File') which shows 'No file chosen' (Max: 2,048KiB). It also allows dragging and dropping files. The 'Character set of the file:' dropdown is set to 'utf-8'. The 'Partial import:' section has a checked checkbox for allowing interruptions. The 'Other options:' section has a checked checkbox for enabling foreign key checks. The 'Format:' dropdown is set to 'SQL'. The 'Format-specific options:' section includes a 'SQL compatibility mode:' dropdown set to 'NONE' and a checked checkbox for 'Do not use AUTO\_INCREMENT for zero values'. At the bottom left are 'Go' and 'Console' buttons.

The import section can be used to import an SQL archive file into the tables.

**Click on the Privileges Tab**

Server: 127.0.0.1 » Database: wpdatabase » Table: wp\_users

Browse Structure SQL Search Insert Export Import Privileges Operations Triggers

Users having access to "wpdatabase.wp\_users"

User name	Host name	Type	Privileges	Grant	Action
root	127.0.0.1	global	ALL PRIVILEGES	Yes	Edit privileges Export
root	27ae0e316043	global	ALL PRIVILEGES	Yes	Edit privileges Export
root	::1	global	ALL PRIVILEGES	Yes	Edit privileges Export
root	localhost	global	ALL PRIVILEGES	Yes	Edit privileges Export
wpdatabase	localhost	database-specific	ALL PRIVILEGES	Yes	Edit privileges Export

New Add user account

The privileges to the wpdatabase table are mentioned on the tab.

### Click on the Operations tab.

Move table to (database.table)

wpdatabase . wp\_users

Add AUTO\_INCREMENT value  
Adjust privileges

Table options

Rename table to wp\_users  
Adjust privileges

Table comments

Storage Engine InnoDB

Collation utf8mb4\_unicode\_ci  
Change all column collations

AUTO\_INCREMENT 2

ROW\_FORMAT COMPACT

Operations such as moving the table, creating a copy, renaming it, adding comments can be performed from the operations tab.

## Click on Triggers

← Server: 127.0.0.1 » Database: wpdatabase » Table: wp\_users

Browse Structure SQL Search Insert Export Import Privileges Operations Triggers

Triggers ?

There are no triggers to display.

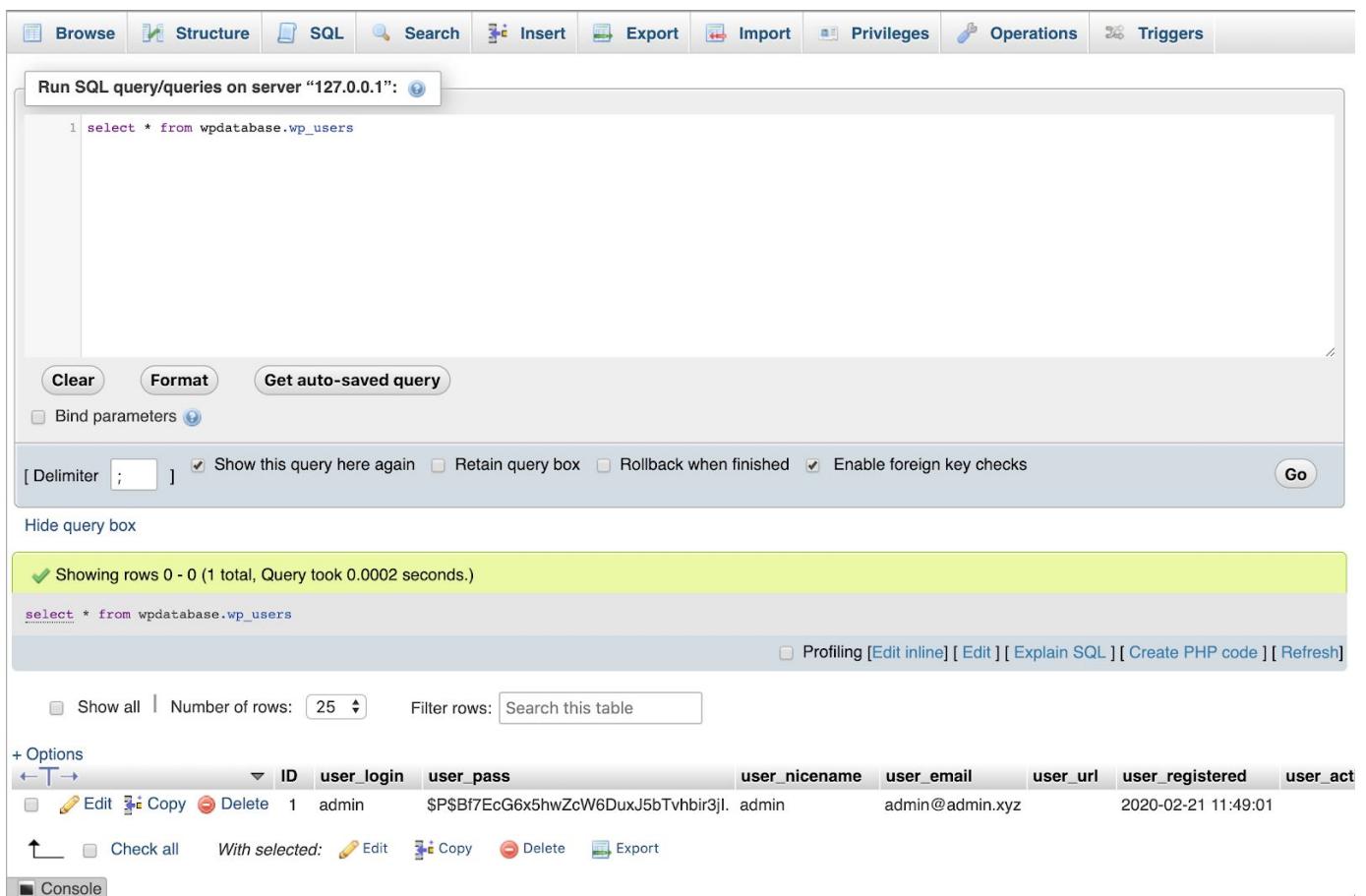
New Add trigger ?

The triggers are mentioned here. Currently there are no triggers.

## Click on the Dashboard (Home Icon)



## Click on the SQL Tab.



The screenshot shows the OmniDB SQL console interface. At the top, there is a navigation bar with tabs: Browse, Structure, SQL, Search, Insert, Export, Import, Privileges, Operations, and Triggers. Below the navigation bar, a message box says "Run SQL query/queries on server "127.0.0.1":". A code editor window contains the following SQL query:

```
1 select * from wpdatabase.wp_users
```

Below the code editor are several buttons: Clear, Format, Get auto-saved query, Bind parameters, Show this query here again (checked), Retain query box, Rollback when finished, Enable foreign key checks, and a Go button. A "Hide query box" link is also present.

The main content area displays the results of the query execution:

Showing rows 0 - 0 (1 total, Query took 0.0002 seconds.)

```
select * from wpdatabase.wp_users
```

Below the results, there is a table with the following data:

	ID	user_login	user_pass	user_nicename	user_email	user_url	user_registered	user_act
<input type="checkbox"/>	1	admin	\$P\$Bf7EcG6x5hwZcW6DuxJ5bTvhbir3jl.	admin	admin@admin.xyz		2020-02-21 11:49:01	

At the bottom of the table, there are buttons for Edit, Copy, Delete, and Export. There is also a "Console" button.

The SQL statements can be executed from here. All the statements executed above on the OmniDB SQL console can be executed here.

**Click on the User accounts Tab.**

The screenshot shows the MySQL Workbench interface with the title bar "Server: 127.0.0.1". The top menu bar includes "Databases", "SQL", "Status", "User accounts" (which is selected), "Export", "Import", "Settings", "Binary log", "Replication", and "More". The main content area is titled "User accounts overview". A warning message in a yellow box states: "⚠ A user account allowing any user from localhost to connect is present. This will prevent other users from connecting if the host part of their account allows a connection from any (%) host." Below the message is a table with the following data:

User name	Host name	Password	Global privileges	Grant	Action
Any	%	No	USAGE	No	Edit privileges  Export
Any	27ae0e316043	No	USAGE	No	Edit privileges  Export
Any	localhost	No	USAGE	No	Edit privileges  Export
root	127.0.0.1	No	ALL PRIVILEGES	Yes	Edit privileges  Export
root	27ae0e316043	No	ALL PRIVILEGES	Yes	Edit privileges  Export
root	::1	No	ALL PRIVILEGES	Yes	Edit privileges  Export
root	localhost	No	ALL PRIVILEGES	Yes	Edit privileges  Export
wpdatabase	localhost	Yes	USAGE	No	Edit privileges  Export

Below the table, there are buttons for "Check all" and "With selected: Export". The "New" button is highlighted. The "Add user account" button is shown below it. A "Remove selected user accounts" button is also present. A note says "(Revoke all active privileges from the users and delete them afterwards.)" and a checkbox "Drop the databases that have the same names as the users." is available. At the bottom right is a "Go" button.

Access to various User accounts are mentioned on the web page.

**Click on the Binary Log**

Server: 127.0.0.1

Databases SQL Status User accounts Export Import Settings Binary log Replication More

## Binary log

Select binary log to view

mysql-bin.000001 (86.91 KiB) 2 Files, 91,278 B

**Go**

Your SQL query has been executed successfully.

SHOW BINLOG EVENTS LIMIT 0, 25

Profiling [Edit inline] [ Edit ] [ Create PHP code ] [ Refresh ]

Log name	Position	Event type	Server ID	Original position	Information
mysql-bin.000001	4	Format_desc	1	107	Server ver: 5.5.56-log, Binlog ver: 4
mysql-bin.000001	107	Query	1	202	create database wpdatabase
mysql-bin.000001	202	Query	1	322	CREATE USER 'wpdatabase'@'localhost' IDENTIFIED BY 'password'
mysql-bin.000001	322	Query	1	493	GRANT ALL ON wpdatabase.* TO 'wpdatabase'@'localhost' IDENTIFIED BY 'password' WITH GRANT OPTION
mysql-bin.000001	493	Query	1	568	FLUSH PRIVILEGES
mysql-bin.000001	568	Query	1	700	use `wpdatabase`; DROP TABLE IF EXISTS `wp_commentmeta` /* generated by server */
mysql-bin.000001	700	Query	1	1197	use `wpdatabase`; CREATE TABLE `wp_commentmeta` ( `meta_id` bigint(20) unsigned NOT NULL AUTO_INCREMENT, `comment_id` bigint(20) unsigned NOT NULL DEFAULT '0', `meta_key` varchar(255) COLLATE utf8mb4_unicode_ci DEFAULT NULL, `meta_value` longtext COLLATE utf8mb4_unicode_ci, PRIMARY KEY (`meta_id`), KEY `comment_id`(`comment_id`), KEY `meta_key` (`meta_key` (191)) ) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_unicode_ci
mysql-bin.000001	1569	Query	1	3087	use `wpdatabase`; CREATE TABLE `wp_comments` ( `comment_ID` bigint(20) unsigned NOT NULL AUTO_INCREMENT, `comment_post_ID` bigint(20) unsigned NOT NULL DEFAULT '0', `comment_author` tinytext COLLATE utf8mb4_unicode_ci NOT NULL, `comment_author_email` varchar(100) COLLATE utf8mb4_unicode_ci NOT NULL DEFAULT '', `comment_author_url` varchar(200) COLLATE utf8mb4_unicode_ci NOT NULL DEFAULT '', `comment_date` datetime NOT NULL DEFAULT '0000-00-00 00:00:00', `comment_date_gmt` datetime NOT NULL DEFAULT '0000-00-00 00:00:00', `comment_content` text COLLATE utf8mb4_unicode_ci NOT NULL, `comment_karma` int(11) NOT NULL DEFAULT '0', `comment_approved` varchar(20) COLLATE utf8mb4_unicode_ci NOT NULL DEFAULT '1', `comment_agent` varchar(255) COLLATE utf8mb4_unicode_ci NOT NULL DEFAULT '', `comment_type` varchar(20) COLLATE utf8mb4_unicode_ci NOT NULL DEFAULT '' ) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_unicode_ci
mysql-bin.000001	3087	Query	1	3206	use `wpdatabase`; /*!40000 ALTER TABLE `wp_comments` DISABLE KEYS */
mysql-bin.000001	3206	Query	1	3280	BEGIN
mysql-bin.000001	3280	Query	1	3742	use `wpdatabase`; INSERT INTO `wp_comments` VALUES (1,1,'A WordPress Commenter','wapuu@wordpress.example','https://wordpress.org/','','2020-02-21 11:49:01','2020-02-21 11:49:01','Hi, this is a comment.\nTo get started with moderating, editing, and deleting comments, please visit the Comments screen in the dashboard.\nCommenter avatars come from <a href=\"https://gravatar.com\">Gravatar</a>.',0,'1','','',0,0)

Console

The Binary log tab displays all the SQL queries executed.

## **References:**

1. OmniDB (<https://omnidb.org/en/>)
2. PHPMyAdmin (<https://www.phpmyadmin.net/>)