

## Create User Sample Output:

The screenshot shows the Postman application interface. The top bar includes navigation links (Home, Workspaces, Explore), a search bar, and user options (Sign In, Create Account). The main workspace is titled "Working locally in Scratch Pad. Switch to a Workspace". On the left, the "Collections" sidebar shows a message: "You don't have any collections. Collections let you group related requests, making them easier to access and run." The central editor shows a GET request to `http://127.0.0.1:5000/users/create`. The "Body" tab is selected, displaying a JSON payload: 

```
{  "name": "Umer",  "email": "umer@salman123.com",  "password": "umersalman"}
```

. The "Send" button is visible. Below the request, the "Test Results" section shows a status of "200 OK", a time of "359 ms", and a size of "186 B". The response body is displayed in the "Pretty" view: 

```
{  "Success": "User inserted successfully"}
```

## Output from DB:

The screenshot shows the MySQL Workbench interface. The top bar includes the "AVEN" logo and standard menu options (File, Edit, View, Query, Database, Server, Tools, Scripting, Help). The left sidebar contains navigation panels for "MANAGEMENT" (Server Status, Client Connections, Users and Privileges, Status and System Variables, Data Export, Data Import/Restore), "INSTANCE" (Startup / Shutdown, Server Logs, Options File), and "PERFORMANCE" (Dashboard, Performance Reports, Performance Schema Setup). The main editor displays a SQL query: 

```
CREATE TABLE users (  name VARCHAR(80) NOT NULL,  email VARCHAR(120) UNIQUE NOT NULL,  password_hash VARCHAR(128) NOT NULL,  last_login DATETIME DEFAULT CURRENT_TIMESTAMP)  select * from users
```

. The "Result Grid" shows the output of the query, displaying columns: id, name, email, password\_hash, and last\_login. The results are as follows:

id	name	email	password_hash	last_login
9	Umer	umer@salman.com	\$argon2d\$v=19\$m=102400,t=2,p=8\$4oTied...	2024-04-27 12:04:08
10	Umer	umer@salman123.com	\$argon2d\$v=19\$m=102400,t=2,p=8\$8oe8Y6...	

The bottom section of the interface shows the "Output" tab, which displays the "Action Output" for the query. It lists the execution steps, including the query execution and the number of rows returned (1 row(s) returned).

## Update User Sample Output:

The screenshot shows the Postman application interface. The top bar indicates 'Working locally in Scratch Pad. Switch to a Workspace'. The left sidebar shows 'Collections' with a message: 'You don't have any collections. Collections let you group related requests, making them easier to access and run. Create Collection'. The main area shows a GET request to 'http://127.0.0.1:5000/users/update/10'. The 'Body' tab is selected, showing a JSON payload: 

```
{  \"name\": \"Umer\",  \"email\": \"umer@salman123456.com\",  \"password\": \"umersalman\"}
```

. The 'Test Results' tab shows a status of '200 OK', 'Time: 316 ms', and 'Size: 190 B'. The 'Response' tab shows the body: 

```
{  \"Success\": \"User data updated successfully\"}
```

.

## Output from DB:

The screenshot shows the MySQL Workbench interface. The 'Query Editor' displays a query: 

```
select * from users
```

. The 'Result Grid' shows two rows of data: 

id	name	email	password_hash	last_login
9	Umer	umer@salman.com	\$argon2d\$y=19\$m=102400,t=2,p=8\$4o7eId...	2024-04-27 12:04:08
10	Umer	umer@salman123456.com	\$argon2d\$y=19\$m=102400,t=2,p=8\$94%JBO...	

. The bottom panel shows the 'Action Output' log with the following entries: 

#	Time	Action	Message	Duration / Fetch
13	12:01:50	select * from users LIMIT 0, 1000	1 row(s) returned	0.015 sec / 0.000 sec
14	12:04:02	select * from users LIMIT 0, 1000	1 row(s) returned	0.016 sec / 0.000 sec
15	12:04:12	select * from users LIMIT 0, 1000	1 row(s) returned	0.016 sec / 0.000 sec
16	12:34:11	select * from users LIMIT 0, 1000	1 row(s) returned	0.015 sec / 0.000 sec
17	12:35:22	select * from users LIMIT 0, 1000	2 row(s) returned	0.015 sec / 0.000 sec
18	12:36:57	select * from users LIMIT 0, 1000	2 row(s) returned	0.015 sec / 0.000 sec

.

## Delete User Sample Output:

The screenshot shows the Postman interface with a DELETE request to `http://127.0.0.1:5000/users/delete/10`. The request is successful, returning a 200 OK status. The response body is a JSON object: `{"Success": "User data deleted successfully"}`.

Request Details:

- Method: DELETE
- URL: `http://127.0.0.1:5000/users/delete/10`
- Body: `1`

Response Details:

- Status: 200 OK
- Time: 279 ms
- Size: 190 B
- Body: 

```
1 {
2   "Success": "User data deleted successfully"
3 }
```

## Output from DB:

The screenshot shows the MySQL Workbench interface with a query result for the 'users' table. The query is `select * from users`. The result grid shows one row with the following data:

#	id	name	email	password_hash	last_login
9	Umer	umer@salman.com	\$argon2d\$vr=19\$w=102400,t=2,p=8\$4oTieId...	2024-04-27 12:04:08	

The bottom panel shows the Action Output, which lists the actions performed during the query execution:

#	Time	Action	Message	Duration / Fetch
14	12:04:02	select * from users LIMIT 0, 1000	1 row(s) returned	0.016 sec / 0.000 sec
15	12:04:12	select * from users LIMIT 0, 1000	1 row(s) returned	0.016 sec / 0.000 sec
16	12:34:11	select * from users LIMIT 0, 1000	1 row(s) returned	0.015 sec / 0.000 sec
17	12:35:22	select * from users LIMIT 0, 1000	1 row(s) returned	0.015 sec / 0.000 sec
18	12:36:57	select * from users LIMIT 0, 1000	2 row(s) returned	0.015 sec / 0.000 sec
19	12:38:28	select * from users LIMIT 0, 1000	1 row(s) returned	0.031 sec / 0.000 sec

## List User Sample Output:

The screenshot shows the Postman interface. The request is a GET to `http://127.0.0.1:5000/users/list`. The response is a JSON object with the following structure:

```
{  "users": [    {      "email": "umez@salman.com",      "id": 9,      "last_login": "2024-04-27 12:04:08",      "name": "Umei"    }  ]}
```

## Output from DB:

The screenshot shows the MySQL Workbench interface. The query being executed is:

```
select * from users
```

The results are displayed in a table with the following columns: `id`, `name`, `email`, `password_hash`, and `last_login`.

id	name	email	password_hash	last_login
9	Umei	umez@salman.com	\$argon3d\$v=19\$m=102400,t=2,p=8\$0toTedd...	2024-04-27 12:04:08

The bottom panel shows the Action Output, which lists the execution details of the query, including the time taken and the number of rows returned.