

MySQL Database

school_db* quizdb

Limit to 1000 rows

```
1 • CREATE DATABASE quizdb;
2 • USE quizdb;
3 • SHOW TABLES;
4 • SHOW DATABASES;
5 • SELECT * FROM questions;
6
7
8
```

Result Grid

	id	category	correct_answer	difficulty_level	option1	option2	option3	option4	question_title
▶	2	SPRING	JavaScript	MEDIUM	Java	C	Python	JavaScript	Which language runs in a web browser?
	3	GITHUB	O(log n)	HARD	O(n)	O(log n)	O(n log n)	O(1)	What is the time complexity of binary search?
*		NULL	NULL	NULL	NULL	NULL	NULL	NULL	

questions 7 questions 10

Output

Action Output

#	Time	Action	Message
✓ 15	14:05:51	SHOW TABLES	3 row(s) returned
✓ 16	14:05:55	SHOW DATABASES	10 row(s) returned
✓ 17	14:06:00	SELECT * FROM questions LIMIT 0, 1000	3 row(s) returned
✓ 18	14:10:01	SELECT * FROM questions LIMIT 0, 1000	3 row(s) returned
✓ 19	14:10:11	SHOW DATABASES	10 row(s) returned
✓ 20	14:10:14	SELECT * FROM questions LIMIT 0, 1000	3 row(s) returned

POSTMAN:

ADD:

The screenshot displays the Postman application interface. The top navigation bar includes a hamburger menu, navigation arrows, and tabs for 'Home', 'Workspaces', and 'API Network'. A search bar labeled 'Search Postman' and a keyboard shortcut 'Ctrl K' are also present.

The left sidebar contains icons for 'Collections', 'Environments', 'Flows', and 'History'.

The main workspace shows a POST request to the URL `http://localhost:8080/api/questions`. The request body is configured as JSON and contains the following data:

```
1 {
2   "questionText": "What is Java?",
3   "option1": "Programming language",
4   "option2": "coffee",
5   "option3": "Database",
6   "option4": "Food",
7 }
```

Below the request editor, the 'Body' tab is selected, showing the response in JSON format:

```
1 {
2   "id": 5,
3   "questionTitle": null,
4   "option1": "Programming language",
5   "option2": "coffee",
6   "option3": "Database",
7   "option4": "Food",
8   "correctAnswer": "Programming language",
9   "difficultyLevel": "EASY",
10  "category": "JAVA"
11 }
```

DELETE:

DEL

http://localhost:8080/api/

●

+

HTTP

http://localhost:8080/api/questions/3

DELETE

▼

http://localhost:8080/api/questions/3

Params

Authorization

Headers (9)

Body ●

Scripts

Tests

Settings

☐ none

☐ form-data

☐ x-www-form-urlencoded

☒ raw

☐ binary

☐ GraphQL

JSON ▼

1

}

Body

Cookies

Headers (3)

Test Results

🕒

Raw ▼

▶ Preview

🔄 Visualize ▼

1

PUT:

PUT http://localhost:8080/api/questions/1

HTTP http://localhost:8080/api/questions/1

PUT http://localhost:8080/api/questions/1

Params Authorization Headers (9) Body Scripts Tests Settings

☐ none ☐ form-data ☐ x-www-form-urlencoded ☒ raw ☐ binary ☐ GraphQL JSON

```
1 {
2   "questionTitle": "Which command is used to create a new branch in Git?",
3   "option1": "git branch new-branch",
4   "option2": "git checkout new-branch",
5   "option3": "git commit new-branch",
6   "option4": "git init new-branch",
7   "correctAnswer": "git branch new-branch",
8   "difficultyLevel": "MEDIUM",
9   "category": "GITHUB"
10 }
```

Body Cookies Headers (5) Test Results

{ JSON Preview Visualize

```
1 {
2   "id": 1,
3   "questionTitle": "Which command is used to create a new branch in Git?",
4   "option1": "git branch new-branch",
5   "option2": "git checkout new-branch",
6   "option3": "git commit new-branch",
7   "option4": "git init new-branch",
8   "correctAnswer": "git branch new-branch",
9   "difficultyLevel": "MEDIUM",
10  "category": "GITHUB"
11 }
```

GETALLQUESTIONS:

The screenshot shows the Postman interface with a GET request to `http://localhost:8080/api/questions` executed successfully. The response status is **200 OK** with a response time of 28 ms and a body size of 1.09 KB. The response body is displayed in JSON format, showing an array of two question objects.

```
17  {
18    "id": 2,
19    "questionTitle": "Which language runs in a web browser?",
20    "option1": "Java",
21    "option2": "C",
22    "option3": "Python",
23    "option4": "JavaScript",
24    "correctAnswer": "JavaScript",
25    "difficultyLevel": "MEDIUM",
26    "category": "SPRING"
27  },
28  {
29    "id": 3,
30    "questionTitle": "What is the time complexity of binary search?",
31    "option1": "O(n)",
32    "option2": "O(log n)",
33    "option3": "O(n log n)",
34    "option4": "O(1)",
35    "correctAnswer": "O(log n)",
36    "difficultyLevel": "HARD",
37    "category": "GITHUB"
38  }
39 ],
40 "number": 0,
41 "sort": {
42   "empty": true,
43   "sorted": false,
```

PAGINATION:

GET http://localhost:8080/api/

HTTP http://localhost:8080/api/questions?page=0&size=20

GET http://localhost:8080/api/questions?page=0&size=20

Params Authorization Headers (7) Body Scripts Tests Settings

Body Cookies Headers (5) Test Results

{ } JSON Preview Visualize

```
17 {
18   "id": 2,
19   "questionTitle": "Which language runs in a web browser?",
20   "option1": "Java",
21   "option2": "C",
22   "option3": "Python",
23   "option4": "JavaScript",
24   "correctAnswer": "JavaScript",
25   "difficultyLevel": "MEDIUM",
26   "category": "SPRING"
27 },
28 {
29   "id": 3,
30   "questionTitle": "What is the time complexity of binary search?",
31   "option1": "O(n)",
32   "option2": "O(log n)",
33   "option3": "O(n log n)",
34   "option4": "O(1)",
35   "correctAnswer": "O(log n)",
36   "difficultyLevel": "HARD",
37   "category": "GITHUB"
38 }
39 ],
40 "number": 0,
41 "sort": {
42   "empty": true,
43   "sorted": false,
```

Find and replace Console

GET QUESTION BY ID:

The screenshot displays a REST client interface with two requests. The first request is a GET to `http://localhost:8080/api/questions/3`, which returns a JSON object for a specific question. The second request is a GET to `http://localhost:8080/api/questions?page=0&size=20`, which returns a paginated list of questions.

Request 1:

Method: GET
URL: `http://localhost:8080/api/questions/3`

Body (JSON):

```
1 {
2   "id": 3,
3   "questionTitle": "What is the time complexity of binary search?",
4   "option1": "O(n)",
5   "option2": "O(log n)".
```

Request 2:


Method: GET
URL: `http://localhost:8080/api/questions?page=0&size=20`


Body (JSON):

```
36   "difficultyLevel": "HARD",
37   "category": "GITHUB"
38 },
39 ],
40 "number": 0,
41 "sort": {
42   "empty": true,
43   "sorted": false,
44   "unsorted": true
45 },
46 "numberOfElements": 3,
47 "first": true,
48 "last": true,
49 "pageable": {
50   "pageNumber": 0,
51   "pageSize": 20,
52   "sort": {
53     "empty": true,
54     "sorted": false,
55     "unsorted": true
56   },
57   "offset": 0,
58   "paged": true,
59   "unpaged": false
60 },
61 "empty": false
62 }
```


GET QUESTION BY ID:






GET http://localhost:8080/api/... +

 http://localhost:8080/api/questions/2

GET  http://localhost:8080/api/questions/2

Params Authorization Headers (7) Body Scripts Tests Settings

Body Cookies Headers (5) Test Results 

 JSON   Preview  Visualize 

```
1  {
2    "id": 2,
3    "questionTitle": "Which language runs in a web browser?",
4    "option1": "Java",
5    "option2": "C",
6    "option3": "Python",
7    "option4": "JavaScript",
8    "correctAnswer": "JavaScript",
9    "difficultyLevel": "MEDIUM",
10   "category": "SPRING"
11 }
```


PRETTY PRINT:

Pretty-print ☒

```
{
  "id": 2,
  "questionTitle": "Which language runs in a web browser?",
  "option1": "Java",
  "option2": "C",
  "option3": "Python",
  "option4": "JavaScript",
  "correctAnswer": "JavaScript",
  "difficultyLevel": "MEDIUM",
  "category": "SPRING"
},
{
  "id": 3,
  "questionTitle": "What is the time complexity of binary search?",
  "option1": "O(n)",
  "option2": "O(log n)",
  "option3": "O(n log n)",
  "option4": "O(1)",
  "correctAnswer": "O(log n)",
  "difficultyLevel": "HARD",
  "category": "GITHUB"
}
],
"number": 0,
"sort": {
  "empty": true,
  "sorted": false,
  "unsorted": true
},
"numberOfElements": 3,
"first": true,
"last": true,
"pageable": {
  "pageNumber": 0,
  "pageSize": 20,
  "sort": {
    "empty": true,
    "sorted": false,
    "unsorted": true
  },
  "offset": 0,
  "paged": true,
  "unpaged": false
},
"empty": false
}
```

