

Dear (applicant)

We are excited to have you make it to the next round of our interview process for the web developer position. This next phase will be a testing phase to see the quality and competence of your work in a time sensitive situation. You will have 12 hours to complete this process of producing and completing the following tasks.

1. Utilizing the table below:
 1. Create a MySQL or PostgreSQL database schema
 2. Create a query to pull employee performance, arranged highest to lowest
 3. Create a dashboard to show the above report built with a backend API server written in Go, and a frontend created with React.js
 4. Allow the user to sort by name, performance, and date

ID #	Name	Performance	Date
1	John	58	12/11/2022
2	Daniel	87	11/15/2022
3	Sally	34	6/15/2022
4	Tiffany	99	2/28/2022

2. Use HTML and CSS (or SCSS) to design a new Eightvape.com homepage with a slider that incorporates the core components of an ecommerce website with a high CTR, low bounce rate, and user friendliness approach.
3. (optional) Create a Shopify script that enables orders over \$100 being shipped to Brazil to automatically receive free shipping.
4. Design a complete set of wireframes (nonfunctional mockups) for a mobile version of the Eightvape storefront to show user friendliness. Include the following screens:
 - ☐ Landing screen
 - ☐ Product category screen with filters
 - ☐ Product screen with variant selector
 - ☐ Cart
 - ☐ Checkout screen with fields for shipping address, shipping options, and payment

Start time: 11:00 AM

End time: 7:00 PM

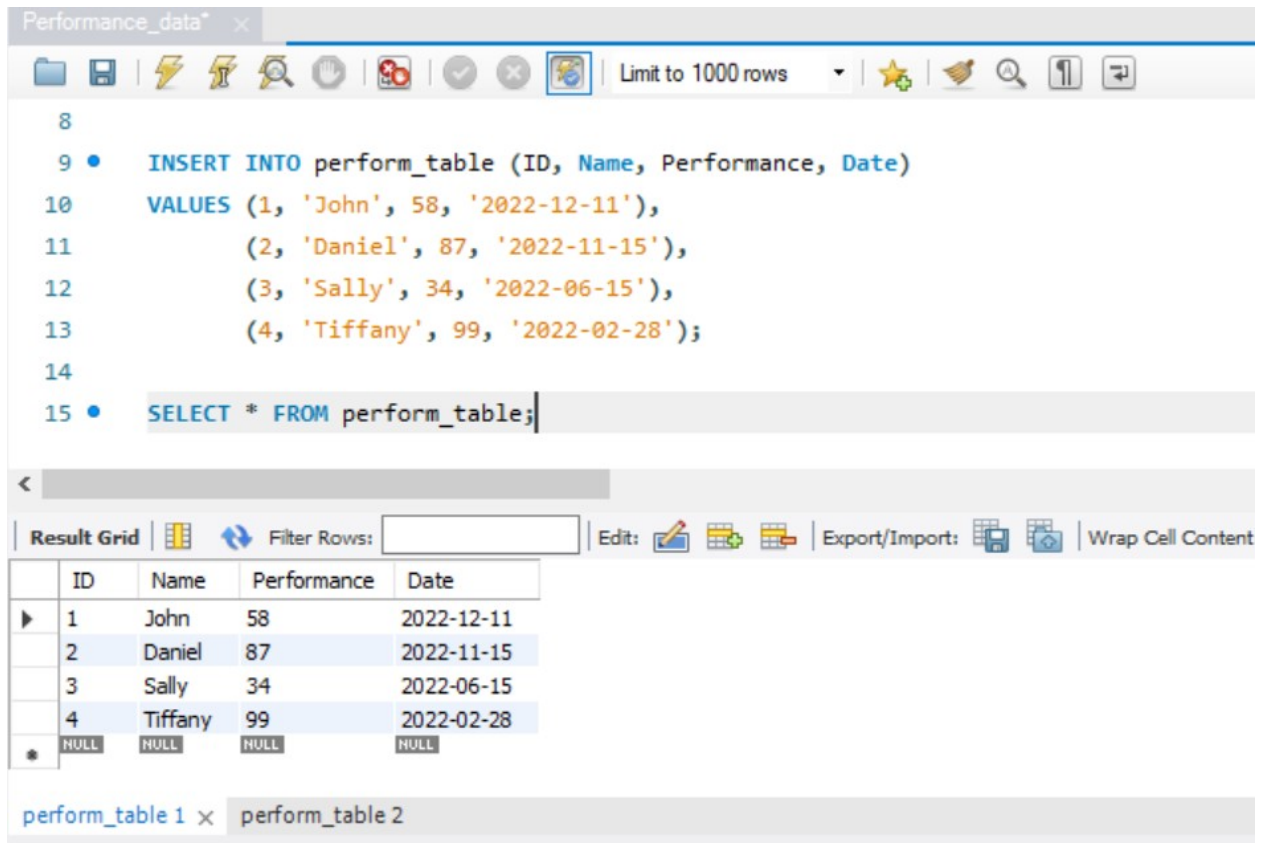
Please acknowledge receipt of this e-mail.

Thank you and good luck!

I appreciate the opportunity to interview at Magma Holding Inc. This is my documentation for the whole assessment.

Task 1:

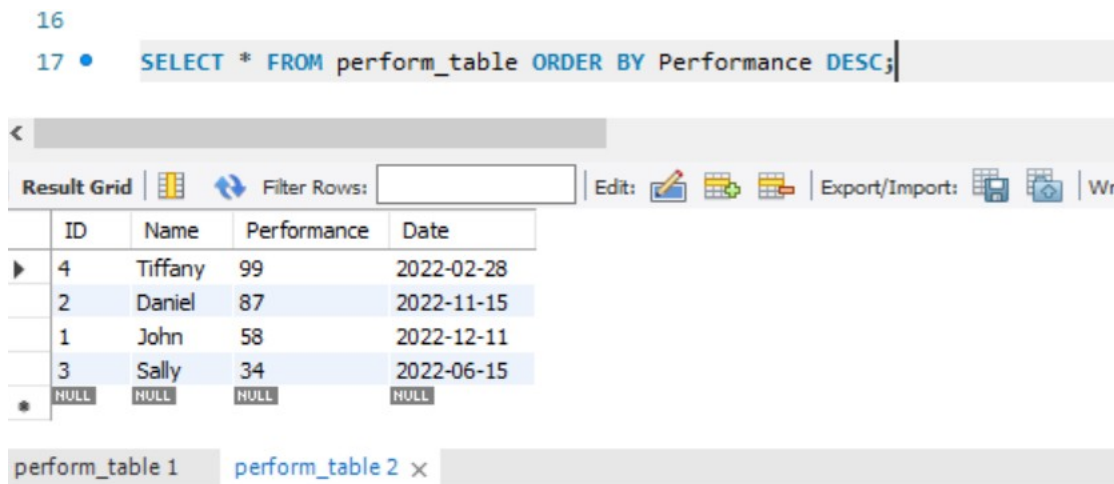
1. Creating Database Schema



The screenshot shows a database IDE window titled "Performance_data". The SQL editor contains two queries. The first query is an INSERT statement that adds four records to a table named "perform_table". The second query is a SELECT statement that retrieves all records from "perform_table". Below the editor, the "Result Grid" shows the output of the first query, which is an empty table. The output of the second query is a table with four rows, each representing an employee's performance record. The table has columns for ID, Name, Performance, and Date. The data is as follows:

ID	Name	Performance	Date
1	John	58	2022-12-11
2	Daniel	87	2022-11-15
3	Sally	34	2022-06-15
4	Tiffany	99	2022-02-28

2. Query to pull employee performance, arranged highest to lowest



The screenshot shows a database IDE window titled "Performance_data". The SQL editor contains a query that retrieves all records from "perform_table" ordered by the "Performance" column in descending order. Below the editor, the "Result Grid" shows the output of the query, which is a table with four rows, each representing an employee's performance record. The table has columns for ID, Name, Performance, and Date. The data is as follows:

ID	Name	Performance	Date
4	Tiffany	99	2022-02-28
2	Daniel	87	2022-11-15
1	John	58	2022-12-11
3	Sally	34	2022-06-15

3. Create a dashboard to show the above report built with a backend API server written in Go, and a frontend created with React.js

Back-end:

Connection to MySQL Database Successful

```
PS C:\Users\arvin\OneDrive\Desktop\gooo> go get -u github.com/go-sql-driver/mysql
>> go run main.go
Connected to the MySQL database!
PS C:\Users\arvin\OneDrive\Desktop\gooo> █
```

```
[GIN-debug] GET    /employees          --> main.getEmployees (3 handlers)
[GIN-debug] [WARNING] You trusted all proxies, this is NOT safe. We recommend you to set a value.
Please check https://pkg.go.dev/github.com/gin-gonic/gin#readme-don-t-trust-all-proxies for details.
[GIN-debug] Listening and serving HTTP on :6060
[GIN] 2023/07/22 - 17:38:49 | 404 |      0s |      127.0.0.1 | GET      "/employees1"
[GIN] 2023/07/22 - 17:39:43 | 200 | 535.2µs |      127.0.0.1 | GET      "/employees"
█
```

Status 200 to fetch all the employees using CURL command.

```
aniket@Aniket-Arsenal MINGW64 ~/OneDrive/Desktop/gooo
$ curl -X GET http://localhost:6060/employees
[{"id":4,"name":"Tiffany","performance":99,"date":"2022-02-28"}, {"id":2,"name":"Daniel","performance":87,"date":"2022-11-15"}, {"id":1,"name":"John","performance":58,"date":"2022-12-11"}, {"id":3,"name":"Sally","performance":34,"date":"2022-06-15"}]
aniket@Aniket-Arsenal MINGW64 ~/OneDrive/Desktop/gooo
$ █
```

Front-end:

Table Structure rendering

Employee Performance Dashboard

ID	Name	Performance	Date
----	------	-------------	------

4. On click Sorting Functionality

Employee Performance Dashboard

ID	Name	Performance ↓	Date
4	Tiffany	99	2022-02-28
2	Daniel	87	2022-11-15
1	John	58	2022-12-11
3	Sally	34	2022-06-15