

Backend FastAPI Assessment

1. SQL Assessment Question

Scenario: You are given a database for an e-commerce platform. The database includes the following tables:

1. Customers

- customer_id (INT, Primary Key)
- customer_name (VARCHAR)
- email (VARCHAR)
- signup_date (DATE)

2. Orders

- order_id (INT, Primary Key)
- customer_id (INT, Foreign Key referencing Customers)
- order_date (DATE)
- total_amount (DECIMAL)

3. Order_Items

- order_item_id (INT, Primary Key)
- order_id (INT, Foreign Key referencing Orders)
- product_id (INT)
- quantity (INT)
- price_per_unit (DECIMAL)

4. Products

- product_id (INT, Primary Key)
- product_name (VARCHAR)
- category (VARCHAR)

Question: Write a SQL query to find the top 5 customers who have spent the most money on the platform. The output should include the following columns: customer_id, customer_name, email, total_spent, and most_purchased_category (the category of products they spent the most money on).

PS: Feel free to create a dummy data for the above tables and add them in the repo.

Additional Requirements:

- The total amount spent should be calculated as the sum of quantity * price_per_unit for each order item.
- Consider only orders that have been placed in the last year from the current date.
- The most_purchased_category should be determined based on the total spending in each category.
- The query should be optimized for performance.

2. Python Assessment Question (Use FastAPI)

Scenario: You are given a dataset containing information about various products sold on an e-commerce platform. The dataset is in CSV format with the following columns:

- product_id
- product_name
- category
- price
- quantity_sold
- rating
- review_count (number of reviews)

Question: Write a Python script to perform the following tasks:

1. Database Connection:

- Connect to a relational database (e.g., PostgreSQL, MySQL, SQLite).
- Create a table to store the product data if it does not already exist.

2. Data Upload:

- Load the data from the CSV file into a pandas DataFrame.
- Upload the data from the DataFrame to the database table.

3. Login and Sign-Up System:

- Implement a simple login and sign-up system using JWT tokens.
- Create two endpoints: /signup and /login.
- /signup should accept username and password, hash the password, and store the user details in a Users table.
- /login should accept username and password, verify the credentials, and return a JWT token if successful.

4. Data Cleaning:

- Handle missing values: For price and quantity_sold, replace missing values with the median value of the respective column. For rating, replace missing values with the average rating of that category.
- Ensure that price, quantity_sold, and rating are numeric values.

5. Summary Report:

- Create a React view to display the results of the data analysis.
- Generate a summary report in the form of a CSV file containing the following columns:
 - category
 - total_revenue
 - top_product (the product name of the highest selling product in that category)
 - top_product_quantity_sold

Additional Requirements:

- Use appropriate exception handling to manage potential errors during file operations, database connections, and data processing.
- Include comments and docstrings to explain the functionality of your code.
- Ensure the script is modular, with functions handling different tasks.
- Use SQLAlchemy or another ORM for database operations.

Instructions for Submission

- The assignment should be completed within one day. Upload the code & supporting files to a private GitHub repository and share the collaboration link.
- For any query reach out to: deepanshu@couture.ai