LA Clippers Basketball Full-Stack Developer Question Set

Background Questions

1. How did you end up at your undergraduate school? How did you choose your major? What appealed to you about it?
2. Please list the computer programs, full stack libraries, and other computational tools you are familiar with including your proficiency and a brief description on what you used the tool for.
3. List 3 websites or apps that annoy you (functionally/aesthetically). In your opinion what's wrong with them? How would you fix them?
4. What is the hardest thing you ever programmed? Why was it difficult? How did you overcome the difficulty?
5. What online (sports/programming/statistics) communities do you read and/or participate in on a regular basis?
6. Have you ever taught yourself something? What was it? Why did you learn it? How did you learn it?
7. Over the last 4 years, what single skill did you develop that you think would be the most valuable for this role and what single skill did you develop that you think would be the least valuable for this role? Why? How did you go about developing those two skills?
8. What interests you about working for a basketball team? How do you think a Basketball Front Office could maximize your development?

Technical Questions

In the attached zip folder, you will find JSON files in the "dev_test_data" folder which will be used for this question. All your completed code should be runnable and testable. To ensure this, please send your finished code in a fully dockerized container image with a corresponding Dockerfile and/or docker-compose configuration file. Please make sure to include any documentation that may be needed to properly run your code (including packages/libraries used). If you deploy the resulting web apps to a managed deployment instance (on Vercel, Render, DigitalOcean, Cloudflare, etc.), please send the corresponding link. You can assume that any new data will have the same names as the given files. For all the questions, please describe your approach from start to finish and issues that came up along the way.

1. Database creation (Python, SQL, other scripting languages)
   a. Write code to transfer the files from a directory called dev_test_data to a SQL database called lac_fullstack_dev (code can be Python, SQL, etc.)
      i. The tables created should be named team, team_affiliate, game_schedule, player, lineup, and roster
      ii. Make sure your code creates tables if needed and that it can handle data reloads, merges, and/or updates
   b. What should be the Primary Keys across the tables?
2. Basic Queries (SQL)
   a. Write a SQL query that can calculate team win-loss records, sorted by win percentage (defined as wins divided by games played)
      i. Final table should include team name, games played, wins, losses, win percentage
   b. In the same table, show how the team ranks (highest to lowest) in terms of games played, home games, and away games during this month of the season? Make sure your code can extend to additional months as data is added to the data set. For each, show both the number of games and the rank
3. Schedule SQL Queries (SQL):
   a. The NBA has a concept of a Back-to-Back (B2B) which is if a team played 2 days in a row (regardless of start time). For the data given which team had the most Home-Home B2Bs? Which had the most Away-Away B2Bs? For example, LAC's game on 1/27 at BOS would be an Away-Away B2B since LAC played 1/26 at TOR.
   b. Which team(s) had the longest rest between 2 games and what were the days of the 2 games?

   c. Additionally, write a query that ranks the teams based on the number of 3-in-4s (3 games over 4 days that is regardless of start time).

4. Lineups Queries (SQL): In answering any of these items, feel free creating intermediate temp tables, inline tables, or CTEs as needed.
   a. Notice that in the lineup data each row corresponds to a given player, game, lineup_num, period. Write a SQL query that creates a "wide" table for the team (so a given row is now game_id, team_id, lineup_num, period, time_in, time_out, and the 5 players on the court)
      i. Notice that time_in and time_out are in seconds, starting at 12 minutes (720) and going down to 0 minutes
   b. The field lineup_num changes as a player on either team gets substituted. Write a SQL query with the resultant table that stores when a player is continuously on the court for a given period (call this a stint)
      i. Final table should be game date, team, opponent, player_name, period, stint_number, stint_start_time, stint_end_time
      ii. Format the stint times in mm:ss so the start of the period is 12:00 and the end of the period is 00:00
   c. From you answer to 4.b, for each player, calculate the average number of stints a player has and average stint length for a player for a given game.
   d. Extend the query from 4.c to show columns for all games, in wins, in losses as well as a column that shows the difference in wins and losses
      i. each set (all/wins/losses) should have # of games, average stint length, average number of stints

5. Front-End Lineup Visualization (any frontend tool(s)/libraries you'd like)
   a. You will now make an application that has at least two different types of data visualizations utilizing any frontend tools you'd like. The data will be based on any combination of the jsons, tables, and/or queries you have written/used in questions 1-4. Think about your app or frontend visualizations being used in the context of a basketball operations staff member. The visualization you show on your screen will depend on which staff member is using it. This is an open-ended question so take creative freedom in the data visualizations you build. The technical requirements are listed below:
      i. The two or more visualizations should be variants of the same data and/or seeing different lineup stints (see question 4b for definition of a "stint"). To illustrate a few possibilities: the data visualization for a team's schedule, lineup stints for a game, or a roster board all have different possible views based on a user
      ii. There should be two user login and password combinations, and each should show the corresponding visualization that is appropriate. Please describe in your documentation for each how the visualization differs:

User	User ID	User Password
User 1	user_one	password123
User 2	user_two	123password

      iii. You have the freedom to choose the visualization; it can be displayed as tables, a figure, graph, or something else.
      iv. You can make the visualization app using any full-stack libraries you'd like (Flask, Django, shiny app with R or Python, Dash, Observable Framework, React, etc.)

Final Question

Answer this question after doing the rest of the questionnaire. For what you have done and submitted so far in applying for this position (Resume, Questionnaire) did you use any AI programs? If so, on what parts and why?