***Los Angeles Clippers - Front-End Developer Interview Project***

*This file includes both answers to the Background Questions as well as a guide for the Docker File and the Technical questions*

Table of Contents

[Background Questions 2](#_Toc177517074)

[Question #1: 2](#_Toc177517075)

[Question #2 2](#_Toc177517076)

[Question #3: 3](#_Toc177517077)

[Question #4: 4](#_Toc177517078)

[Question #5: 4](#_Toc177517079)

[Question #6: 4](#_Toc177517080)

[Question #7 4](#_Toc177517081)

[Question #8: 5](#_Toc177517082)

[Technical Questions 6](#_Toc177517083)

[Question #1: 6](#_Toc177517084)

[Question #2: 7](#_Toc177517085)

[Question #3: 8](#_Toc177517086)

[Question #4: 10](#_Toc177517087)

[Question #5: 11](#_Toc177517088)

# Background Questions

## Question #1:

How did you end up at your undergraduate school? How did you choose your major? What appealed to you about it?

* *Math and Science are the two subjects that bring out my curiosity. I decided to major in Biomedical Engineering because it was “the Best of Both Worlds”. This lead me to apply to programs with strong Engineering programs in SoCal. I ended up at California State, Long Beach where I was able to stay locally where I grew up and not take a large financial burden.*

## Question #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.

*The following list is in order of proficiency*

* *Python: 5/5*
  + *Libraries: pandas, numpy, simple\_salesforce, google*
  + *Work Application: Currently at the Dodgers I use python on a daily basis. All my DE projects begin with python scripts. I am familiar with troubleshooting both my own scripts + the imported modules.*
* *Google Cloud Platform:* 
  + *BigQuery 5/5*
  + *Work Application: Currently at the Dodgers most of our databases are in BQ. I work through python scripts to build the data pipelines to ingest the data into BQ.*
* *Microsoft SQL Server 4/5:*
  + *Work Application: Similar to BQ, some of our data pipelines are used to send data to SQL Server.I have developed some tools to assist in this upload process, which were also used for this assignment.*
* *Virtual Machines (Linux/Windows):*
  + *Work Application: At the Dodgers, we use virtual machines to assist the regular extracts running throughout the day.*
* *Litmus Edge: 4/5*
  + *Work Application: At Niagara Bottling, I worked on Litmus Edge to build real-time data pipelines on our continuous data feeds. Additionally, within Litmus Edge I used Node.js to build real-time analytics.*
* *PowerBI: 4/5*
  + *Work Application: At Niagara Bottling PowerBI was one of the main visualization platforms.*
* *Grafana: 4/5*
  + *Work Application: At Niagara Bottling PowerBI was one of the main visualization platforms.*
* *Github: 3/5*
  + *Work Application: At the Dodgers, we use Github as our source control for our scripts.*
* *Django: 3/5*
  + *Application: I have not used Django to build a production application, but I have built a few one-off visualization projects to learn the framework. I know I have room to grow and further understand the techniques used to develop a fully functional application, but I will continue to sharpen my skills to reach a 5/5.*

## Question #3:

List 3 websites or apps that annoy you (functionally/aesthetically). In your opinion what’s wrong with them? How would you fix them?

* *Website Aesthetic: College Sports Homepages - Throughout the last couple of years I have periodically looked at the homepages of many college sports programs and every time I am meet with the same general website frontend. Its very surprising the websites for most high profile sports programs in the country are all either built by the same company or use the exact same template. I am a big fan of cross-collaboration, I think the schools each have their own unique flair and usually the social media teams are at the forefront of building new creative ideas to represent the school each year. This collaboration could shake up the monotonous look of these homepages. A few examples:*
  + *<https://longbeachstate.com>*
  + [*https://gocards.com*](https://gocards.com)
  + [*https://usctrojans.com*](https://usctrojans.com)
* *Website Aesthetic: Google - I love Google, but the format for google results hasn’t changed in a very long time. Once a google search is made we get one of two screens: a generic google results list of 10+ links + a container of information on the top right (ex.* [*https://www.google.com/search?q=University+of+San+Francisco+Athletics*](https://www.google.com/search?q=University+of+San+Francisco+Athletics)*)*

*, or a container of images/information on the left + a container of information at the top right (ex.* [*https://www.google.com/search?q=la+clippers*](https://www.google.com/search?q=la+clippers)*). In both these examples, once you scroll down to the middle of the page there is blank space on the right side of the screen. I think the theory is not many people scroll all the way down in their searched, but I think there is opportunity in the space to attract people to look at all the google results that have been loaded in. Potentially a table of statistics on the search made.*

* *App Functionality: In both Instagram and Tik Tok, the videos or images that are captured with HDR enabled allow for the image brightness to be increased without the user changing any of the settings within their device. This is a small nuisance but this could deter a user from continuing to use the product for prolonged periods of time. I would disable this functionality.*

## Question #4:

What is the hardest thing you ever programmed? Why was it difficult? How did you overcome the difficulty?

* *In grad school, I was tasked to build an app that would facilitate a company’s process to apply one of their drug products for FDA approval. In this process there are many forms + nuisances that small companies could easily miss a form or two. The goal for the app was to centralize all possible forms and make a process to give a user all the forms they needed for their application. The process of building the app (in hindsight) is not that difficult, but this was the first time I was building my own coding project without the additional help of others. Additionally, at that point in time I was only familiar with Matlab (which is not the most straightforward way to build an app). I enjoyed the process of figuring out how to code the frontend, how to build the nested functionalities for the forms, and ultimately being able to deliver a project that I didn’t know I could do on my own. This project was a milestone in my journey to become more than just a Biomedical Engineer. It taught me how to be dynamic, learn how to find answers where there isn’t any yet, and it set me on the path to become a “swiss-army knife”.*

## Question #5:

What online (sports/programming/statistics) communities do you read and/or participate in on a regular basis?

* *This is a topic that I would like to improve on. Over the last few years, I have dedicated my time to learn on many topics on my own. But now that I have my feet solidly in the sports industry I want to hopefully be able to share my knowledge with those on a similar path.*

## Question #6:

Have you ever taught yourself something? What was it? Why did you learn it? How did you learn it?

* *Everything I have learned in my coding journey I have taught myself. There were some fundamentals I was taught in undergrad, but the vast majority of my coding knowledge came from pure curiosity of learning. I enjoy learning topics I never thought I would know. I always knew I would be an engineer, but I never imagined learning the ins and outs of data pipelines or visualization platforms. I enjoy the process of learning and that never stops. The next topics on my journey will be Machine Learning and full application development.*

## Question #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?

* *Besides my coding experience over the last four years, I have also grown as a communicator. When it comes to working with data, the main goal is to showcase this to a user who does not know the behind the scenes of the pipelines. They only want to know what the data is saying (in their terms). When I was working at Niagara Bottling I learned to talk to many people from all walks of life and showcase to them how data can improve their processes. In tandem, they helped me learn the “behind the scenes” of the machines they worked with and what the data meant in their terms. Being able to communicate to stakeholders is the most important thing outside of the hard skills needed for this position.*
* *I don’t think skills can ever not be valuable. The lessons to learn those skills are foundational for future adventures. But I have developed my Rocket League skills over the last four years. I might not be able to play with co-workers or apply my skill to work scenarios, but it is a game I never thought I could consistently play. I have continuously improved over the last few years where I can say my highest rank has been Champion I.*

## Question #8:

What interests you about working for a basketball team? How do you think a Basketball Front Office could maximize your development?

* *Growing up I was apart of the kids that saw Kobe Bean Bryant day-in day-out developing his basketball skills. His relentless and constant drive to perfection, showed many of us all we need is a will to learn something and we are capable of anything. I am on this sports journey in part to him. I am also on this journey because I love the aspect of becoming a better engineer. At Niagara Bottling, I was able to lead a project from start to finish and enjoy learning topics I never knew I would learn. Here at the Dodgers, I am working on data projects within the Business Analytics department and enjoy being able to use my skills for the purpose of growing the game of baseball. The next step is to apply all the above skills and work within a basketball organization for the purpose of winning a championship. The Clippers are currently growing their identity and opening opportunities like this to find people who want to continue to learn and improve. This means new and innovative projects are coming in the horizon. I want to be a part of that driving force and also gain a greater understanding of what questions are unanswered within the Basketball Operations world and hopefully apply a new lens/find solutions.*

# Technical Questions

*For the following technical questions, Questions #1 – 4 have been fully answered across multiple python files + sql queries (as indicated below).*

*My initial approach for the SQL Queries was to create a view through Python, but I got errors trying this approach. Instead, I created the necessary ‘temp tables’ as real tables through python and derived the answers. All associated SQL Queries are within the extract\_clips\_data.py file. In the zip file you will also see .sql files that are the initial approach to each question.*

*For question #5, I have run out of time to fully customize the visualization and functionality. I will continue to build upon the Django app (even after I submit this file) in case any further questions arise.*

*Zip File:*

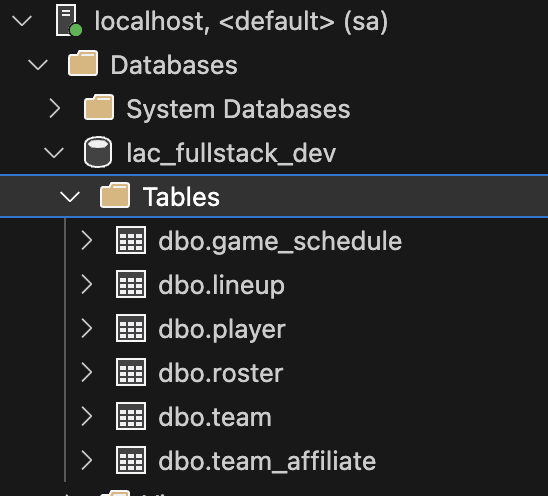
* *Folders:*
  + *Clips\_app (Django App)*
  + *Django\_app (Django Project)*
  + *Dev\_test\_app (Given Data)*
  + *SQL Sever (Placeholder in case I needed a Dockerfile) (Used the SQL Server Image)*
  + *Utilities (User made library to communicate with SQL Server)*
* *Files:*
  + *Docker-compose (Main File to Run)*
  + *Dockerfile (for extract\_clips\_data.py)*
  + *extract\_clips\_data.py (main script for data/sql queries)*
  + *manage.py (for Django Project)*
  + *.sql files (initial code for SQL queries) (not used)*
  + *SaraCruz\_Questionnaire.docx*
  + *SaraCruz\_Resume\_2024\_V2.pdf*

## Question #1:

Database creation (Python, SQL, other scripting languages)

1. 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.)
   1. The tables created should be named team, team\_affiliate, game\_schedule, player, lineup, and roster.
   2. Make sure your code creates tables if needed and that it can handle data reloads, merges, and/or updates

* *Database Location:*
  + *Host: LocalHost*
  + *Username: ‘sa’*
  + *Password: ‘tEST1234’*
  + *Server: ‘sql\_server’*
* *Python Script: ‘final\_clips\_proj/extract\_clips\_data.py’*
  + *Structure of Script:*
    - *Library Imports*
      * *Utilities Library is user created and I have used it for all data upload/extraction to SQL Server.*
      * *SQL Server Connection: connection = sql\_server()*
      * *Create database: connection.create\_datebase()*
      * *Upload Data: connection.sql\_query\_bt()* 
        + *It first builds queries associated to creating, and merging tables.*
        + *Then creates the staging table + production table through the SQL Server connection.*
        + *The staging table is first created. Then the merging statement is used for the final upload to the production table.*
        + *This function takes into account any merging or updates to records. (If any errors occur during merging and error is displayed).*



1. What should be the Primary Keys across the tables?

* *Game\_schedule – game\_id*
* *Lineup - team\_id, lineup\_num, game\_id, player\_id*
* *Player – player\_id*
* *Roster - team\_id, player\_id*
* *Team - teamId*
* *Team\_affiliate - nba\_teamId (if a team had more than one affiliate we could add glg\_teamId to the primary keys)*

## Question #2:

Basic Queries (SQL)

1. Write a SQL query that can calculate team win-loss records, sorted by win percentage (defined as wins divided by games played)
   1. Final table should include team name, games played, wins, losses, win percentage
2. 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

* *Final Table: select \* from lac\_fullstack\_dev.dbo.q2\_final*

## Question #3:

Schedule SQL Queries (SQL):

1. 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 example, LAC’s game on 1/27 at BOS would be an Away-Away B2B since LAC played 1/26 at TOR.

*Final Table: select \* from lac\_fullstack\_dev.dbo.q3\_a\_final*



* 1. For the data given which team had the most Home-Home B2Bs?
     + *This is a tie between Charlotte, Detroit, Golden State, Houston, Indiana, Memphis, Milwaukee, Philadelphia, San Antonio, and Washington.*
  2. Which had the most Away-Away B2Bs?
     + *Also a tie. This tie is between Houston, Indiana, Minnesota, Orlando, Philadelphia, Portland, and Utah.*

1. Which team(s) had the longest rest between 2 games and what were the days of the 2 games?

*Final Table: select \* from lac\_fullstack\_dev.dbo.q3\_b\_final*



1. 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).

*Final Table: select \* from lac\_fullstack\_dev.dbo.q3\_b\_final*

* *For this query, I would love feedback to understand if my approach to define 3 in 4s was correct. Or if there was as simpler approach.*

## Question #4:

Lineups Queries (SQL): In answering any of these items, feel free creating intermediate temp tables, inline tables, or CTEs as needed.

1. 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)

*Final Table: select \* from lac\_fullstack\_dev.dbo.q4\_a\_final*

1. 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)

*Final Table: select \* from lac\_fullstack\_dev.dbo.q4\_b\_final*

1. 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.

*Final Table: select \* from lac\_fullstack\_dev.dbo.q4\_c\_final*

1. 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
   1. each set (all/wins/losses) should have # of games, average stint length, average number of stints

*Final Tables: select \* from lac\_fullstack\_dev.dbo.q4\_d\_all*

*select \* from lac\_fullstack\_dev.dbo.q4\_d\_wins*

*select \* from lac\_fullstack\_dev.dbo.q4\_d\_losses*

## Question #5:

Front-End Lineup Visualization (any frontend tool(s)/libraries you’d like)

1. 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:
2. 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
   1. 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
   2. You have the freedom to choose the visualization; it can be displayed as tables, a figure, graph, or something else.
   3. 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.)

* *For this question, I attempted to start a Django Application. Currently there are two links in that work, but are not finalized.*
* *Links:* 
  + - [*http://0.0.0.0:8000/league/*](http://0.0.0.0:8000/league/)
    - *http://0.0.0.0:8000/team/*

*I was hoping to have two views, one for a coach or team staff and a separate view for someone in the front office. A coach might want to see actual analysis on data of last nights game for the Clippers, while someone in the front office might be working on a trade and wants to know the full analysis of a player on another team.*

*Coaches would have one profile where they can see the team analysis, potentially sensitive player information, while a front office employee only gets high level stats of each player.*

## Question #6:

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?

*I have only used AI whenever I encounter errors that cannot be simply searched. Most of the time (ex. ChatGPT) doesn’t give the correct answer, but it leads me down a specific path I either did not think about or did not know existed. For example, during this assignment I was troubleshooting an SSL error when connecting to SQL Server in Docker. I am familiar with SSL errors, but in this particular case I didn’t know how to ‘dockerize’ the download of the certificates. So ChatGPT gave me a ‘generic’ answer for getting certificates which lead me down the right path to find the specific certificates I needed.*