Data Engineering Interview prep

**1. Python Data Stack**

* Pandas and Numpy for data manipulation
* PySpark or Dask for handling large datasets
* Scikit-learn or similar for featurizing and processing data
* Best practices for automating data tasks (e.g., using Python scripts, scheduling jobs)

**2. Data Onboarding and Reconciliation**

* Data exploration techniques
* Automating data quality checks and validations
* Processes for onboarding datasets and classifying data
* Methods for reconciling data discrepancies (diff checks, error tracing)

**3. ETL Pipelines**

* ETL tools (Airflow, Prefect, Luigi, etc.)
* Building and maintaining automated ETL pipelines
* Handling and transforming financial datasets during ETL processes
* Monitoring and debugging ETL pipelines in a production environment

**4. Data Debugging and Anomaly Detection**

* Techniques for finding anomalies in datasets (e.g., outlier detection, validation logic)
* Tracing issues to the root cause in a data pipeline
* Deductive reasoning to debug complex data flows

**5. Data Analysis and Featurization**

* Cleaning and preprocessing financial data
* Featurization techniques (e.g., time-series analysis, aggregation)
* Data enrichment processes
* Using SQL for data analysis and querying large datasets

**6. SQL and Databases**

* Proficiency in at least one SQL dialect (PostgreSQL, MySQL, MSSQL)
* Writing complex SQL queries (joins, subqueries, window functions)
* Query optimization techniques
* Handling database operations on financial data

**7. Linux Command Line**

* Navigating the Linux file system
* Shell scripting and basic system administration tasks
* Managing processes, logs, and environment variables on Linux

**8. Production Support and Monitoring**

* Techniques for proactive monitoring of data pipelines
* Handling incidents and resolving issues in live trading environments
* Communication with internal customers regarding production issues

**9. Financial Datasets**

* Familiarity with Refinitiv, S&P, and Bloomberg data structures
* Understanding financial data conventions (tickers, pricing, metadata)
* Working with market data APIs

**10. Soft Skills**

* Communicating technical issues and resolutions with stakeholders
* Collaboration across teams in a trading environment

These topics align with the job description and should prepare you for a range of questions during the interview. Let me know if you'd like to dive deeper into any of these areas!

**Questions on Data Pipeline:**

1. What is a data pipeline?
   1. A mechanism to transfer data from point A to point B with multiple steps like data cleaning, data transformation, etc. which are called C, D, E.
   2. [A, [C, D, E], B]
2. Why data pipeline?
   1. It depends on what you want to build
   2. For any business decisions, or trading decisions, we need clean data. We need products that are being traded, in what strategy, etc. To make these informed decisions, we need a data pipeline that will take in raw data and spit out clean and transformed data.
3. Difference between data pipeline and ETL pipeline?
   1. ETL pipelines fall under the umbrella of data pipelines.
   2. It’s an example of a batch data streaming, where you extract, transform and load data.

SQL questions:

* What is a difference between primary key and foreign key?
  1. Primary key is a value with unique values in a table
  2. Foreign key is a column that is used to link 2 tables.

Resume-based questions:

1. **What is an ETL**
   1. ETL stands for Extract Transform Load, which is a mechanism used to read and write data. You start with extracting the data, it could be from a file or an API, etc. you transform it using some statistical tools like pandas and then you load it to a file or a database table.
   2. ETLs are extremely important in data science and data engineering, because we work with data so heavily, this allows us to concatenate data from different resources into one big data structure or a data frame, and we can perform transformations on it.
2. **Difference between ETL and ELT?**
   1. ETL is used when the data is raw and unstructured, so when loading into a data structure or a DB, it needs to be transformed
   2. ELT is used when data is raw but structured, so it’s easier to transform data after loading it. This can be used for reporting purposes, when the data is extracted and uploaded to the DB but is transformed for reporting purposes.
3. **What is a data warehouse?**
   1. Data warehouse is where all data is stored. Readable query
4. **Give me an example of an ETL pipeline you created using airflow**
   1. For the options
5. **Tell me about yourself**

* **Why data engineer?**
  + I graduated from depaul with a bachelor’s in computer science, and while I enjoyed what I did, at XR I was a part of data infrastructure team where I was able to work with large-scale data and understand how it’s structured, processed, etc. I have really enjoyed understanding data and coming up with solutions on how we can leverage that to make business decisions. With this, I also developed an interest in predictive modeling, when given a set of data, predicts an outcome. This was fascinating because there is so much a data can give you and now, it’s become invaluable.
* **Difference between data engineer and data architect**
  + A data architect is someone who’s responsible for managing servers and building out the data system for the company, which is able to handle the data load that the company generates.
  + Data engineer is someone who works with data, and the build-out system, and works on providing high quality data to make informed decisions.
* **Tech-stack needed for data engineers**
  + SQL
  + Python
  + Hadoop (used for big data)
    - Hadoop uses the map reduce algorithm to divide and conquer the data and run the process on chunks of it in parallel.
  + Especially in trading, data science is extremely helpful in understanding our orders, which strategies are making more revenue, etc.
* **How do you onboard a new dataset**
  + To onboard a new dataset, first we need to understand why we need it so we can better add structure to it. Understand the source of data, is it through an API, csv, xml, txt, parquet, etc.
  + Get the specifications, on what each column means, and its type.
  + Then make an informed decision for the structure of the DB
* **Tell me a time you worked on a challenging process at work** 
  + As a data engineer, one of the challenges I had was working with an API that spits out unordered data. We used one of the analytics tools called corvil which had all the exchange-order data, as well as network data. The goal was to build out a tool that gets the number of orders for a given time for a given exchange and symbol and create visualizations on graphana. I had to make calls to Corvils API to get the required data, but the output was unordered. As in, at every call it would return a random number of columns which made the batch processing complicated. It did come with headers, but parsing headers at each call with make it inefficient and slow. I did struggle with this issue quite a bit, and after asking other employees who have worked with packet data, I found out that we had access to Corvils API code. I had to figure out a way to tweak their code and add the ability to only return a certain set of required columns. This made the process streamlined and processing much faster.
* **Working in teams**
* **Can you describe a time when you had to collaborate with a cross-functional team to complete a project?**
  + Most of the work I did was for internal stakeholders, I like to call them internal clients. I would get a ticket to build out a process that does this, this, and this. I would need to start by reaching out to the person who has created the ticket and understand their requirements. Most of the times, it’s a non-technical person so they can’t explain in technical terms so it’s best to always hear their thought process for the end goal. There have been times, that what the ticket was for, and what came out of the call with the person is completely different, because we are better able to understand the technical side of things and ask better questions. Collaborating with other teams gave me a lot of industry knowledge than I would have just with being a data engineer
* **Walk me through a project you worked on** 
  + I worked on many projects and one of the most time sensitive projects I have worked on is Options span optimizer. A process to calculate the SPAN margin which identifies the overall risk for a portfolio. This is to make sure we don’t have long and short positions of a same option in different trading accounts. According to the margin in the account, desired positions file was created based on the settlement file of the options desk which had the list of stocks and their corresponding accounts. This was determined using the black scholes algorithm. The process I created was to collect data from different data sources to understand the existing symbols in an account, and top-day activity, and create a csv file with instructions to transfer the positions if they don’t match the desired list, thus fixing the long/short positions. This was an extremely time sensitive process, so a lot of error handling had been done. We integrated slack API so the output and error messages could directly be sent to the appropriate channels.

**Data Engineering & ETL**

1. **ETL Pipelines:** Can you walk us through an ETL pipeline you've designed and implemented? What were the main challenges, and how did you handle data validation?
   * One of the examples of an ETL pipeline I have built is an order action table for crypto. The main goal was to streamline data from all crypto exchanges we were listed on into one golden source of data. The extraction part of the process was getting data from API which was in raw form, transform it to fit into an order action format, and load into a database. This sounds very simple but was it was complicated because of all the nitpicks, and security concerns. There were ofcourse data inconsistencies between values such as timezones, currencies, value mappings, naming conventions, pricing units, and just standardizing all exchange data into one big source.
2. **Data Onboarding:** How do you typically onboard a new dataset? What steps do you take to classify, clean, and validate the data?
3. **Python for Data Engineering:** How do you automate data processing tasks in Python? What libraries or frameworks do you prefer for this?
   * Based on the kind of task to automate, it’s either finding data from different data sources, and performing calculations, and spitting out a daily report or load data to a DB. We mostly use python’s sqlite to connect to sql server, pandas, csv, excel for data manipulating, cleaning and transformation, and smtplib for sending emails.
4. **Financial Datasets:** Have you worked with financial datasets like Refinitiv, Bloomberg, or S&P? How did you structure and reconcile data from these sources?
   * I worked on Finra CAT reporting for which we had to tap into a lot of sources, and we also worked with third party firms like spiderrock/dealerweb who generated or submitted data on our behalf.
5. **SQL & Data Debugging:** When dealing with anomalies in a large SQL-based dataset, how do you approach debugging and identifying root causes?
   * I owned the technical side of some of the reporting processes which took almost several hours to run, and finding an anomaly in trading data can be a tedious task. the best way to handle is to add reconciliation checks and creating stored procedures to spit out the entries with missing data or errors. Especially in trading data, we also add volume and order action count checks to make sure we aren’t missing any data, between the drop source and our internal source. Having these reconciliation stored procedures help debug most of the anomalies. Despite the sprocs, the best way to debug would be to break down the dataset into smaller chunks, mainly using time, and conditions to find the exact record with the issue. This also requires patience.
6. **Describe your experience with low-latency systems and how you have optimized them in the past. OR Discuss a time when you improved the performance of an existing system. What tools did you use?**
   * When working with large-scale datasets, it’s important to pay attention to their latency. With SQL, the best way to optimize these systems is to use indexes for selecting a table, and only selecting the necessary columns. Using temp databases for redundant calculations and clearing it up when necessary. For one of the processes, after optimizing the SQL queries, the time to run it went down from 7 hours to 4 hours.
7. **Can you describe a situation where you had to balance competing priorities in a fast-paced environment?**
   * Ofcourse, as we know trading firms are fast paced and data integrity is extremely crucial. There are some processes that load data from external sources to our internal sources and that data is crucial for daily trading. Some data is crucial for start of day reconciliations and there have been times when multiple processes have failed, because of different or common issues. Multi-tasking and time management skills come into play when this happens, understanding which processes are top priority and balancing out the time it takes for each to run have helped me in this situation.
8. **What methods do you use to test and validate trading algorithms before deployment?**
   * I like validating these algorithms on historical data, and ensuring the result is accurate. With pricing, and volume, we need to be able to match it down to a cent. Having a test server that is a clone of the prod has proven to have helped a lot with all the testing since it has all the historical data.
9. **Describe your experience with real-time data processing and analytics.**
   * As I mentioned, I have worked on a python process for entering order actions data for crypto exchanges, and it was in real-time, as we want to get all the trades down to the minute or whatever’s available on their API. For this real-time processing, we divided the data in chunks and processed each chunk for faster and accurate processing.
10. **How do you ensure code quality and reliability in critical software systems** 
    * Extensive error handling and there’s no harm in adding debug statements when necessary
    * Test driven development is important for a reliable code so we can ensure we covered all the edge cases.
    * I am pro code review, because I think they help a lot, professionally as we as personally. Because of those, my coding skills have definitely improved.

**Data Analysis & Validation**

1. **Data Validation:** How do you perform data reconciliations and quality checks when working with large datasets? Can you provide an example where you detected and resolved inconsistencies?
2. **Featurizing Data:** How do you assist in cleaning and featurizing data for a machine learning model? What steps do you take to ensure the features are relevant and high-quality?
3. **Data Anomalies:** When you encounter anomalies in derived datasets, how do you trace them back to their source? Can you give a concrete example?

**Python & Machine Learning**

1. **Python Debugging:** If a Python script processing trading data throws an error in a live environment, how would you go about debugging it efficiently while under time pressure?
2. **Machine Learning:** You mention collaborating with a Data Science team on machine learning models. Can you explain a machine learning model you helped implement and how it was integrated into a data pipeline?
3. **Generative AI Chatbot:** How did you implement the generative AI-powered chatbot for USMLE prep? What specific challenges did you face with the retrieval-augmented generation technique?

**Financial Data & Production Support**

1. **Trading Environment:** How do you ensure the integrity of data used in a live trading environment? What steps do you take to mitigate risk?
2. **Real-Time Data Integration:** How did you handle the integration of real-time trading data using DropCopy or FIX? What were the biggest challenges?
3. **Data Production Support:** In a trading environment, what steps would you take to proactively monitor a data pipeline to avoid disruptions?

**General Technical Knowledge**

1. **SQL Optimization:** How do you optimize SQL queries in high-volume environments? Can you provide an example of a query you optimized for better performance?
2. **Linux:** How comfortable are you with the Linux command line? Can you describe how you've used it in your data engineering tasks?
3. **Kafka & Airflow:** How have you used Kafka or Airflow in your previous projects to manage real-time data or schedule tasks? What are the pros and cons of each?