Trogiue

Download

Interview Analytics

Henry
ETL Developer

Overall Interview Score | 3/10
Technical Score (Including Coding) | 3/10

In Review

3
3
3

maybe a temporary issue, I will set to retry, so it will retry, and then if it still fails, the alert will trigger, like, it will, the alert triggers and the pipeline will stop so that whatever bad data it is, it will not move across the stage layers. So by doing this, it will ensure that this will make the ETL failures easy to detect, easy to detect, and then quick to fix, quick to catch this issues on time and then recover from them.

Answer Match 80% | Proficiency N/A | Duration 04:48

Category: ETL | Difficulty Level: Hard

Explain your approach to designing ETL for real-time or near-real-time processing.

Like a career. Engineer of implemented every time in visit, I could remember when I was trying. Had actually handled real time data differently. I had to implement the Spark structure streaming to handle retirement, then when I was in the NHS, I could remember using it. To handle every time, so depending on the. Depending on the on what I'm working on, like depending on the stock, what I'm working on, like for the real time, for the real time processing. I will first ensure that like I design the pipeline in a way that the data would move in like small blocks or small batches. To avoid delay like to ensure that the when the data is streaming it will go. It will just stream with a very low delay although. Or or those why very slow delay, like very low delay, I'll still ensure that it stays relax. So and then I just ensure that I choose the right thing. Injection, injection to. So if like if let's say if a true streaming is needed, I can use it. I've used, I've used this Kafka. So I could use Kafka. I could also use a I could also use a Kinesis as well. So Kafka or AWS Kinesis, we capture these events as they are happening. As it happened then for taking about for nearly time, I can use a spark like I I can use a micro batching with a spark structure screaming so that data is processed every seconds or minutes. That's in that way the data were catching the data in every time as well. So and then I ensure that the transformations, I would design the transformations in a way that is going to be incremental and it will be lightweighted as well. So by showing that it's like we take that way ensure that the transformation work in the way that we avoid using a heavy joint. Using a full table scan 'cause they're working with billions of data scanning through the entire tables with the job to slow down, so I will ensure that. I use a look up tables. I can use a broker joints as well or and then I use a greater detail. To ensure that my. My streams are lightweight so I'm I'm sure it will ensure that the stream will keep up like with the. Incremental data that is coming on like with the rates of the income and data incoming so and then I ensure that the data format as well. I ensure that I store the data formats in like ADI store the data in the format that would support quick writes like fast writes and then fast update so, I could use it for the for my data movements and my retired data movement. I could also use a delta leak. Because it's very ideal in in such scenarios we got there. It's aside just being ideal, it handles. Transaction was like ACID transactions on there as I just handling our ACID transaction. He also handles A schema enforcement, he handles A checkpoints and then it does this exactly what's processing. So and you know that checkpointing is very important because it will it will checkpoint is very important because it's going to like. Without the processing, the entire Sure that the pipeline is elastic. 'Cause every time loads can spark, you can spark at any point in time, so I'll ensure that. Your dedicated clusters like. Ensure the job doesn't lag behind so and therefore my output layers. I ensure that. What I do is that I always write the clean records to a silver table and then I create aggregate. Have the educated data then study created here. The downstream Downstream systems with the. It is it will get quick of this. And then monitoring is also ensure that I implement monitoring because it's very. Critical here. I'm sure I also track latency as well because. And how data is processed per second and there are. A lot. I also implement a lot in as well. Check for field batches and then I will focus on the latest Very key, very important. Thank you.

Answer Match 60% | Proficiency N/A | Duration 05:33

Category: Python | Difficulty Level: Easy

How would you structure Python code to handle large ETL jobs efficiently with clear error handling?

Working with Python, I ensure that this, like working with Python codes, I ensure that like I talk with me with that they handle my ETL jobs efficiently with the clear error handling. I do them in a way that, like I show, that my codes are modular. So I will ensure that I cleanse my codes in a very modular way so that it's easy to manage, easy to maintain, and also easy to debug in times where you need to debug and ensure that it's modular. It's easy to debug and also ensure that it's scalable when you're dealing with large jobs. So what I do is that I will start by breaking down the several separate models. I will have different modules; one could be for config, one for extraction, one for transformations, one for loading the tables, one for utilities like logging. Several modules could be one could be the wrapper, which could be the main structure. Then the pipeline is easy to reuse across environments. For performance, I ensure that I use chunking or batching when pulling data from my data sources and ensure that I avoid overloading everything. Especially when I'm working with Spark jobs, I ensure that I tune my partitions where necessary. I would use broadcast joins for small lookup tables. Then for error handling, I ensure that I wrap each main step in try and except blocks. So in case of a failure, if an extraction fails, I will log the exact error and stop the pipeline. If a transformation fails, I ensure that I log the records of the particular batch that caused the error and quarantine it. Either quarantine it or skip the entire batch, depending on the business rules predefined. I ensure that errors get logged to a central log table, with timestamps and IDs. I keep the job ID. I also use custom exceptions when dealing with common issues like a schema mismatch or missing configurations. This makes debugging much easier because the error messages will be very clear. Finally, I ensure that my main ETL script works in a way that it will flow; the data will flow and extract data, load, and then every step will return status codes. By doing that, if anything breaks, I will be able to catch it on time. The pipeline will fail fast; log it, and send a notification because I also ensure I implement alerting and monitoring. This structure keeps my code modular, scalable, and easy to troubleshoot when working on a large job and experiencing failures in the pipeline.

Answer Match 80% | Proficiency N/A | Duration 05:30

Category: Python | Difficulty Level: Medium

What is your approach to packaging and deploying Python ETL scripts across multiple environments?

My approach to deploying a Python script across multiple environments. Just a different one. The one approach I use, like my favorite approach, like what I tend to use is, like I'll ensure that I package and deploy this Python script the same way that's like, I'm sure I do it in the same way I would deploy any production code. So what I mean is that I'm going to ensure that I make it versions, I try and make it very, very portable, and then I will ensure that I make it only through, like, a very specific, only through, like, the config. What I do is, like, firstly, I structure the ETL, like I will structure it like in Python package with the clear layout declare. With the models you to use. We also have a set of five, so all environment-specific things. Parts, the table names, the secrets, they were all going to separate config files for development and then for the testing team as well to go into separate config files, and then for the production team as well. So what I do is that I use it like a requirement file to. To like I use the requirements fine. So by using the requirement for ensure that dependencies will stay consistent everywhere and then I then push everything into it and then my CI will build the package lasting wheel by presently like a wheel or like. To be in different way it does depends on the technology. So for the like a few. Presently what I use is. I use this a Python set of the PYP, this thing that's called wheel that is working with the DISDIQ module. So and therefore deployment. What I do is like I don't change the code. I only the config is what I change for each environment. So and then I ensure that in my CD pipeline I will place my CD pipeline in. Promotes the same package from development from. Thing to production. So by doing that I will use ensure that I use the correct configs and correct secrets for each step so deployment. I also ensure that deployment is usually done through CLI or in job scheduler. Just depends on what platform I'm working on so. In that way, I will ensure that my ETL will stay identical. Across all environments, whether they're testing or production. And the only I only the config will change. So by doing it like that it will ensure that I keep things predictable, easy to debug, and it will be very very safe for production so that.

Answer Match 80% | Proficiency N/A | Duration 04:11

Category: Python | Difficulty Level: Hard

How would you design a fault-tolerant ETL framework in Python that scales dynamically?

OK, in Python, like I'll, I'll design a photo around to eat your framework in a way that I ensure that it's virtual. That ensure that his skills is like he's scalable and I show that my codes are clean, my cools are modular. Way that it followed the configs will follow is a good structure, like it will follow. Structure, uh, as I went dealing with. Dealing with. Skill, data skill. Dealing with the task here. So doing doing this, I will show that. I use like. I follow through a process like what I did. I just I start to like designing it in a way that the architecture like, I will just make sure that I split. Split it into a plane with those like. The data injection, the buffering, the queuing the processing persistence, even the monitoring, the utility configure ensure that I I also keep the orchestration separate, like the. Depends on the occasional thing you use, but maybe. Air flow, Prefect, or database job, so workers can only run on processing logic. Then while in buffering or ingesting. So that I push events into a durable queue like a Kafka kindnesses or whatever streaming device I'm using import at that point. So by doing that, the queue will give the coupling, it's going to give a back pressure, it's going to give me replay. So for each. We use an insertion layer that we write into like is that it's going to write data across the layers and if you write into even the staging folders as well, which could be my S3 or my ADLS Gen. 2. And then after writing it will also emit events. So and then I will ensure that for. In in my processing mode there I will use a stream processing like a. Or lightweight consumers when dealing with micro batch workers, depending on the latency needs, like depending on the latency needs to ensure that I make the workers like the importance. So by doing that I will just ensure that I include. Unique key or like a unique event and. Use this duplication logic to avoid double processing. So I also ensure that I implemented checkpoints in like ensure that I persist the offsets. I also checkpoints in a durable store try like a walking way my Kafka consumer like I will handle the offset properly. And then for this. Transforms and we ensure that we use like a state stores within my. Delta tables so so I can also restart or resume from where and I where I stopped. Like I showed that it's a city, it's a acidic nature. Also ensure that I implement a error handling and retries. I'm sure that I set retries and remains error handling logic. Be meant like an exponential back off, like for, like an inspirational back off retry for. Arrow that I that they occurs and then I'll ensure that I classified my errors. I retry and then I did. In terms of retrying whatever is bad at quarantine, whatever is good will continue across digital layers. So I also ensure that where I sync some data, like dealing with the transaction, I ensure that I sync some data where possible. And then working with skill and I show that my. Autoscaler workers and. And also very important and ensure that they are also based on my CPU memory. So and then I will just ensure that I use. Orchestration which could be like a Kubernetes or. Money services with with my with the authors. So and then our insurance also ensure that I partition top. So like I should have. For example, maybe I'm working with Cathcart office and so that I will partition this topics so parallel consumers can scale horizontally without any contention. So by doing that I will ensure that I will use separate clusters for.

Answer Match 60% | Proficiency N/A | Duration 05:00

Category: Unstructured | Difficulty Level: Hard

What strategies do you use to stay updated with evolving data engineering technologies?

I use a. There are lots of strategies I used to still dictate. There are some channels I sign up to on YouTube to stay updated with the. Commit upcoming technologies and you know, data engineering is evolving very fast. So I have some YouTube channels I subscribe to. I also subscribe to TLDR. To Medium as well. I also follow do some like ensure that I follow like official documentations as well. Releases as well for tools like. I use. Database, Spark, Cloud, and the VSA Cloud platform that you so and then I should follow some newsletters like I mentioned some YouTube channels, Medium somewhere newsletters as well and then, in my free time I just said cy like a new try. New things like those do like this. I just do any small sandbox project just to. Myself updated with trend and I'm also very active. I ensure that I still very active on my community from community forums like guitar discussions and all, and then sometimes I join internal knowledge sharing as well. Like Italian sharing sessions within my an organization, I ensure that I joined those sessions or any tech talks that are scheduled so I enjoy them by then. By doing that, I ensure that I still did it like that. And also I I, I bring out time in the week for learning. Some hours in the week. Every with what I do present is every week I bring her six hours for learning so and then I split it across this cross money to Saturdays. So I try to learn something new to stay updated with the evolving trends in big data engineering

Answer Match 80% | Proficiency N/A | Duration 02:51

Category: Unstructured | Difficulty Level: Medium

How do you handle team disagreements during architectural decisions?

Team arguments and team arguments and then arguments or do from disagreements actually doing architectural decisions. So there are various ways I do it. Just depends if there's a disagreement on a particular decision, an architectural decision, I will just ensure that I keep it structured and objective in a way that I will ensure that everybody, I will make sure that I give everybody a rope or give everybody opportunity to like explain their view clearly. In a way that aside them just explaining, I ensure that we listen to like everybody will listen to understand one another to get their view and then we can then list the options for solution. Then we compare them based on the facts. And the facts we look at here are costs. We always look at the cost if cost effectiveness, we look at the performance, we look at the maintenance. We also look at the security. So and then we ask ourselves questions like with this be long term so. And then sometimes if needed, we could just do like a quick POC like we also a quick POC to validate this like to gidgets like or to let it to benchmark. To do a benchmark and then get real real data like by doing the POC. But do you need to review data? Trying the experimental to ensure that instead of just depicting opinions, we are keeping the discussion focused on the problem, not just the people's opinions. We still can't that line. They will have to take it up to the possibly the team leader, the people, the people leader, the people manager and therefore a final call. So and then maybe once decision is made, we won't commit to it, the goal. The goal of the job is to keep collaboration active within the team and so mock up and to always move on. Without the move on to move, ensure that we move on, we move. Without the blocking and delivery. That's just how I handle disagreements within a team when dealing with that.

Answer Match 80% | Proficiency N/A | Duration 02:48