Q5) You need to join data from two different sources in an AWS Glue job. Explain how you would do this and any considerations you need to take into account.

Step 1: Define Data Sources

In this instance, we are joining data from two different sources: student enrollment records stored in an Amazon S3 bucket and curriculum information from an external API. These are two data sources.

Step 2: Data Source Configuration

In the AWS Glue job script, configure the data sources. Use AWS Glue DynamicFrames to work with the data. Create DynamicFrames for each data source, specifying their schemas and locations.

```python

# Create DynamicFrames for data sources

enrollment\_data = glueContext.create\_dynamic\_frame.from\_catalog(database="enrollment\_db", table\_name="enrollment\_records")

curriculum\_data = glueContext.create\_dynamic\_frame.from\_catalog(database="curriculum\_db", table\_name="curriculum\_info")

```

Step 3: Data Transformation and Join

Prepare the data for the join operation. Select specific columns, rename columns, or filter rows for data transformation. Then, use the `join` transformation to join the two DynamicFrames based on a common key, such as the student's id.

```python

# Join the two DynamicFrames

student\_curriculum\_data = enrollment\_data.join(

frame1=enrollment\_data,

frame2=curriculum\_data,

keys=["student\_id"],

transformation\_ctx="student\_curriculum\_data"

)

```

In this example, we are joining student enrollment records with curriculum information based on the `student\_id` column.

Step 4: Data Sink Configuration

Determine where the joined data will be stored.

```python

# Create a DynamicFrame for the target data location

joined\_data\_destination = glueContext.create\_dynamic\_frame.from\_catalog(database="joined\_db", table\_name="joined\_data\_location")

```

Step 5: Data Write Operation

Write the joined data to the target data location using the `write\_dynamic\_frame` method. Specify the format and compression options.

```python

# Write the joined data to the target location

glueContext.write\_dynamic\_frame.from\_catalog(

frame=student\_curriculum\_data,

database="joined\_db",

table\_name="joined\_data\_location",

transformation\_ctx="joined\_data\_destination",

format="parquet"

)

```

Step 6: Error Handling and Logging

Implement error handling and logging to track the progress of your AWS Glue job and handle potential issues gracefully.

Considerations :

1. Data Compatibility

Ensure that the data types and formats of the columns you're joining match between the two data sources. For instance, the `student\_id` should have the same data type in both sources. A common mismatched in data type might be varchar and int in the case ‘student\_id’

2. Data Granularity

With two sources passing through the pipeline, assuming that the data would be inserted to a table in the data lake, we need to consider the granularity of the sources. We would need to consider modeling the data to fit into a table. In this example, the granularity is on per student-wise:

Registration table

| fact\_id | student\_id | enrollment\_date | curriculum | grade |
| --- | --- | --- | --- | --- |
| 123dwe-3123 | 2 | 12/9/2023 | math | 3 |
| 2313-fggh54 | 3 | 13/9/2023 | science | 4 |

3. Data Volume

With more than 800k active users with Koobits, the volume of data can be substantial. Optimize the data processing and consider partitioning would be a strong consideration.

5. Data Security

Protect student and curriculum data with encryption and access controls to comply with PDPA.

. Testing

Thoroughly test your AWS Glue job with real-world data to ensure that the join operation produces accurate results. Join operation if done wrongly could result in duplication of records easily. A tool called DBT is useful for testing and version-control complex queries.

5. Documentation

Document your AWS Glue job, including the data sources, transformations, and the join logic. This documentation is essential for maintaining and troubleshooting the glue job.

By following these steps and considerations, we can successfully join data from two different sources in an AWS Glue job while ensuring data accuracy, performance, and security for tasks like student enrollment and curriculum management.