These instructions guide you through creating the ParsedResumes DynamoDB table, emphasizing efficient schema design for optimal data storage and retrieval.

I. Creating the DynamoDB Table:

1. **Open the DynamoDB Console:** Log in to the AWS Management Console and navigate to the DynamoDB service.
2. **Create Table:** Click "Create table."
3. **Table Name:** Enter ParsedResumes.
4. **Primary Key:** Specify ResumeID as the primary key. Set the data type to String.

5. **Attribute Definitions:** Define the table's attributes. This is crucial for efficient data retrieval and analysis:

Attribute Name	Attribute Type	Key Type
ResumeID	String	HASH
ApplicantName	String	-
ContactInfo	String	-
Skills	List	-
Experience	List	-
Education	List	-
Projects	List	-
Keywords	List	-

6.

Important Notes on Data Types and Structure:

- **Skills, Experience, Education, Projects, Keywords (List):** These fields store lists of items. It's best practice to store these as JSON arrays within the DynamoDB table. This allows for flexible data structures within each item. For example, the Experience attribute could store a JSON array of objects where each object represents a work experience entry containing fields like CompanyName, JobTitle, StartDate, EndDate, and Description.
- **Data Types:** DynamoDB uses specific data types (String, Number, Binary, etc.). Ensure that the data types you define in your DynamoDB table match the data types of the data your Lambda function is extracting and storing.
- **Data Transformation:** Your Lambda function will need to structure your extracted data appropriately before storing it in DynamoDB.

1. **Provisioned Capacity:** This controls the read and write throughput. Start with low values (e.g., 5 for both read and write capacity units) and allow DynamoDB to scale automatically as needed. You can increase this later if necessary.
2. **Create Table:** Click "Create table" to finish.

II. Verifying Table Creation:

1. After the table is created, navigate to your ParsedResumes table.
2. Verify that the primary key and all the attributes have been defined correctly and that the data types match your expectations.

This detailed guide ensures the correct setup of your DynamoDB table. The structured approach, using lists of JSON objects, provides significant flexibility and enhances the effectiveness of your data analysis using QuickSight. Always test your DynamoDB table configuration before moving on to other parts of the project. Incorrectly defined data types can cause issues later in the process.