A Project Report on

ETL Pipeline for Data Processing using AWS S3, PySpark, and Snowflake

PREPARED BY:

ASSOCIATE ID	**ASSOCIATE NAME**
2387739	**Aryan Khokle**

GUIDED BY:

Mentor	**Coach**
Govindaraj, Kavin Kumar	Yalin Maria
Associate	CPO L&D GenC Training,
AIA Cloud Data Integration	Human Resource – GenC

Contents

List Of Figures

Introduction

ETL Pipeline for Data Processing using AWS S3, PySpark, and Snowflake

This project aims to develop a strong ETL (Extract, Transform, Load) pipeline for efficient data processing and loading into a cloud data warehouse. The pipeline retrieves raw CSV data from AWS S3, cleans and transforms it with PySpark on AWS Glue, and then loads it into Snowflake using its robust COPY command. The primary objectives are to maintain data quality by eliminating duplicates, managing null values, rearranging columns, casting data types, and truncating excessively long text values. This report details the system architecture, data flow, database design, ETL implementation, and additional aspects.

Tools and Technologies Used

- **AWS S3:** Used as the data storage layer for both raw and processed data.
- **AWS Glue:** A managed ETL service that supports PySpark based transformations.
- **PySpark:** Used for data cleaning and transformation.
- **Snowflake:** The cloud data warehouse where the final processed data is loaded for further analytics.

Purpose of Project

The main goal of this project is to create a robust ETL pipeline for data processing that can:

1. **Efficiently Process and Store Employee Data:**
    - The system automates the extraction, transformation, and loading of sales data from various sources stored in AWS S3. This ensures quick and accurate processing, reducing manual effort and minimizing errors. The processed data is stored in a structured format, ready for analysis.
2. **Ensure Data Quality:**
    - Using PySpark on AWS Glue, the system performs data cleaning and transformation tasks such as removing duplicates, handling null values, reordering columns, performing data type casts, and truncating overly long text values. This ensures high data quality and consistency.
3. **Enable Comprehensive Analytics and Reporting:**
    - The system supports advanced analytics by loading processed data into Snowflake. It generates detailed reports on key performance indicators, such as top-performing employees and department-wise employee performance. These insights help managers make informed decisions.
4. **Support Data-Driven Decision-Making:**
    - By providing real-time insights and historical analysis, the system empowers decision-makers to respond quickly to market changes, optimize operations, and develop effective business strategies. Data-driven decisions enhance competitiveness and drive growth.

About the Software System

The ETL Pipeline for Data Processing is a comprehensive solution that integrates various AWS services and Snowflake to manage sales data effectively. The system performs the following key functions:

- **Data Extraction:** Extracts raw CSV data from AWS S3 for initial storage.
- **Data Transformation:** Cleans and transforms the raw data using PySpark on AWS Glue, ensuring data quality by removing duplicates, handling null values, reordering columns, performing data type casts, and truncating overly long text values.

- **Data Loading:** Loads the transformed data into Snowflake using its powerful COPY command for efficient storage and analysis.
- **Reporting:** Generates detailed reports on key performance indicators, such as top-selling products and store-wise sales performance, to support comprehensive analytics and informed decision-making.

Scope of the System

1. **Data Ingestion:**
    - Automating the process of uploading employee data to AWS S3. The data is initially stored in the "starting" folder as untransformed CSV files.
2. **Data Processing:**
    - Implementing data validation, transformation, and aggregation using AWS Glue with PySpark. The transformed data is stored in the "transformed" folder in S3. The transformation process includes:
3. **Data Storage:**
    - Loading the processed data into Snowflake as a cloud warehouse. The connection setup includes:
4. **Analytics and Reporting:**
    - Providing SQL queries and reports to analyze employee data and support business decisions.
5. **Scalability:**
    - Ensuring the system can handle increasing volumes of employee data as the business grows.
6. **Security:**
    - Implementing appropriate access controls and permissions to protect sensitive employee data.

Flow Chart

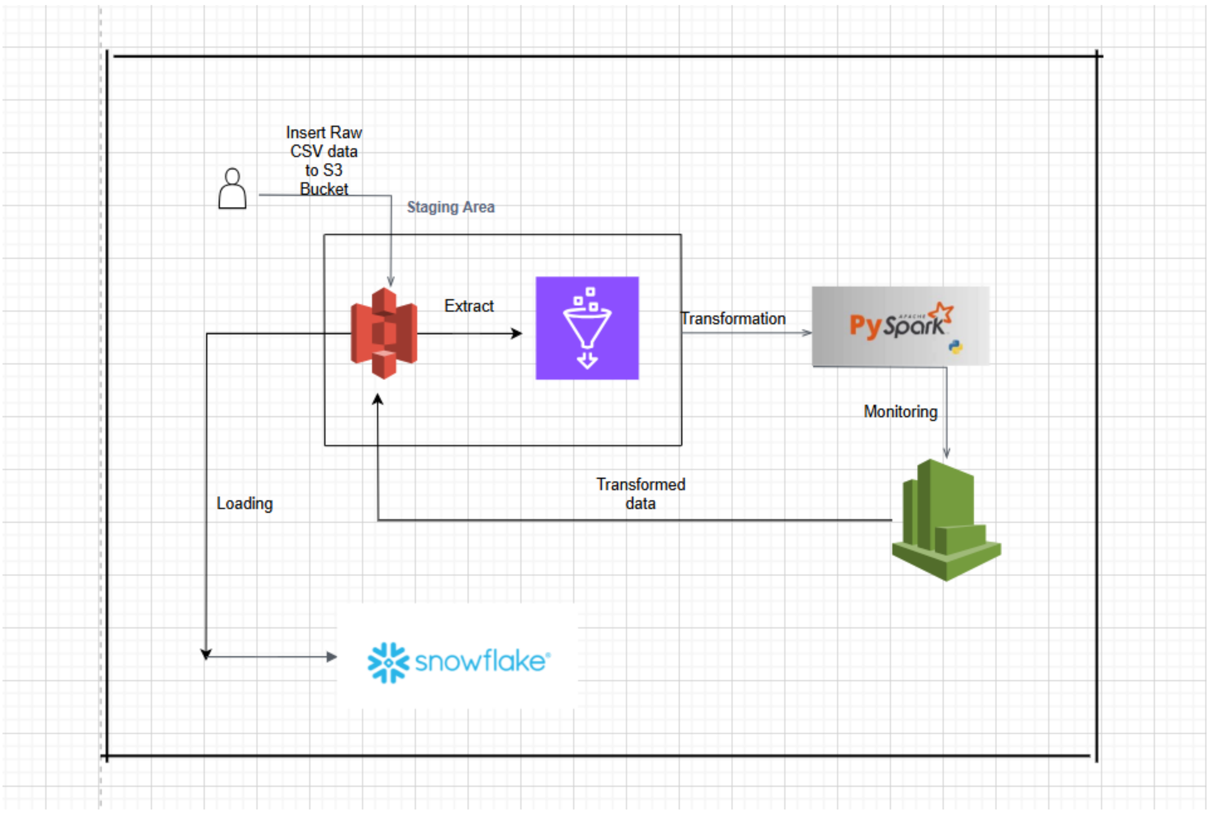


Fig 1. Flow Chart

Chapter 1: Data Extraction

1.1 Definition

Data extraction is the initial step in the data processing pipeline. In this phase, an employee CSV file is read from a local directory and uploaded to AWS S3 for raw data storage. This process ensures that the raw employee data is securely stored in a centralized location, making it readily available for subsequent processing and analysis.

.

1.2 Types of Data Extraction

Manual Extraction: This method involves manually uploading files to the storage system. While it is straightforward, it is labour-intensive and prone to human error, making it less suitable for handling large volumes of data.

Automated Extraction: This method uses scripts or tools to automate the data upload process. It is efficient, reduces the risk of errors, and can handle large volumes of data without manual intervention, making it ideal for modern data processing needs.

1.3 Categories of Data Extraction

Batch Extraction: This approach involves extracting data in batches at scheduled intervals. It is suitable for periodic data uploads, such as daily or weekly sales data, and helps in managing data loads effectively.

Real-Time Extraction: This approach continuously extracts data as it becomes available. It is ideal for systems that require immediate data availability, such as real-time analytics and monitoring systems.

1.4 Advantages

Efficiency: Automating the data upload process saves significant time and effort, allowing the system to handle large volumes of data seamlessly.

Scalability: Automated extraction can scale to accommodate increasing data volumes without requiring additional manual effort.

Reliability: By reducing the risk of human error, automated extraction ensures that data is consistently and accurately uploaded to the storage system.

1.5 Disadvantages

Initial Setup: Setting up automated data extraction requires initial configuration of scripts and tools, which can be time-consuming and require technical expertise.

Maintenance: Regular maintenance is needed to ensure the smooth operation of the extraction process, including updates and troubleshooting.

Dependency: The extraction process relies on the availability of the source data and network connectivity, which can be a limitation in some scenarios.

1.6 Examples

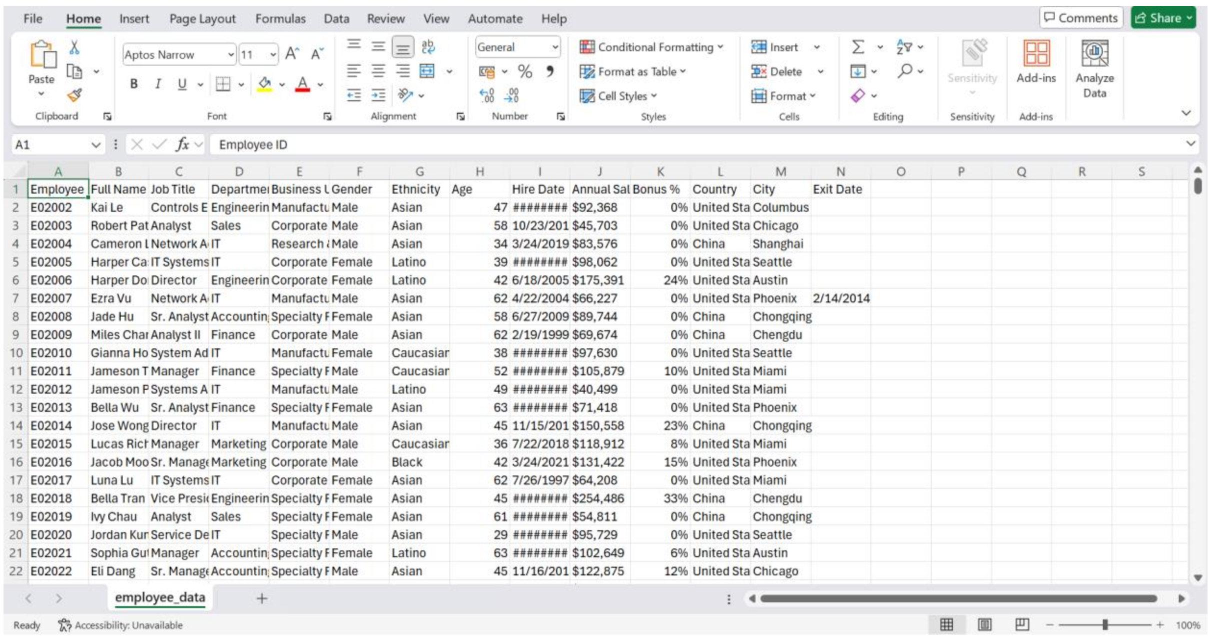
Employee Records Management: A company uploads daily updates of employee records, including new hires, terminations, and changes in employment status, to AWS S3. This ensures that all employee information is centrally stored and accessible for further processing and compliance.

Employee Activity Logs: A company extracts activity logs from its internal systems, such as login times, application usage, and project contributions, and stores them in a centralized location for monitoring and analysis. This helps in identifying productivity trends and troubleshooting issues promptly.

1.7 Steps Involved

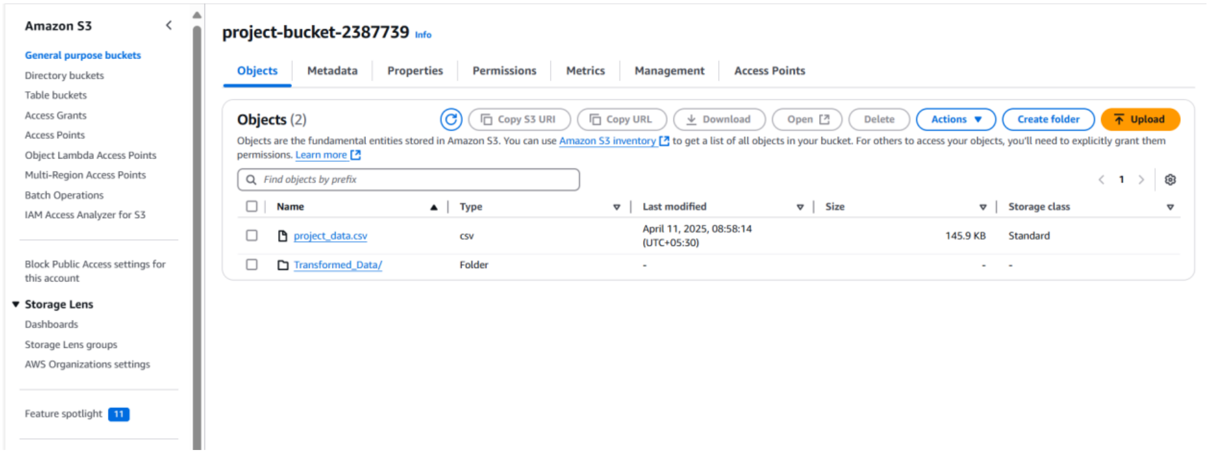
Create S3 Bucket: Set up an S3 bucket to store raw sales data, ensuring that the bucket is properly configured for data storage and access.

Generate Policies: Define access policies for the S3 bucket to control who can read and write data, ensuring data security and compliance.



	A	B	C	D	E	F	G	H	I	J	K	L	M	N
1	Employee	Full Name	Job Title	Departme	Business U	Gender	Ethnicity	Age	Hire Date	Annual Sal	Bonus %	Country	City	Exit Date
2	E02002	Kai Le	Controls E	Engineerin	Manufactu	Male	Asian	47	########	$92,368	0%	United Sta	Columbus	
3	E02003	Robert Pat	Analyst	Sales	Corporate	Male	Asian	58	10/23/201	$45,703	0%	United Sta	Chicago	
4	E02004	Cameron L	Network A	IT	Research &	Male	Asian	34	3/24/2019	$83,576	0%	China	Shanghai	
5	E02005	Harper Ca	IT Systems	IT	Corporate	Female	Latino	39	########	$98,062	0%	United Sta	Seattle	
6	E02006	Harper Do	Director	Engineerin	Corporate	Female	Latino	42	6/18/2005	$175,391	24%	United Sta	Austin	
7	E02007	Ezra Vu	Network A	IT	Manufactu	Male	Asian	62	4/22/2004	$66,227	0%	United Sta	Phoenix	2/14/2014
8	E02008	Jade Hu	Sr. Analyst	Accountin	Specialty F	Female	Asian	58	6/27/2009	$89,744	0%	China	Chongqing	
9	E02009	Miles Char	Analyst II	Finance	Corporate	Male	Asian	62	2/19/1999	$69,674	0%	China	Chengdu	
10	E02010	Gianna Ho	System Ad	IT	Manufactu	Female	Caucasiar	38	########	$97,630	0%	United Sta	Seattle	
11	E02011	Jameson T	Manager	Finance	Specialty F	Male	Caucasiar	52	########	$105,879	10%	United Sta	Miami	
12	E02012	Jameson P	Systems A	IT	Manufactu	Male	Latino	49	########	$40,499	0%	United Sta	Miami	
13	E02013	Bella Wu	Sr. Analyst	Finance	Specialty F	Female	Asian	63	########	$71,418	0%	United Sta	Phoenix	
14	E02014	Jose Wong	Director	IT	Manufactu	Male	Asian	45	11/15/201	$150,558	23%	China	Chongqing	
15	E02015	Lucas Rich	Manager	Marketing	Corporate	Male	Caucasiar	36	7/22/2018	$118,912	8%	United Sta	Miami	
16	E02016	Jacob Moo	Sr. Manag	Marketing	Corporate	Male	Black	42	3/24/2021	$131,422	15%	United Sta	Phoenix	
17	E02017	Luna Lu	IT Systems	IT	Corporate	Female	Asian	62	7/26/1997	$64,208	0%	United Sta	Miami	
18	E02018	Bella Tran	Vice Presi	Engineerin	Specialty F	Female	Asian	45	########	$254,486	33%	China	Chengdu	
19	E02019	Ivy Chau	Analyst	Sales	Specialty F	Female	Asian	61	########	$54,811	0%	China	Chongqing	
20	E02020	Jordan Kur	Service De	IT	Specialty F	Male	Asian	29	########	$95,729	0%	United Sta	Seattle	
21	E02021	Sophia Gu	Manager	Accountin	Specialty F	Female	Latino	63	########	$102,649	6%	United Sta	Austin	
22	E02022	Eli Dang	Sr. Manag	Accountin	Specialty F	Male	Asian	45	11/16/201	$122,875	12%	United Sta	Chicago	

Fig 1. Raw Employee Data CSV File



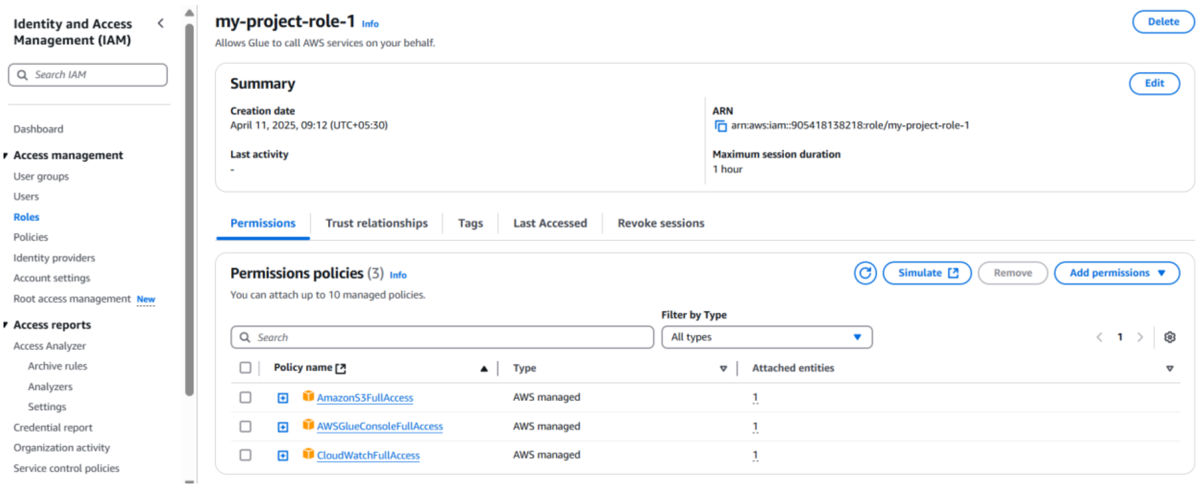
project-bucket-2387739

Objects (2)

Objects are the fundamental entities stored in Amazon S3. You can use Amazon S3 inventory to get a list of all objects in your bucket. For others to access your objects, you'll need to explicitly grant them permissions. Learn more

Name	Type	Last modified	Size	Storage class
project_data.csv	csv	April 11, 2025, 08:58:14 (UTC+05:30)	145.9 KB	Standard
Transformed_Data/	Folder	-	-	-

Fig 2. Creating S3 bucket



my-project-role-1 Info

Allows Glue to call AWS services on your behalf.

Summary

Creation date: April 11, 2025, 09:12 (UTC+05:30)

ARN: arn:aws:iam::905418138218:role/my-project-role-1

Last activity: -

Maximum session duration: 1 hour

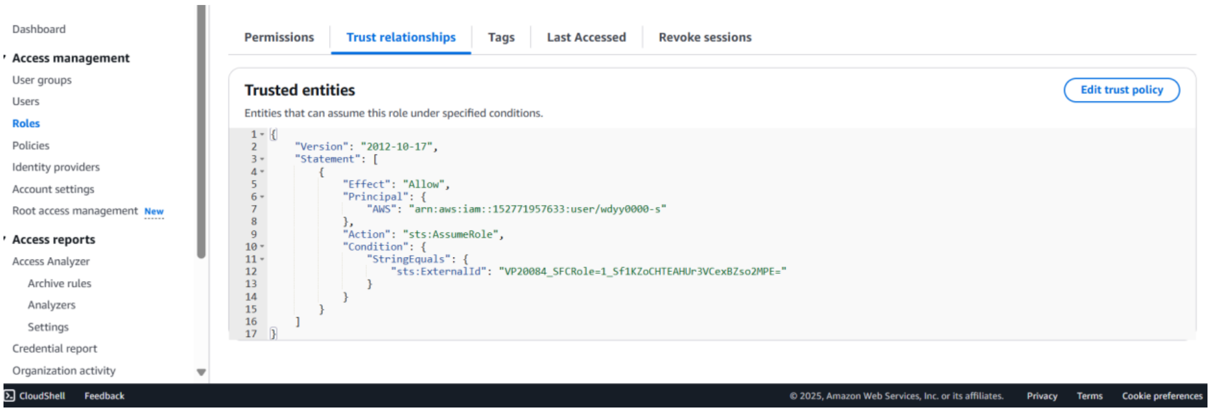
Permissions policies (3) Info

You can attach up to 10 managed policies.

Filter by Type: All types

Policy name	Type	Attached entities
AmazonS3FullAccess	AWS managed	1
AWSGlueConsoleFullAccess	AWS managed	1
CloudWatchFullAccess	AWS managed	1

Fig 3. Creating IAM role for Glue

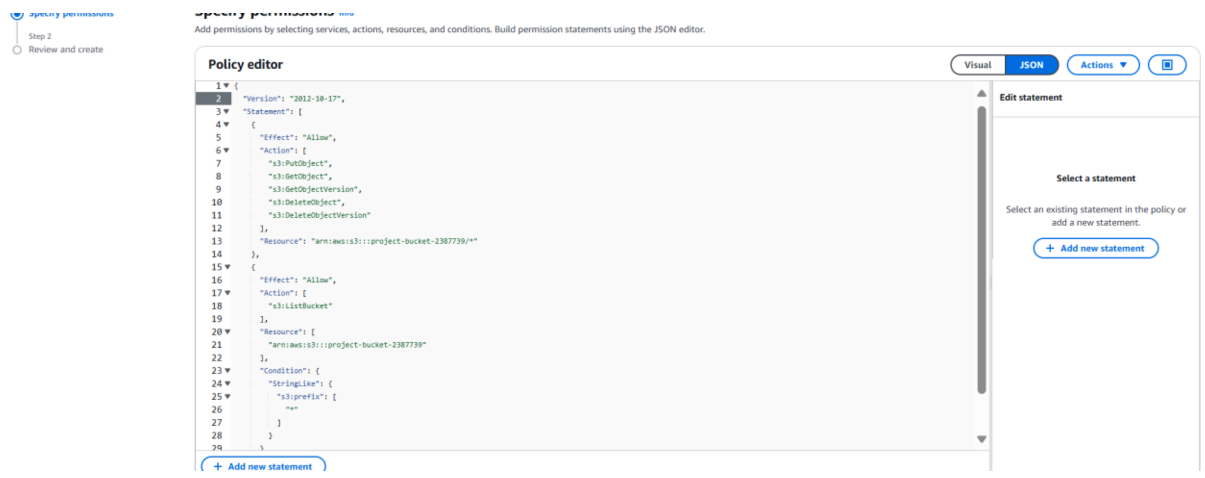


Trusted entities

Entities that can assume this role under specified conditions.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Principal": {
                "AWS": "arn:aws:iam::152771957633:user/wdyy0000-s"
            },
            "Action": "sts:AssumeRole",
            "Condition": {
                "StringEquals": {
                    "sts:ExternalId": "VP20084_SFCRole=1_Sf1KZoCHTEAHUr3VCexBZso2MPE="
                }
            }
        }
    ]
}
```

Fig 4. Created IAM role for Snowflake



Specify permissions Info

Add permissions by selecting services, actions, resources, and conditions. Build permission statements using the JSON editor.

Policy editor

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "s3:PutObject",
        "s3:GetObject",
        "s3:GetObjectVersion",
        "s3:DeleteObject",
        "s3:DeleteObjectVersion"
      ],
      "Resource": "arn:aws:s3:::project-bucket-2387739/*"
    },
    {
      "Effect": "Allow",
      "Action": [
        "s3:ListBucket"
      ],
      "Resource": [
        "arn:aws:s3:::project-bucket-2387739"
      ],
      "Condition": {
        "StringLike": {
          "s3:prefix": [
            "*"
          ]
        }
      }
```

Fig 5.Creating Policy for Snowflake named policyforgluee

Chapter 2: Data Transformation

2.1 Definition

Data transformation is a critical step in the data processing pipeline, where raw sales data is validated and aggregated in AWS Glue using PySpark. This process ensures data quality by removing or correcting invalid data and generating processed datasets that are ready for analysis.

2.2 Types of Data Transformation

- **Data Cleaning:** This involves removing or correcting invalid data to ensure accuracy. It includes tasks such as handling missing values, correcting data types, and removing duplicates.
- **Data Aggregation:** This involves summarizing data to provide insights, such as calculating total sales per store or product. Aggregation helps in reducing data complexity and making it more manageable for analysis.
- **Data Enrichment:** This involves adding additional information to the data, such as product categories or customer demographics, to enhance its value and provide deeper insights.

2.3 Categories of Data Transformation

- **Synchronous Transformation:** This approach processes data in real-time as it is received, ensuring that the data is immediately available for analysis. It is suitable for applications that require up-to-date information.
- **Asynchronous Transformation:** This approach processes data in batches at scheduled intervals, allowing for more efficient use of resources. It is suitable for applications that can tolerate some delay in data availability.

2.4 Advantages

- **Data Quality:** By validating and cleaning the data, transformation ensures that the data is accurate and consistent, which is essential for reliable analysis.
- **Insights:** Aggregating data provides valuable insights by summarizing complex data into meaningful metrics, such as total sales or average order value.
- **Automation:** Automating data transformation tasks reduces manual effort and ensures that the data is processed consistently and efficiently.

2.5 Disadvantages

- **Complexity:** Setting up and maintaining data transformation processes requires technical expertise and can be complex, especially for large datasets.

- **Resource Intensive:** Data transformation can consume significant computational resources, especially for large datasets or complex transformations.
- **Latency:** Depending on the approach used, data transformation may introduce delays in data availability, which can be a limitation for real-time applications.

2.6 Examples

- **Employee Data Aggregation:** Summarizing employee data by department and role to calculate total headcount and identify top-performing employees.

- **Data Validation:** Checking for missing values in employee records and correcting data types to ensure that the data is accurate and consistent.

2.7 Steps Involved

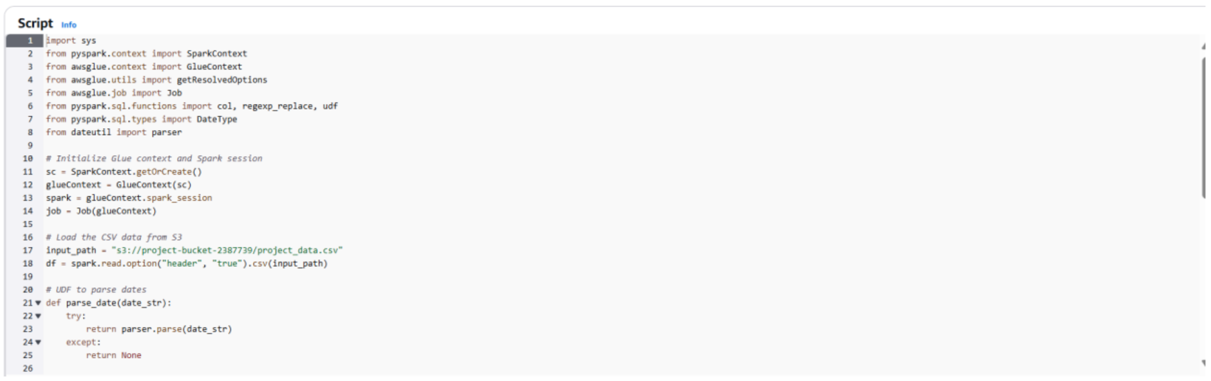
1. **Create ETL Job:**
    - Configure the job name and select the appropriate Glue version (e.g., Glue 3.0) to match the specific requirements.
    - Adjust timeout settings to avoid errors, ensuring that the job has enough time to process the data.

2. **Set Permissions:**
    - Attach policies such as AWSGlueConsoleFullAccess, AmazonS3FullAccess, CloudWatchFullAccess and to grant the necessary permissions for data processing.

3. **Add Trigger:**
    - Choose the S3 bucket as the trigger source and acknowledge the configuration to ensure that the Glue job is triggered when new data is uploaded.

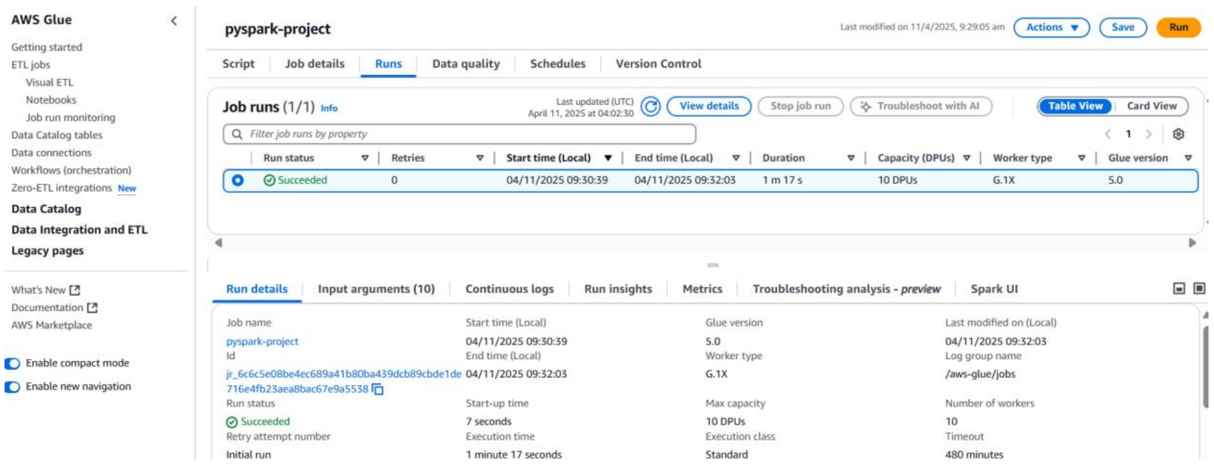
4. **Add Libraries:**
    - Ensure that the Glue job has access to these libraries by specifying them in the job configuration.

5. **PySpark Script:**
    - Edit the script to specify the bucket name, folder, and default date.
    - Generate cleaned employee CSV and ensuring that the data is validated and aggregated correctly.
    - Deploy and test the job to ensure that it processes the data as expected.



```python
import sys
from pyspark.context import SparkContext
from awsglue.context import GlueContext
from awsglue.utils import getResolvedOptions
from awsglue.job import Job
from pyspark.sql.functions import col, regexp_replace, udf
from pyspark.sql.types import DateType
from dateutil import parser

# Initialize Glue context and Spark session
sc = SparkContext.getOrCreate()
glueContext = GlueContext(sc)
spark = glueContext.spark_session
job = Job(glueContext)

# Load the CSV data from S3
input_path = "s3://project-bucket-2387739/project_data.csv"
df = spark.read.option("header", "true").csv(input_path)

# UDF to parse dates
def parse_date(date_str):
    try:
        return parser.parse(date_str)
    except:
        return None

```

Fig 6. Spark code for data transformation



pyspark-project

Last modified on 11/4/2025, 9:29:05 am

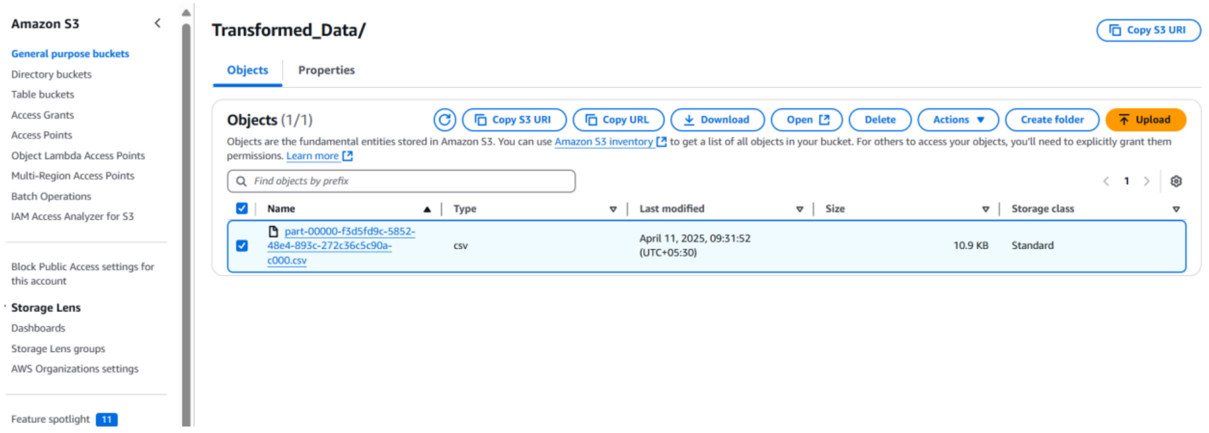
Job runs (1/1) Info

Last updated (UTC) April 11, 2025 at 04:02:30

Run status	Retries	Start time (Local)	End time (Local)	Duration	Capacity (DPUs)	Worker type	Glue version
Succeeded	0	04/11/2025 09:30:39	04/11/2025 09:32:03	1 m 17 s	10 DPUs	G.1X	5.0

Job name	Start time (Local)	Glue version	Last modified on (Local)
pyspark-project	04/11/2025 09:30:39	5.0	04/11/2025 09:32:03
Id	End time (Local)	Worker type	Log group name
jr_6c6c5e08be4ec689a41b80ba439dcb89cbde1de716e4fb23aea8bac67e9a5538	04/11/2025 09:32:03	G.1X	/aws-glue/jobs
Run status	Start-up time	Max capacity	Number of workers
Succeeded	7 seconds	10 DPUs	10
Retry attempt number	Execution time	Execution class	Timeout
Initial run	1 minute 17 seconds	Standard	480 minutes

Fig 7. Run the job



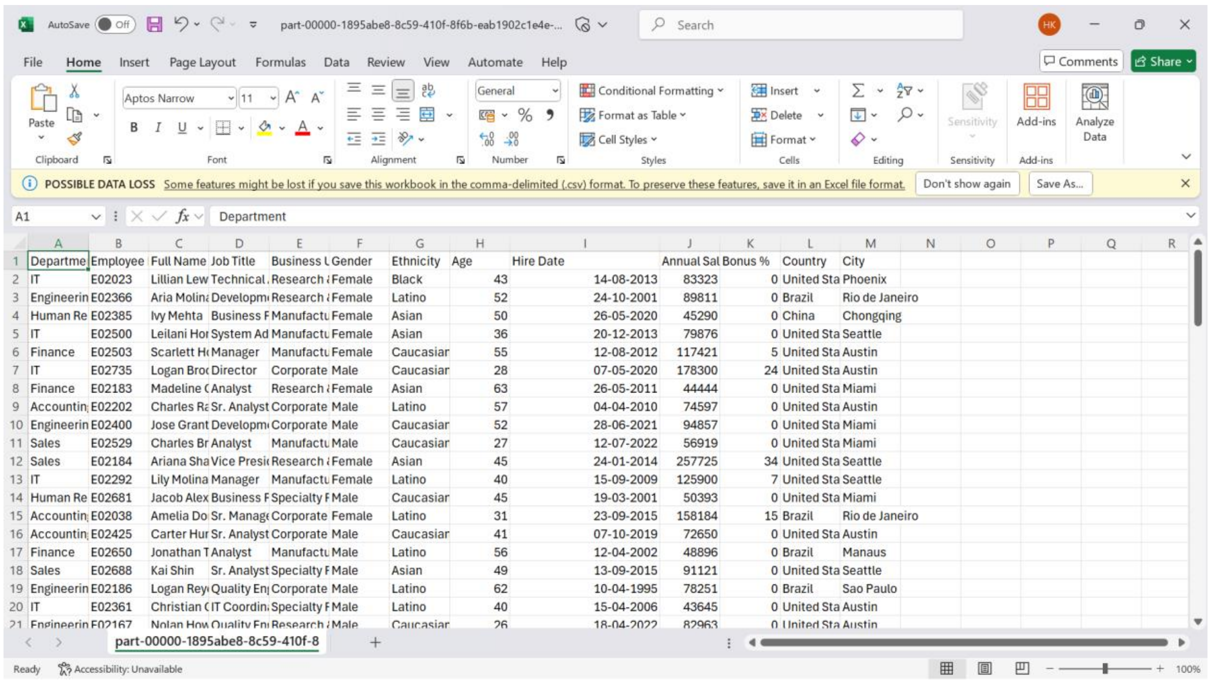
Transformed_Data/

Objects (1/1)

Objects are the fundamental entities stored in Amazon S3. You can use Amazon S3 inventory to get a list of all objects in your bucket. For others to access your objects, you'll need to explicitly grant them permissions. Learn more

Name	Type	Last modified	Size	Storage class
part-00000-f3d5fd9c-5852-48e4-893c-272c36c5c90a-c000.csv	csv	April 11, 2025, 09:31:52 (UTC+05:30)	10.9 KB	Standard

Fig 8. Checking for transformed CSV



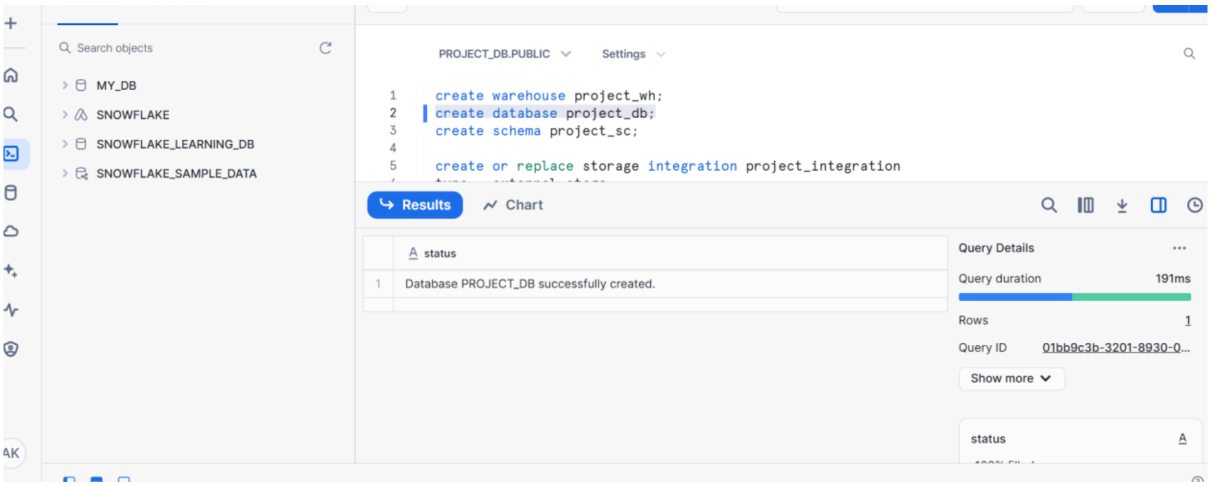
POSSIBLE DATA LOSS Some features might be lost if you save this workbook in the comma-delimited (.csv) format. To preserve these features, save it in an Excel file format.

A1 Department

	A	B	C	D	E	F	G	H	I	J	K	L	M
1	Departme	Employee	Full Name	Job Title	Business U	Gender	Ethnicity	Age	Hire Date	Annual Sal	Bonus %	Country	City
2	IT	E02023	Lillian Lew	Technical	Research	Female	Black	43	14-08-2013	83323	0	United Sta	Phoenix
3	Engineerin	E02366	Aria Molin	Developm	Research	Female	Latino	52	24-10-2001	89811	0	Brazil	Rio de Janeiro
4	Human Re	E02385	Ivy Mehta	Business F	Manufactu	Female	Asian	50	26-05-2020	45290	0	China	Chongqing
5	IT	E02500	Leilani Hor	System Ad	Manufactu	Female	Asian	36	20-12-2013	79876	0	United Sta	Seattle
6	Finance	E02503	Scarlett H	Manager	Manufactu	Female	Caucasiar	55	12-08-2012	117421	5	United Sta	Austin
7	IT	E02735	Logan Broc	Director	Corporate	Male	Caucasiar	28	07-05-2020	178300	24	United Sta	Austin
8	Finance	E02183	Madeline (	Analyst	Research	Female	Asian	63	26-05-2011	44444	0	United Sta	Miami
9	Accountin	E02202	Charles Ra	Sr. Analyst	Corporate	Male	Latino	57	04-04-2010	74597	0	United Sta	Austin
10	Engineerin	E02400	Jose Grant	Developm	Corporate	Male	Caucasiar	52	28-06-2021	94857	0	United Sta	Miami
11	Sales	E02529	Charles Br	Analyst	Manufactu	Male	Caucasiar	27	12-07-2022	56919	0	United Sta	Miami
12	Sales	E02184	Ariana Sha	Vice Presi	Research	Female	Asian	45	24-01-2014	257725	34	United Sta	Seattle
13	IT	E02292	Lily Molina	Manager	Manufactu	Female	Latino	40	15-09-2009	125900	7	United Sta	Seattle
14	Human Re	E02681	Jacob Alex	Business F	Specialty F	Male	Caucasiar	45	19-03-2001	50393	0	United Sta	Miami
15	Accountin	E02038	Amelia Do	Sr. Manag	Corporate	Female	Latino	31	23-09-2015	158184	15	Brazil	Rio de Janeiro
16	Accountin	E02425	Carter Hur	Sr. Analyst	Corporate	Male	Caucasiar	41	07-10-2019	72650	0	United Sta	Austin
17	Finance	E02650	Jonathan T	Analyst	Manufactu	Male	Latino	56	12-04-2002	48896	0	Brazil	Manaus
18	Sales	E02688	Kai Shin	Sr. Analyst	Specialty F	Male	Asian	49	13-09-2015	91121	0	United Sta	Seattle
19	Engineerin	E02186	Logan Rey	Quality En	Corporate	Male	Latino	62	10-04-1995	78251	0	Brazil	Sao Paulo
20	IT	E02361	Christian (	IT Coordin	Specialty F	Male	Latino	40	15-04-2006	43645	0	United Sta	Austin
21	Engineerin	E02167	Nolan How	Quality En	Research	Male	Caucasiar	26	18-04-2022	82963	0	United Sta	Austin

part-00000-1895abe8-8c59-410f-8

Fig 9. Transformed Data Preview



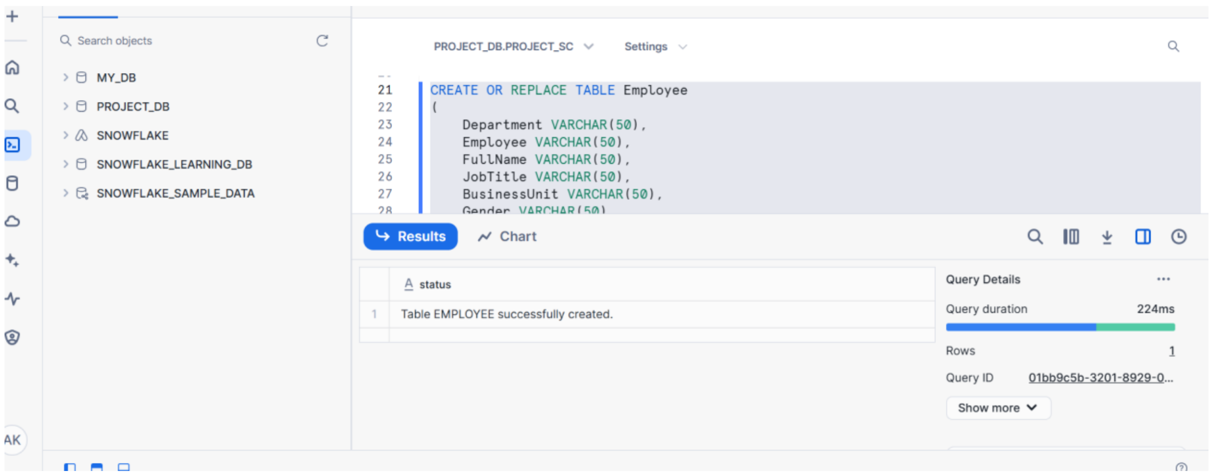
PROJECT_DB.PUBLIC Settings

```
create warehouse project_wh;
create database project_db;
create schema project_sc;

create or replace storage integration project_integration
```

	status
1	Database PROJECT_DB successfully created.

Query Details
Query duration 191ms
Rows 1
Query ID 01bb9c3b-3201-8930-0...

Fig 10. Setting up roles, warehouses and databases



```
CREATE OR REPLACE TABLE Employee
(
    Department VARCHAR(50),
    Employee VARCHAR(50),
    FullName VARCHAR(50),
    JobTitle VARCHAR(50),
    BusinessUnit VARCHAR(50),
    Gender VARCHAR(50)
```

	status
1	Table EMPLOYEE successfully created.

Fig 11. Creating employee table in Snowflake

Chapter 3: Data Loading

3.1 Definition

Data loading is the process of loading the transformed data into Snowflake for analytics and reporting. This step ensures that the data is available in a structured format, ready for querying and analysis.

3.2 Types of Data Loading

- **Full Load:** This method involves loading the entire dataset into the target system. It is suitable for initial data loads or when a complete refresh of the data is required.
- **Incremental Load:** This method involves loading only the new or updated data. It is ideal for ongoing data updates, as it minimizes the amount of data transferred and reduces the load on the system.

3.3 Categories of Data Loading

- **Batch Loading:** This approach involves loading data in batches at scheduled intervals. It is suitable for periodic data updates, such as daily or weekly loads, and helps in managing data loads effectively.
- **Real-Time Loading:** This approach continuously loads data as it becomes available. It is ideal for systems that require immediate data availability, such as real-time analytics and monitoring systems.

3.4 Advantages

- **Performance:** Optimizing data loading processes ensures faster data processing and reduces the time required to make the data available for analysis.
- **Scalability:** Data loading processes can scale to handle large volumes of data efficiently, making them suitable for growing datasets.
- **Flexibility:** Various data loading methods can be used to meet different requirements, such as full loads for initial data setup and incremental loads for ongoing updates.

3.5 Disadvantages

- **Complexity:** Setting up and configuring data loading processes requires careful planning and technical expertise, especially for large datasets.
- **Resource Intensive:** Data loading can consume significant computational resources, particularly for large datasets or complex transformations.
- **Dependency:** The loading process relies on the availability of the target system, which can be a limitation if the system is unavailable or experiencing issues.

3.6 Examples

- **Employee Data Loading:** Loading processed employee data into AWS Redshift to enable querying and analysis of employee performance and other HR metrics.

- **Activity Log Data Loading:** Loading employee activity log data into a centralized database for monitoring and analysis, helping to identify productivity trends and troubleshoot issues promptly.

3.7 Steps Involved

1. **Set Up Warehouse and Database:**
    - Create a warehouse to provide the required resources (CPU, memory, and temporary storage) for DML operations:
    - Create a database and schema.
2. **Create IAM Role:**
    - Create an IAM role of AWS Account and attach an S3 inline policy for Snowflake to access the necessary permissions for data loading.
    - Edit the role permissions to ensure that the role has the appropriate access to the required resources.
3. **Create storage integration:**

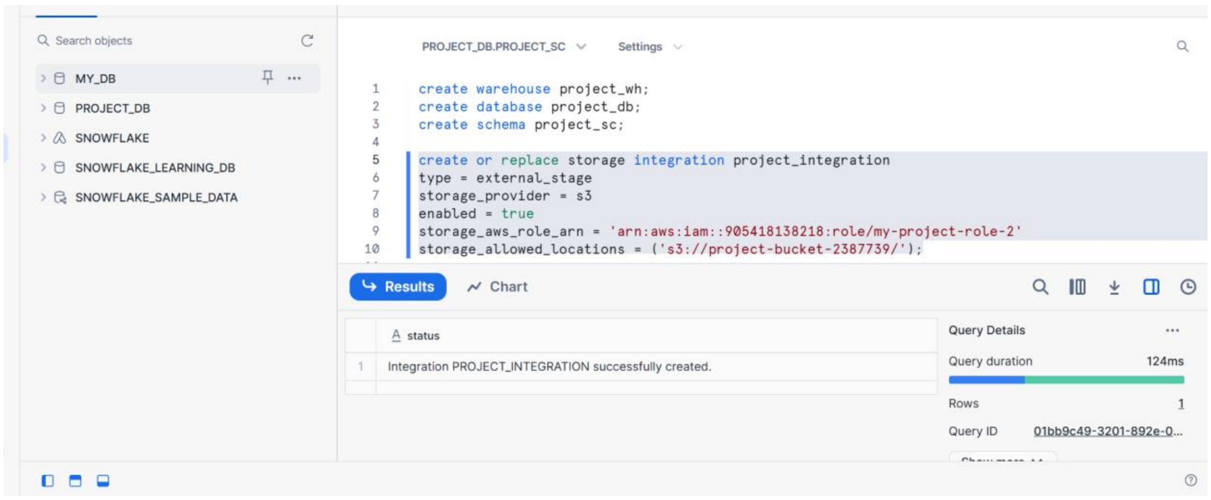
    - Describe the storage integration after its creation using AWS ARN and S3 bucket path.
    - Use the describe command to get the Amazon Resource Name (ARN).

4. **Create a file format for CSV:**

    - Create a file format for CSV by properly mentioning the header and null options.

5. **Create Stage and Target Table:**

    - Create a stage and target table specifying where the files are stored.
    - Load data into that table using COPY command.



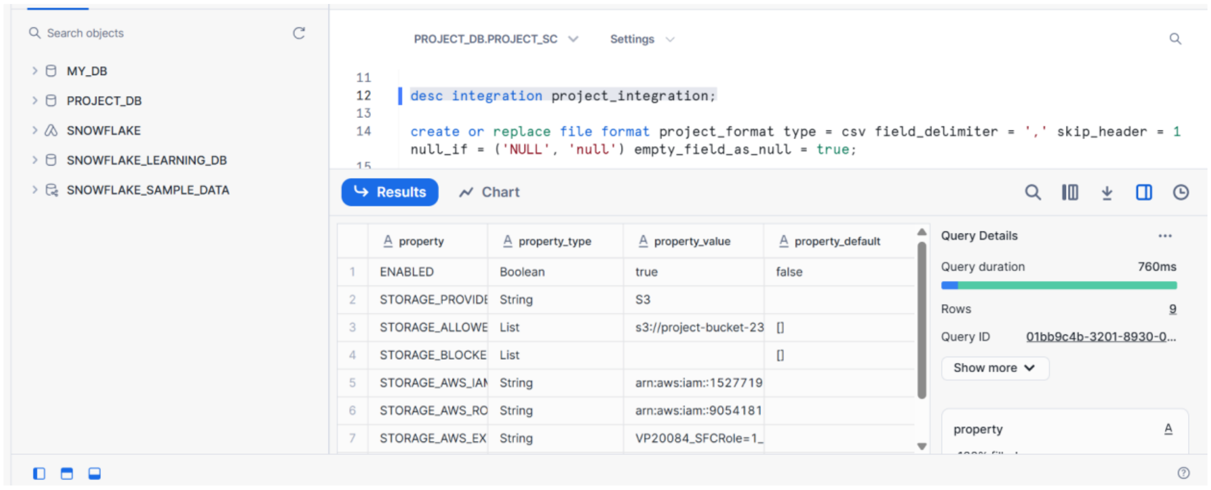
```
create warehouse project_wh;
create database project_db;
create schema project_sc;

create or replace storage integration project_integration
type = external_stage
storage_provider = s3
enabled = true
storage_aws_role_arn = 'arn:aws:iam::905418138218:role/my-project-role-2'
storage_allowed_locations = ('s3://project-bucket-2387739/');
```

	status
1	Integration PROJECT_INTEGRATION successfully created.

Query Details

Query duration	124ms
Rows	1
Query ID	01bb9c49-3201-892e-0...

Fig 12. Integrating S3 and Snowflake



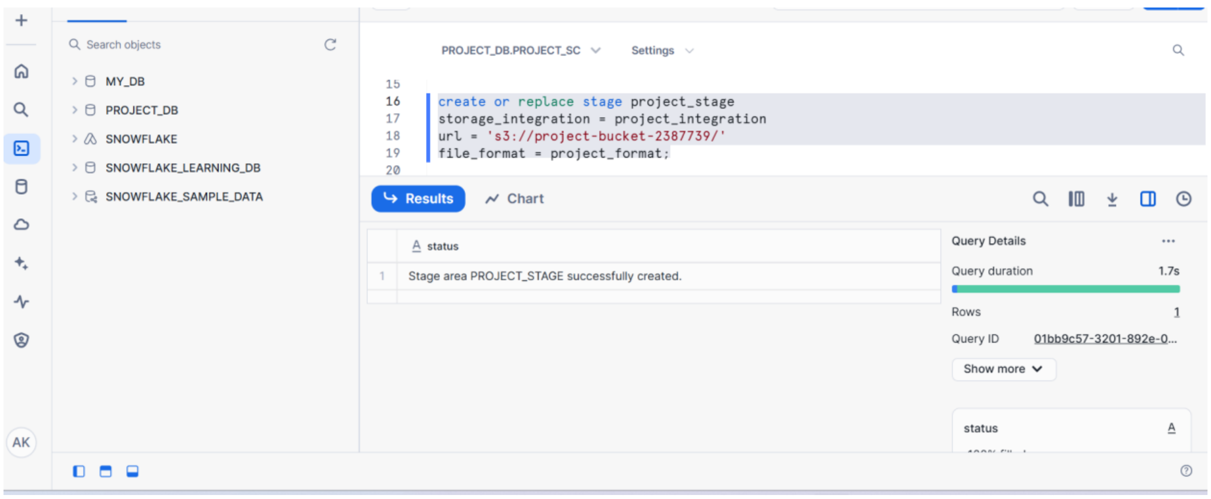
```
desc integration project_integration;

create or replace file format project_format type = csv field_delimiter = ',' skip_header = 1
null_if = ('NULL', 'null') empty_field_as_null = true;
```

	property	property_type	property_value	property_default
1	ENABLED	Boolean	true	false
2	STORAGE_PROVIDE	String	S3	
3	STORAGE_ALLOWE	List	s3://project-bucket-23	[]
4	STORAGE_BLOCKE	List		[]
5	STORAGE_AWS_IAI	String	arn:aws:iam::1527719	
6	STORAGE_AWS_RO	String	arn:aws:iam::9054181	
7	STORAGE_AWS_EX	String	VP20084_SFCRole=1_	

Query Details

Query duration	760ms
Rows	9
Query ID	01bb9c4b-3201-8930-0...

Fig 13. Editing IAM arn and external ID



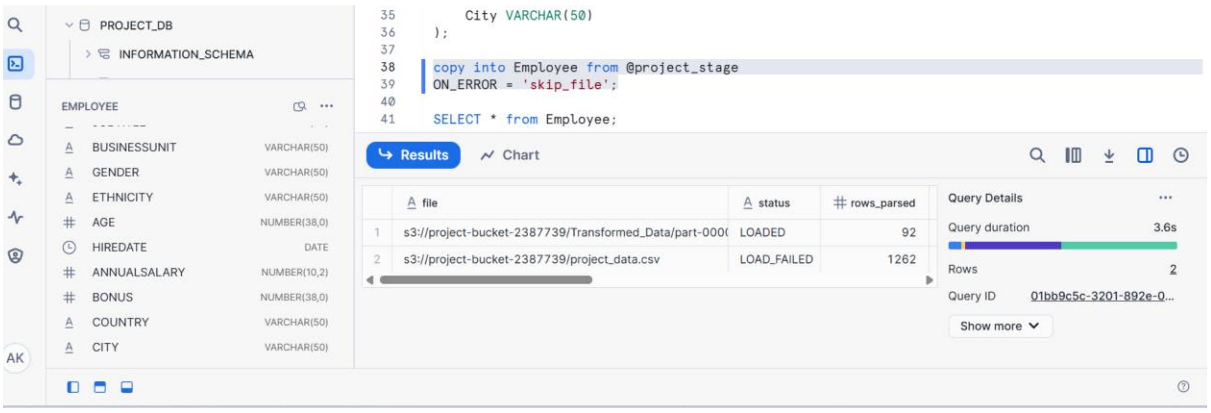
PROJECT_DB.PROJECT_SC   Settings

```
create or replace stage project_stage
storage_integration = project_integration
url = 's3://project-bucket-2387739/'
file_format = project_format;
```

	status
1	Stage area PROJECT_STAGE successfully created.

Query Details
Query duration 1.7s
Rows 1
Query ID 01bb9c57-3201-892e-0...

Fig 14. Accessing the external storage



EMPLOYEE

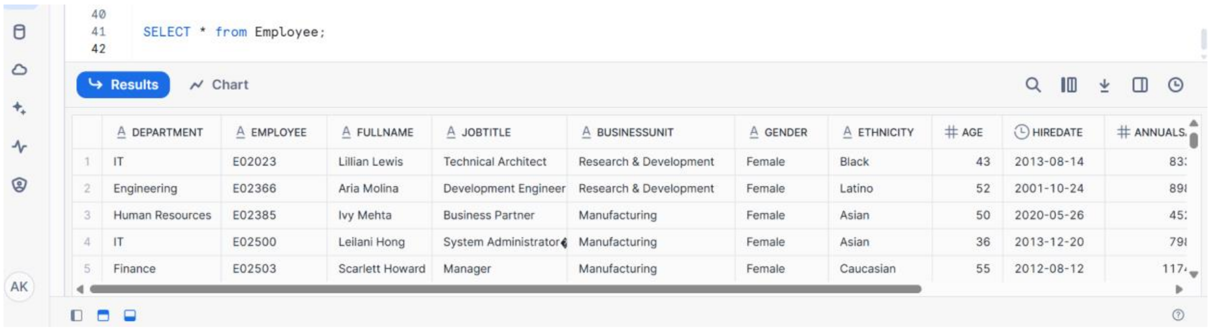
BUSINESSUNIT	VARCHAR(50)
GENDER	VARCHAR(50)
ETHNICITY	VARCHAR(50)
AGE	NUMBER(38,0)
HIREDATE	DATE
ANNUALSALARY	NUMBER(10,2)
BONUS	NUMBER(38,0)
COUNTRY	VARCHAR(50)
CITY	VARCHAR(50)

```
    City VARCHAR(50)
);

copy into Employee from @project_stage
ON_ERROR = 'skip_file';

SELECT * from Employee;
```

	file	status	rows_parsed
1	s3://project-bucket-2387739/Transformed_Data/part-000(	LOADED	92
2	s3://project-bucket-2387739/project_data.csv	LOAD_FAILED	1262

Query Details
Query duration 3.6s
Rows 2
Query ID 01bb9c5c-3201-892e-0...

Fig 15. Load data to employee table



```
SELECT * from Employee;
```

	DEPARTMENT	EMPLOYEE	FULLNAME	JOBTITLE	BUSINESSUNIT	GENDER	ETHNICITY	AGE	HIREDATE	ANNUALS.
1	IT	E02023	Lillian Lewis	Technical Architect	Research & Development	Female	Black	43	2013-08-14	83:
2	Engineering	E02366	Aria Molina	Development Engineer	Research & Development	Female	Latino	52	2001-10-24	89!
3	Human Resources	E02385	Ivy Mehta	Business Partner	Manufacturing	Female	Asian	50	2020-05-26	45:
4	IT	E02500	Leilani Hong	System Administrator	Manufacturing	Female	Asian	36	2013-12-20	79!
5	Finance	E02503	Scarlett Howard	Manager	Manufacturing	Female	Caucasian	55	2012-08-12	117.

Fig 16. Table in snowflake

Chapter 4: Real Time Data Storage

Real-time data storage is a critical component of modern data processing systems, enabling organizations to capture, store, and analyse data as it is generated. This capability is essential for applications that require immediate insights and rapid response to changing conditions. AWS offers a suite of services that facilitate real-time data storage, ensuring scalability, low latency, and robust data management. In this section, we will explore the concept of real-time data storage, its benefits, challenges, and how AWS services can be leveraged for effective implementation.

4.1 Definition

Real-time data storage refers to the continuous and immediate capture and storage of data as it is generated. Unlike batch processing, which involves collecting and processing data at scheduled intervals, real-time data storage ensures that data is available for analysis and decision-making almost instantaneously.

4.2 Benefits

1. **Immediate Insights**: Real-time data storage allows organizations to gain immediate insights into their operations. This capability is crucial for applications such as monitoring, fraud detection, and real-time analytics, where timely information is essential for effective decision-making.
2. **Enhanced Responsiveness**: By capturing and storing data in real-time, organizations can respond more quickly to changing conditions. For example, in retail, real-time data storage enables dynamic pricing and inventory management, helping businesses stay competitive and meet customer demands.
3. **Improved Customer Experience**: Real-time data storage supports personalized and timely interactions with customers. For instance, e-commerce platforms can use real-time data to recommend products based on current browsing behaviour, enhancing the customer experience and driving sales.
4. **Operational Efficiency**: Real-time data storage streamlines operations by providing up-to-date information. This capability is valuable in industries such as manufacturing, where real-time data can be used to monitor equipment performance and prevent downtime.

4.3 Challenges

1. **Scalability**: Real-time data storage systems must be able to handle large volumes of data generated at high velocity. Ensuring scalability requires robust infrastructure and efficient data processing mechanisms.
2. **Data Quality**: Maintaining data quality in real-time environments can be challenging. Organizations must implement validation and cleansing processes to ensure that the data being stored is accurate and reliable.
3. **Latency**: Minimizing latency is critical for real-time data storage. Any delays in data capture, processing, or storage can impact the timeliness of insights and decision-making.
4. **Cost**: Real-time data storage can be resource-intensive, requiring significant investment in infrastructure and technology. Organizations must balance the benefits of real-time capabilities with the associated costs.

4.4 AWS Services for Real-Time Data Storage

1. **AWS Lambda**: AWS Lambda allows you to run code in response to events, enabling real-time data processing without provisioning or managing servers. It can be used to process data streams from Kinesis and trigger actions based on real-time data.
2. **Amazon S3**: Amazon S3 provides scalable object storage for real-time data. It can be used to store raw and processed data, ensuring durability and availability. S3's integration with other AWS services makes it a versatile choice for real-time data storage.
3. **AWS Glue:** AWS Glue is a serverless data integration service that simplifies the process of discovering, preparing, moving, and integrating data from multiple sources for analytics, machine learning, and application development.
4. **Snowflake**: Amazon Snowflake is a data warehousing service that supports real-time data loading and querying for large amount of data. It can be used to analyse real-time data streams and generate insights quickly.

4.5 Use Cases

1. **Retail**: Real-time data storage enables dynamic pricing, inventory management, and personalized marketing in retail. By capturing sales and customer data in real-time, retailers can optimize their operations and enhance the customer experience.
2. **Finance**: In the finance industry, real-time data storage supports fraud detection, algorithmic trading, and risk management. Financial institutions can analyse transactions as they occur, identifying suspicious activity and making informed decisions.
3. **Healthcare**: Real-time data storage is critical for patient monitoring, telemedicine, and medical research. Healthcare providers can capture and analyse patient data in real-time, improving diagnosis, treatment, and patient outcomes.
4. **Manufacturing**: Real-time data storage supports predictive maintenance, quality control, and supply chain optimization in manufacturing. By monitoring equipment performance and production processes in real-time, manufacturers can prevent downtime and improve efficiency.

Chapter 5: Analytical Data Storage

Analytical data storage is crucial for organizations that need to process and analyze large volumes of data to derive actionable insights. AWS offers a range of services that facilitate efficient and scalable analytical data storage, ensuring that data is readily available for querying and analysis. In this section, we will explore the concept of analytical data storage, its benefits, challenges, and how AWS services can be leveraged for effective implementation.

5.1 Definition

Analytical data storage refers to the storage of processed data in a structured format that can be queried using analytical tools. This type of storage is optimized for read-heavy operations and supports complex queries, making it ideal for business intelligence, reporting, and data analysis.

5.2 Benefits

1. **Enhanced Query Performance**: Analytical data storage systems are designed to handle complex queries efficiently. By organizing data in a way that optimizes query performance, these systems enable faster data retrieval and analysis.
2. **Scalability**: Analytical data storage solutions can scale to accommodate growing data volumes. This scalability ensures that organizations can continue to analyse data effectively as their data needs expand.
3. **Data Integration**: Analytical data storage systems often support the integration of data from multiple sources. This capability allows organizations to consolidate data from various systems, providing a comprehensive view of their operations.
4. **Advanced Analytics**: By storing data in a structured format, analytical data storage systems enable advanced analytics, such as machine learning and predictive modelling. These capabilities help organizations uncover deeper insights and make data-driven decisions.

5.3 Challenges

1. **Data Volume**: Managing large volumes of data can be challenging. Analytical data storage systems must be able to handle high data throughput and ensure data integrity.
2. **Data Quality**: Ensuring data quality is critical for accurate analysis. Organizations must implement data validation and cleansing processes to maintain the quality of their analytical data.
3. **Cost**: Analytical data storage solutions can be resource-intensive, leading to higher costs. Organizations must balance the benefits of advanced analytics with the associated expenses.

4. **Latency**: Minimizing latency is important for timely data analysis. Analytical data storage systems must be optimized to reduce delays in data processing and querying.

5.4 AWS Services for Analytical Data Storage

1. **Amazon Redshift**: Amazon Redshift is a fully managed data warehousing service that enables fast querying and analysis of large datasets. It supports complex queries and integrates with various data sources, making it ideal for analytical data storage.
2. **Amazon S3**: Amazon S3 provides scalable object storage for analytical data. It can store raw and processed data, ensuring durability and availability. S3's integration with other AWS services makes it a versatile choice for analytical data storage.
3. **AWS Glue**: AWS Glue is a fully managed ETL (extract, transform, load) service that prepares data for analysis. It can integrate data from multiple sources and transform it into a format suitable for analytical storage.
4. **Amazon Athena**: Amazon Athena is an interactive query service that allows you to analyse data directly in Amazon S3 using standard SQL. It is serverless, meaning there is no infrastructure to manage, and you only pay for the queries you run.
5. **Amazon EMR**: Amazon EMR is a cloud big data platform that provides a managed Hadoop framework. It can process large amounts of data quickly and cost-effectively, making it suitable for analytical data storage and processing.

5.5 Use Cases

1. **Business Intelligence**: Analytical data storage supports business intelligence applications by providing a structured and optimized environment for querying and reporting. Organizations can generate insights from their data to inform strategic decisions.
2. **Data Warehousing**: Analytical data storage is essential for data warehousing, where large volumes of data are stored and analysed. Data warehouses enable organizations to consolidate data from various sources and perform complex queries.
3. **Machine Learning**: Analytical data storage systems provide the foundation for machine learning applications. By storing data in a structured format, these systems enable the training and deployment of machine learning models.
4. **Financial Analysis**: In the finance industry, analytical data storage supports activities such as risk assessment, fraud detection, and portfolio management. Financial institutions can analyse large datasets to identify trends and make informed decisions.

Challenges Faced and Solutions

Implementing a robust Employee Data Analysis System involves overcoming several challenges in data processing. These challenges can impact the efficiency, accuracy, and scalability of the system. Here, we explore some common challenges and their solutions, particularly in the context of using AWS services and Snowflake.

1. **Data Volume**:
   - **Challenge**: Handling large volumes of employee data generated daily from multiple departments can be overwhelming. The sheer amount of data can create bottlenecks in processing and storage.
   - **Solution**: Utilize scalable storage solutions like AWS S3 and Snowflake. AWS S3 provides virtually unlimited storage capacity, while Snowflake can handle large datasets with its columnar storage and parallel processing capabilities. Implementing data partitioning and compression techniques can further optimize storage and query performance.
2. **Data Quality**:
   - **Challenge**: Ensuring the accuracy and consistency of data is critical. Incomplete, incorrect, or inconsistent data can lead to erroneous insights and decisions.
   - **Solution**: Implement data validation and cleansing processes using AWS Glue. It will ensure high data quality before it is loaded into Snowflake. Regular audits and monitoring can also help maintain data integrity.
3. **Data Integration**:
   - **Challenge**: Integrating data from various sources and formats can be complex. Employee data may come in different formats, requiring transformation before analysis.
   - **Solution**: Use AWS Glue for ETL (extract, transform, load) processes. AWS Glue can automatically discover and catalog data. Standardizing data formats and using consistent schemas can also simplify integration.
4. **Latency**:
   - **Challenge**: Minimizing latency in data processing is essential for timely insights. Delays in data extraction, transformation, or loading can hinder real-time analysis.
   - **Solution**: Leverage AWS Kinesis for real-time data streaming and processing. AWS Kinesis Data Streams can capture and process data in real-time, reducing latency. Additionally, using AWS Glue for event-driven processing can ensure that data is processed as soon as it is available.

Conclusion

The Employee Data Analysis System represents a major leap forward in how a company can manage, process, and analyze its employee data. By harnessing the capabilities of AWS services such as S3, Glue, and Snowflake, this system offers a robust and scalable solution tailored to meet diverse data management and analytics requirements.

The process begins with data extraction, where daily employee CSV files are uploaded from a local directory to AWS S3. This ensures that raw employee data is securely stored in a centralized location, ready for subsequent processing. Automating this step allows the system to efficiently handle large volumes of data, minimizing human error and ensuring consistent and accurate data uploads.

The next phase involves data transformation, where AWS Glue with PySpark is used to validate and aggregate the raw employee data. This step is crucial for maintaining data quality by removing or correcting invalid entries and preparing datasets for analysis. Automation of these tasks reduces manual effort and ensures consistent and efficient data processing. AWS Glue's scalability and flexibility make it ideal for handling varying data volumes and complexities.

Once the data is transformed, it is loaded into Snowflake for analytics and reporting. This ensures that the data is structured and optimized for querying and analysis. By streamlining the data loading process, the system can efficiently manage large datasets, making it suitable for growing data needs. Snowflake provides a powerful and scalable data warehousing solution that supports complex queries and analytics.

Additionally, CloudWatch logs are utilized to monitor and log errors during the ETL process, ensuring timely detection and resolution of issues. This enhances the reliability and robustness of the entire data pipeline.

In summary, the Employee Data Analysis System is a comprehensive and scalable solution that addresses various data management and analytics needs for a company. Leveraging AWS services, the system ensures efficient data extraction, transformation, loading, and reporting. This robust system not only enhances operational efficiency but also provides valuable insights that support data-driven decision-making and strategic planning. As the company grows, the Employee Data Analysis System will be instrumental in managing and analyzing employee data, driving business success and growth.

THANK YOU