COLLEGE OF CREATIVE INNOVATION

Programme

Diploma in Cloud Computing and Cyber Security
(120 Credits)

Course

CCC602: Data Engineering in the Public Cloud
(Level 6, 30 Credits, Version 1.1)

Assessment Title

Scenario-Based Activity Design and Implementation (AWS)
CCC602 | Assessment-4

Weighting within the course

40%

Objective:

The objective of this task is to demonstrate a comprehensive, automation-driven approach to deploying a cloud-based data pipeline within a **multi-cloud environment** using **scripting** to orchestrate cloud-native services. Leveraging AWS-native services such as IAM, S3, Glue, Athena, and QuickSight, combined with **scripting** for event-driven automation via AWS Lambda and SNS, this solution effectively simulates an enterprise-grade ingestion and notification framework.

A key component involves implementing a real-time file monitoring mechanism through a **scripted Lambda function**, triggered by .csv uploads to S3, which publishes alerts via Amazon SNS. This exemplifies best practices in cloud-native observability, operational responsiveness, and **scripting-driven automation** aligned with DevOps methodologies.

Moreover, the seamless migration of curated datasets from AWS to Microsoft Azure, with integration into Azure Synapse Analytics, highlights the ability to design and operate federated data architectures across **multi-cloud providers**. The extensive use of scripting throughout the deployment pipeline ensures reusability, scalability, and reduced operational complexity.

Overall, this task aligns with Graduate Profile Outcome (GPO1) and Learning Outcome (LO3) by showcasing proficiency in designing, automating, and explaining cross-platform IT infrastructure using **multi-cloud orchestration, scripting, and service integration**. It evidences the learner's readiness to operate in real-world, multi-vendor cloud environments focused on automation, monitoring, and data interoperability.

Course Learning Outcomes (LOs) covered:

LO3: Design an automated system for deploying IT infrastructure on multi-vendor cloud services using user-friendly programming language.

Qualification Graduate Profile Outcomes (GPOs) covered:

GPO1: Plan and use services, technologies, and tools to automate the deployment and management of devices, applications, and infrastructure by way of scripts to automate standard system procedures.

Assessment Tasks to Learning Outcome and GPOs Mapping:

LO	GPO	Task	Task Component	Weighting
LO 3	GPO1	**Task 1:** IAM and S3 Configuration + AWS Glue for Automated Data Cataloguing	**Task 1:** (Practical Implementation)	30%
LO 3	GPO1	**Task 2:** Query and Visualize the Dataset Using Athena and QuickSight.	**Task 2:** (Practical Implementation)	20%
LO3	GPO1	**Task 3:** Multi-Cloud Integration (GitHub ➜ AWS ➜ Azure Synapse Analytics)	**Task 3:** (Practical Implementation and Multi-cloud comparison)	30%
LO3	GPO 1	**Task 4:** Final Report on AWS Services and Architecture	**Task 4:** (Report writing)	20%
			Total	100%

Recommended Tasks Completion Timeline:

Week Full Time	Part Time Week	Progress	Submission
Week 13	Week 25,26	Start working on the Assessment	
Week 14	Week 27,28	Complete Task 1 and Task 2	
Week 15	Week 29,30	Complete Task 3	
Week 16	Week 31,32	Complete Task 4 and submit	Assessment due by Week 16 (Full Time) Assessment due by Week 32 (Part Time)

Grading:

The final grade will be determined by the score achieved in this assessment based on the following table. Should a second or third attempt be required, the maximum contribution toward the overall mark for the tasks that required a second or third assessment attempt is 50%. **A late submission is considered a second attempt, so the contribution will be capped at 50%.**

To pass this assessment, you must meet the requirements of each of the learning outcomes (irrespective of the numerical grade awarded).

Grade	Range
A	Meet all course requirements, range (80—100%)
B	Meet all course requirements, range (65—79%)
C	Meet all course requirements, range (50—64%)
D	Did not meet all course requirements, range (40—49%)
E	Did not meet all course requirements, mark range (0—39%)

Candidate's Assessment Instructions:

- This assessment is an **open-book activity**; you can use your course and review notes, and offline or online resources, such as textbooks or online journals.
- You can always ask your online tutor if you need further explanation if the instructions are unclear.
- Your work should not be plagiarised. Plagiarism includes copying material without acknowledging it, copying from another student, getting another person to help you with your assessment, using material from commercial essays or assignment services, or using AI to create the answers.
- The purpose of this assessment is to assess your knowledge. In the event Yoobee suspects collusion, this will be addressed. For more information on plagiarism, please refer to the Student Handbook.
- Submit your completed assessment online in the correct space provided.
- Marks and feedback will be returned within 15 days of the submission date.
- By completing and submitting an assessment, you are authenticating that the work is original and does not violate plagiarism or copyright law. Authenticity is checked where any breaches of academic integrity are suspected. Please refer to the Student Handbook for further information.

Submission Instructions:

Submit **one PDF report** document to the LMS by the specified due date.

Your report should:

- Include your name and ID number
- Include the Azure account login details, a cover page, and a report index for verification purposes in your report.
- Use a standard citation format if external sources are referenced
- Clearly label tasks and subtasks, and Diagrams must be clear and labeled properly.
- Include screenshots of each practical step in sequence, naming and numbering the screenshots. Screenshots must display the relevant settings or outputs for each step.
- Include your answers to the assessment questions for each task, describing choices, configurations, and learned insights with an appropriate practical and theoretical understanding.
- **Submission must be in PDF format only because other formats may cause issues with accessing screenshots.**

Assessment Tasks: Scenario-Based Activity Design and Implementation (AWS)

Scenario:

Yoobee Data Organization is implementing a cloud-based data pipeline for its analytics operations. The company seeks a **automated** solution for data ingesting, processing, querying, and visualizing data using key AWS services such as **IAM, Amazon S3, AWS Glue, Amazon Athena,** and **Amazon QuickSight**. The pipeline also includes migrating log data to **Microsoft Azure** for multi-cloud data analysis with **Azure Synapse Analytics**. This solution employs **scripting** and event-driven automation using **AWS Lambda** and **Amazon SNS** to simulate an enterprise-grade, automated deployment within a **multi-cloud environment**, ensuring scalability, observability, and operational efficiency through advanced cloud service orchestration.

Your Tutor has provided a folder named Assessment_4_Azure_Project_Codes as a Git GitHub repository at the link below for require datasets for this Assessment and SQL commands your reference.

Git Repository Link: https://github.com/yoobeecloud/Assessment_4_AWS_Project_Codes

Delete all resources after completing the Assessment, including the S3 bucket. IAM Role, Tables in AWS Glue service, etc. To avoid any billing on your AWS attached credit card.

Task 1: Cloud Infrastructure Setup and Automation for Data Ingestion, Cataloguing, and Real-Time Notifications Using AWS

Activities: 1.1: IAM and S3 Setup

- Create an **IAM user** (e.g., your first name) with **AdministratorAccess**.
- Create an **S3 bucket** (e.g., yourfirstname2025) and set up folders:
    - /orders/snapshot_day=2017-01-01
- Upload a manually prepared CSV file containing January 2017 order data.

Deliverables:

- Screenshot of IAM user and attached policy.
- Screenshot of the S3 bucket and folder structure.
- Screenshot showing the uploaded file.
- Submit a screenshot as shown in **Appendix 1**

Activities: 1.2: AWS Glue Configuration

- Create a **Glue database** named db_yoobee.
- Configure a **Glue Crawler,** name it as yoobee_crawler.
    - Target the orders/ folder in your S3 bucket.
    - Enable it to **crawl new subfolders** automatically.
    - Assign an IAM role (e.g., AWSGlueServiceRole-assessmentsql).
- Create a new folder in the S3 bucket named (snapshot_day=2017-01-04) with a filter data file of April 2017 and re-run the crawler.

Deliverables:

- Screenshot of the Glue database and crawler settings.
- Screenshot of schema and partitions after re-crawling.
- Submit a screenshot as shown in **Appendix 1**

Activities 1.3: Real-Time Notification Using AWS Lambda and SNS

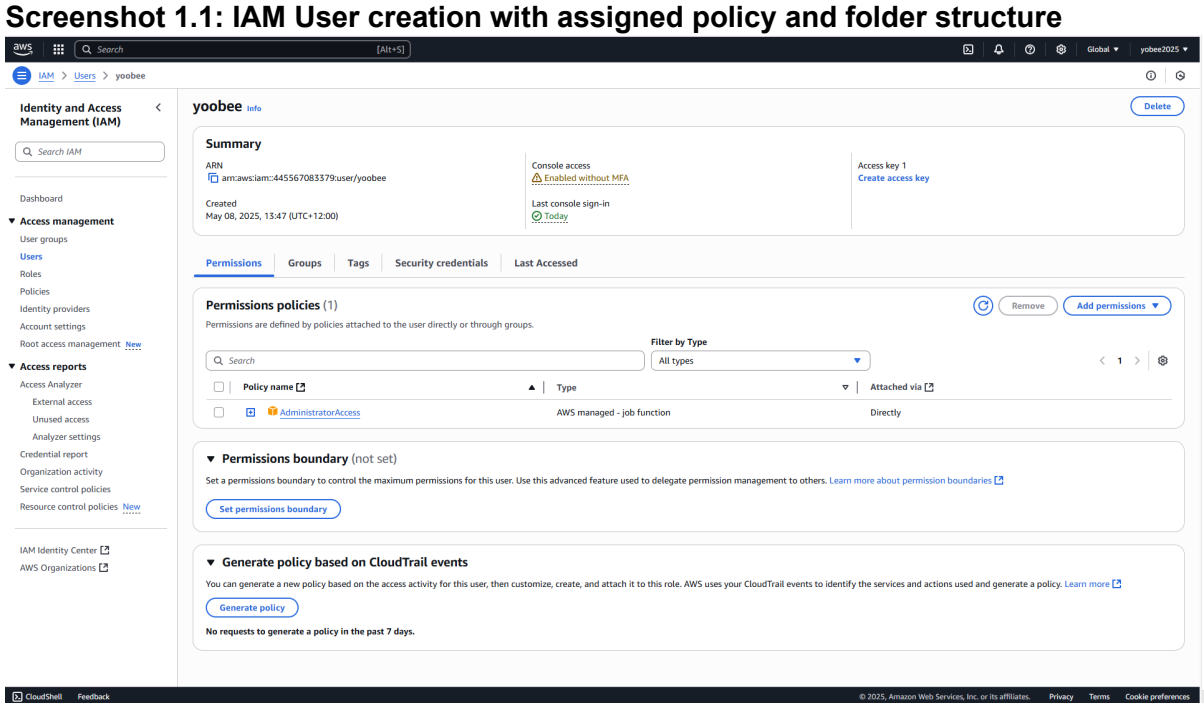
- Create an **SNS topic** named Notify CSVUploads
- Write and deploy an AWS **Lambda function** that:
  - Is triggered when a file is uploaded to the S3 bucket.
  - Checks if the uploaded file has a .csv extension.
  - Publishes a structured notification message to the Notify CSVUploads topic.
- Add an S3 event trigger to invoke the Lambda function on ObjectCreated:* events.

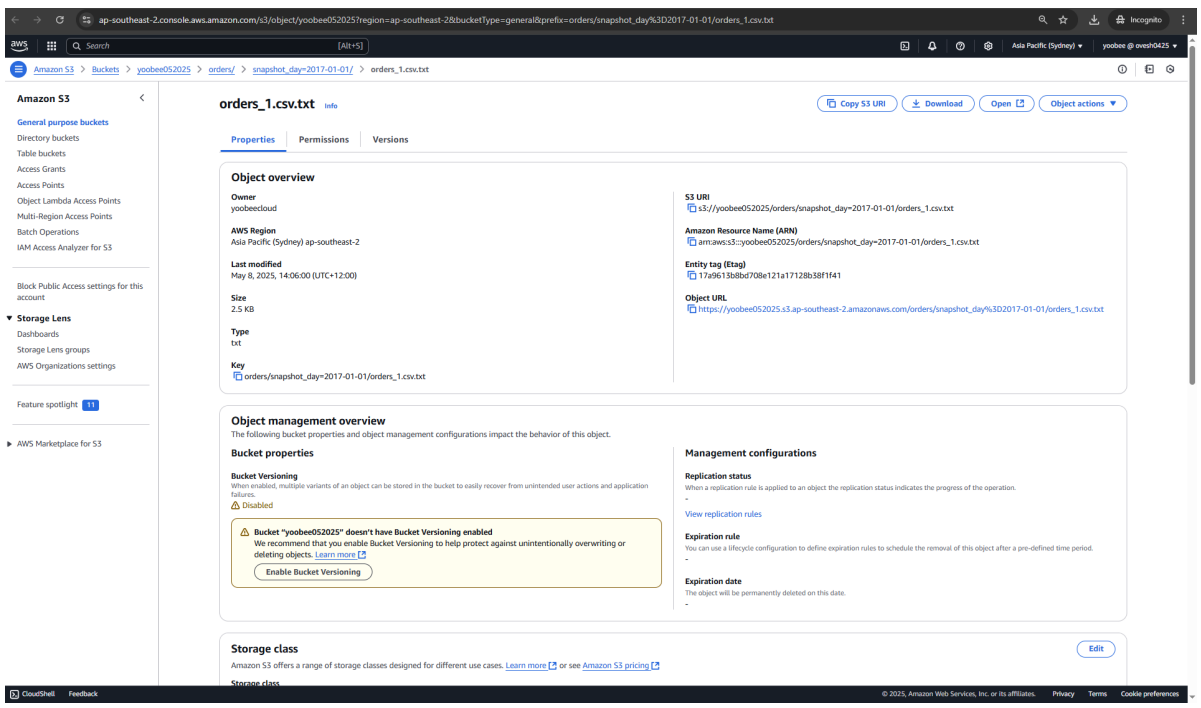
Deliverables:

- Screenshot of the SNS topic configuration.
- Screenshot of the Lambda function code and successful execution log.
- Screenshot showing the email (or CloudWatch log) notification triggered by a CSV upload.

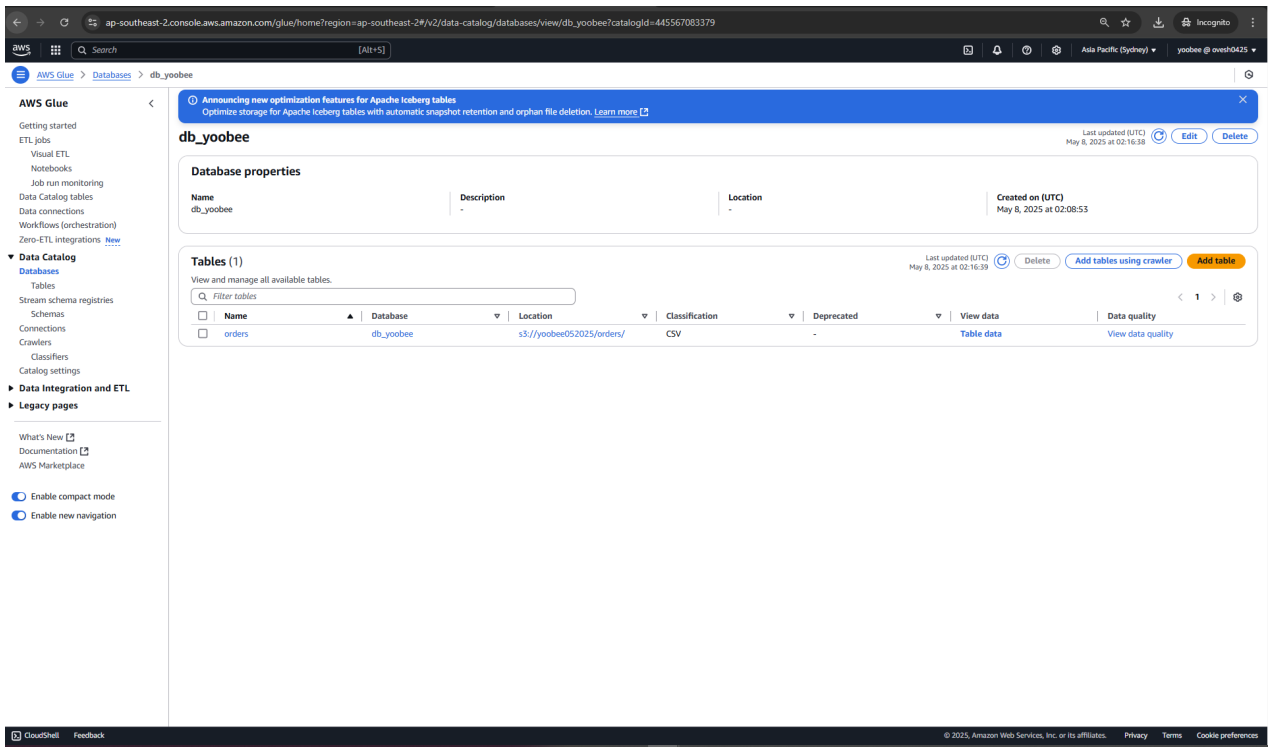
Appendix-1:

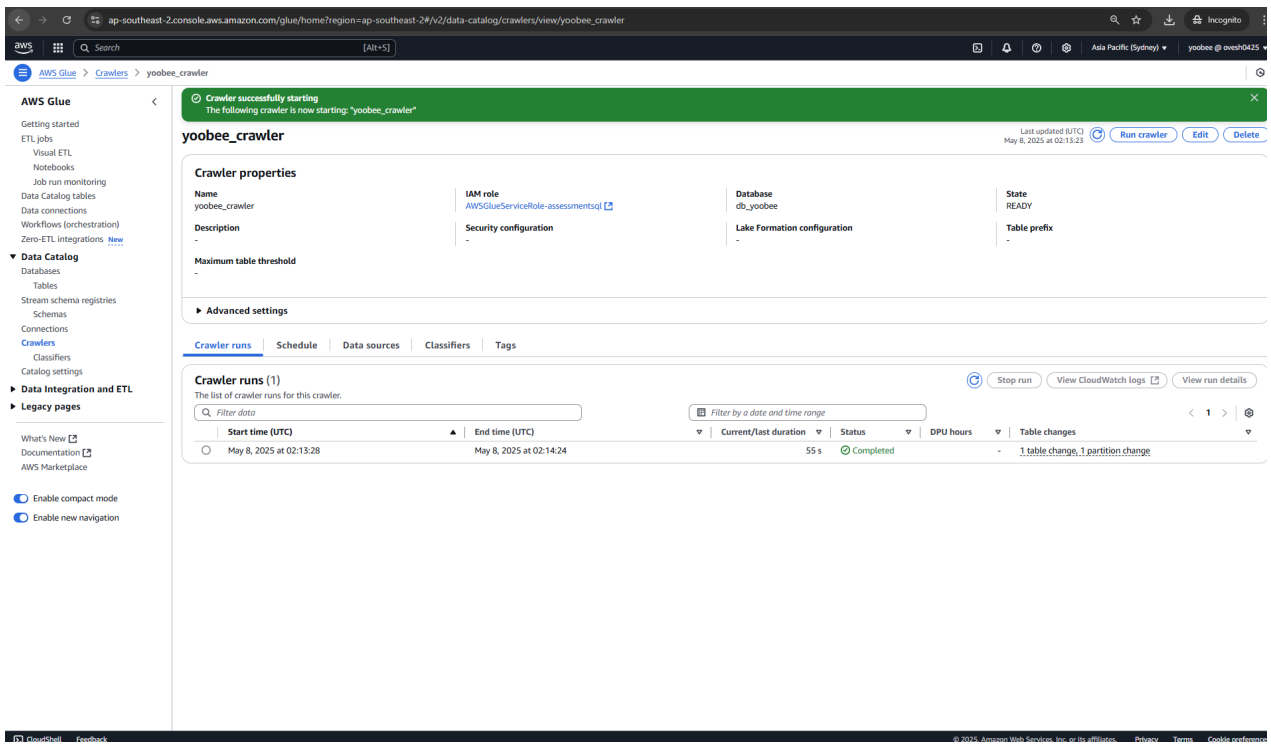
Screenshot 1.1: IAM User creation with assigned policy and folder structure



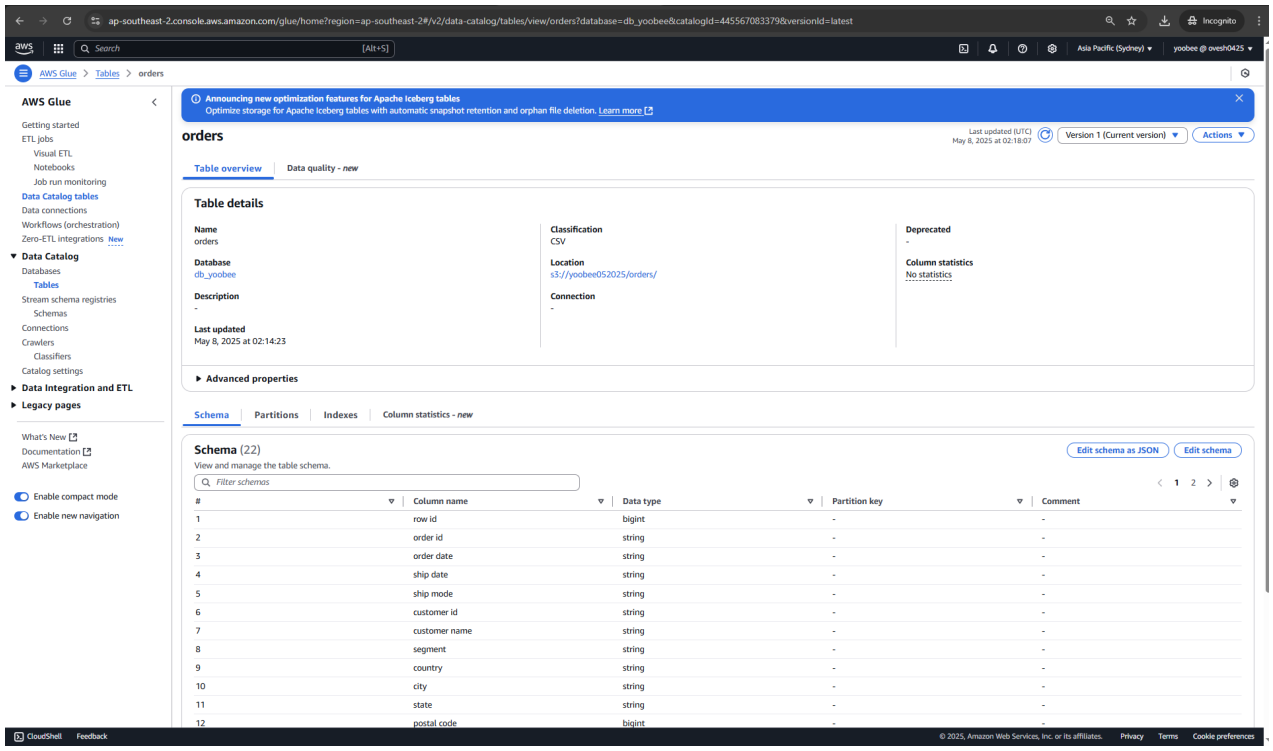


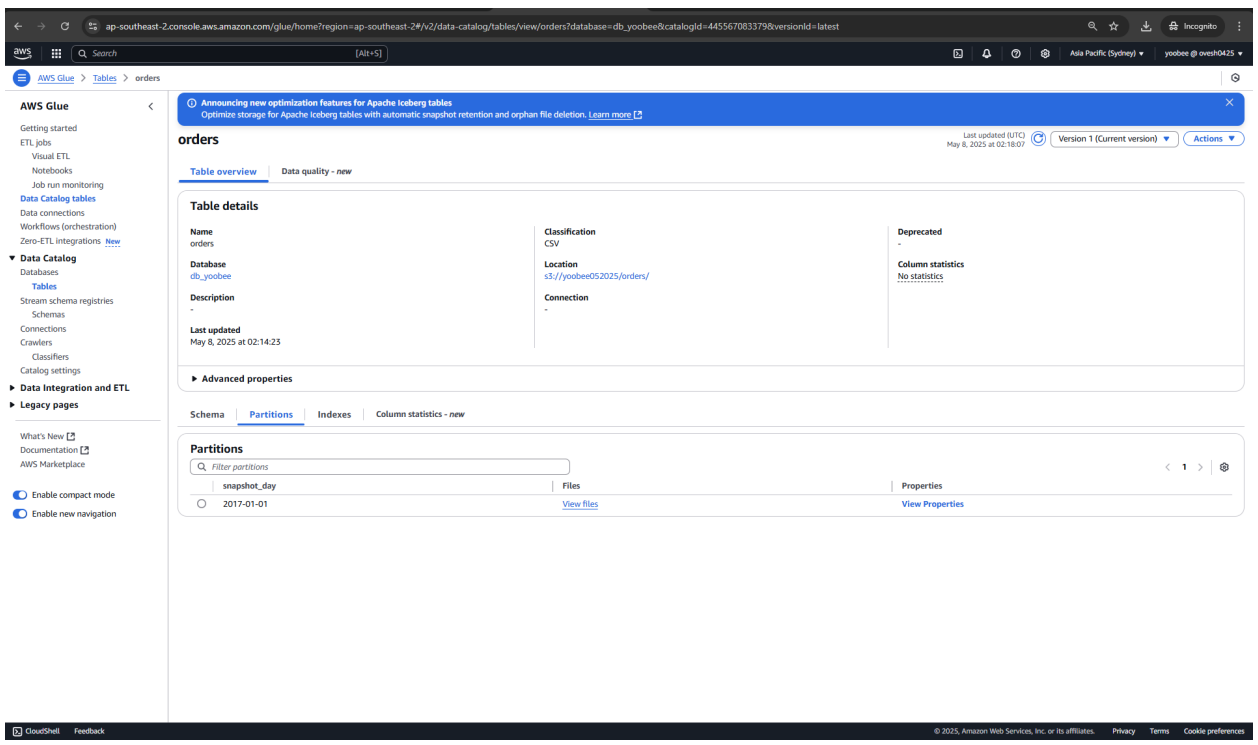
Screenshot 1.2: Using a Glue service database created, and a crawler run outcome

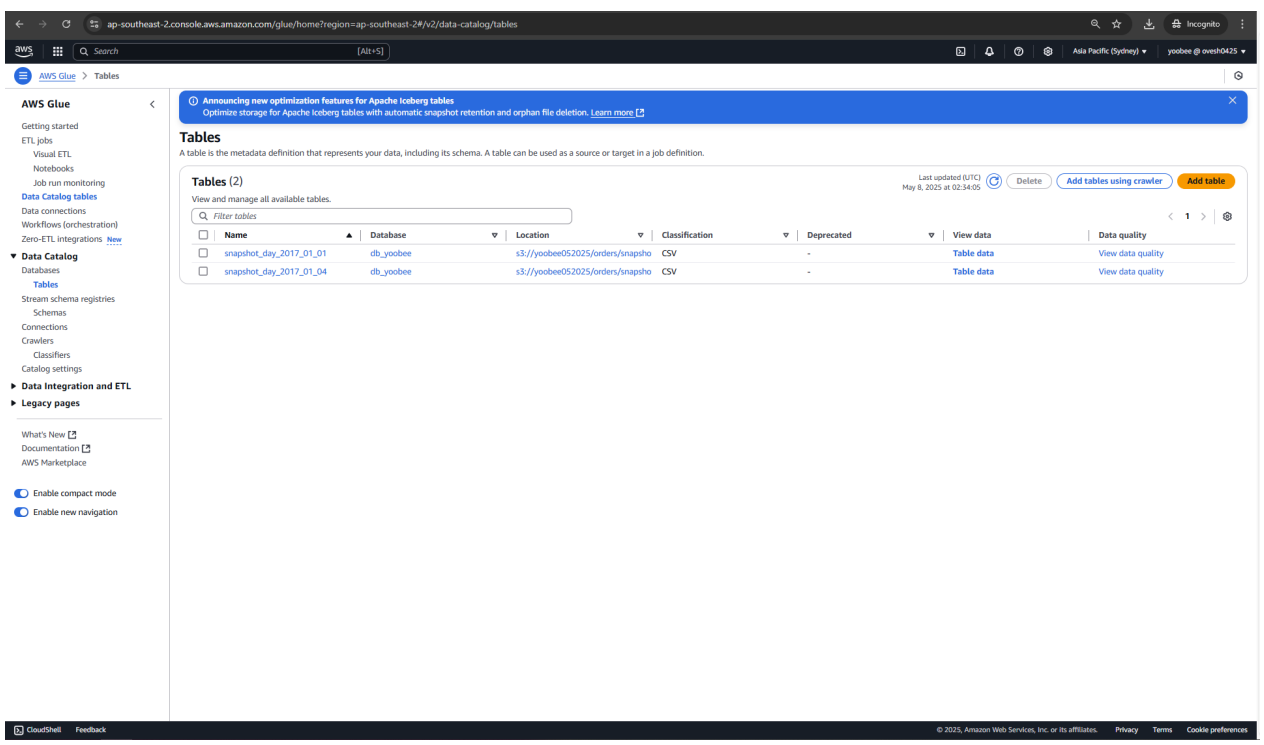




Screenshots of schema and partitions after re-crawling.


ap-southeast-2.console.aws.amazon.com/glue/home?region=ap-southeast-2#/v2/data-catalog/tables/view/orders?database=db_yoobee&catalogId=445567083379&versionId=latest
AWS Glue > Tables > orders
Announcing new optimization features for Apache Iceberg tables
Optimize storage for Apache Iceberg tables with automatic snapshot retention and orphan file deletion. Learn more
orders
Table overview
Data quality - new
Table details
Name
orders
Database
db_yoobee
Description
-
Last updated
May 8, 2025 at 02:14:23
Classification
CSV
Location
s3://yoobee052025/orders/
Connection
-
Deprecated
-
Column statistics
No statistics
Advanced properties
Schema
Partitions
Indexes
Column statistics - new
Schema (22)
View and manage the table schema.
Edit schema as JSON
Edit schema
Filter schemas
| Column name | Data type | Partition key | Comment
1 | row id | bigint | - | -
2 | order id | string | - | -
3 | order date | string | - | -
4 | ship date | string | - | -
5 | ship mode | string | - | -
6 | customer id | string | - | -
7 | customer name | string | - | -
8 | segment | string | - | -
9 | country | string | - | -
10 | city | string | - | -
11 | state | string | - | -
12 | postal code | bigint | - | -
CloudShell
Feedback
© 2025, Amazon Web Services, Inc. or its affiliates.
Privacy
Terms
Cookie preferences


ap-southeast-2.console.aws.amazon.com/glue/home?region=ap-southeast-2#/v2/data-catalog/tables/view/orders?database=db_yoobee&catalogId=445567083379&versionId=latest
AWS Glue > Tables > orders
Announcing new optimization features for Apache Iceberg tables
Optimize storage for Apache Iceberg tables with automatic snapshot retention and orphan file deletion. Learn more
orders
Table overview
Data quality - new
Table details
Name
orders
Database
db_yoobee
Description
-
Last updated
May 8, 2025 at 02:14:23
Classification
CSV
Location
s3://yoobee052025/orders/
Connection
-
Deprecated
-
Column statistics
No statistics
Advanced properties
Schema
Partitions
Indexes
Column statistics - new
Partitions
Filter partitions
snapshot_day | Files | Properties
2017-01-01
View files
View Properties
CloudShell
Feedback
© 2025, Amazon Web Services, Inc. or its affiliates.
Privacy
Terms
Cookie preferences


ap-southeast-2.console.aws.amazon.com/glue/home?region=ap-southeast-2#/v2/data-catalog/tables
AWS Glue > Tables
Announcing new optimization features for Apache Iceberg tables
Optimize storage for Apache Iceberg tables with automatic snapshot retention and orphan file deletion. Learn more
Tables
A table is the metadata definition that represents your data, including its schema. A table can be used as a source or target in a job definition.
Tables (2)
View and manage all available tables.
Last updated (UTC)
May 8, 2025 at 02:34:05
Delete
Add tables using crawler
Add table
Filter tables
Name | Database | Location | Classification | Deprecated | View data | Data quality
snapshot_day_2017_01_01 | db_yoobee | s3://yoobee052025/orders/snapsho | CSV | - | Table data | View data quality
snapshot_day_2017_01_04 | db_yoobee | s3://yoobee052025/orders/snapsho | CSV | - | Table data | View data quality
CloudShell
Feedback
© 2025, Amazon Web Services, Inc. or its affiliates.
Privacy
Terms
Cookie preferences

Task 2: Query and Visualize the Dataset Using Athena and QuickSight.

Activities: 2.1: Amazon Athena Queries

- Set up an S3 bucket for Athena logs (e.g., yoobeeathena_logs).
- In Athena, connect to your Glue Catalog and run queries like:
- Execute pre-defined queries:

  - **SQL Example 1**

    ```
    SELECT category, SUM(sales)

    FROM "db_yoobee"."orders"

    GROUP BY category;
    ```

  - **SQL Example 2**

    ```
    SELECT category, SUM(sales)  as catrgory_sales
    FROM "db_yoobee"."orders"
    GROUP BY category;
    ```

  - **SQL Example 3**

    ```
    SELECT *
    FROM "db_yoobee"."orders";
    ```

  - **SQL Example 4**

    ```
    SELECT *
    FROM "db_yoobee"."orders"
    WHERE snapshot_day = '2017-01-01';
    ```

Deliverables:

- Screenshot of Athena setup and query editor.
- Screenshot of query results.
- Copy of SQL queries used.
- Submit a screenshot as shown in Appendix 2

Activities: 2.2: Amazon QuickSight Dashboards

- Sign up for **Amazon QuickSight** (Standard Edition).
- Authorize QuickSight to access your **S3 bucket** and connect to Athena and assign require permission to access s3 bucket in QuickSight.
- Use **SPICE** to import data and build:

  - Bar chart showing **Sales by Category**
  - Chart showing Profit by City

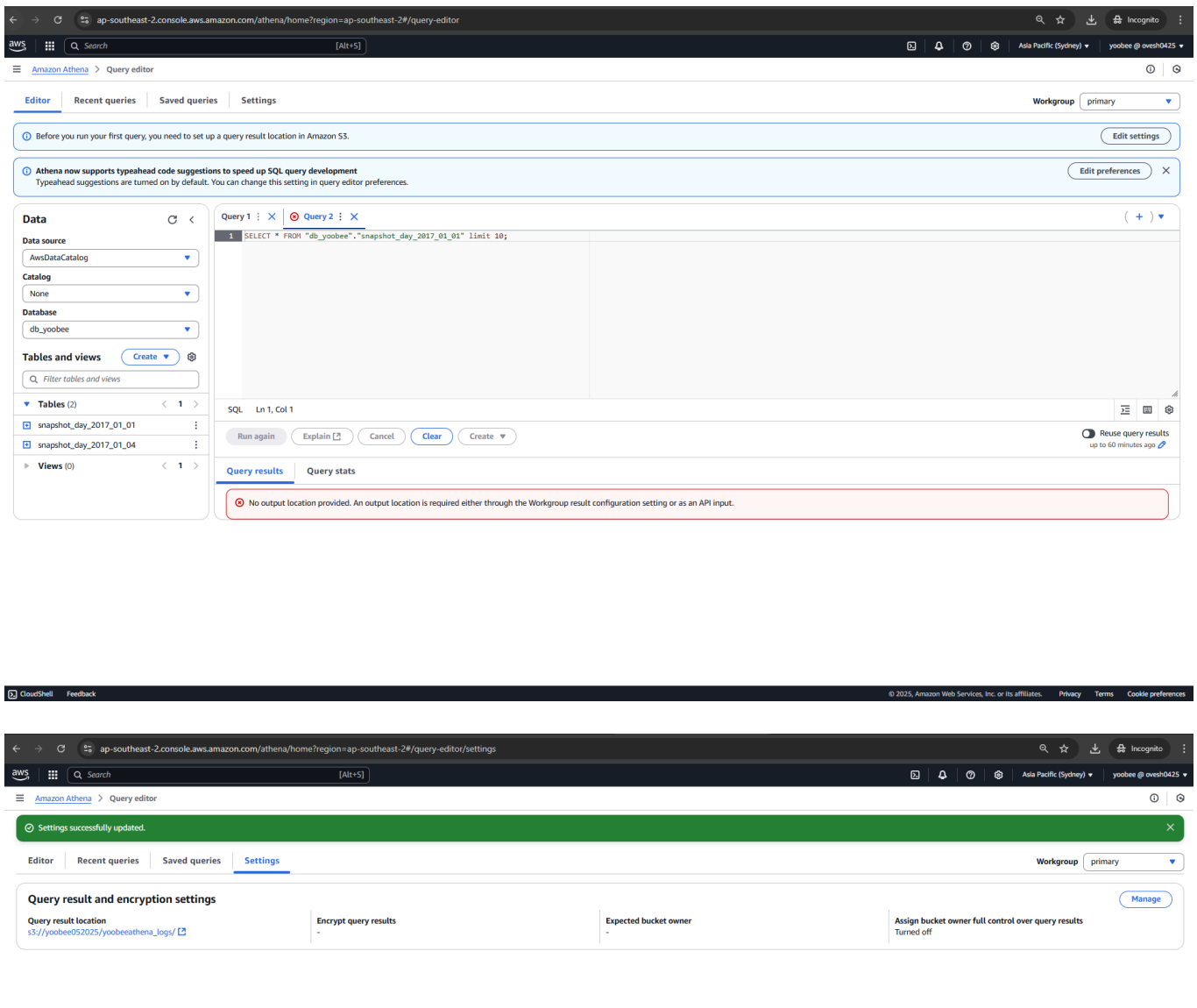
- Resolve any S3 access issues via QuickSight security settings.

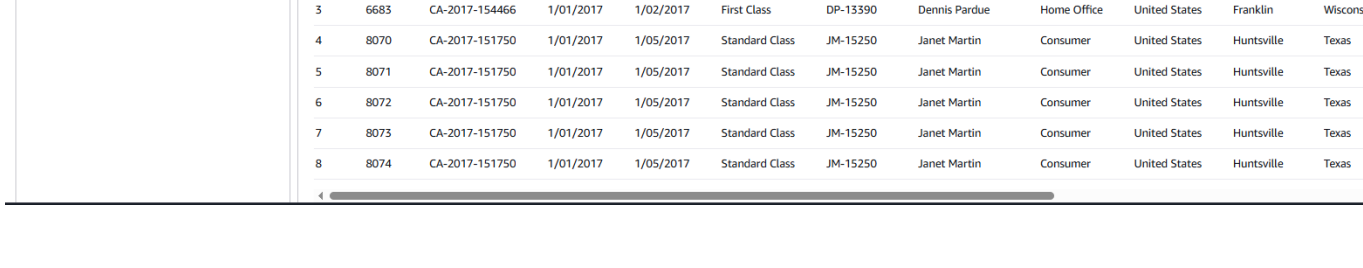
Deliverables:

- Screenshots of created visuals.
- Summary of key business insights from the charts.
- Submit a screenshot as shown in Appendix 2

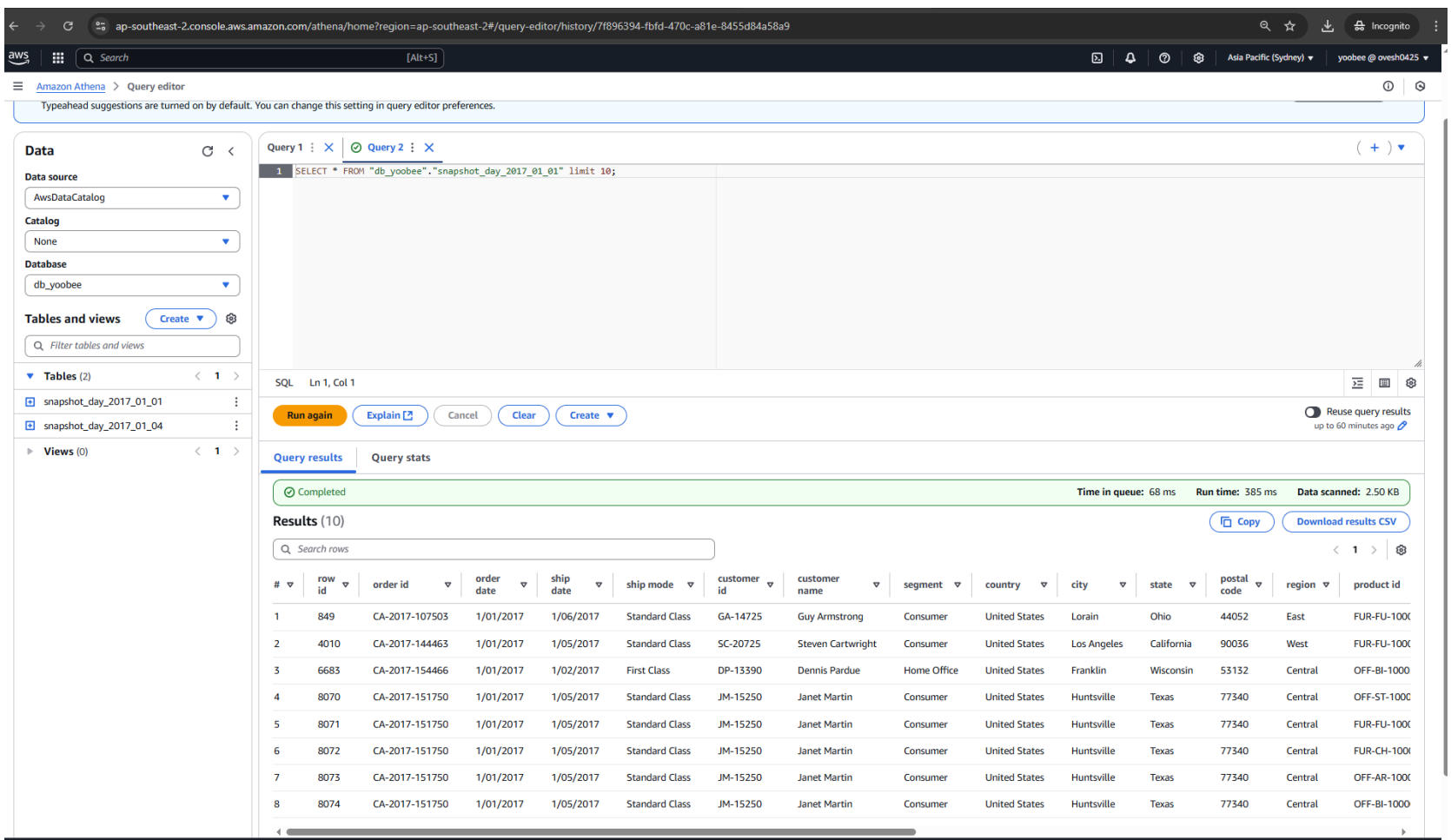
Appendix-2:

Screenshots 2.1 with all SQL queries using the Athena cloud service:

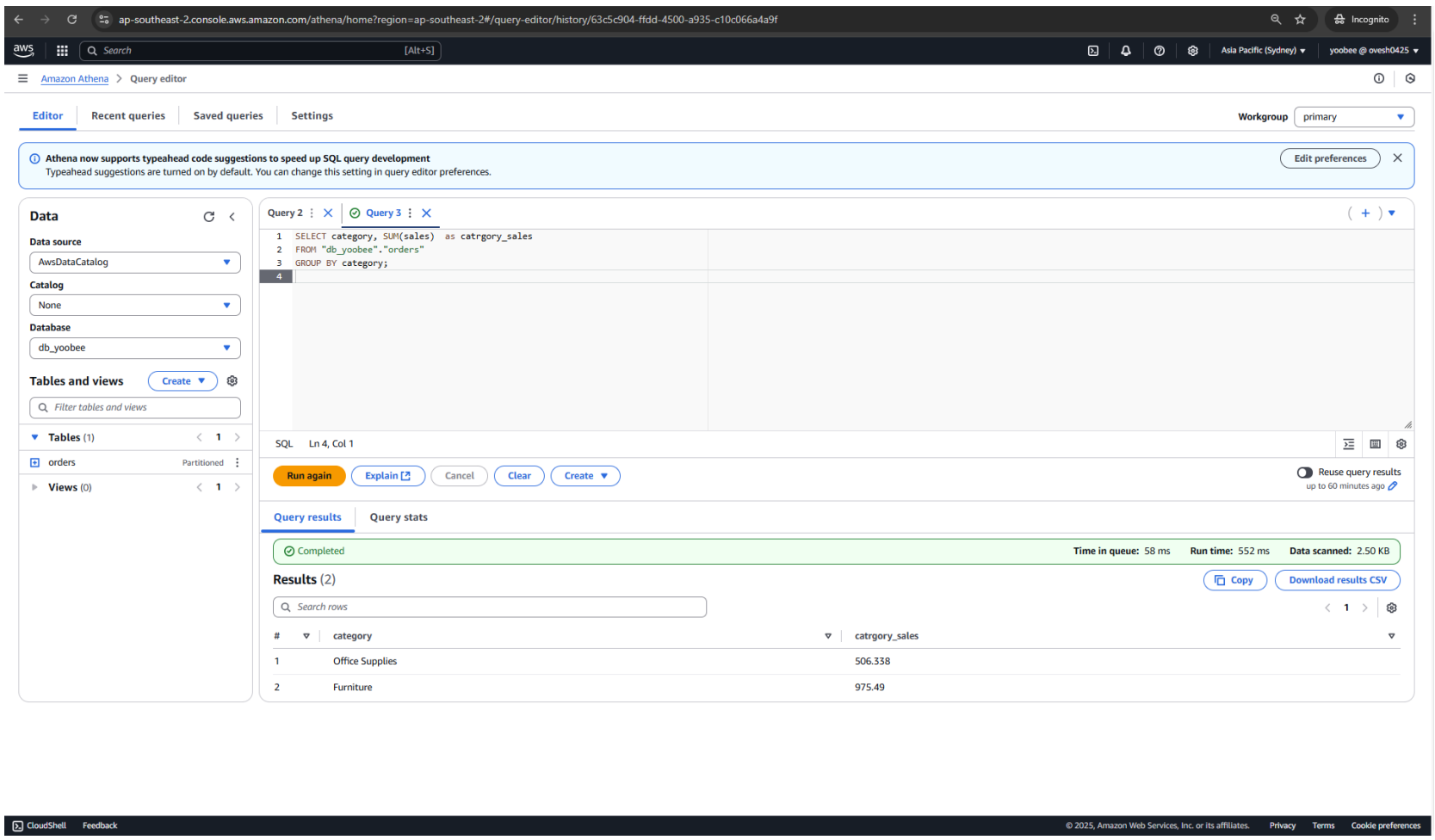




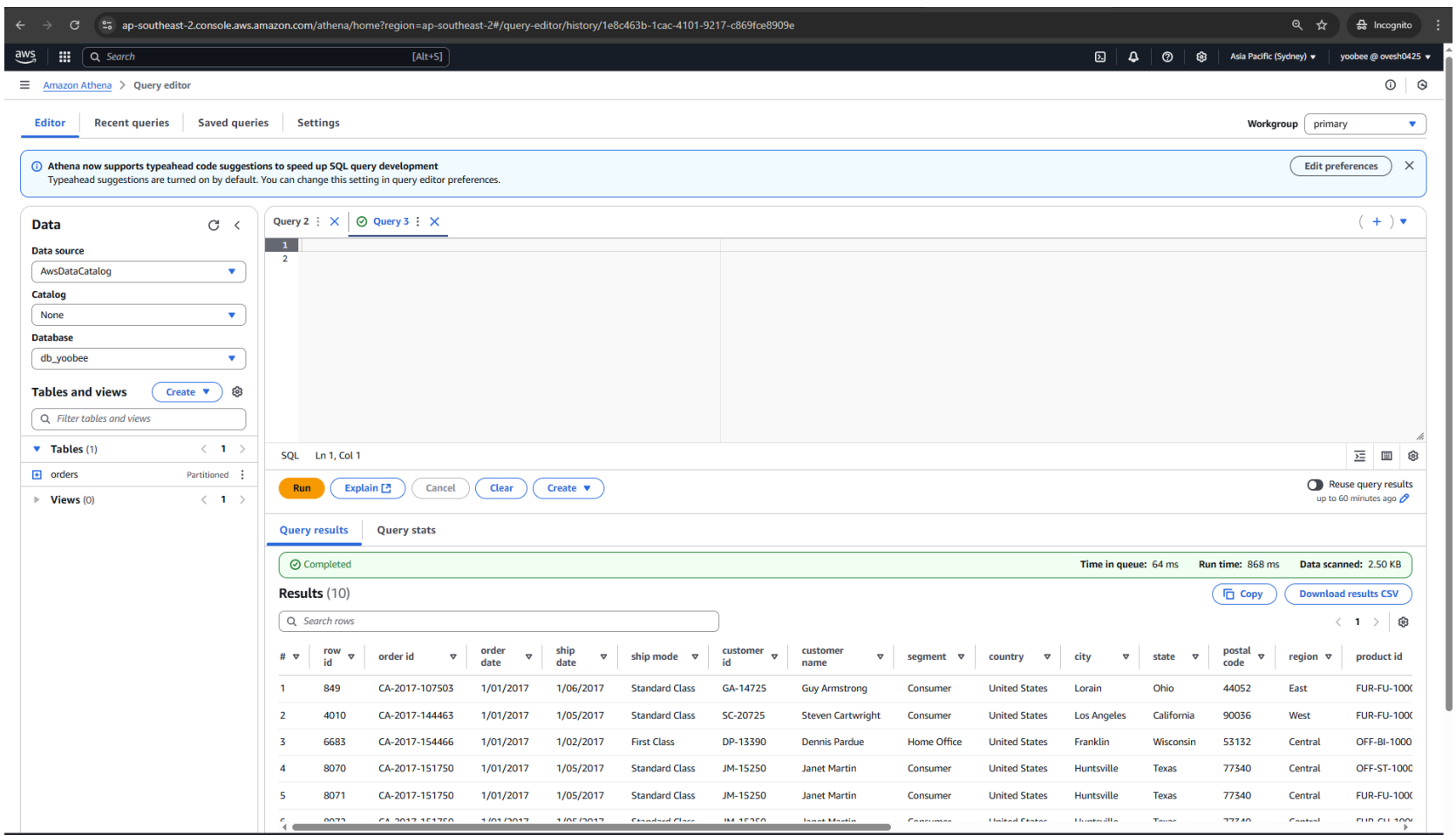
Example 1 Query Result



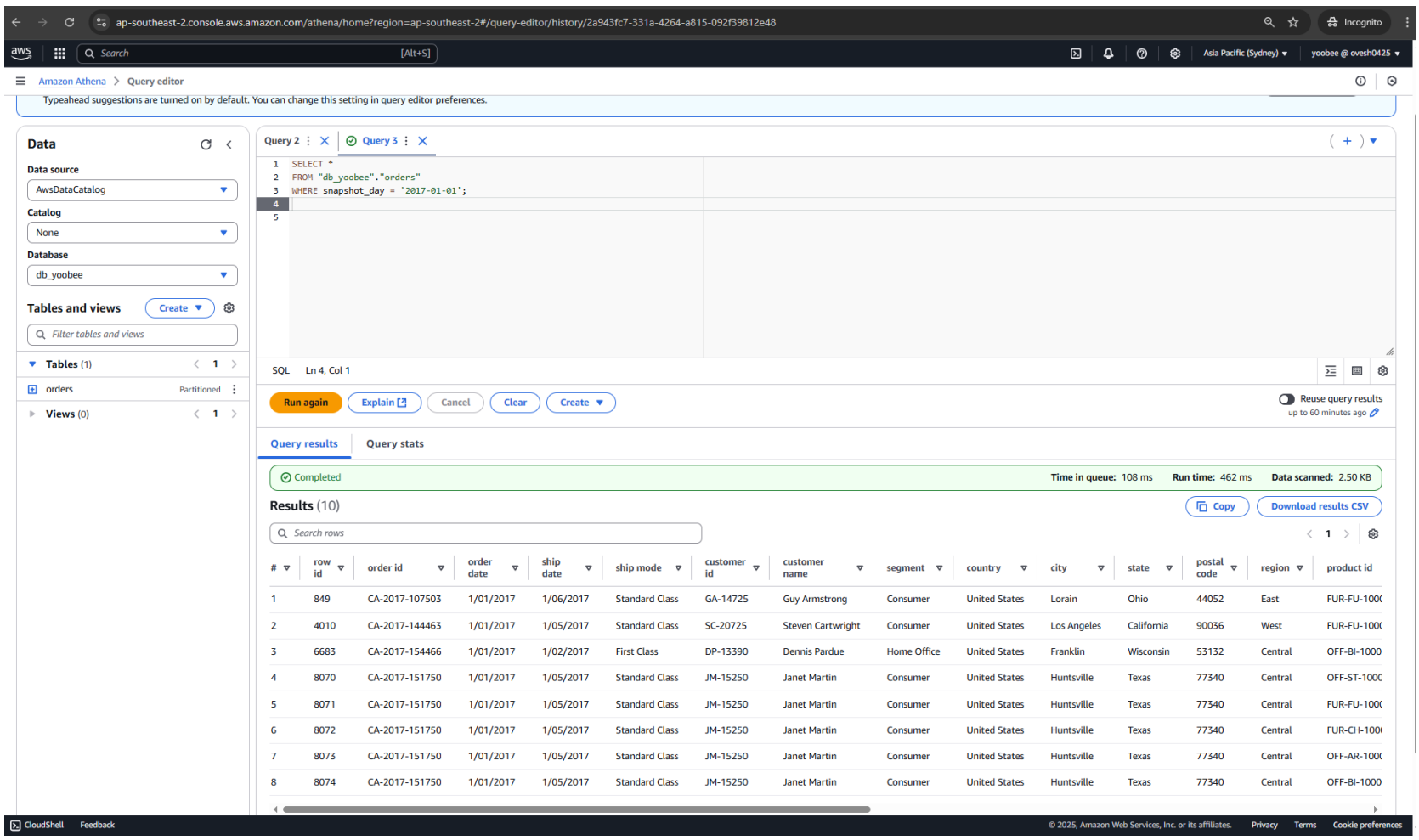
Example 2 Query Result



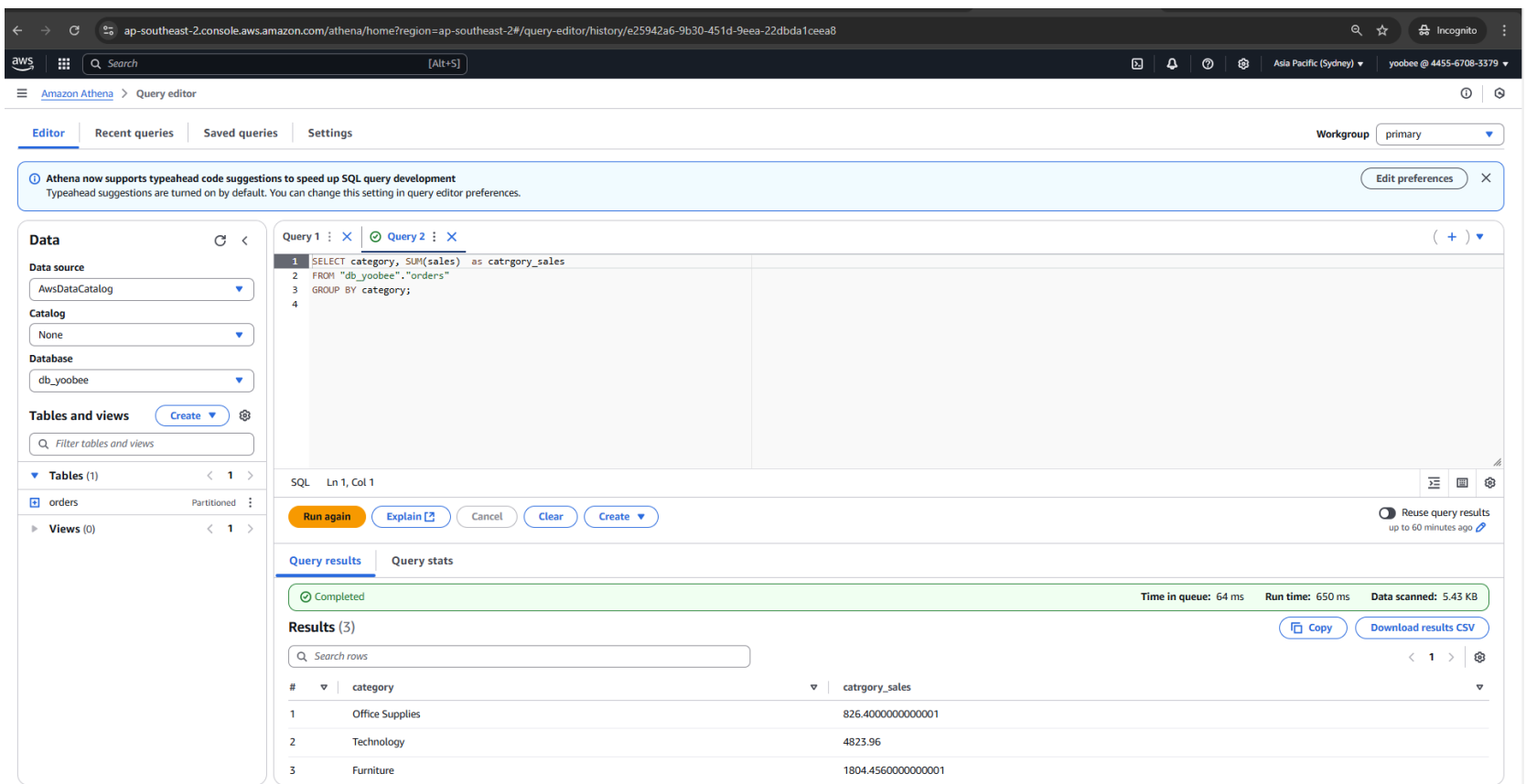
Example 3 Query Result



Example 4 Query Result

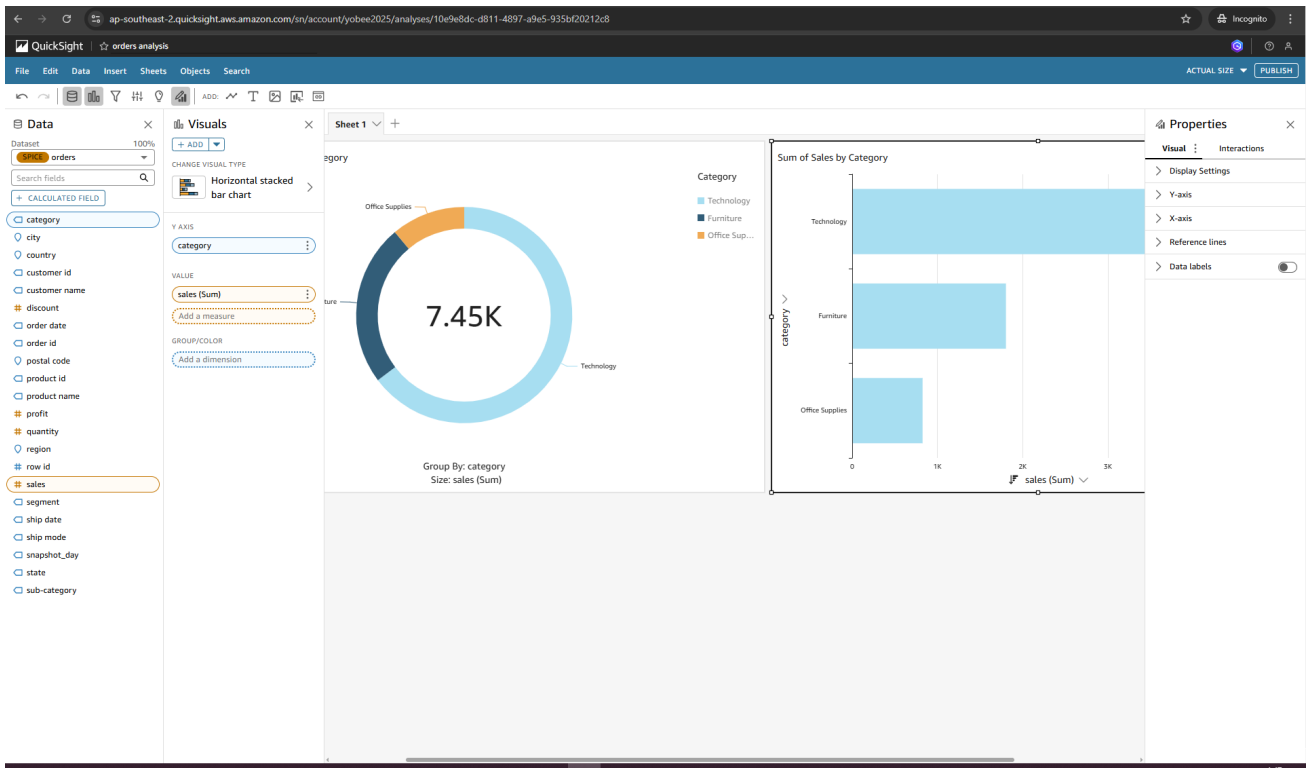


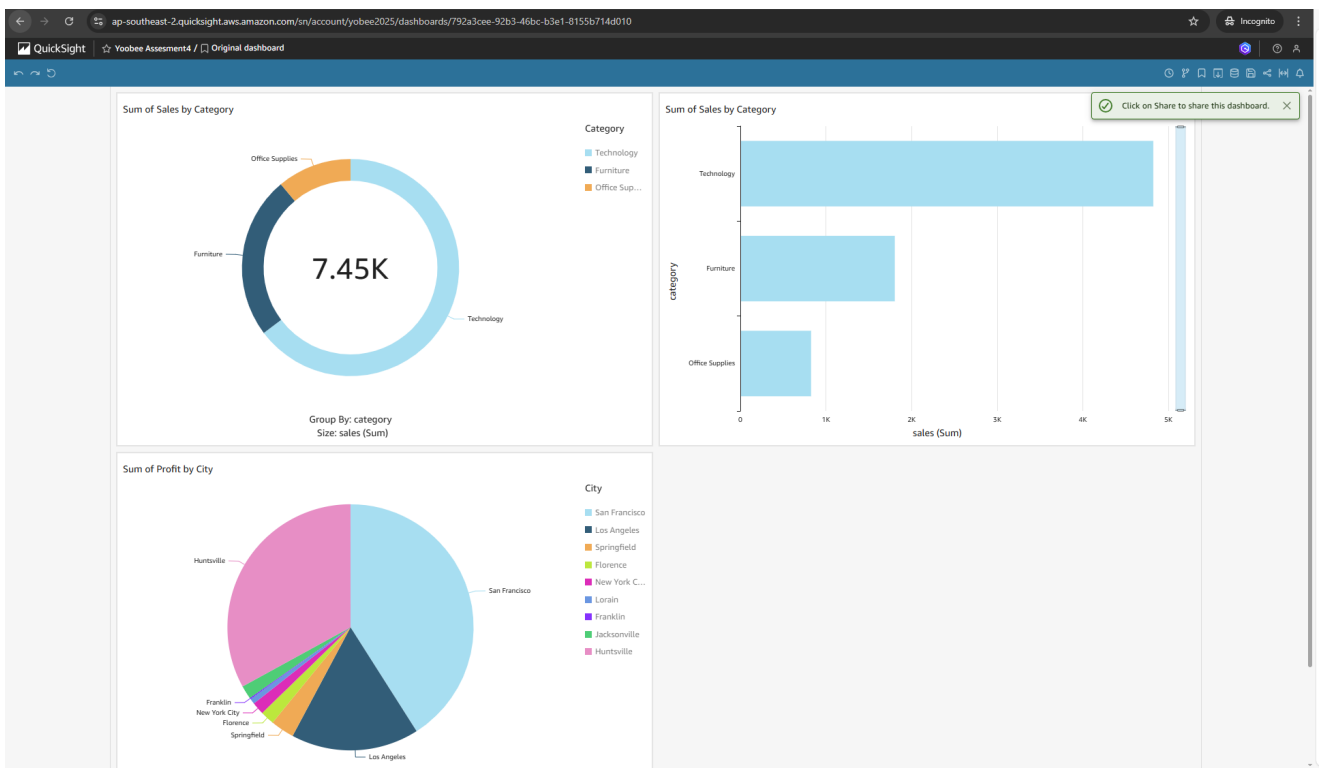
Verify the difference in the Data scanned size 5.43 due to the second file uploaded and running Scrawler.



Screenshots 2.2

- Bar chart showing **Sales by Category**
- Chart showing Profit by City





Task 3: Multi-Cloud Integration (GitHub → AWS → Azure Synapse Analytics)

Purpose:

Demonstrate basic cross-cloud interoperability by transferring curated datasets from AWS S3 to Azure Synapse and performing SQL analytics on Azure, using Azure-native low-code tools like Synapse Studio and Upload relevant screenshots as outlined in **Appendix-3.**

Activity 3.1: Data Export from AWS S3

- After transforming and querying your data using Athena, in **Task-2 download the result sets (CSV)** from Athena output (stored in your Athena S3 bucket, e.g., yoobeeathena_logs/).
- **Rename and organize CSVs**: e.g., orders_jan2017.csv, orders_apr2017.csv

Activities 3.2: Upload CSV files from Azure Portal

- Upload CSV files to an **Azure Blob Storage container** (e.g., yoobeecontainer/data/)
- Optional: Mount this Blob container to Synapse Workspace as a Linked Service.

Activities 3.3: Ingest Data into Synapse

- Use **Azure Synapse Studio**:
  - Create a **Linked Service** for the Blob container.
  - Use **Data Flow or Copy Data Tool** to ingest into a **Synapse SQL table**.
  - Use **Serverless SQL Pool** or **Dedicated SQL Pool** to query the ingested data.

Activities 4.4: Run SQL Queries in Synapse

Replicate (or modify) your AWS Athena queries:

Query Example:

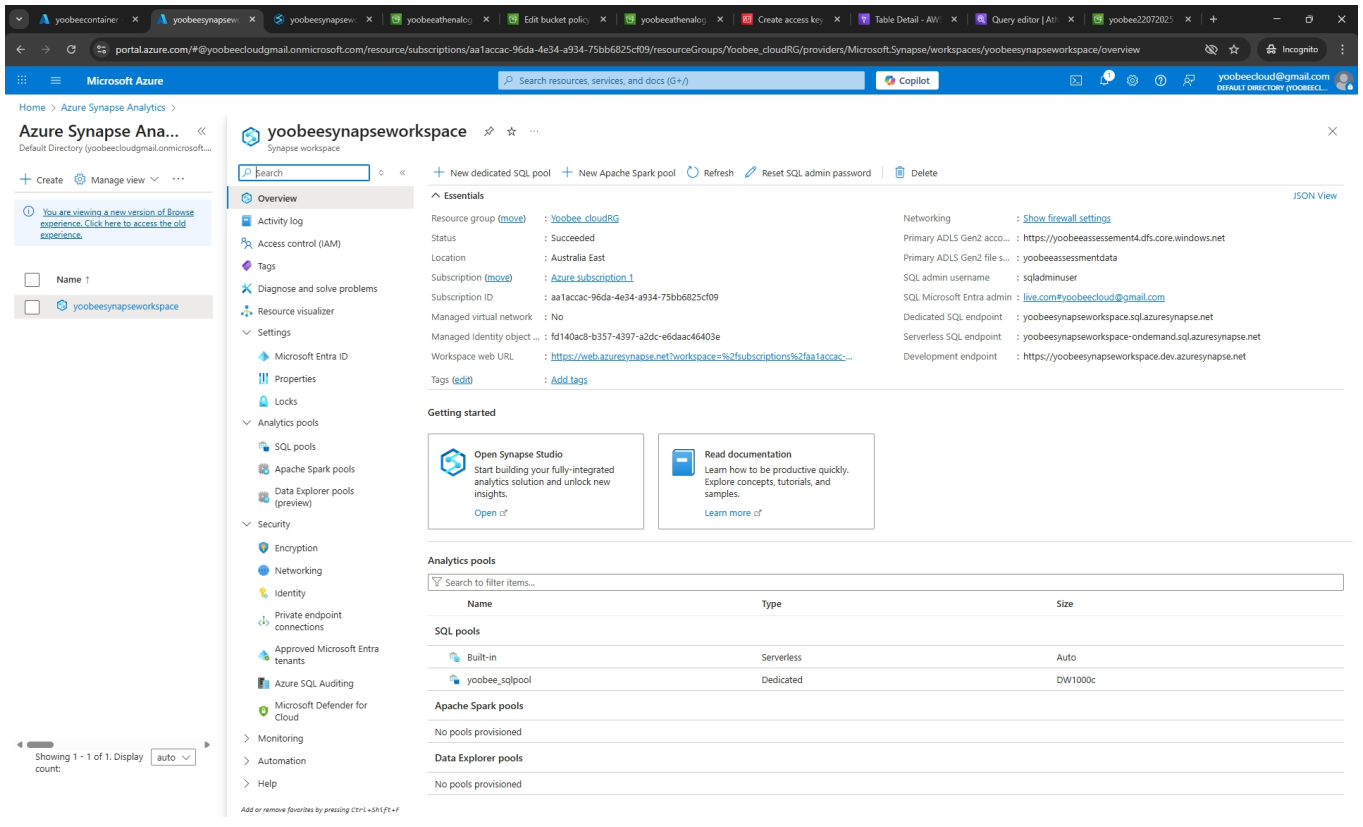
```
SELECT category, SUM(sales) AS total_sales
FROM orders_external
GROUP BY category;
```

Deliverables:

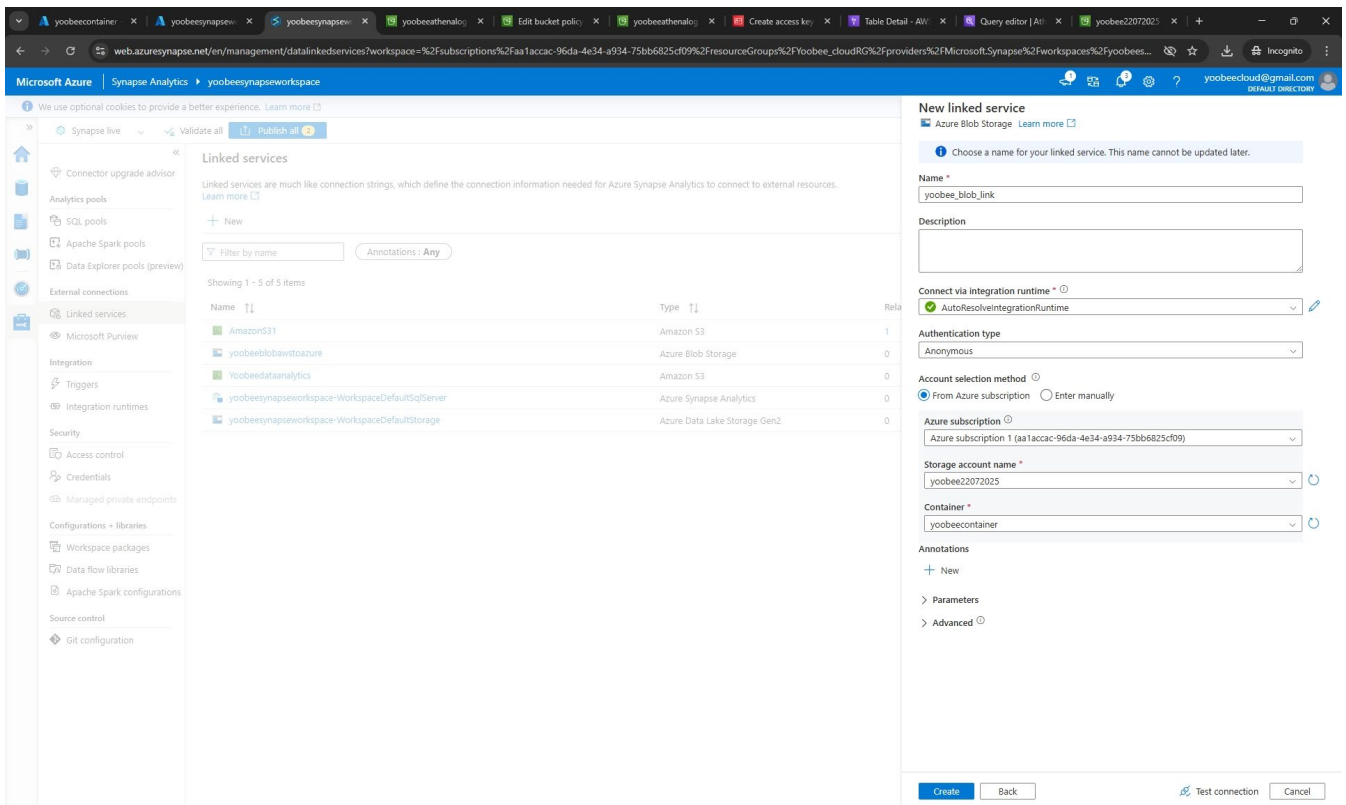
- Screenshot of Azure Blob container with uploaded CSVs.
- Screenshot of Linked Service and pipeline settings.
- Screenshot of Synapse SQL query results.
- Screenshot of Synapse Studio SQL Editor.

Appendix:3

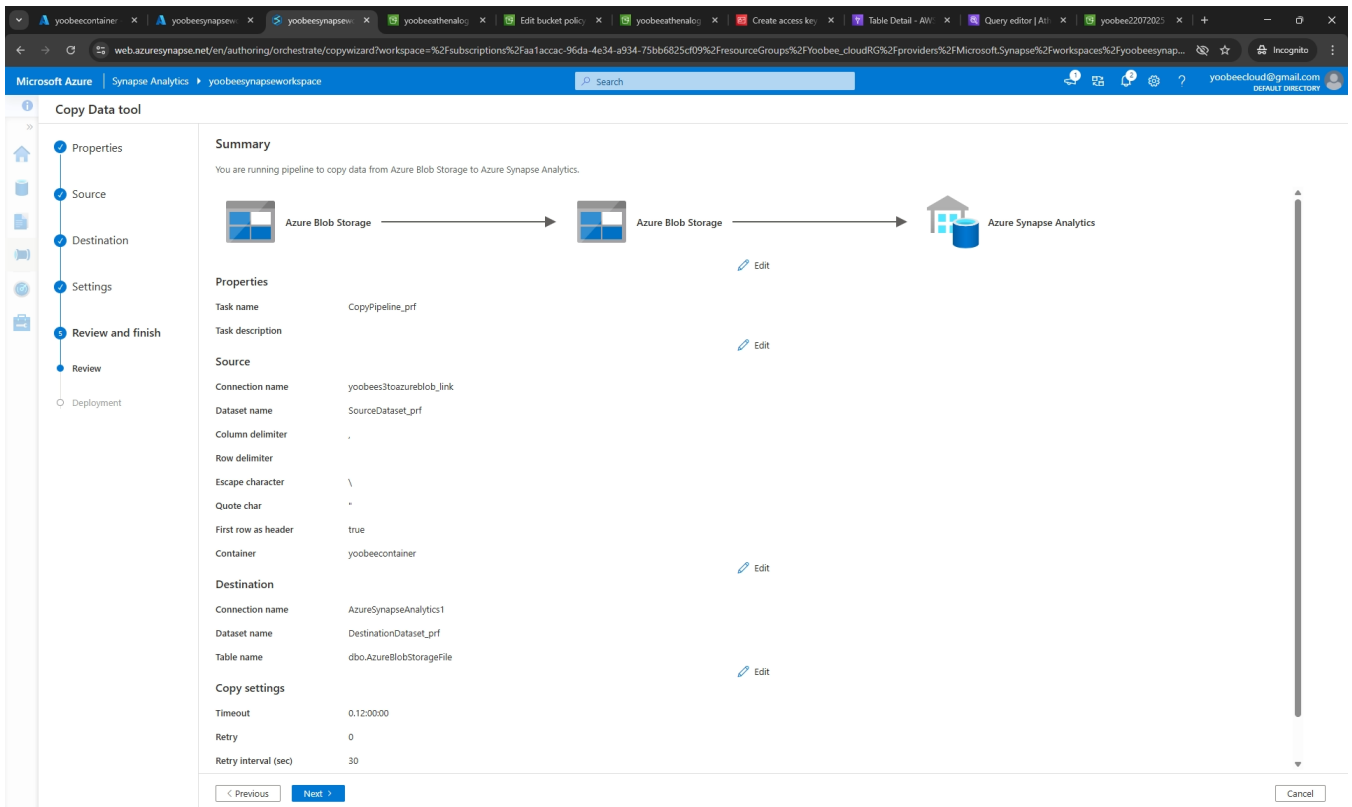
Screenshot - 3.1: Azure synapse analytics workspace



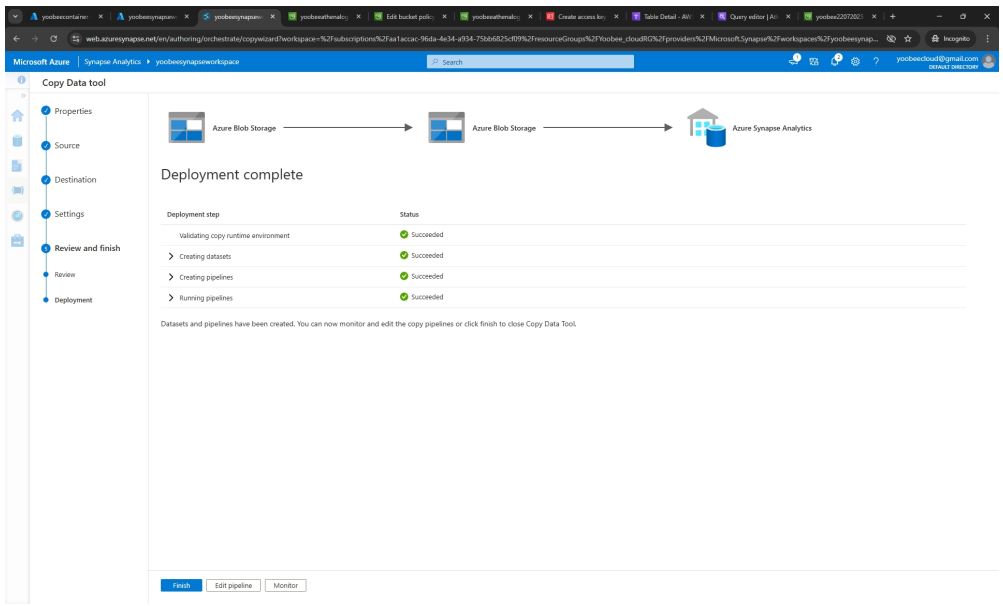
Screenshot - 3.2: Azure link service configuration



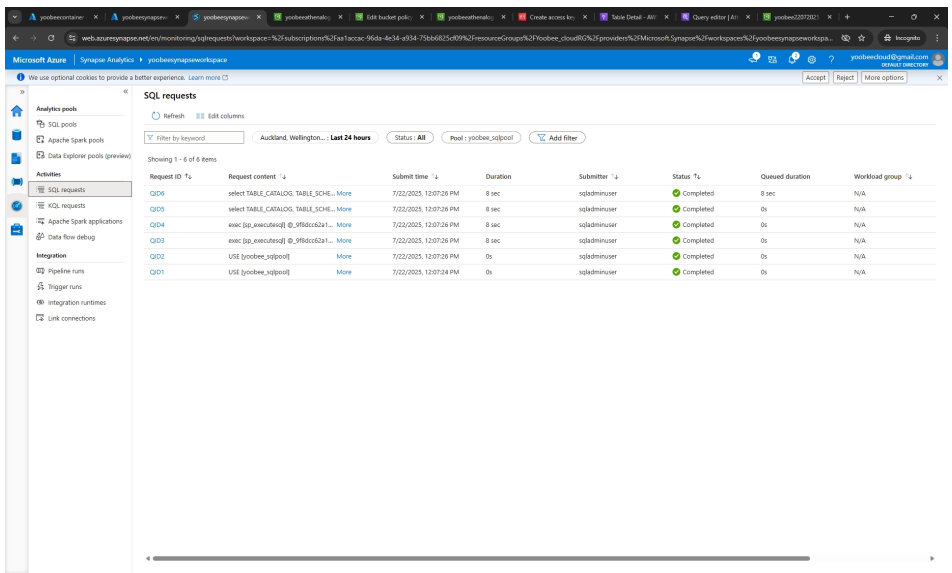
Screenshots - 3.3 Data migration from Azure Blob storage to Synapse Analytics



Screenshots - 3.3 Data migration status verification



Screenshots - 3.5 SQL request data completion process



Task 4: Final Report on AWS Services and Architecture

Activities 4.1: Report writing

"Summarize your understanding of how AWS services were used to automate the data engineering workflow and illustrate this with an architectural diagram. Reflect on how this approach supports the learning outcome of designing automated IT infrastructure using user-friendly, service-based tools."

Include the following sections:

- **Overview of AWS Services Used**
  Describe the purpose of IAM, S3, Glue, Athena, and QuickSight.
- **How These Services Work Together**
  Explain the end-to-end flow and integration points.
- **Automation and No-Code Features**
  Show how tasks were completed without custom scripts.
- **Data Engineering Perspective**
  Relate the work to real-world data engineering practices.
- **Athena vs. Synapse SQL** experience brief comparison.
- **Challenges and Solutions**
  Describe any issues faced (e.g., schema errors, IAM permissions) and how you resolved them.

Deliverables:

- A PDF report (1000 words ±10%).
- An architecture diagram showing the AWS components and data flow.
- Use clear section headings and finish with a summary paragraph on how your project meets the learning outcome.

Marking Rubric

To pass this assessment, you must meet the requirements of each of the learning outcomes (irrespective of the numerical grade awarded).

Criterion		Evidence				
Task and Weightage		**A (80-100%)**	**B (65-79%)**	**C (50-64%)**	**D (40-49%)**	**E (0-39%**
Task 1: **Cloud Infrastructure Setup and Automation for Data Ingestion, Cataloguing, and Real-Time Notifications Using AWS** **(LO1)**	**1.1** IAM and S3 Setup	All components correctly implemented: IAM user with proper policy, correctly named bucket and folder structure, accurate CSV file upload, and well-documented screenshots.	Minor errors (e.g., slight misnaming of folder), but main components implemented correctly. Screenshots are mostly provided.	Basic setup done. One or two missing elements or incomplete screenshots.	The setup attempted but contains significant errors or lacks key elements.	Setup not attempted or incorrectly configured. No deliverables submitted.
	1.2 AWS Glue Configuration	The glue DB and crawler are correctly configured with automation options, and partitions reflect the new snapshot. Well-explained screenshots.	The glue setup is mostly correct, but some minor configuration or documentation issues.	Crawler and database exist but lack full automation or partition updates.	Setup incomplete or incorrect (e.g., crawler not rerun, missing schema updates).	Not submitted or major setup flaws.
	1.3 Real-Time Notification Using AWS Lambda and SNS **(30%)**	Demonstrates full completion with a correctly created SNS topic and a well-scripted, deployed Lambda function that triggers on all S3 object creation events, accurately checks for .csv files, and publishes structured notifications. All required screenshots are clear, showing successful execution and notifications. Code is clean, documented, and follows best practices.	SNS topic and Lambda function are created and mostly configured correctly. The Lambda function triggers as expected but may have minor logic or formatting issues. Deliverables are mostly complete and demonstrate functionality, though some details or documentation may be lacking	SNS topic and Lambda function are partially implemented with noticeable errors in triggering, file extension checks, or message publishing. Some deliverables are missing or unclear. Code works but is inefficient or incomplete.	SNS topic or Lambda function is poorly implemented or largely non-functional. Key requirements such as event triggers or notifications are missing. Deliverables are insufficient or irrelevant.	Little to no evidence of SNS topic creation or Lambda deployment. Deliverables are absent or unrelated, showing no understanding of the task requirements.
Task 2: **Query and Visualize the Dataset Using Athena and QuickSight.** **(LO1)**	**2.1** Amazon Athena Queries	All queries were correctly written and executed. Log bucket configured. Screenshots show valid results. Queries demonstrate understanding of the dataset	Queries mostly correct. Log setup done. Minor SQL or documentation errors	Some queries correct. Incomplete logs/screenshots. SQL lacks clarity.	Queries attempted but mostly incorrect. Incomplete or unclear screenshots.	Athena setup or queries are not submitted or are fully incorrect.
	2.2 Amazon QuickSight Dashboards **(20%)**	QuickSight is fully configured. Charts are accurate, well-labeled, and reflect insights. S3 access resolved.	The charts are mostly correct. Some visualization errors, but the setup is sound.	Charts are basic and may lack clear insights. The setup is incomplete or incorrect.	QuickSight attempted, but the charts do not reflect the proper use of data.	QuickSight has not encountered or critical setup errors.

Criterion		Evidence				
Task and Weightage		**A (80-100%)**	**B (65-79%)**	**C (50-64%)**	**D (40-49%)**	**E (0-39%**
Task 3: **Multi-Cloud Integration (GitHub ➜ AWS ➜ Azure Synapse Analytics)** **(LO1)**	**3.1** Data Export from AWS S3 **3.2:** Upload CSV files from Azure Portal **3.3:** Ingest Data into Synapse **3.4:** Run SQL queries in Synapse **(30%)**	CSV files exported correctly from Athena, renamed clearly (e.g., orders_jan2017.csv) and well-organized. Files uploaded to Blob Storage with proper structure and naming; container hierarchy is logical. Linked Service configured; data ingested into Synapse table using Copy Data Tool or Data Flow successfully. SQL queries run correctly with valid syntax and accurate results. All screenshots of editor and output provided.	CSVs downloaded with minor naming or organizational inconsistencies. Files uploaded but container structure could be improved. Linked Service created; minor issues with mapping or ingestion but corrected. SQL queries mostly correct; minor syntax or result accuracy issues. Most screenshots included.	CSVs exported but not clearly renamed or inconsistently organized. Upload to Blob completed but filenames or structure unclear. Linked Service attempted but ingestion partially failed or table not usable. SQL queries attempted but with syntax or logical errors. Output not fully interpretable. Partial screenshots provide	CSVs missing, incorrect, or poorly named. Export process unclear Upload process flawed (wrong files, missing uploads, incorrect location). Linked Service not configured, or ingestion failed with no valid table created. SQL queries incomplete, failed to execute, or produce invalid results. Few or unclear screenshots.	No evidence of CSV export or incorrect process followed. No valid upload to Blob container or irrelevant files used. No ingestion into Synapse; Linked Service missing or misuse No valid SQL queries submitted or run. Screenshots absent or irrelevant.
Task 4: **Final Report on AWS Services and Architecture** **(LO1)**	**4.1** Report writing **(20%)**	Clear, structured report (within word limit), explaining each AWS service and integration. Well-labeled architecture diagram. Strong reflection on automation and expected outcome. All technical terms and AWS concepts are used appropriately and accurately throughout. Clear formatting, consistent naming, complete documentation, and screenshots. Highly professional.	The report covers all sections, minor issues with structure or clarity. Diagram included but may lack detail. Minor technical inaccuracies. Terminology was mostly correct. Mostly well-presented. Some minor formatting or labelling issues.	Basic report submitted. Addresses the main sections but lacks depth or misses diagram clarity. Some technical confusion, but key ideas are present. Basic formatting. Screenshots present but may be unorganized.	Incomplete or vague report. Poor structure, unclear reflection. Weak or missing diagram. Frequent misuse of technical terms. Limited understanding shown. Poor formatting, unclear documentation. Hard to follow.	No report submitted, or the report lacks relevance to the task or learning outcome. Major inaccuracies or lack of technical understanding. Unprofessional or no formatting. Missing documentation.