

Banking Data Platform Project: End-to-End Enterprise Solution

1. Project Overview

1.1 Title

Banking Data Platform Project using Microsoft Azure : End-to-End Enterprise Solution

1.2 Objective

The main goal of this project is to design and implement an end-to-end data engineering solution for a bank using Azure Cloud.

The platform should:

- Ingest ATM and UPI transactions, and other banking data in real-time and batch.
- Store data safely in a data lake and Cosmos DB.
- Detect high-value and suspicious transactions and raise fraud alerts.
- Build a data warehouse for analytics and reporting.
- Provide Power BI dashboards for business users.
- Ensure security using network rules and firewalls.
- Automate deployments using CI/CD pipelines.

1.3 High-Level Workflow

1. Raw banking data (ATM, UPI, customer, etc.) lands into Azure Blob Storage / ADLS.
2. Event Grid + Azure Functions react when new files arrive or when events occur.
3. Data is validated, transformed and written to:
 - Cosmos DB (operational store + fraud alerts)
 - ADLS Bronze, Silver, Gold layers (Delta format via Databricks)
4. From Gold layer, data is loaded into Azure SQL Database as dimensional model (facts & dimensions).
5. Timer-trigger Functions keep dim tables updated.
6. Event-driven fraud logic pushes alerts to Cosmos DB and notifies via Service Bus.
7. Power BI connects to the SQL warehouse for Customer 360 and other dashboards.
8. Network security and CI/CD pipelines ensure production readiness.

2. Azure Resource Setup

2.1 Resource Group

- A dedicated Resource Group was created to keep all banking-related resources together.
- This makes it easy to manage, monitor and clean up everything for the project.

2.2 Storage Account – Raw & Data Lake

Purpose: This is the landing zone for all incoming data and also the data lake for analytics.

You created an Azure Storage Account (with hierarchical namespace enabled for ADLS Gen2) and defined multiple containers, for example:

- raw – for original source files (CSV, JSON etc.)
 - raw/atm/ – ATM transaction CSV files
 - raw/upi/ – UPI transaction CSV files
 - raw/customers/ – customer master data
- bronze – raw but structured copies for analytics
- silver – cleaned and standardized data
- gold – final analytical tables (facts/dimensions) in Delta format

Why this step matters:

- Centralized source-of-truth for all input data.
- Separation of raw vs processed data (Bronze/Silver/Gold).
- Used later by Databricks for ETL.

Name	Last modified	Anonymous access level	Lease state
\$logs	12/7/2025, 9:08:21 PM	Private	Available
azure-webjobs-hosts	12/7/2025, 9:44:56 PM	Private	Available
azure-webjobs-secrets	12/7/2025, 9:44:52 PM	Private	Available
bronze	12/8/2025, 1:09:55 AM	Private	Available
gold	12/8/2025, 1:10:17 AM	Private	Available
raw	12/7/2025, 9:08:50 PM	Private	Available
scm-releases	12/7/2025, 9:38:51 PM	Private	Available
silver	12/8/2025, 1:10:09 AM	Private	Available

2.3 Service Bus Namespace & Queues

Purpose: Service Bus acts as a message broker between components to decouple ingestion and processing.

You created:

- **A Service Bus Namespace.**
- A queue, e.g., ingestion-queue, used to send messages whenever a new file or transaction needs processing.

Usage in the project:

- The Event Grid-trigger Function sends a message into ingestion-queue whenever a new transaction file is created.
- The Queue-trigger Function listens to that queue, reads the blob, processes rows, and writes to Cosmos DB & ADLS.

The screenshot shows the Microsoft Azure portal with the URL <https://portal.azure.com/#@infosynergicsoftware.onmicrosoft.com/resource/subscriptions/337f2b3a-68b6-4a2e-befd-01a13f20c1d0/resourceGroups/mazenet-databricks-02/providers/Microsoft.ServiceBus/namespaces/chandrashekarservicebus>. The page title is "chandrashekarservicebus | Queues". The left sidebar shows the navigation path: Home > chandrashekarservicebus. The main content area displays a table of queues with one entry:

Name	Status	Message count	Active messages	Dead-letter messages	Scheduled messages	Max size	Enable partitioning
ingestion-queue	Active	0	0	0	0	1024 MB	false

2.4 Application Insights

Purpose: To monitor the Function App – requests, failures, performance, logs.

You enabled Application Insights for the Function App, which allows:

- Viewing server request count.
- Monitoring response time and failures.
- Filtering logs based on function name (e.g., handleEventGrid, QueueProcessor).
- Debugging runtime behaviour without logging into the VM.

The screenshot shows the Microsoft Azure portal with the URL <https://portal.azure.com/#@infosynergycosmosdb.onmicrosoft.com/resource/subscriptions/337f2b3a-68b6-4a2e-befd-01a13f20c1d0/resourceGroups/mazennet-databricks-02/providers/Microsoft.ServiceBus/namespaces/chandrashekarservicebus>. The page title is "chandrashekarservicebus | Queues". The left sidebar shows "Overview", "Activity log", "Access control (IAM)", "Tags", "Diagnose and solve problems", "Resource visualizer", "Settings" (with "Shared access policies", "Geo-Recovery", "Networking", "Encryption", "Configuration", "Properties", "Locks"), and "Entities" (selected, with "Queues" highlighted). The main content area displays a table of queues:

Name	Status	Message count	Active messages	Dead-letter messages	Scheduled messages	Max size	Enable partitioning
ingestion-queue	Active	0	0	0	0	1024 MB	false

2.5 Azure Cosmos DB – Operational Store

Purpose: To store real-time transactional and fraud data in a scalable NoSQL database.

Created:

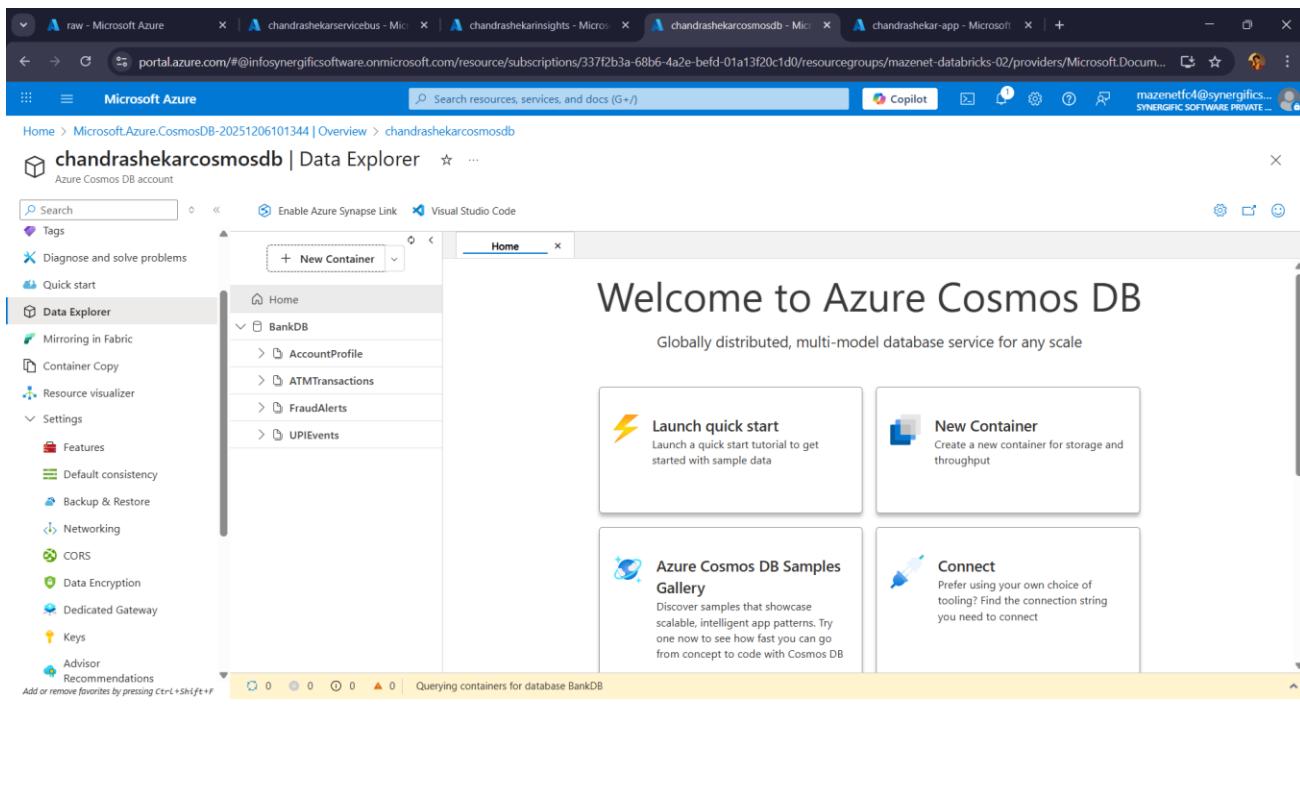
- Cosmos DB Account (e.g., azurebankcosmosdb)
- Database: BankDB
- Containers:
 - ATMTransactions – per-transaction records from ATM files
 - UPIEvents – UPI transfer events
 - FraudAlerts – each fraud alert raised for flagged transactions

Each container has:

- A partition key (for scalability), e.g.:
 - /txn_type for ATM/UPI transactions
 - /alertType for FraudAlerts (e.g., "High-value transaction", "Unusual UPI Transfer")

Why this is important:

- Supports fast reads for operational apps and fraud analytics.
- Works great with JSON structure produced by Functions.
- Serves as a source for Databricks to ingest data into the lake.



2.6 Configuration Using Application Settings

Instead of hard-coding secrets and connection strings, you used Application Settings in the Function App:

Examples of settings (names only, not values):

- AzureWebJobsStorage
- SERVICE_BUS_CONN
- SERVICE_BUS_QUEUE_NAME
- COSMOS_CONN_STRING
- COSMOS_DB_NAME
- COSMOS_ATM_CONTAINER
- COSMOS_UPI_CONTAINER
- COSMOS_ALERTS_CONTAINER

Benefits:

- Secrets are not inside the code.
- Allows smooth CI/CD deployment across environments (dev/test/prod).
- Easy rotation of keys without redeploying code.

Name	Value	Deployment slot setting	Type	Source	Delete
AzureWebJobsStorage	Show value		Custom	App Service	
COSMOS_ALERTS_CONTAINER	Show value		Custom	App Service	
COSMOS_ATM_CONTAINER	Show value		Custom	App Service	
COSMOS_CONN_STRING	Show value		Custom	App Service	
COSMOS_DB_NAME	Show value		Custom	App Service	
COSMOS_UPL_CONTAINER	Show value		Custom	App Service	
SERVICE_BUS_CONN	Show value		Custom	App Service	
SERVICE_BUS_QUEUE_NAME	Show value		Custom	App Service	

3. Azure Functions – Ingestion & Fraud Detection

3.1 Function 1 – Event Grid / HTTP Trigger (Ingestion Orchestrator)

What it does:

- Gets triggered when:
 - A new blob is created (Event Grid trigger), or
 - A manual API call is made (HTTP trigger used with Postman for testing).
- Extracts blob URL and metadata from the event.
- Constructs a message with details such as:
 - Blob URL (where the file is stored)
 - Type of file (ATM/UPI)
- Sends that message into the Service Bus queue (ingestion-queue).

3.2 Function 2 – Service Bus Queue Trigger (Main Processor)

What it does:

- Listens to ingestion-queue.
- For each message:
 1. Reads the blob URL.
 2. Downloads the corresponding CSV file from raw container.
 3. Identifies whether it is an ATM file or a UPI file.

4. Converts each row into a transaction document with:

- Transaction ID
- Customer ID
- Account Number
- Amount
- Type (ATM/UPI)
- Time and Location

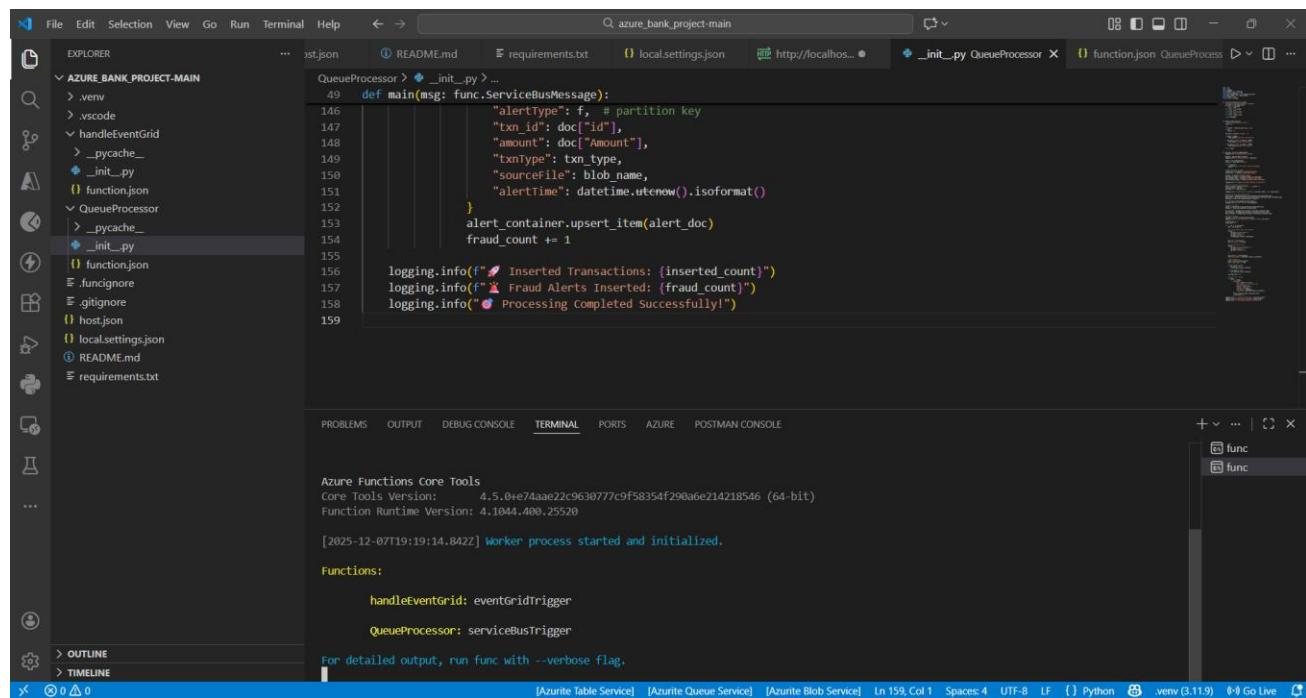
5. Inserts:

- Normalized transaction data into Cosmos DB (ATMTransactions / UPIEvents).
- High-risk items also into FraudAlerts` container.

Fraud detection logic used here:

- If amount > ₹50,000 → mark as High-value transaction.
- If ATM withdrawal above certain threshold → Large ATM withdrawal.
- If UPI transfer above threshold → Unusual UPI transfer.

These flagged records are saved as separate documents under FraudAlerts.



The screenshot shows the Azure Functions Core Tools interface within VS Code. The Explorer sidebar on the left lists the project structure, including files like .venv, handleEventGrid, QueueProcessor, and host.json. The main editor area displays the Python code for the QueueProcessor function. The code defines a main function that processes Service Bus messages, extracts transaction details, and inserts them into a database. It also handles logging for inserted transactions and fraud alerts. The bottom status bar indicates the worker process has started and initialized, and the terminal shows the Azure Functions Core Tools version information.

```
def main(msg: func.ServiceBusMessage):
    alert_type = f # partition key
    tx_id = doc["id"]
    amount = doc["Amount"]
    tx_type = txn_type
    sourcefile = blob_name
    alert_time = datetime.utcnow().isoformat()
}
alert_container.upsert_item(alert_doc)
fraud_count += 1

logging.info(f"Inserted Transactions: {inserted_count}")
logging.info(f"Fraud Alerts Inserted: {fraud_count}")
logging.info("Processing Completed Successfully!")
```

3.3 Testing with Postman

To validate the flow end-to-end without waiting for real files, you:

1. Used the Function's HTTP endpoint.

2. Sent a JSON payload with a sample blob_url.
3. Confirmed:
 - o HTTP response 202 Accepted.
 - o In Function logs: Blob processed → records inserted → alerts inserted.
 - o In Cosmos DB: new documents present in containers.

The screenshot shows the Azure Functions developer tools interface. The left sidebar displays the project structure for 'AZURE_BANK_PROJECT-MAIN' with files like host.json, requirements.txt, local.settings.json, and function.json. The main area shows a POST request to 'http://localhost:7081/runtime/webhooks/eventgrid?functionName=handleEventGrid'. The request body contains a JSON payload:

```

    {
        "eventType": "Microsoft.Storage.BlobCreated",
        "subject": "",
        "eventTime": "2025-01-01T10:00:00Z",
        "data": {
            "url": "https://azurereadystorage01.blob.core.windows.net/raw/atm/atm_transaction_records.csv"
        },
        "dataVersion": "1.0"
    }
  
```

The response status is 202 Accepted. Below the request, the function's logs are displayed, showing various X-MS headers and the insertion of a document into the database:

```

  'x-ms-resource-quota': 'documentsSize=51200;documentsSize=52428800;documentsCount=-1;collectionSize=52428800',
  'x-ms-resource-usage': 'documentsSize=3;documentsSize=2458;documentsCount=3337;collectionSize=3338',
  'x-ms-schemaversion': '1.29',
  'lsn': '3339',
  'x-ms-request-charge': '9.9',
  'x-ms-alt-content-path': 'dbs/bank00/colls/ATMTransactions',
  'x-ms-content-path': 'jklh@Wifgocw='
  ...
  'x-ms-current-write-quorum': '3',
  'x-ms-current-replica-set-size': '4',
  'x-ms-documentdb-partitionkeyrangeid': '0',
  'x-ms-role': '1',
  'x-ms-global-committed-lsn': '3338',
  'x-ms-number-of-read-regions': '0',
  'x-ms-transport-request-id': '3339',
  'x-ms-cosmos-lsn': '3339'
  
```

3.4 Deploy the Functions into Function App

All the deployed functions are shown in the Function App.

These Functions are Triggered when the File is inserted in to the Blob Storage

The screenshot shows the Azure Functions developer tools interface. The left sidebar displays the project structure for 'AZURE' and 'RESOURCES' (Remote). The main area shows the code for the 'QueueProcessor' function in 'function.json' and 'init_.py'. The code handles a ServiceBusMessage and inserts transactions into a database:

```

  def main(msg: func.ServiceBusMessage):
      alert_type = f # partition key
      txnid = doc["id"],
      amount = doc["Amount"],
      txntype = txntype,
      sourcefile = blob_name,
      alerttime = datetime.utcnow().isoformat()
      alert_container.upsert_item(alert_doc)
      fraud_count += 1
  
```

The 'PROBLEMS' tab shows deployment status: 'Deploy to app "azurereadyapp" Succeeded in 26s', 'Zip and deploy workspace "c:\Users\chand\azureready_bank_project-main" 2s', and 'Build app "azurereadyapp" in Azure 0s'.

3.5 Testing the functions in the cloud

To Test the function in the cloud:

Upload the CSV into the ATM Container

It will Trigger the function by using the Event Grid Trigger With the Help of the Event Subscription

It can also display the Logs and Messages in the Application Insights and Service Bus

The Trigger function data will be stored in the Cosmos DB Containers

Name	Last modified	Access tier	Blob type	Size	Lease state
atm_transactions_records.csv	12/6/2025, 12:03:10 PM	Hot (Inferred)	Block blob	872.43 KB	Available

Microsoft Azure | portal.azure.com/#view/Microsoft_OperationsManagementSuite/Workspace/LogsReactView/query/requests%20%7C%20extend%20functionNameFromCustomDimension%20%3D%20tostrin... | Copilot | Search resources, services, and docs (G+) | +

Home > Function App > chandrashekhar-app > EventGridTrigger | Invocations >

Logs chandrashekharinsights

New Query 1+ Time range: Set in query Show: 500000 results Add Simple mode

Results Chart

timestamp [UTC]	id	operation_Name	success	resultCode	duration	operation_Id	cloud_RoleName	invocationId
> 6/12/2025, 6:48:01.335 am	40b94fc15de4fd04	EventGridTrigger	True	0	3.9475	50f926b485189c1ac495fbbae447bb9	chandrashekhar-app	645be19-1b44-4832-8553-15eb2bd669ff
> 6/12/2025, 6:48:00.637 am	b5c1e063b59758bd	EventGridTrigger	True	0	9.7498	9000ea2f153103d774f2a0a26405e92f	chandrashekhar-app	8e3c2855-76cb-4a7f-98fe-90228e24131b
> 6/12/2025, 6:46:01.093 am	977fe8710cb3de6e6	EventGridTrigger	True	0	6.4161	e2df415f9b303a9771b886474dfdbbc2	chandrashekhar-app	7107937a-d305-4229-b1b4-0ca5a5a434a7
> 6/12/2025, 6:46:00.623 am	bbdb119280740ab65	EventGridTrigger	True	0	6.9963	977759546bd2aba8cca168723ba347ad	chandrashekhar-app	7c798041-3ff4-c456-9287-8e346101c11c
> 6/12/2025, 6:44:01.098 am	5763124c619d4667	EventGridTrigger	True	0	8.9734	d2bd04175906b9af5f888bb2acc55a7	chandrashekhar-app	f66e5071-f384-4fbc-9061-5133996e0edb
> 6/12/2025, 6:44:00.589 am	315ea01ce7fbfd6c	EventGridTrigger	True	0	11.9588	dc465fc40412309c0bc6fe64a7d1f	chandrashekhar-app	201c6cbff-a3cf-4b76-8720-e466461f7255
> 6/12/2025, 6:42:01.146 am	ad03a7abc4aceee9	EventGridTrigger	True	0	3.5597	9467117252463f3881d81bc506b2a6e6	chandrashekhar-app	fffd1a3d-c314-4913-bcca-d7463401c5b6
> 6/12/2025, 6:42:00.589 am	3e77fd7d854de109	EventGridTrigger	True	0	4.1517	ab0e50e9875373d7e696c96661e131	chandrashekhar-app	d7f20590-0cfb-4743-9836-9061-5133996e0edb
> 6/12/2025, 6:40:01.076 am	d85b5ac05f59a4f5	EventGridTrigger	True	0	9.5571	15233f15a863c0ee5f820f04e3c311f	chandrashekhar-app	4c4560a4-9968-4dfe-b3ef-32a7c5e897c
> 6/12/2025, 6:40:00.564 am	505ea2d4b0b62a6f	EventGridTrigger	True	0	8.1007	98cf0761c9484f142d56abad7c4618	chandrashekhar-app	2c256cc6-08d7-4d81-9e42-117325c188d9
> 6/12/2025, 6:38:01.050 am	e948ef907fdb23aa3	EventGridTrigger	True	0	8.8646	e35871c050a6b9af0004e197a6b2ef	chandrashekhar-app	88ae9971-fa32-46e4-a129-15ca7c6a7006
> 6/12/2025, 6:38:00.616 am	6cd8b089b64d6a88	EventGridTrigger	True	0	9.3761	02e75631ebf2d647993ca70ff7a2	chandrashekhar-app	981404dc-5fbf-465a-a5aa-9ae4da6760da
> 6/12/2025, 6:36:01.188 am	cbf1ab3f2d7756bf	EventGridTrigger	True	0	3.7141	2fa6ead36699fd6879ed31dce048d071	chandrashekhar-app	f1e723d7-7600-4def-959f-8e96d0c896fb

1s 414ms | Display time (UTC+00:00) ▾

Query details | 1 - 13 of 20

Microsoft Azure | portal.azure.com/#view/WebsitesExtension/FunctionTabMenuBlade/~/invocations/resourceId%2Fsubscriptions%2F3372b3a-68b6-4a2e-befd-01a13f20c1d0%2FresourceGroups%2Fmazene... | Copilot | Search resources, services, and docs (G+) | +

Home > Function App > chandrashekhar-app > EventGridTrigger

EventGridTrigger | Invocations

chandrashekhar-app

Code + Test Integration Function Keys **Invocations** Logs Metrics

Open in Application Insights Refresh Send us your feedback

Query
Up to 20 of the most recent function invocation traces. For more advanced analysis, run the query in Application Insights.

Success count	Error count
59	80
Last 30 days	Last 30 days

Search

Date	Status	Result Code	Duration (ms)	Operation ID
12/6/2025, 12:18:01 PM	Success	0	4	50f926b485189c1ac495fbbae447bb9
12/6/2025, 12:18:00 PM	Success	0	10	9000ea2f153103d774f2a0a26405e92f
12/6/2025, 12:16:01 PM	Success	0	6	e2df415f9b303a9771b886474dfdbbc2
12/6/2025, 12:16:00 PM	Success	0	7	9f7759546bd2aba8cca168723ba347ad
12/6/2025, 12:14:01 PM	Success	0	9	d2bd04175906b9af5f888bb2acc55a7
12/6/2025, 12:14:00 PM	Success	0	12	dc465fc40412309c0bc6fe64a7d1f

Microsoft Azure | portal.azure.com/#@kluniversity.in/resource/subscriptions/4fb3a09a-9941-4890-9642-6a50016aa912/resourceGroups/Azure-bank-RG/providers/Microsoft.ServiceBus/namespaces/azurebank... | Copilot | Search resources, services, and docs (G+) | +

Home > azurebankservicebus | Queues >

ingestion-queue (azurebankservicebus/ingestion-queue)

Service Bus Queue

Overview Access control (IAM) Diagnose and solve problems Service Bus Explorer Resource visualizer Settings Automation Help

Refresh Delete Give feedback I U (change) 2.4 KB I GB (change) 14 DAYS (changed) NEVER (change) I MINUTE (change) I 100,0 % (add metadata)

Message Counts Active 1 MESSAGES Scheduled 0 MESSAGES Dead-letter 3 MESSAGES Transfer 0 MESSAGES Transfer dead-letter 0 MESSAGES

Metrics Show data for the last: 1 hour 6 hours 12 hours 1 day 7 days 30 days

Requests

Incoming Requests (Sum), azurebankservicebus : 73
Successful Requests (Sum), azurebankservicebus : 73
Server Errors. (Sum), azurebankservicebus : 0

Messages

Incoming Messages (Sum), azurebankservicebus : 4
Outgoing Messages (Sum), azurebankservicebus : 31

Add or remove favorites by pressing **ctrl + shift + F**

Raw - Microsoft Azure ingestion-queue (azurebankservicebus/ingestion-queue) | Service Bus Explorer

Home > azurebankservicebus | Queues > ingestion-queue (azurebankservicebus/ingestion-queue)

Microsoft Azure portal.azure.com/#@kluniversity.in/resource/subscriptions/4fb3a09a-9941-4890-9642-6a50016aa912/resourceGroups/Azure-bank-RG/providers/Microsoft.ServiceBus/namespaces/azurebank... Copilot Search resources, services, and docs (G+) 2100032603@kluniversity.kluniversity.in

Service Bus Queue

Search Queue Mode Send messages Refresh Export messages Show message body Settings Learn more Give feedback

Queue (5) Dead-letter (9)

Peek from start Peek next messages Peek with options Re-send selected messages Download selected message body

Showing 5 of 5 messages

Sequence Number	Message ID	Enqueued Time	Delivery Count	State	Body ...	Label/Subject	Message Text
9	887060d4eee14176bfce548148a0...	Sun, Dec 07, 25, 11:55:31 PM GM...	9	Active	106 B		{"blob_url": "https://azurebankst...
10	a6ba3711b58d44f3ab340f2fd43...	Sun, Dec 07, 25, 11:59:41 PM GM...	8	Active	106 B		{"blob_url": "https://azurebankst...
11	05fce3a9a8e949e79274162abd4...	Mon, Dec 08, 25, 12:06:13 AM G...	7	Active	106 B		{"blob_url": "https://azurebankst...
12	c64c92081059456c940b7a577187...	Mon, Dec 08, 25, 12:24:50 AM G...	3	Active	106 B		{"blob_url": "https://azurebankst...
14	92946fab9899442193f5daea6226...	Mon, Dec 08, 25, 12:30:04 AM G...	2	Active	97 B		{"blob_url": "https://azurebankst...

Message Body Message Properties

Select a message to see its details.

Add or remove favorites by pressing Ctrl+Shift+F

Raw - Microsoft Azure ingestion-queue (azurebankservicebus/ingestion-queue) | Service Bus Explorer

Home > azurebankservicebus | Queues > ingestion-queue (azurebankservicebus/ingestion-queue)

Microsoft Azure portal.azure.com/#@kluniversity.in/resource/subscriptions/4fb3a09a-9941-4890-9642-6a50016aa912/resourceGroups/Azure-bank-RG/providers/Microsoft.ServiceBus/namespaces/azurebank... Copilot Search resources, services, and docs (G+) 2100032603@kluniversity.kluniversity.in

Service Bus Queue

Search Queue Mode Send messages Refresh Export messages Show message body Settings Learn more Give feedback

Queue (5) Dead-letter (9)

Peek from start Peek next messages Peek with options Re-send selected messages Download selected message body

Showing 5 of 5 messages

Sequence Number	Message ID	Enqueued Time	Delivery Count	State	Body ...	Label/Subject	Message Text
9	887060d4eee14176bfce548148a0...	Sun, Dec 07, 25, 11:55:31 PM GM...	9	Active	106 B		{"blob_url": "https://azurebankst...
10	a6ba3711b58d44f3ab340f2fd43...	Sun, Dec 07, 25, 11:59:41 PM GM...	8	Active	106 B		{"blob_url": "https://azurebankst...
11	05fce3a9a8e949e79274162abd4...	Mon, Dec 08, 25, 12:06:13 AM G...	7	Active	106 B		{"blob_url": "https://azurebankst...
12	c64c92081059456c940b7a577187...	Mon, Dec 08, 25, 12:24:50 AM G...	3	Active	106 B		{"blob_url": "https://azurebankst...
14	92946fab9899442193f5daea6226...	Mon, Dec 08, 25, 12:30:04 AM G...	2	Active	97 B		{"blob_url": "https://azurebankst...

Message Body Message Properties

Fit message body Off

```
{  
  "blob_url": "https://azurebankstorage01.blob.core.windows.net/raw/atm/atm_transaction_records.csv"  
}
```

Add or remove favorites by pressing Ctrl+Shift+F

Raw - Microsoft Azure ingestion-queue (azurebankservicebus/ingestion-queue) | Service Bus Explorer

Home > azurebankservicebus | Queues > ingestion-queue (azurebankservicebus/ingestion-queue)

Microsoft Azure portal.azure.com/#@kluniversity.in/resource/subscriptions/4fb3a09a-9941-4890-9642-6a50016aa912/resourceGroups/Azure-bank-RG/providers/Microsoft.ServiceBus/namespaces/azurebank... Copilot Search resources, services, and docs (G+) 2100032603@kluniversity.kluniversity.in

Service Bus Queue

Search Queue Mode Send messages Refresh Export messages Show message body Settings Learn more Give feedback

Queue (5) Dead-letter (9)

Peek from start Peek next messages Peek with options Re-send selected messages Download selected message body

Showing 5 of 5 messages

Sequence Number	Message ID	Enqueued Time	Delivery Count	State	Body ...	Label/Subject	Message Text
9	887060d4eee14176bfce548148a0...	Sun, Dec 07, 25, 11:55:31 PM GM...	9	Active	106 B		{"blob_url": "https://azurebankst...
10	a6ba3711b58d44f3ab340f2fd43...	Sun, Dec 07, 25, 11:59:41 PM GM...	8	Active	106 B		{"blob_url": "https://azurebankst...
11	05fce3a9a8e949e79274162abd4...	Mon, Dec 08, 25, 12:06:13 AM G...	7	Active	106 B		{"blob_url": "https://azurebankst...
12	c64c92081059456c940b7a577187...	Mon, Dec 08, 25, 12:24:50 AM G...	3	Active	106 B		{"blob_url": "https://azurebankst...
14	92946fab9899442193f5daea6226...	Mon, Dec 08, 25, 12:30:04 AM G...	2	Active	97 B		{"blob_url": "https://azurebankst...

Message Body Message Properties

Fit message body Off

```
{  
  "blob_url": "https://azurebankstorage01.blob.core.windows.net/raw/atm/atm_transaction_records.csv"  
}
```

Add or remove favorites by pressing Ctrl+Shift+F

The screenshot shows the Azure Service Bus Explorer interface for an 'ingestion-queue' under the 'azurebankservicebus' namespace. The left sidebar has 'Service Bus Explorer' selected. The main area shows a table of 5 messages in the queue, each with a checkbox, sequence number, message ID, enqueue time, delivery count, state, body size, label, and message text. Message 14 is selected. At the bottom, there are tabs for 'Message Body' and 'Message Properties', with the message body content being a JSON object with a single key-value pair: 'blob_url': 'https://azur...'. The top navigation bar shows the full URL of the service bus endpoint.

Sequence Number	Message ID	Enqueued Time	Delivery Count	State	Body ...	Label/Subject	Message Text
9	887060d4eef14176bfce548148a...	Sun, Dec 07, 25, 11:55:31 PM GM...	9	Active	106 B		{ "blob_url": "https://azur..."}
10	a68a3711b58d44f3ab340f21da43...	Sun, Dec 07, 25, 11:59:41 PM GM...	8	Active	106 B		{ "blob_url": "https://azur..."}
11	05fce3a9a8e049e79274162abd4...	Mon, Dec 08, 25, 12:06:13 AM G...	7	Active	106 B		{ "blob_url": "https://azur..."}
12	c64c92081059456c940b7a57718...	Mon, Dec 08, 25, 12:24:50 AM G...	3	Active	106 B		{ "blob_url": "https://azur..."}
14	92946fab9899442193f5dae6226...	Mon, Dec 08, 25, 12:30:04 AM G...	2	Active	97 B		{ "blob_url": "https://azur..."}

The screenshot shows the Microsoft Azure Data Explorer interface for the azurebankcosmosdb database. The left sidebar navigation includes Home, Overview, Activity log, Access control (IAM), Tags, Diagnose and solve problems, Quick start, Data Explorer (selected), Mirroring in Fabric, Container Copy, Resource visualizer, Settings, Features, Default consistency, Backup & Restore, Networking, CORS, and Data Encryption. The main area displays a table titled 'ATMTr_items' with columns: id, ... /txn_typ ..., and ATM. A dropdown menu above the table lists 'New Item', 'Update', 'Discard', 'Delete', and 'Upload Item'. The table contains 28 rows of ATM transaction data, with the first few rows shown below:

	Type	Value
1	id	"8344af7c-dd20-4757-a705-3b...
2	...	/txn_typ ...
3	ATM	"ATM"
4	...	
5	AccountNumber	"1002003183"
6	CardID	"jk1hAWMfqcyT427"
7	TransactionAmount	"8000"
8	TransactionType	"WITHDRAWAL"
9	TransactionTime	"2025-01-01T02:31:00Z"
10	Location	"Chennai"
11	Status	"SUCCESS"
12	txId	"344af7c-dd20-4757-a705-3bcadfb04087"
13	suspicious_flag	[]
14	is_suspicious	false
15	_rid	"jk1hAWMfqcyYAAAAAAA=="
16	_self	"obs/jk1hAWMfqcyYAAAAAAA=="
17	_etag	"\"0000c900-0000-5900-0000-6935b2440000\""
18	_attachments	"attachments/";
19	_ts	"1765126724"
20		

```

SELECT * FROM c
    Type a query predicate (e.g., WHERE cid='1'), or choose one from the drop down list, or leave empty to query all documents.
    ↓
    Apply Filter
    1 {
    2     "id": "ATM000005_High-value transaction",
    3     "alertType": "High-value transaction",
    4     "txId": "ATM000005",
    5     "amount": 60000,
    6     "txType": "ATM",
    7     "sourceFile": "atm/atm_transaction_records.csv",
    8     "alertTime": "2025-12-07T18:29:56.460753",
    9     "_rid": "jk1hALUHT3EBAAAAAAAA--",
   10    "_self": "https://jk1hALUHT3EBAAAAAAAA-/",
   11    "_etag": "\\"0e0049bd-0000-0000-6935c7a40000\\\"",
   12    "_attachments": "attachments/",
   13    "_ts": 1765132196
   14 }

```

4. Databricks & Data Lake – Bronze, Silver, Gold

4.1 Databricks Workspace & Cluster

You created an Azure Databricks workspace, and inside it:

- A cluster with appropriate Spark version.
- Notebooks to perform data engineering steps.

Setting	Value
Status	: Active
Resource group	: Azure-bank-RG
Location	: East Asia
Subscription	: Azure for Students
Subscription ID	: f4b3a09a-9941-4890-9642-6a50016aa912
Tags (edit)	: Add tags

[Launch Workspace](#) [Upgrade to Premium](#)

4.2 Secure Access to ADLS (Mounting)

To make ADLS paths easy to use in Spark, you:

- Configured access either using:
 - Account key or
 - Secret scope (Databricks secrets) and dbutils.secrets.get.
- Mounted containers like bronze, silver, gold to convenient mount points:
 - /mnt/bronze
 - /mnt/silver
 - /mnt/gold

This allows you to read/write using standard paths like:

- /mnt/bronze/ATMTransactions
- /mnt/silver/UPIEvents

4.3 Ingesting from Cosmos DB into Bronze

Because the direct Cosmos Spark connector was not compatible with your Spark version, you used an alternative:

- Azure Cosmos DB Python SDK inside Databricks.
- For each container (ATMTransactions, UPIEvents, FraudAlerts):
 - Connected using endpoint + key.

- Read all items from the container into a Python list.
- Loaded into a Pandas DataFrame, then converted it into a Spark DataFrame.

Then, for each Spark DataFrame:

- Wrote data into the Bronze layer in Delta format, for example:
 - /mnt/bronze/ATMTransactions
 - /mnt/bronze/UPIEvents
 - /mnt/bronze/FraudAlerts

The screenshot shows the Databricks workspace interface. On the left is the sidebar with various notebooks and clusters. The main area shows a notebook titled "Azure-Bank". A message box at the top says "FraudAlerts written to Bronze successfully!". Below it, a code cell displays the command `display(dbutils.fs.ls("/mnt/bronze"))` and its output, which is a table showing three entries: ATMTransactions, FraudAlerts, and UPIEvents, all with size 0 and modification time 0. The table has columns: path, name, size, and modificationTime. The notebook also shows other cells numbered 35 and 36.

The screenshot shows the Microsoft Azure Storage Explorer interface. It's navigating through storage accounts and containers. The current view is of a container named "bronze". The left sidebar shows options like Overview, Diagnose and solve problems, Access Control (IAM), and Settings. The main area lists blobs with the prefix "bronze". There are four items: ATMTransactions, FraudAlerts, UPIEvents, and _\$azurertmpfolder\$. Each item has a "Name" column, "Last modified" column, "Access tier" column, "Blob type" column, "Size" column, and an ellipsis column for more options. A search bar at the top allows filtering by prefix.

4.4 Silver Layer – Cleaning & Normalization

Silver is the clean and business-ready layer.

For each data source:

4.4.1 Silver – ATM Transactions

On top of bronze ATM data, you:

- Selected only relevant fields: TransactionID, CustomerID, AccountNumber, Amount, TransactionType, TransactionTime, Location, Status, ATMID.
- Standardised column names to common naming like transaction_id, customer_id, amount, txn_type, etc.
- Converted amount to a numeric type (e.g., double).
- Converted txn_timestamp to a proper timestamp.
- Dropped Cosmos-specific metadata fields (_rid, _etag, etc.).

Result: A clean, consistent ATM transactions table in Silver, stored as Delta under:

- /mnt/silver/ATMTransactions

4.4.2 Silver – UPI Events

On top of bronze UPI data, you:

- Normalized fields like EventID, TxnID, CustomerID, AccountNumber, PayerUPI, PayeeUPI, Amount, TxnTimestamp, Status, DeviceID, GeoLocation.
- Renamed and standardized them to match ATM naming where possible:
 - transaction_id, customer_id, account_number, amount, txn_type, txn_timestamp, etc.
- Added a channel field with value "UPI".
- **Ensured amount and timestamps are in proper types.**

Stored at:

- /mnt/silver/UPIEvents

4.4.3 Silver – Fraud Alerts

From the bronze FraudAlerts:

- Extracted alertType, txn_id, amount, txnType, sourceFile, alertTime.
- Renamed to columns like alert_type, txn_type, alert_time.
- Converted amount to numeric, alert_time to timestamp.

Stored at:

- /mnt/silver/FraudAlerts

The screenshot shows the Databricks workspace interface. On the left, the sidebar includes sections like Workspace, Recents, Catalog, Jobs & Pipelines, Compute, Data Engineering, Job Runs, AI/ML, Playground, Experiments, Features, Models, and Serving. The main area displays a notebook titled "Fraud Silver". The code cell contains:

```
display(dbutils.fs.ls("/mnt/silver"))

|
```

Below the code, a table shows the results of the file listing:

<code>path</code>	<code>name</code>	<code>size</code>	<code>modificationTime</code>
dbfs:/mnt/silver/ATMTransactions/	ATMTransactions/	0	0
dbfs:/mnt/silver/FraudAlerts/	FraudAlerts/	0	0
dbfs:/mnt/silver/UPIEvents/	UPIEvents/	0	0

Information at the bottom of the notebook pane:

- [Shift+Enter] to run and move to next cell
- [Ctrl+Shift+P] to open the command palette
- [Esc H] to see all keyboard shortcuts

The screenshot shows the Microsoft Azure Storage Explorer interface. The URL in the browser is https://portal.azure.com/#view/Microsoft_Azure_Storage/ContainerMenuBlade/~/overview/storageAccountId/%2Fsubscriptions%2Ff14b3a09-a994-4890-9642-6a50016aa912%2FresourceGroups%2FAzur.... The storage account is "azurebankstorage01".

The container listed is "silver". The table below shows the contents of the "silver" container:

Name	Last modified	Access tier	Block type	Size	Lease state
ATMTransactions	12/8/2025, 4:28:37 PM				---
FraudAlerts	12/8/2025, 4:29:41 PM				---
UPIEvents	12/8/2025, 4:29:25 PM				---
_azurertmpfolder\$	12/8/2025, 4:28:37 PM				---

At the bottom of the page, there is a note: "Add or remove favorites by pressing Ctrl+Shift+F".

4.5 Gold Layer – Fact & Dim Tables (Analytics)

Gold is the final analytics-ready layer, designed like a star schema.

4.5.1 FactTransactions (Gold)

You combined Silver ATM and Silver UPI into a single, unified transaction fact:

- Selected common columns like:
 - transaction_id, customer_id, account_number, amount, txn_type, txn_timestamp, status, channel, location, plus channel-specific fields.
- Used union to merge ATM and UPI into one FactTransactions Spark DataFrame.
- Saved to Delta at:
 - /mnt/gold/FactTransactions

4.5.2 DimCustomer & DimAccount (Gold)

From Silver data and/or customer master files:

- DimCustomer:
 - Contains each CustomerID, basic attributes (name/segment if available), and flags like IsCurrent to support history.
- DimAccount:
 - Contains AccountNumber, linkage to Customer, account status (active/dormant), and account type (savings/current/loan).

Saved to:

- /mnt/gold/DimCustomer
- /mnt/gold/DimAccount

4.5.3 FactFraudDetection (Gold)

From Silver FraudAlerts:

- Grouped by txn_id, date, and alert types to build:
 - Fraud count per transaction
 - Total fraud amount
 - Fraud severity scores

Saved to:

- /mnt/gold/FactFraudDetection

The screenshot shows a Databricks notebook interface. On the left is a sidebar with various navigation options like Workspace, Recents, Catalog, Jobs & Pipelines, Compute, Data Engineering, Job Runs, AI/ML, Playground, Experiments, Features, Models, and Serving. The main area has tabs for 'Azure-Bronze Layer', 'Azure-Silver Layer', and 'Azure-Gold Layer'. A Python cell is active, showing the output of the command `dbutils.fs.ls("/mnt/gold")`. The output is a table with the following data:

#	path	name	size	modificationTime
1	dbfs:/mnt/gold/DimAccount/	DimAccount/	0	0
2	dbfs:/mnt/gold/DimCustomer/	DimCustomer/	0	0
3	dbfs:/mnt/gold/DimDate/	DimDate/	0	0
4	dbfs:/mnt/gold/FactFraudAlerts/	FactFraudAlerts/	0	0
5	dbfs:/mnt/gold/FactFraudDetection/	FactFraudDetection/	0	0
6	dbfs:/mnt/gold/FactTransactions/	FactTransactions/	0	0

The screenshot shows the Microsoft Azure Storage Explorer interface. At the top, there are tabs for 'Containers', 'Blobs', and 'Tables'. The 'Containers' tab is selected, showing a list of containers. One container, 'gold', is selected and expanded, showing its contents. The 'Overview' section for the 'gold' container includes buttons for 'Add Directory', 'Upload', 'Refresh', 'Delete', 'Copy', 'Paste', 'Rename', 'Acquire lease', 'Break lease', and 'Edit columns'. Below this, there is a search bar for blobs and a table listing blobs with columns: Name, Last modified, Access tier, Blob type, Size, and Lease state. The blobs listed are: '_\$azurertmpfolder\$', 'FactTransactions', 'FactFraudAlerts', 'DimCustomer', 'DimAccount', 'DimDate', and 'FactFraudDetection'. All blobs were last modified on 12/8/2025 at various times between 5:10:08 PM and 5:41:51 PM.

5. Azure SQL Database – Data Warehouse Layer

5.1 SQL Database Setup

You provisioned an Azure SQL Database to act as the data warehouse (DW).

- Example name: BankDW
- Connected using tools like Azure Portal Query Editor / SSMS / Azure Data Studio.

Azure SQL Server Overview:

- Resource group: Azure-bank-RG
- Status: Available
- Location: East Asia
- Subscription: Azure for Students
- Subscription ID: f4b3a09a-9941-4890-9642-6a50016aa912
- Tags: Add tags
- Notifications: 0
- Features: Microsoft Defender for SQL (NOT CONFIGURED), Automatic tuning (CONFIGURED), Auditing, Failover groups, Transparent data encryption

BankDWH Database Properties:

- Resource group: Azure-bank-RG
- Status: Online
- Location: East Asia
- Subscription: Azure for Students
- Subscription ID: f4b3a09a-9941-4890-9642-6a50016aa912
- Tags: Add tags
- Server name: azurebank-server.database.windows.net
- Connection strings: Show database connection strings
- Pricing tier: General Purpose - Serverless: Gen5, 1 vCore
- Auto-pause delay: 1 hour
- Earliest restore point: No restore point available

5.2 Star Schema Design

In the SQL DB, you implemented dimensional modelling:

Dimension Tables

- **DimDate** – contains each calendar date with fields like Year, Month, Day, Quarter.
- **DimCustomer** – one row per customer (possibly with history fields).
- **DimAccount** – one row per account, with status, type, open/close dates.

- **DimBranch (optional placeholder)** – for future branch-level analytics.
- **DimProduct (placeholder)** – to model loans, savings, current accounts.

Fact Tables

- **FactTransactions** – holds all ATM and UPI transactions, referencing DimCustomer, DimAccount, DimDate.
- **FactFraudDetection** – aggregates fraud information per transaction or per day.
- **FactCustomerActivity** – for logins/device usage (conceptual, optional based on data availability).

The screenshot shows the Microsoft Azure portal with the URL: portal.azure.com/#@kluniversity.in/resource/subscriptions/f4b3a09a-9941-4890-9642-6a50016aa912/resourceGroups/Azure-bank-RG/providers/Microsoft.Sql/servers/azurebank-server/databases/BankDWH. The browser title is "BankDWH (azurebank-server/BankDWH) | Query editor (preview)". The left sidebar shows "BankDWH (2100032603@kluniversity.in)" under Overview, Activity log, Tags, and Query editor (preview). The main area contains a query editor with the following code:

```

1 -- ***** DIMENSION TABLES *****
2 -- 1. DimDate
3 CREATE TABLE dbo.DimDate (
4     DateKey INT NOT NULL PRIMARY KEY,
5     FullDate DATE NOT NULL,
6     [Year] INT NOT NULL,
7     [Quarter] INT NOT NULL,
8     [Month] INT NOT NULL,
9     [Day] INT NOT NULL
10 );
11
12 -- 2. DimCustomer (SCD Type 2)
13 CREATE TABLE dbo.DimCustomer (
14     CustomerKey INT IDENTITY(1,1) PRIMARY KEY,
15     CustomerID NVARCHAR(50) NOT NULL, -- business key
16     FullName NVARCHAR(200) NULL,
17     Segment NVARCHAR(50) NULL,
18     EffectiveFrom DATETIME2 NOT NULL,
19     EffectiveTo DATETIME2 NULL,
20     IsCurrent BIT NOT NULL
21 );
22

```

The status bar at the bottom right says "Query succeeded | 0s".

The screenshot shows the Microsoft Azure portal with the same URL and browser title. The left sidebar is identical. The main area continues the query editor with the following code:

```

23 -- 3. DimAccount
24 CREATE TABLE dbo.DimAccount (
25     AccountKey INT IDENTITY(1,1) PRIMARY KEY,
26     AccountNumber NVARCHAR(50) NOT NULL,
27     CustomerID NVARCHAR(50) NULL,
28     AccountType NVARCHAR(50) NULL,
29     Status NVARCHAR(20) NULL,
30     OpenDate DATE NULL,
31     CloseDate DATE NULL
32 );
33
34 -- 4. DimBranch (simple - optional)
35 CREATE TABLE dbo.DimBranch (
36     BranchKey INT IDENTITY(1,1) PRIMARY KEY,
37     BranchCode NVARCHAR(50) NOT NULL,
38     BranchName NVARCHAR(200) NULL,
39     City NVARCHAR(100) NULL,
40     State NVARCHAR(100) NULL
41 );
42
43 |
44

```

The status bar at the bottom right says "Query succeeded | 0s".

BankDWH (azurebank-server/BankDWH) | Query editor (preview)

```
27     CustomerID NVARCHAR(50) NULL,
28     AccountType NVARCHAR(50) NULL,
29     Status NVARCHAR(20) NULL,
30     OpenDate DATE NULL,
31     CloseDate DATE NULL
32 );
33
34 -- 4. DimBranch (simple - optional)
35 CREATE TABLE dbo.dimBranch (
36     BranchKey INT IDENTITY(1,1) PRIMARY KEY,
37     BranchCode NVARCHAR(50) NOT NULL,
38     BranchName NVARCHAR(200) NULL,
39     City NVARCHAR(100) NULL,
40     State NVARCHAR(100) NULL
41 );
42
43 -- 5. DimProduct (simple - optional)
44 CREATE TABLE dbo.dimProduct (
45     ProductKey INT IDENTITY(1,1) PRIMARY KEY,
46     ProductCode NVARCHAR(50) NOT NULL, -- e.g., SAV, CURR, LOAN
47     ProductName NVARCHAR(100) NULL,
48     ProductType NVARCHAR(50) NULL -- Savings / Current / Loan
49 );
```

Query succeeded | 0s

BankDWH (azurebank-server/BankDWH) | Query editor (preview)

```
49 );
50 === FACT TABLES ===
51
52 -- FactTransactions
53 CREATE TABLE dbo.FactTransactions (
54     TransactionKey BIGINT IDENTITY(1,1) PRIMARY KEY,
55     TransactionID NVARCHAR(50) NOT NULL,
56     DateKey INT NOT NULL,
57     CustomerID NVARCHAR(50) NULL, -- or FK to DimCustomer if you join later
58     AccountNumber NVARCHAR(50) NULL, -- or FK to DimAccount
59     Channel NVARCHAR(10) NULL, -- ATM / UPI
60     TxnType NVARCHAR(50) NULL,
61     Amount DECIMAL(18,2) NULL,
62     [Status] NVARCHAR(20) NULL,
63     ATMTD NVARCHAR(20) NULL,
64     PayerUPI NVARCHAR(200) NULL,
65     PayeeUPI NVARCHAR(200) NULL,
66     DeviceID NVARCHAR(100) NULL,
67     Location NVARCHAR(200) NULL
68 );
69
70
71 |
```

Query succeeded | 0s

BankDWH (azurebank-server/BankDWH) | Query editor (preview)

```
70 -- FactFraudDetection (from gold FactFraudDetection)
71 CREATE TABLE dbo.FactFraudDetection (
72     FraudKey BIGINT IDENTITY(1,1) PRIMARY KEY,
73     AlertID NVARCHAR(100) NOT NULL,
74     AlertType NVARCHAR(200) NOT NULL,
75     TxnID NVARCHAR(50) NULL,
76     TxnType NVARCHAR(50) NULL,
77     Amount DECIMAL(18,2) NULL,
78     Sourcefile NVARCHAR(200) NULL,
79     AlertDateKey INT NULL, -- link to DimDate
80     AlertTime DATETIME2 NULL
81 );
82
83 -- FactCustomerActivity (you can fill later from logs or leave empty)
84 CREATE TABLE dbo.FactCustomerActivity (
85     ActivityKey BIGINT IDENTITY(1,1) PRIMARY KEY,
86     CustomerID NVARCHAR(50) NOT NULL,
87     ActivityDateKey INT NOT NULL,
88     Channel NVARCHAR(20) NULL, -- MOBILE / WEB / ATM / BRANCH
89     ActivityType NVARCHAR(50) NULL, -- LOGIN / PASSWORD_RESET / etc.
90     ActivityCount INT NULL
91 );
92
```

Query succeeded | 0s

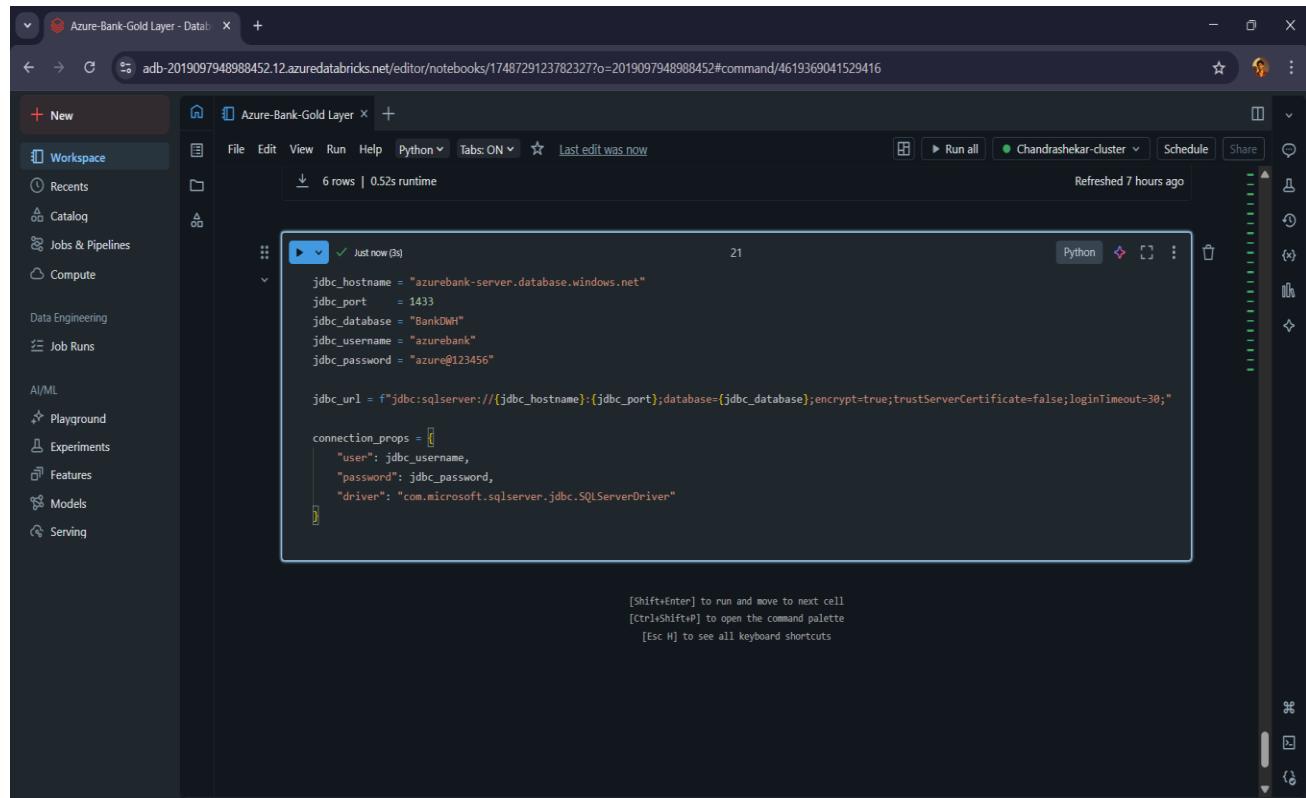
5.3 Loading Gold Data into SQL (via Databricks)

From Databricks notebooks:

1. Read Gold Delta tables (FactTransactions, DimCustomer, DimAccount, FactFraudDetection).
2. For DimDate:
 - o Derived unique dates from transaction timestamps.
 - o Calculated DateKey (YYYYMMDD), Year, Month, Quarter, Day.
 - o Wrote into DimDate.
3. Wrote other tables into SQL:
 - o DimCustomer → DimCustomer table.
 - o DimAccount → DimAccount table.
 - o FactTransactions → FactTransactions table.
 - o FactFraudDetection → FactFraudDetection table.

This was done using JDBC connection, but the document hides code and focuses on the process:

- Connect from Spark → SQL.
- Map columns from Gold dataset → SQL table columns.
- Use append or overwrite as needed.



The screenshot shows a Databricks notebook interface. The left sidebar contains navigation links like Workspace, Recents, Catalog, Jobs & Pipelines, Compute, Data Engineering, Job Runs, AI/ML, Playground, Experiments, Features, Models, and Serving. The main area has a title bar 'Azure-Bank-Gold Layer - Data' and a URL 'adb-201909794898452.12.azuredatabricks.net/editor/notebooks/1748729123782327?o=201909794898452#command/4619369041529416'. Below the title bar are tabs for 'File', 'Edit', 'View', 'Run', 'Help', 'Python', 'Tabs: ON', and 'Last edit was now'. To the right are buttons for 'Run all', 'Chandrashekhar-cluster', 'Schedule', and 'Share'. A status message 'Refreshed 7 hours ago' is shown. The main workspace contains a single code cell with the following Python code:

```
jdbc_hostname = "azurebank-server.database.windows.net"
jdbc_port = 1433
jdbc_database = "BankDWH"
jdbc_username = "azurebank"
jdbc_password = "azure@123456"

jdbc_url = f"jdbc:sqlserver://{{jdbc_hostname}}:{{jdbc_port}};database={{jdbc_database}};encrypt=true;trustServerCertificate=false;loginTimeout=30;"
```

Below the code cell, there is a note: '[Shift+Enter] to run and move to next cell [Ctrl+Shift+P] to open the command palette [Esc H] to see all keyboard shortcuts'.

Azure-Bank-Gold Layer - DataBricks

```
# Load DimDate from Gold to SQL
from pyspark.sql.functions import to_date, year, month, dayofmonth, quarter, col

fact = spark.read.format("delta").load("/mnt/gold/FactTransactions")

dim_date_df = (
    fact
        .select(to_date(col("txn_timestamp")).alias("FullDate"))
        .dropna()
        .dropDuplicates()
        .withColumn("Year", year("FullDate"))
        .withColumn("Month", month("FullDate"))
        .withColumn("Day", dayofmonth("FullDate"))
        .withColumn("Quarter", quarter("FullDate"))
        .withColumn("DateKey",
                   ((col("Year")*10000 + col("Month")*100 + col("Day")).cast("int")))
        .select("DateKey", "FullDate", "Year", "Quarter", "Month", "Day")
)

dim_date_df.write.jdbc(
    url=jdbc_url,
    table="dbo.DimDate",
    mode="overwrite",
    properties=connection_props
)
```

(2) Spark Jobs

```
> dim_date_df: pyspark.sql.dataframe.DataFrame = [DateKey: integer, FullDate: date ... 4 more fields]
> fact: pyspark.sql.dataframe.DataFrame = [Transaction_id: string, customer_id: string ... 7 more fields]
```

Azure-Bank-Gold Layer - DataBricks

```
# Load DimCustomer and DimAccount (initial full load)
# DimCustomer GOLD -> SQL
dim_customer = spark.read.format("delta").load("/mnt/gold/DimCustomer")

dim_customer_sql = (
    dim_customer
        .withColumnRenamed("customer_id", "CustomerID")
        .withColumnRenamed("full_name", "FullName")
        .withColumnRenamed("segment", "Segment")
        .withColumnRenamed("effective_from", "EffectiveFrom")
        .withColumnRenamed("effective_to", "EffectiveTo")
        .withColumnRenamed("is_current", "IsCurrent")
)
```

```
dim_customer_sql.write.jdbc(
    url=jdbc_url,
    table="dbo.DimCustomer",
    mode="overwrite",
    properties=connection_props
)
```

```
# DimAccount GOLD -> SQL
dim_account = spark.read.format("delta").load("/mnt/gold/DimAccount")

dim_account_sql = (
    dim_account
        .withColumnRenamed("account_number", "AccountNumber")
        .withColumnRenamed("customer_id", "CustomerID")
        .withColumnRenamed("status", "Status")
        .withColumnRenamed("account_type", "AccountType")
)
```

Azure-Bank-Gold Layer - DataBricks

```
dim_account_sql.write.jdbc(
    url=jdbc_url,
    mode="overwrite",
    properties=connection_props
)
```

```
# DimAccount GOLD -> SQL
dim_account = spark.read.format("delta").load("/mnt/gold/DimAccount")

dim_account_sql = (
    dim_account
        .withColumnRenamed("account_number", "AccountNumber")
        .withColumnRenamed("customer_id", "CustomerID")
        .withColumnRenamed("status", "Status")
        .withColumnRenamed("account_type", "AccountType")
)
```

```
dim_account_sql.write.jdbc(
    url=jdbc_url,
    table="dbo.DimAccount",
    mode="overwrite",
    properties=connection_props
)
```

(2) Spark Jobs

```
> dim_account: pyspark.sql.dataframe.DataFrame = [account_number: string, account_type: string ... 1 more field]
> dim_account_sql: pyspark.sql.dataframe.DataFrame = [AccountNumber: string, AccountType: string ... 1 more field]
> dim_customer: pyspark.sql.dataframe.DataFrame = [customer_id: string, customer_name: string]
> dim_customer_sql: pyspark.sql.dataframe.DataFrame = [CustomerID: string, customer_name: string]
```

```

# Load FactTransactions from Gold to SQL
from pyspark.sql.functions import year, month, dayofmonth, col

fact_gold = spark.read.format("delta").load("/mnt/gold/FactTransactions")

# Create Datekey
fact_sql = (
    fact_gold
    .withColumn("DateKey",
                (year("txn_timestamp")*10000
                 + month("txn_timestamp")*100
                 + dayofmonth("txn_timestamp").cast("int")))
    .select(
        col("transaction_id").alias("TransactionID"),
        col("DateKey"),
        col("customer_id").alias("CustomerID"),
        col("account_number").alias("AccountNumber"),
        col("txn_type").alias("TxnType"),
        col("amount").alias("Amount"),
        col("status").alias("Status"),
        col("atm_id").alias("ATMID"),
        col("location").alias("Location")
    )
)

```

> fact_gold: pyspark.sql.dataframe.DataFrame = [transaction_id: string, customer_id: string ... 7 more fields]
> fact_sql: pyspark.sql.dataframe.DataFrame = [TransactionID: string, DateKey: integer ... 7 more fields]

```

# Load FactFraudDetection from Gold to SQL
from pyspark.sql.functions import year, month, dayofmonth, col

fraud_gold = spark.read.format("delta").load("/mnt/gold/FactFraudDetection")

fraud_sql = [
    fraud_gold
    .withColumn("AlertDateKey",
                (year("fraud_date")*10000 +
                 month("fraud_date")*100 +
                 dayofmonth("fraud_date").cast("int")))
    .select(
        col("txn_id").alias("TransactionID"),
        col("fraud_date").alias("FraudDate"),
        col("fraud_severity").alias("FraudSeverity"),
        col("fraud_count").alias("FraudCount"),
        col("total_fraud_amount").alias("TotalFraudAmount"),
        col("AlertDateKey")
    )
]

```

> fraud_gold: pyspark.sql.dataframe.DataFrame = [txn_id: string, fraud_date: date ... 3 more fields]
> fraud_sql: pyspark.sql.dataframe.DataFrame = [transactionID: string, FraudDate: date ... 4 more fields]

Showing limited object explorer here. For full capability please click here to open Azure Data Studio.

```

91 );
92 -- Implement MERGE / UPSERT for daily sync (SCD Type-2)
93 -- Create staging tables
94 CREATE TABLE dbo.StgCustomer (
95     CustomerID VARCHAR(50),
96     FullName VARCHAR(200)
97 );
98
99 CREATE TABLE dbo.StgAccount (
100     AccountNumber VARCHAR(50),
101     CustomerID VARCHAR(50),
102     Status VARCHAR(50),
103     AccountType VARCHAR(50)
104 );
105
106
107
108
109
110
111
112
113

```

Query succeeded | 0s

Azure-Bank-Gold Layer - DataBricks

Azure-Bank-Gold Layer

```
jdbc_url = "jdbc:sqlserver://azuredatabricks-test.database.windows.net:1433;database=BankDW"

db_props = {
    "user": "azuredatabricks",
    "password": "azuredatabricks",
    "driver": "com.microsoft.sqlserver.jdbc.SQLServerDriver"
}

# Load GOLD DimAccount
dim_account_path = "/mnt/gold/DimAccount"
dim_account = spark.read.format("delta").load(dim_account_path)

# Prepare staging
stg_account = dim_account.select(
    col("account_number").alias("AccountNumber"),
    col("customer_id").alias("CustomerID"),
    col("Status").alias("Status"),
    col("AccountType").alias("AccountType")
)

# Write to SQL staging
stg_account.write.jdbc(
    url=jdbc_url,
    table="dbo.StgAccount",
    mode="overwrite",
    properties=db_props
)

print("🔥 StgAccount successfully loaded to SQL")
```

The screenshot shows the Microsoft Azure Query editor (preview) interface. On the left, there's a sidebar with navigation links like Overview, Activity log, Tags, Diagnose and solve problems, Query editor (preview), Mirror database in Fabric (preview), Resource visualizer, Settings, Data management, Integrations, Power Platform, Security, Intelligent performance, Monitoring, Automation, Help, Tables, Views, and Stored Procedures. The 'Query editor (preview)' link is highlighted.

The main area has a search bar at the top with the placeholder 'Search resources, services, and docs (G+/-)'. Below the search bar is a Copilot button. The URL in the address bar is: portal.azure.com/#@kluniversity.in/resource/subscriptions/f4b3a09a-9941-4890-9642-6a50016aa912/resourceGroups/Azure-bank-RG/providers/Microsoft.Sql/servers/azurebank-server/databases/BankDWH.

The title bar says 'BankDWH (azurebank-server/BankDWH) | Query editor (preview)'. The current query is 'Query 1'.

The query code is:

```
100 CREATE TABLE dbo.StgAccount (
101     AccountNumber VARCHAR(50),
102     CustomerID VARCHAR(50),
103     Status VARCHAR(50),
104     AccountType VARCHAR(50)
105 );
106
107
108 SELECT * FROM dbo.StgAccount;
```

The results section shows the following table:

AccountNumber	CustomerID	Status	AccountType
1002003242	CUST129	Active	Savings
1002003035	CUST041	Active	Savings
1002003186	CUST440	Active	Savings
1002003470	CUST461	Active	Savings

At the bottom, a message says 'Query succeeded | 1s'.

Azure-Bank-Gold Layer - DataFrames

File Edit View Run Help Python Tabs: ON Last edit was 4 minutes ago

6 CUST438 CUST438
7 CUST103 CUST103
8 CUST356 CUST356
9 CUST404 CUST404
10 CUST238 CUST238
11 CUST316 CUST316
12 CUST212 CUST212
13 CUST239 CUST239
14 CUST307 CUST307
15 CUST004 CUST004

500 rows | 0.52s runtime Refreshed 5 minutes ago

```

stg_customer.write.jdbc(
    url=jdbc_url,
    table="dbo.StgCustomer",
    mode="overwrite",
    properties=db_props
)

print("StgCustomer loaded successfully to SQL")

```

(1) Spark Jobs
StgCustomer loaded successfully to SQL.

BankDWH (azurebank-server/BankDWH) | Query editor (preview)

CREATE TABLE dbo.StgAccount (
 AccountNumber VARCHAR(50),
 CustomerID VARCHAR(50),
 Status VARCHAR(50),
 AccountType VARCHAR(50)
);

SELECT * FROM [dbo].[StgCustomer];

CustomerID	FullName
CUST073	CUST073
CUST158	CUST158
CUST303	CUST303
CUST359	CUST359

Query succeeded | 0s

BankDWH (azurebank-server/BankDWH) | Query editor (preview)

```

-- MERGE For DimCustomer (SCD Type-2)
MERGE dbo.DimAccount AS target
USING dbo.StgAccount AS source
ON target.AccountNumber = source.AccountNumber
WHEN MATCHED THEN
    UPDATE SET
        target.CustomerID = source.CustomerID,
        target.is_active = source.is_active

```

AccountNumber	AccountType	is_active
1002003092	Savings	1
1002003338	Savings	1
1002003041	Savings	1
1002003409	Savings	1

Query succeeded | 0s

5.4 Scheduled Sync (Timer Trigger Functions)

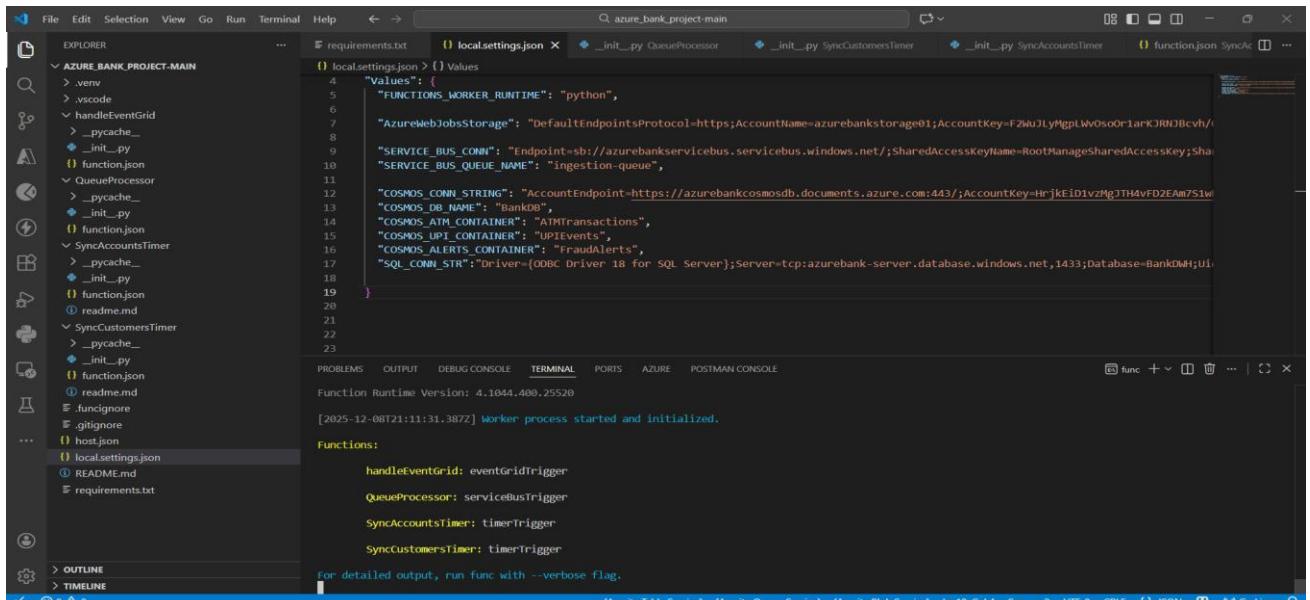
To match your requirement:

"Timer-trigger Azure Function – Daily – Synchronize full customer master data from source.
Timer-trigger Azure Function – Daily – Update account status (e.g., active/dormant) in DimAccount."

You designed:

- A Timer-trigger Azure Function that runs once per day.
 - It pulls latest customer/account data from ADLS or upstream source.
 - Loads it into staging tables StgCustomer and StgAccount in SQL.
 - Executes SQL logic or stored procedures to:
 - Insert new customers/accounts.
 - Update status (e.g., set accounts to dormant).

This implements a simple SCD (Slowly Changing Dimension) style sync.

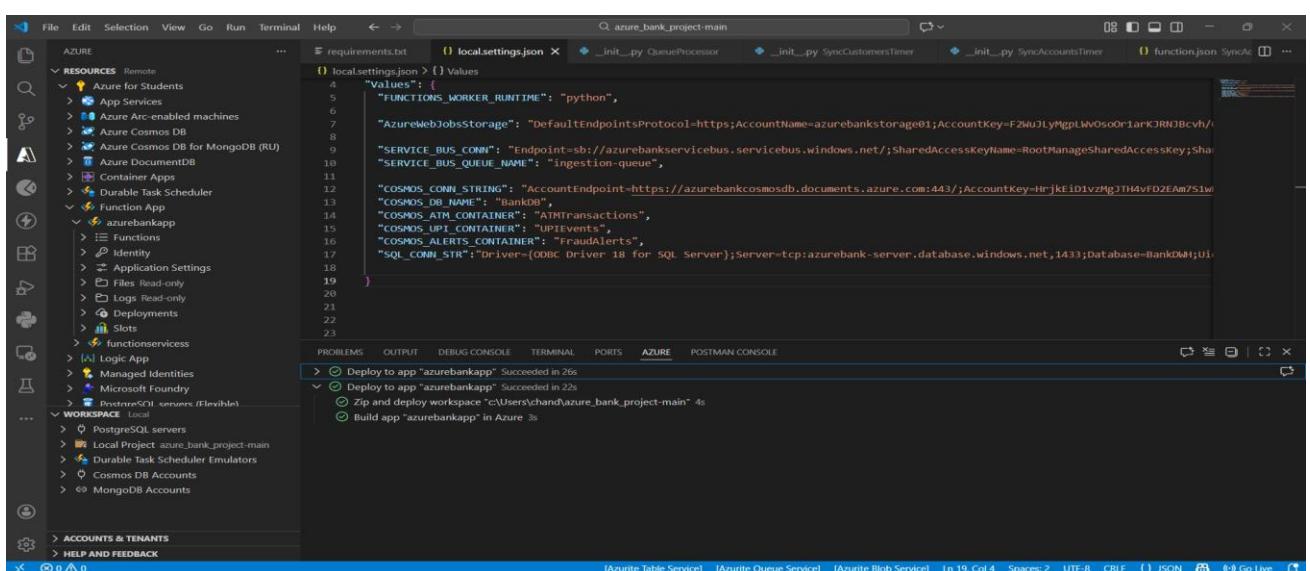


The screenshot shows the Azure Functions project structure in Visual Studio Code. The Explorer sidebar shows files like requirements.txt, local.settings.json, and function.json. The local.settings.json file is open, displaying configuration values for worker runtime, storage, and service bus. The terminal shows the function runtime version and a message indicating the worker process has started and initialized. The functions listed are handleEventGrid, QueueProcessor, SyncAccountsTimer, and SyncCustomersTimer.

```
values: {
    "FUNCTIONS_WORKER_RUNTIME": "python",
    "AzureWebJobsStorage": "DefaultEndpointsProtocol=https;AccountName=azurebankstorage01;AccountKey=F2WuJLyMgplWvoso0r1arKJRNBcvh/",
    "SERVICE_BUS_CONN": "Endpoint=sb://azurebankservicebus.servicebus.windows.net/;SharedAccessKeyName=RootManageSharedAccessKey;SharedAccessSignature=HrjkEiD1vzMgJTH4vFD2EAmtS1wI",
    "SERVICE_BUS_QUEUE_NAME": "ingestion-queue",
    "COSMOS_CONN_STRING": "AccountEndpoint=https://azuredbankcosmosdb.documents.azure.com:443/;AccountKey=HrjkEiD1vzMgJTH4vFD2EAmtS1wI",
    "COSMOS_DB_NAME": "BankDB",
    "COSMOS_ATM_CONTAINER": "ATMTransactions",
    "COSMOS_UPT_CONTAINER": "UPEvents",
    "COSMOS_ALERTS_CONTAINER": "FraudAlerts",
    "SQL_CONN_STR": "Driver={ODBC Driver 18 for SQL Server};Server=tcp:azuredbank-server.database.windows.net,1433;Database=BankDB;Trusted_Connection=True;Encrypt=True;Connection Timeout=30"
}
```

Functions:

- handleEventGrid: eventGridTrigger
- QueueProcessor: serviceBusTrigger
- SyncAccountsTimer: timerTrigger
- SyncCustomersTimer: timerTrigger



The screenshot shows the Azure portal's resources page. The local.settings.json file is open in the code editor. The terminal shows deployment logs for the 'azurebankapp' function app, indicating successful deployment and zip deployment. The workspace sidebar shows various Azure services like App Services, Durable Task Scheduler, and Function App.

```
Deploy to app "azurebankapp" Succeeded in 26s
Deploy to app "azurebankapp" Succeeded in 22s
Zip and deploy workspace "c:\Users\chand\azure_bank_project-main" 4s
Build app "azurebankapp" in Azure 3s
```

functionservices - Microsoft Azure

Home > Function App > functionservices

Overview

- Activity log
- Access control (IAM)
- Tags
- Diagnose and solve problems
- Microsoft Defender for Cloud
- Events (preview)
- Log stream
- Resource visualizer

Resource group (move) : mazanet-pro2

Status : Running

Location (move) : Malaysia West

Subscription (move) : Azure for Students

Subscription ID : f4b3a09a-9941-4890-9642-6a50016aa912

Tags (edit) : Add tags

Default domain : functionservices-hrbng0h4g6cbdugy.malaysiawest1.azurewebsites.net

Health Check : Not Configured

Operating System : Linux

App Service Plan : ASP-mazanetpro2-b16e (B1:1)

Runtime version : 4.1045.100.0

Functions

Name	Trigger	Status	Monitor
handleEventGrid	Event Grid	Enabled	Invocations and more
QueueProcessor	Service Bus	Enabled	Invocations and more
SyncAccountsTimer	Timer	Enabled	Invocations and more
SyncCustomersTimer	Timer	Enabled	Invocations and more

Test/Run - Microsoft Azure

Home > Function App > functionservices > SyncAccountsTimer | Code + Test

Code + Test

Save Discard Refresh Test/Run Get function URL Disable Delete Upload Resource JSON Send us your feedback

```
functionservices / SyncAccountsTimer / __init__.py
```

```
1 import datetime
2 import logging
3 import pyodbc
4 import os
5
6 import azure.functions as func
7
8 def main(timer: func.TimerRequest) -> None:
9     ... logging.info("Starting Daily Account Sync Job...")
10
11     try:
12         ... conn_str = os.getenv("SQL_CONN_STR")
13         ... conn = pyodbc.connect(conn_str)
14     
```

Logs

```
Connected! You are now viewing logs of Function runs in the current Code + Test panel. To see all the logs for this Function, please go to https://[FunctionApp].azurewebsites.net/api/Logs?functionName=SyncAccountsTimer
[2025-12-08T21:20:24Z] [Information] Request [0516a0aa-5c03-41fe-afae-c86f8f012449] GET https://vinithablobser.blob.core.windows.net/timers/functionservicehost/Host.Functions.SyncAccountsTimer/status
x-ms-version:2023-11-03
Accept:application/xml
x-ms-client-request-id:0516a0aa-5c03-41fe-afae-c86f8f012449
x-ms-return-client-request-id:1
x-ms-request-id:40000000-0000-0000-0000-000000000000
x-ms-date:Mon, 08 Dec 2025 21:20:24 GMT
Authorization:REDACTED
```

Test/Run

Input Output

HTTP response code : 202 Accepted

HTTP response content :

Test/Run - Microsoft Azure

Home > Microsoft.SQLDatabase.newDatabaseExistingServer_19a12780237f4f0e | Overview > BankDWH (azurebank-server/BankDWH)

BankDWH (azurebank-server/BankDWH) | Query editor (preview)

Query 1

Run Cancel query Save query Export data as Show only Editor

```
1 SELECT TOP 20 * FROM dbo.DimCustomer ORDER BY StartDate DESC;
```

Results

CustomerID	customer_name	StartDate	EndDate	IsActive
CUST444	CUST444	2025-12-08T20:23:34.21300...		True
CUST441	CUST441	2025-12-08T20:23:34.21300...		True
CUST117	CUST117	2025-12-08T20:23:34.21300...		True
CUST053	CUST053	2025-12-08T20:23:34.21300...		True

Query succeeded | 0s

The screenshot shows the Microsoft Azure Query editor (preview) for the BankDWH database. The left sidebar contains navigation links such as Overview, Activity log, Tags, Diagnose and solve problems, Query editor (preview), Mirror database in Fabric (preview), Resource visualizer, Settings, Compute + storage, Connection strings, Maintenance, Properties, Locks, Data management, Replicas, and Sync to other databases. The main area has a search bar, a login button, and a feedback link. Below that is a 'Query 1' section with a 'Run' button and options to cancel, save, export, or show only the editor. The query itself is:

```

1 SELECT TOP 20 * FROM dbo.DimCustomer ORDER BY StartDate DESC;
2
3 SELECT TOP 20 * FROM dbo.DimAccount;
4

```

The results table shows four rows of data:

AccountNumber	AccountType	is_active
1002003092	Savings	1
1002003338	Savings	1
1002003041	Savings	1
1002003409	Savings	1

A message at the bottom indicates 'Query succeeded | 0s'.

6. Security & Network Configuration

6.1 Network Segmentation & Firewalls

You applied network restrictions to critical components:

- Restricted Cosmos DB to allow traffic only from:
 - Specific IPs (e.g., Databricks public IP).
 - Or from selected subnets (if VNet integration used).
- Restricted SQL Database to:
 - Allow access only from:
 - Your client machine (for development),
 - Azure services, or
 - Specific VNet subnets / private endpoints.

This ensures:

- No open public access to databases.
- Only trusted services communicate with them.

6.2 Fraud Alert Event Flow

Fraud detection to include an event-based alert flow:

- For each high-value transaction:
- An event is generated (e.g., of type HighValueTransaction).
- This event is routed via Event Grid.
- A downstream Function or subscriber can:

- ☐ Insert into FraudAlerts (already done).
- ☐ Send notifications by pushing messages to a Fraud Alert Service Bus queue (for SMS/Email).

This showcases a real-time alert architecture suitable for banking risk teams.

Service Bus Namespace

Queues 2

Name	Status	Message count	Active messages	Dead-letter mess...	Scheduled mess...	Max size	Enable partitioning
fraud-alerts	Active	0	0	0	0	1024 MB	false
ingestion-queue	Active	14	0	14	0	1024 MB	false

```

import logging
import json
import datetime as dt
import azure.functions as func

def main(event: func.EventGridEvent,
        fraudAlertOut: func.Out[func.Document],
        serviceBusMsg: func.Out[str]) -> None:
    logging.info("FraudDetectionHandler triggered")

    data = event.get_json()
    amount = data.get("amount", 0)
    customer_id = data.get("customer_id")
    location = data.get("location")
    device_id = data.get("device_id")
    txn_time_str = data.get("txn_timestamp")

    # very simple parsing

```

PROBLEMS 2 **OUTPUT** **DEBUG CONSOLE** **TERMINAL** **PORTS** **AZURE** **POSTMAN CONSOLE**

FUNCTIONS:

- FraudDetectionHandler: eventGridTrigger
- handleEventGrid: eventGridTrigger
- HighValueTxToEventGrid: eventGridTrigger
- QueueProcessor: serviceBusTrigger
- SyncAccountsTimer: timerTrigger
- SyncCustomersTimer: timerTrigger

For detailed output, run func with --verbose flag.
[2025-12-09T08:28:53.827Z] Host lock lease acquired by instance ID '0000000000000000000000AC11EBCB'.

```

{
  "Values": {
    "AzureWebJobsStorage": "DefaultEndpointsProtocol=https;AccountName=azurebankstorage01;AccountKey=F2NuJLyMgplLwOsoOr1arKJRNTBvh",
    "SERVICE_BUS_CONN": "Endpoint=sb://azurbankservicebus.servicebus.windows.net/;SharedAccessKeyName=RootManageSharedAccessKey;SharedAccessKey=HrjkeId1vzMgJTH4vFD2EAm7S1",
    "SERVICE_BUS_QUEUE_NAME": "ingestion-queue",
    "COSMOS_CONN_STRING": "AccountEndpoint=https://azurbankcosmosdb.documents.azure.com:443/;AccountKey=02EgcWAHPHu5VymafXX1IREVU94q6jQaZmOnb3cyEPTrvdi9tqJQQJ99BLACsw3vEXJ3w3AAABAEZegItMH",
    "COSMOS_DB_NAME": "BankDB",
    "COSMOS_ATM_CONTAINER": "ATMTransactions",
    "COSMOS_UPI_CONTAINER": "UPIEvents",
    "COSMOS_ALERTS_CONTAINER": "FraudAlerts",
    "SQL_CONN_STR": "Driver={ODBC Driver 18 for SQL Server};Server=tcp:azurbank-server.database.windows.net,1433;Database=BankDB;User ID=sa;Password=StrongPass@123;Encrypt=True;TrustServerCertificate=False;Connection Timeout=30",
    "EVENTGRID_TOPIC_ENDPOINT": "https://bank-tx-events.malaysiaeast-1.eventgrid.azure.net/api/events",
    "EVENTGRID_TOPIC_KEY": "02EgcWAHPHu5VymafXX1IREVU94q6jQaZmOnb3cyEPTrvdi9tqJQQJ99BLACsw3vEXJ3w3AAABAEZegItMH"
  }
}

```

PROBLEMS 2 OUTPUT DEBUG CONSOLE TERMINAL PORTS AZURE POSTMAN CONSOLE

- Deploy to app "azurbankapp" Succeeded in 24s
- Zip and deploy workspace "c:\Users\chand\azur_bank_project-main" 4s
- Build app "azurbankapp" in Azure 6s
- Deploy to app "azurbankfunctionapp" Succeeded in 1m 59s
- Zip and deploy workspace "c:\Users\chand\azur_bank_project-main" 1m 16s
- Build app "azurbankfunctionapp" in Azure 13s

Deployment to "azurbankfunctionapp" completed.

Source: Azure Functions Stream logs Upload settings View output

Name	Trigger	Status	Monitor
FraudDetectionHandler	Event Grid	Enabled	Invocations and more
handleEventGrid	Event Grid	Enabled	Invocations and more
HighValueTxToEventGrid	Event Grid	Enabled	Invocations and more
QueueProcessor	Service Bus	Enabled	Invocations and more
SyncAccountsTimer	Timer	Enabled	Invocations and more
SyncCustomersTimer	Timer	Enabled	Invocations and more

timestamp [UTC]	message	severityLevel	itemType	customDimensions	operation_Id	operation_Parent
> 9/12/2025, 8:49:06.111 am	The next 5 occurrences of the 'SyncAccountsTimer' scheduled...	1	trace	{"prop__functionName":"SyncAccountsTimer","ProcessId":2...		
> 9/12/2025, 8:49:06.105 am	The next 5 occurrences of the 'SyncCustomersTimer' scheduled...	1	trace	{"prop__functionName":"SyncCustomersTimer","ProcessId":2...		
> 9/12/2025, 8:48:21.495 am	Host Status: {"id": "azurbankfunctionapp", "state": "Runnin...	1	trace	{"ProcessId":27,"HostInstanceId":e8d30f7b-40b7-4e9f-91...	151b67546dbfceacb391afebe4c2da3	81a699b5e55a
> 9/12/2025, 8:48:21.64 am	Host Status: {"id": "azurbankfunctionapp", "state": "Runnin...	1	trace	{"ProcessId":27,"HostInstanceId":e8d30f7b-40b7-4e9f-91...	7e7948482ccad9cb96cdccc216a3e673	8bbeb0555e0
> 9/12/2025, 8:48:08.054 am	ConcurrencyOptions {"DynamicConcurrencyEnabled": false,...	1	trace	{"ProcessId":27,"HostInstanceId":e8d30f7b-40b7-4e9f-91...		
> 9/12/2025, 8:48:08.047 am	ServiceBusOptions {"ClientRetryOptions": {"Mode": "Expon...	1	trace	{"ProcessId":27,"HostInstanceId":e8d30f7b-40b7-4e9f-91...		

13s 148ms | Display time (UTC+00:00) | Query details | 1 - 7 of 1138

Service Bus Queue

Peek Mode | Send messages | Refresh | Export messages | Show message body | Settings | Learn more | Give feedback

Queue (15) Dead-letter (0)

Peek from start → Peek next messages ↗ Peek with options ↘ Re-send selected messages ↘ Download selected message body

Showing 15 of 15 messages

Sequence Number	Message ID	Enqueued Time	Delivery Count	State	Body ...	Label/Subject	Message Text
1	371208468d7441808557f3f536...	Tue, Dec 09, 25, 02:42:39 PM GM...	0	Active	398 B		[{"id": "EVT-PST-001", "eventTyp...
2	f3754e1af6664d1eb9d993a095c5...	Tue, Dec 09, 25, 02:43:10 PM GM...	0	Active	398 B		[{"id": "EVT-PST-002", "eventTyp...
3	dfc328e0f4ef4469b83a1e272f3e...	Tue, Dec 09, 25, 02:43:58 PM GM...	0	Active	398 B		[{"id": "EVT-PST-003", "eventTyp...
4	6b50bc780fc647b908bf74ec7f5...	Tue, Dec 09, 25, 02:43:58 PM GM...	0	Active	398 B		[{"id": "EVT-PST-003", "eventTyp...
5	9a6d86fcfeedf45ecac8b439e1648...	Tue, Dec 09, 25, 02:43:58 PM GM...	0	Active	398 B		[{"id": "EVT-PST-003", "eventTyp...
6	5d82615f2a6c43419fc95f112bb...	Tue, Dec 09, 25, 02:43:58 PM GM...	0	Active	398 B		[{"id": "EVT-PST-003", "eventTyp...
7	8bcd8130715e45e9a645243c365...	Tue, Dec 09, 25, 02:43:58 PM GM...	0	Active	398 B		[{"id": "EVT-PST-003", "eventTyp...

Message Body | Message Properties

Select a message to see its details.

Azure Cosmos DB account

New Item | Update | Discard | Delete | Upload Item

Home | Fraud_Items

SELECT * FROM c Type a query predicate (e.g., WHERE c.id='1'), or choose one from the drop down list, or leave empty to query all documents. Apply Filter

id	... /alertT
TXN005484_High-value transa...	High-valu...	1
TXN005484_Unusual UPI Tran...	Unusual U...	2
TXN005489_High-value transa...	High-valu...	3
TXN005489_Unusual UPI Tran...	Unusual U...	4
TXN005498_High-value transa...	High-valu...	5
TXN005498_Unusual UPI Tran...	Unusual U...	6
TXN005502_High-value transa...	High-valu...	7
TXN005502_Unusual UPI Tran...	Unusual U...	8
TXN005506_High-value transa...	High-valu...	9
TXN005506_Unusual UPI Tran...	Unusual U...	10
TXN005510_High-value transa...	High-valu...	11
TXN005510_Unusual UPI Tran...	Unusual U...	12

Load more

6.3 Azure Firewall / vSRX Conceptual Integration

The project document specifies:

All traffic is routed through Juniper vSRX firewall (or Azure Firewall) and private endpoints.

For learning and demo purposes, you:

- Explained how in production:
 - Databricks, Function Apps, and SQL/Cosmos would be placed in a VNet.

- Firewall would control ingress/egress.
- Private endpoints would be used instead of public endpoints.

Even if full enterprise network was not completely implemented, your design acknowledges best practices and partially simulates them using IP-based restrictions and Azure networking options you configured.

Azurebanks-firewalls Firewall

Overview

Resource group (move) : [Azure-bank-RG](#)

Location : Malaysia West

Subscription (move) : [Azure for Students](#)

Subscription ID : f4b3a09a-9941-4890-9642-6a50016aa912

Virtual network : [azurebank-virtual](#)

Firewall policy : [azurebank-policy](#)

Provisioning state : Succeeded

Tags (edit) : Add tags

Firewall policy

Visit Azure Firewall Manager at the link below to edit the Firewall Policy on this firewall

Policy : [azurebank-policy \(change\)](#)

Auto-learn IP Prefixes : [Disabled](#)

azurebank-vn Virtual network

Overview

Resource group (move) : [Azure-bank-RG](#)

Location (move) : Malaysia West

Subscription (move) : [Azure for Students](#)

Subscription ID : f4b3a09a-9941-4890-9642-6a50016aa912

Tags (edit) : Add tags

Capabilities (5)

DDoS protection : Configure additional protection from distributed denial of service attacks. Status: Not configured

Azure Firewall : Protect your network with a stateful L3-L7 firewall. Status: AZUREBANK-FIREWALLS

Peerings : Seamlessly connect two or more virtual networks. Status: Not configured

Microsoft Defender for Cloud : Strengthen the security posture of your environment.

Microsoft Azure | Overview | azurebank-vn - Microsoft Network virtualNetworks/azureban...

Subnets

Create subnets to segment the virtual network address space into smaller ranges for use by your applications. When you deploy resources into a subnet, Azure assigns the resource an IP address from the subnet.

Name	IPv4	IPv6	Available IPs	Delegated to	Security group	Route table
AzureFirewallSubnet	10.0.0.0/26	-	59	Microsoft...	-	-
AzureFirewallManage...	10.0.1.0/27	-	27	Microsoft...	-	-
Azurefirewallsubnet2	10.0.2.0/25	-	123	Microsoft...	-	-

Add or remove favorites by pressing **Ctrl+Shift+F**

Microsoft Azure | Overview | Function App > azurebankfunctionapp | Networking >

Virtual Network Integration

You can control what traffic goes through the virtual network integration. Application routing defines what traffic is routed from your app and into the virtual network. Configuration routing affects operations that happen before or during startup of your app. Virtual network routing handles how both app and configuration traffic are routed from your virtual network and out. [Learn more](#)

Virtual network configuration

Virtual network name	azur...
Subnet name	Azurefirewalls...
Subnet IP address availability	0 available (123 total)
Subnet connected sites	azur...
Subnet connected plans	ASP-mazan...

Application routing

Outbound internet traffic

Configuration routing

Container image pull
Content storage
Backup/restore

Virtual network routing

Microsoft Azure | Overview | Microsoft.PrivateEndpoint-20251209182305 | Overview >

Deployment

Your deployment is complete

Deployment name : Microsoft.PrivateEndpoint-20251209182305
Subscription : Azure for Students
Resource group : Azure-bank-RG

Start time : 9/12/2025, 6:26:29 pm
Correlation ID : 07950e2c-9b4d-49f1-9d96-6e1adcca5ba3

Deployment details
Next steps

Go to resource

Cost management
Get notified to stay within your budget and prevent unexpected charges on your bill.
[Set up cost alerts >](#)

Microsoft Defender for Cloud
Secure your apps and infrastructure
[Go to Microsoft Defender for Cloud >](#)

Free Microsoft tutorials
[Start learning today >](#)

Work with an expert
Azure experts are service provider partners who can help manage your assets on Azure and be your first line of support.
[Find an Azure expert >](#)

Microsoft Azure | portalAzure.com/#@kluniversity.in/resource/subscriptions/f4b3a09a-9941-4890-9642-6a50016aa912/resourceGroups/Azure-bank-RG/providers/Microsoft.Network/privateEndpoints/pe-bankdwh... | +

Home > Microsoft.PrivateEndpoint-20251209182305 | Overview

pe-bankdwh Private endpoint

Search Delete Refresh

Overview

Activity log Access control (IAM) Tags Diagnose and solve problems Resource visualizer Settings Monitoring Automation Help

How to make my private endpoint highly available Is my private endpoint resilient

Essentials

Resource group (move)	: Azure-bank-RG
Location	: Malaysia West
Subscription (move)	: Azure for Students
Subscription ID	: f4b3a09a-9941-4890-9642-6a50016aa912
Provisioning state	: Succeeded
Tags (edit)	: Add tags
Virtual network/subnet	: azurebank-virtual/default
Network interface	: pe-bankdwh-nic
Private link resource	: azurebank-server
Target sub-resource	: sqlServer
Connection status	: Approved
Request/Response	: Auto-approved

JSON View

Add or remove favorites by pressing Ctrl+Shift+F

Give feedback

Microsoft Azure | portalAzure.com/#@kluniversity.in/resource/subscriptions/f4b3a09a-9941-4890-9642-6a50016aa912/resourceGroups/Azure-bank-RG/providers/Microsoft.Sql/servers/azurebank-server/networking | +

Home > azurebank-server

azurebank-server | Networking

SQL server

Search

Feedback

Public access Private access Connectivity

Public network access

Public Endpoints allow access to this resource through the internet using a public IP address. An application or resource that is granted access with the following network rules still requires proper authorization to access this resource. [Learn more](#)

Public network access

Disable
 Selected networks
 Only approved private endpoint connections will be accepted by this resource. Any existing firewall rules or virtual network endpoints will be retained, but disabled. [Learn more](#)

Successfully updated server public network access

Successfully updated server public network access for server azurebank-server

Add or remove favorites by pressing Ctrl+Shift+F

Microsoft Azure | portalAzure.com/#view/HubsExtension/DeploymentDetailsBlade/-/overview/id%2fsubscriptions%2ff4b3a09a-9941-4890-9642-6a50016aa912%2fresourceGroups%2fAzure-bank-RG%2fprovide... | +

Home >

Microsoft.PrivateEndpoint-20251209183116 | Overview

Deployment

Search Delete Cancel Redeploy Download Refresh

Overview

Your deployment is complete

Deployment name : Microsoft.PrivateEndpoint-20251209183116
Subscription : Azure for Students
Resource group : Azure-bank-RG

Start time : 9/12/2025, 6:33:57 pm
Correlation ID : d6792096-6854-4a1e-a2db-cf99ff6b8401

Deployment details

Next steps

Go to resource

Cost management

Get notified to stay within your budget and prevent unexpected charges on your bill.
[Set up cost alerts](#)

Microsoft Defender for Cloud

Secure your apps and infrastructure
[Go to Microsoft Defender for Cloud](#)

Free Microsoft tutorials

[Start learning today](#)

Work with an expert

Azure experts are service provider partners who can help manage your assets on Azure and be your first line of support.
[Find an Azure expert](#)

Add or remove favorites by pressing Ctrl+Shift+F

Microsoft Azure | portal.azure.com/#@kluniversity.in/resource/subscriptions/f4b3a09a-9941-4890-9642-6a50016aa912/resourceGroups/Azure-bank-RG/providers/Microsoft.Network/privateEndpoints/pe-bank-cosmos

pe-bank-cosmos | Overview

How to make my private endpoint highly available | Is my private endpoint resilient?

Search | Delete | Refresh | Overview | Essentials | JSON View

Resource group (move) : Azure-bank-RG
Location : Malaysia West
Subscription (move) : Azure for Students
Subscription ID : f4b3a09a-9941-4890-9642-6a50016aa912
Provisioning state : Succeeded
Tags (edit) : Add tags

Virtual network/subnet : azurebank-virtual/default
Network interface : pe-bank-cosmos-nic
Private link resource : azurebankfunctionapp
Target sub-resource : sites
Connection status : Approved
Request/Response : -

Add or remove favorites by pressing Ctrl+Shift+F | Give feedback

Microsoft Azure | portal.azure.com/#@kluniversity.in/resource/subscriptions/f4b3a09a-9941-4890-9642-6a50016aa912/resourceGroups/Azure-bank-RG/providers/Microsoft.DocumentDB/databaseAccounts/azureb...

azurebankcosmosdb | Networking

Azure Cosmos DB account

Search | Save | Discard | Public access | Private access | Network Security Perimeter | Connectivity

Mirroring in Fabric
Container Copy
Resource visualizer
Settings
Features
Default consistency
Backup & Restore
Networking
CORS
Data Encryption
Dedicated Gateway
Keys
Advisor Recommendations
Identity
Locks
Integrations

Public network access
 All networks Selected networks Disabled Secured by perimeter

No public traffic will be able to access this resource. [Learn more](#).

Give Feedback | Help improve this page

Add or remove favorites by pressing Ctrl+Shift+F

Microsoft Azure | portal.azure.com/#@kluniversity.in/resource/subscriptions/f4b3a09a-9941-4890-9642-6a50016aa912/resourceGroups/Azure-bank-RG/providers/Microsoft.Network/routeTables/rt-func-snet/overvi...

rt-func-snet | Route tables

Diagnose routing issues with this route table | Retrieve detailed routing information for troubleshooting | Check the status of routes in this route table

Search | Move | Delete | Refresh | Give feedback | Overview | Essentials | JSON View

Resource group (move) : Azure-bank-RG
Location : Malaysia West
Subscription (move) : Azure for Students
Subscription ID : f4b3a09a-9941-4890-9642-6a50016aa912
Tags (edit) : Add tags

Associations : 0 subnet associations

Routes
Name ↑↓ Address prefix ↑↓ Next hop type ↑↓ Next hop IP address ↑↓

No results.

Subnets
Name ↑↓ Address range ↑↓ Virtual network ↑↓ Security group ↑↓

No results.

Add or remove favorites by pressing Ctrl+Shift+F

Microsoft Azure

Home > Network foundation | Route tables > rt-func-snet

rt-func-snet | Routes

Route table

Search routes

Name	Address prefix	Next hop type	Next hop IP address
default-to-fw	0.0.0.0/0	VirtualAppliance	10.10.0.4

Add | Refresh | Give feedback

Overview | Activity log | Access control (IAM) | Tags | Diagnose and solve problems | Resource visualizer | Settings | Configuration | Routes | Subnets | Properties | Locks | Monitoring | Automation | Help

Add or remove favorites by pressing Ctrl+Shift+F

Microsoft Azure

Home > Network foundation | Route tables > rt-func-snet

rt-func-snet | Subnets

Route table

Associate

Search subnets

Name	Address range	Virtual network	Security group
AzureFirewallSubnet	10.0.0.0/26	azurebank-vn	-

Give feedback

Saved route table for subnet
Successfully saved route table for subnet 'AzureFirewallSubnet'.

Overview | Activity log | Access control (IAM) | Tags | Diagnose and solve problems | Resource visualizer | Settings | Configuration | Routes | Subnets | Properties | Locks | Monitoring | Automation | Help

Add or remove favorites by pressing Ctrl+Shift+F

Microsoft Azure

Home > Microsoft.AzureFirewall-20251209203055 | Overview > Azurebanks-firewalls > azurebank-policy

azurebank-policy | Network rules

Firewall Policy

Add a rule collection | Add rule | Edit | Delete

Rules are shown in the order of execution below. Network rules take precedence over application rules regardless of priority. Within the same rule collection type, inherited rules take precedence over rule collection group priority and rule collection priority.

Rule Collection P...	Rule collection n...	Rule name	Source	Port	Protocol	Destination	Action	Inherited
100	allow-sql-cosmos	sql-pe	10.10.1.0/24	1433	TCP	10.0.0.4	Allow	
100	allow-sql-cosmos	cosmos-pe	10.10.1.0/24	443	TCP	10.0.0.5	Allow	

Search to filter items...

Overview | Activity log | Access control (IAM) | Tags | Resource visualizer | Management | Draft + Deployment (preview) | Rules | Rule collections | DNAT rules | Network rules | Application rules | Settings | Monitoring | Automation | Help

Add or remove favorites by pressing Ctrl+Shift+F

Microsoft Azure Firewall Policy | Application rules

Overview Activity log Access control (IAM) Tags Resource visualizer Management Draft + Deployment (preview) Rules Rule collections DNAT rules Network rules Application rules

Add a rule collection Add rule Edit Delete

Rules are shown in the order of execution below. Network rules take precedence over application rules regardless of priority. Within the same rule collection type, inherited rules take precedence over rule collection group priority and rule collection priority.

Rule Collection P..	Rule collection n..	Rule name	Source	Protocol	Destination	Action	Inherited from
200	allow-azure-services	event-grid	○ 10.10.1.0/24	Https:443	○ *.eventgrid.azure... Allow		...
200	allow-azure-services	service-bus	○ 10.10.1.0/24	Https:443	○ *.servicebus.windo... Allow		...

Azurebanks-firewalls | Diagnostic settings

Maintenance Properties Locks Monitoring Metrics Diagnostic settings Logs Workbooks CLI / PS Tasks Export template Help Resource health Packet capture Support + Troubleshooting

Refresh Feedback

Diagnostic settings are used to configure streaming export of platform logs and metrics for a resource to the destination of your choice. You may create up to five different diagnostic settings to send different logs and metrics to independent destinations. Learn more about diagnostic settings

Name	Storage account	Event hub	Log Analytics workspace	Partner solution	Edit setting
Ds-settings	azurebankrg9e4b	-	DefaultWorkspace-f4b3a09a-994	-	Edit setting

Add or remove favorites by pressing Ctrl+Shift+F

BankDWH (azurebank-server/BankDWH) | Query editor (preview)

SQL database

Search Login New Query Open query Feedback Getting started

Query editor (preview) is a tool to run SQL queries against Azure SQL Database in the Azure portal. It is designed for lightweight querying and object exploration in your database. For more information and troubleshooting, Learn more

Welcome to SQL Database Query Editor

SQL server authentication

Login * azurbank
Password *

OR

Reason: An instance-specific error occurred while establishing a connection to SQL Server. Connection was denied because Deny Public Network Access is set to Yes. For more information, see <https://go.microsoft.com/fwlink/?linkid=2323206>.

Microsoft Entra authentication

Database 'BankDWH' on server 'azurbank-server.database.windows.net' is not currently available. Please retry the connection later. If the problem persists, contact customer support, and provide them the session tracing ID of 'FBBABD849-605E-4E69-9AD2-E22D9935E54A'.

Continue as 2100032603@kluniversity...

OK

7. CI/CD Pipelines

The implemented CI/CD concepts for three main artifact types.

7.1 Azure Functions – CI/CD

- Connected your Function App code repository (GitHub / Azure DevOps).
- Setup a pipeline (e.g., GitHub Actions workflow) that:
 - Triggers on push to main branch.
 - Restores dependencies, builds, and runs basic tests.
 - Deploys new version of Functions to Azure.

For deployment strategy, you referred to:

- Slot-based or blue-green deployment – deploying to a staging slot first, then swapping to production.

```
name: Azure Function CI/CD
on:
  push:
    paths:
      - "functions/**"
    branches:
      - main
jobs:
  build-and-deploy:
    runs-on: ubuntu-latest
    defaults:
      run:
        working-directory: functions
    steps:
      - name: Checkout repo
        uses: actions/checkout@v4
      - name: Setup Python
        uses: actions/setup-python@v5
        with:
          python-version: "3.11"

```

Delta compression using up to 12 threads
Compressing objects: 100% (3/3), done.
Writing objects: 100% (5/5), 434 bytes | 434.00 KiB/s, done.
total 5 (delta 2), reused 0 (delta 0), pack-reused 0 (from 0)
remote: Resolving deltas: 100% (2/2), completed with 2 local objects.
To https://github.com/chandrashekar2845/Banking-Domain-Project.git
 21f9c1..ff9ed8 main -> main
branch 'main' set up to track 'origin/main'.

7.2 Databricks Notebooks – Sync & Deployment

- The code for ETL notebooks was stored in a Git repository.
- You used:
 - Databricks Git integration or
 - Workspace export/import and job scheduling (conceptually)

To align with requirement:

“PySpark Notebooks – Automated sync/deployment to Databricks/Synapse – Workspace API Deployment.”

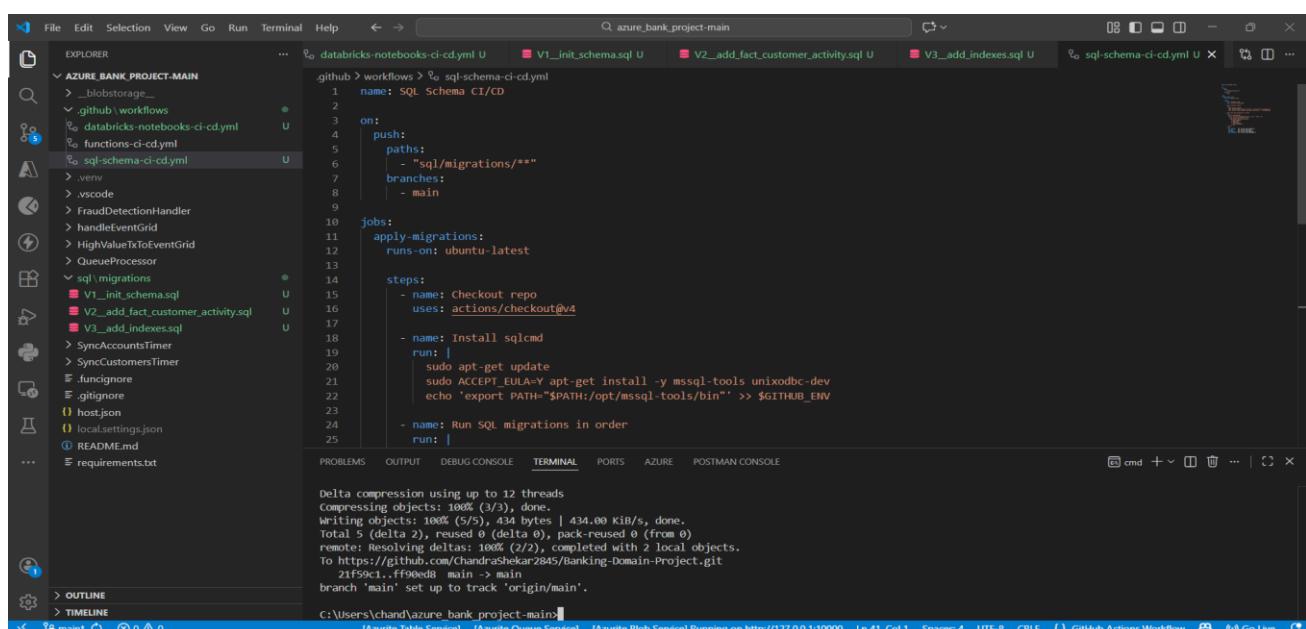
You described:

- How notebooks can be automatically imported or updated via Databricks REST API.
- How scheduled jobs can run ETL pipelines regularly.

7.3 SQL Schema – Migration Approach

For the data warehouse schema:

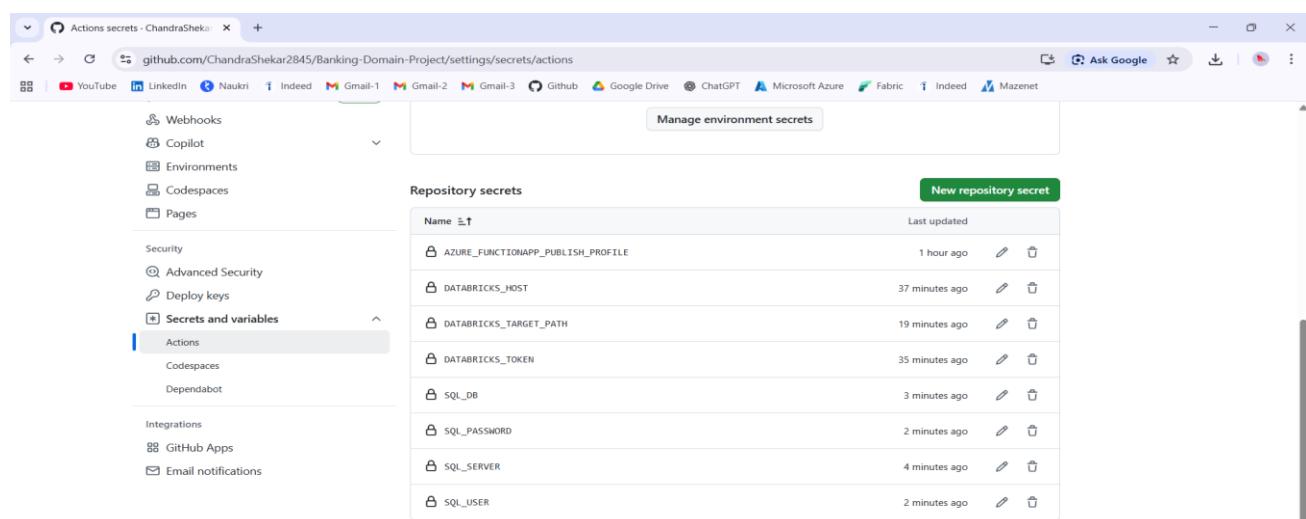
- The DDL scripts (for creating Dim/Fact tables) were stored in source control.
- You referred to tools like Flyway/Liquibase for real CI/CD migration in production, and explained conceptually:
 - Each schema change = a migration script with a version number.
 - Pipelines apply migrations in order to keep DB schema consistent.



```

name: SQL Schema CI/CD
on:
  push:
    paths:
      - "sql/migrations/**"
    branches:
      - main
jobs:
  apply-migrations:
    runs-on: ubuntu-latest
    steps:
      - name: Checkout repo
        uses: actions/checkout@v4
      - name: Install sqlcmd
        run: |
          sudo apt-get update
          sudo ACCEPT_EULA=Y apt-get install -y mssql-tools unixodbc-dev
          echo "export PATH=\"$PATH:/opt/mssql-tools/bin\"" >> $GITHUB_ENV
      - name: Run SQL migrations in order
        run: |
          branch `main` set up to track `origin/main`.

```



Name	Last updated
AZURE_FUNCTIONAPP_PUBLISH_PROFILE	1 hour ago
DATABRICKS_HOST	37 minutes ago
DATABRICKS_TARGET_PATH	19 minutes ago
DATABRICKS_TOKEN	35 minutes ago
SQL_DB	3 minutes ago
SQL_PASSWORD	2 minutes ago
SQL_SERVER	4 minutes ago
SQL_USER	2 minutes ago

8. Analytics & Dashboards

8.1 Customer 360 Analytical View

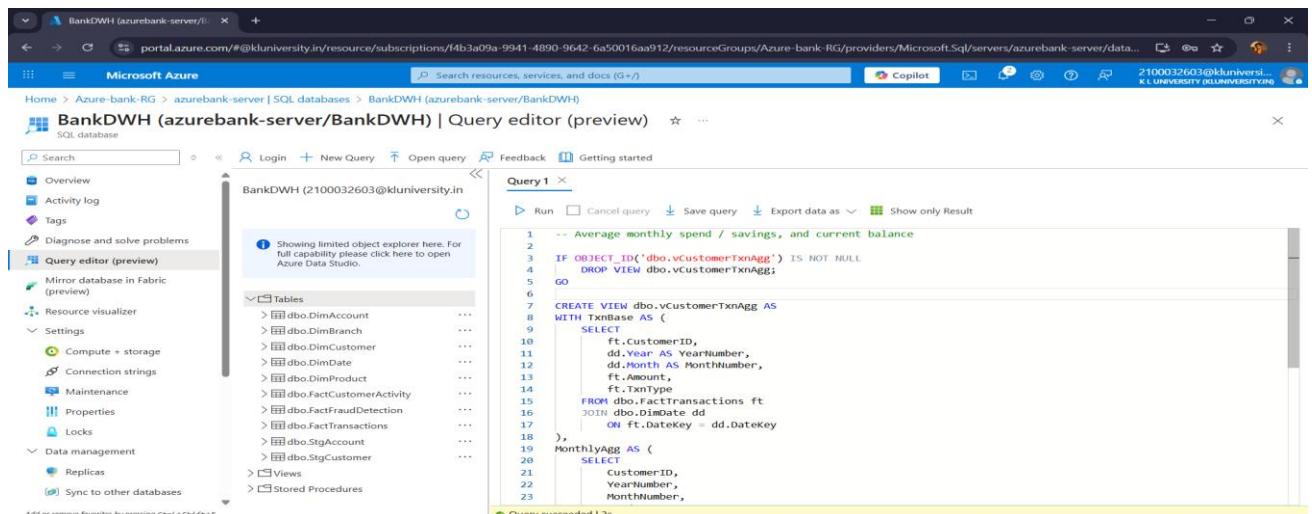
You designed a Customer 360 View that combines:

- Transaction history from FactTransactions.
- Customer master details from DimCustomer.
- Account details from DimAccount.
- Fraud metrics from FactFraudDetection.

Using SQL views, you:

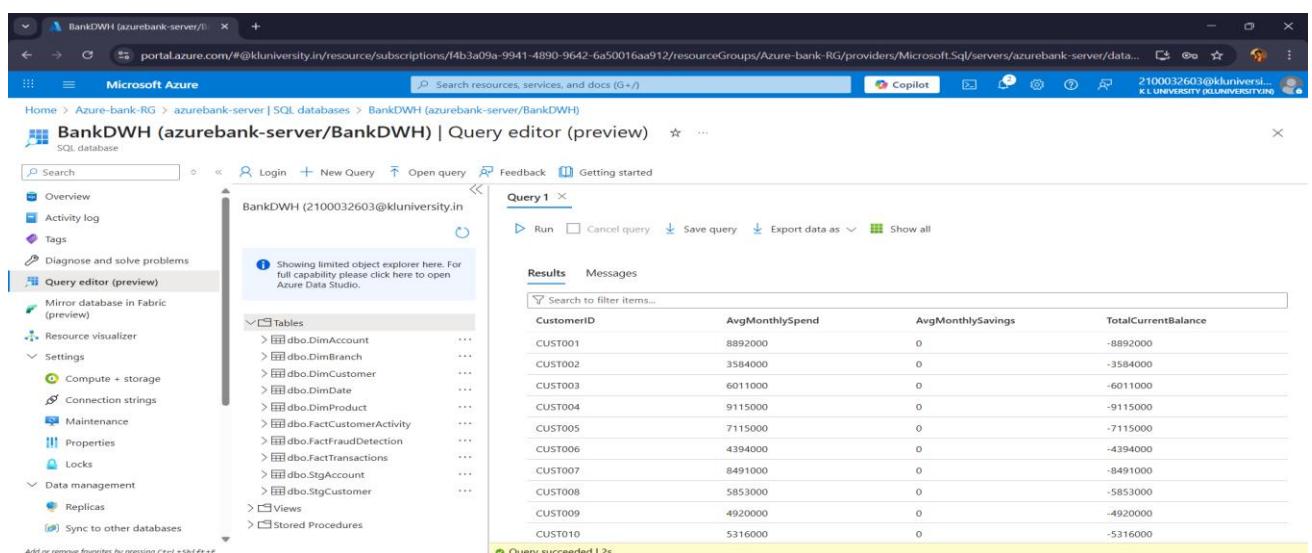
- Calculated average monthly spend and savings per customer.
- Derived current balance approximation based on debits/credits.
- Counted number of accounts (savings, loans) held per customer.
- Calculated simple Risk/Fraud scores based on fraud alerts, frequency, and amount.

This becomes the main dataset for Customer 360 in Power BI.



```
-- Average monthly spend / savings, and current balance
IF OBJECT_ID('dbo.vCustomerTxnAgg') IS NOT NULL
    DROP VIEW dbo.vCustomerTxnAgg;
GO

CREATE VIEW dbo.vCustomerTxnAgg AS
WITH TxnAgg AS (
    SELECT
        ft.CustomerID,
        dd.Year AS YearNumber,
        dd.Month AS MonthNumber,
        ft.Amount,
        ft.AlertType,
        ft.DateKey
    FROM dbo.FactTransactions ft
    JOIN dbo.DimDate dd
        ON ft.DateKey = dd.DateKey
),
MonthlyAgg AS (
    SELECT
        CustomerID,
        YearNumber,
        MonthNumber,
        SUM(Amount) AS AvgMonthlySpend,
        SUM(CASE WHEN AlertType = 'Fraud' THEN -1 ELSE 1 END * Amount) AS AvgMonthlySavings
    FROM TxnAgg
    GROUP BY CustomerID, YearNumber, MonthNumber
)
SELECT
    CustomerID,
    YearNumber,
    MonthNumber,
    AvgMonthlySpend,
    AvgMonthlySavings,
    SUM(CASE WHEN AlertType = 'Fraud' THEN -1 ELSE 1 END * Amount) AS TotalCurrentBalance
FROM MonthlyAgg
```



CustomerID	AvgMonthlySpend	AvgMonthlySavings	TotalCurrentBalance
CUST001	8892000	0	-8892000
CUST002	3584000	0	-3584000
CUST003	6011000	0	-6011000
CUST004	9115000	0	-9115000
CUST005	7115000	0	-7115000
CUST006	4394000	0	-4394000
CUST007	8491000	0	-8491000
CUST008	5853000	0	-5853000
CUST009	4920000	0	-4920000
CUST010	5316000	0	-5316000

BankDWH (azurebank-server/BankDWH) | Query editor (preview)

```

45 -- Product Holdings / Accounts
46 IF OBJECT_ID('dbo.vCustomerProductAgg') IS NOT NULL
47 DROP VIEW dbo.vCustomerProductAgg;
48 GO
49
50
51 CREATE VIEW dbo.vCustomerProductAgg AS
52 SELECT
53     sa.CustomerID,
54     COUNT(*) AS TotalAccounts,
55     SUM(CASE WHEN da.AccountType = 'Savings' THEN 1 ELSE 0 END) AS SavingsAccountsCount,
56     SUM(CASE WHEN da.AccountType = 'Loan' THEN 1 ELSE 0 END) AS LoanAccountsCount
57 FROM dbo.dimAccount da
58 JOIN dbo.stgAccount sa
59     ON da.AccountNumber = sa.AccountNumber
60 WHERE da.is_active = 1
61 GROUP BY sa.CustomerID;
62 GO
63
64 SELECT TOP 10 * FROM dbo.vCustomerProductAgg;
65
66
67

```

Query succeeded | 0s

BankDWH (azurebank-server/BankDWH) | Query editor (preview)

CustomerID	TotalAccounts	SavingsAccountsCount	LoanAccountsCount
CUST001	39	39	0
CUST002	36	36	0
CUST003	45	45	0
CUST004	41	41	0
CUST005	37	37	0
CUST006	42	42	0
CUST007	42	42	0
CUST008	37	37	0
CUST009	41	41	0
CUST010	38	38	0

Query succeeded | 0s

BankDWH (azurebank-server/BankDWH) | Query editor (preview)

```

66 -- Fraud Metrics & Fraud Score
67 IF OBJECT_ID('dbo.vCustomerFraudAgg') IS NOT NULL
68 DROP VIEW dbo.vCustomerFraudAgg;
69 GO
70
71 CREATE VIEW dbo.vCustomerFraudAgg AS
72 SELECT
73     ft.CustomerID,
74     SUM(ff.FraudCount) AS TotalFraudCases,
75     SUM(ff.TotalFraudAmount) AS TotalFraudAmount,
76     MIN(ff.FraudDate) AS FirstFraudDate,
77     MAX(ff.FraudDate) AS LastFraudDate
78 FROM dbo.FactFraudDetection ff
79 JOIN dbo.FactTransactions ft
80     ON ff.TransactionID = ft.TransactionID
81 GROUP BY ft.CustomerID;
82 GO
83
84 SELECT TOP 10 * FROM dbo.vCustomerFraudAgg;
85
86
87
88

```

Query succeeded | 1s

BankDWH (azurebank-server/BankDWH) | Query editor (preview)

Showing limited object explorer here. For full capability please click here to open Azure Data Studio.

Tables

- dbo.DimAccount
- dbo.DimBranch
- dbo.DimCustomer
- dbo.DimDate
- dbo.DimProduct
- dbo.FactCustomerActivity
- dbo.FactFraudDetection
- dbo.FactTransactions
- dbo.StgAccount
- dbo.StgCustomer

CustomerID TotalFraudCases TotalFraudAmount FirstFraudDate LastFraudDate

CustomerID	TotalFraudCases	TotalFraudAmount	FirstFraudDate	LastFraudDate
CUST088	195	14675000	2025-12-07	2025-12-07
CUST292	241	17455000	2025-12-07	2025-12-07
CUST364	256	19010000	2025-12-07	2025-12-07
CUST445	137	11415000	2025-12-07	2025-12-07
CUST033	167	12005000	2025-12-07	2025-12-07
CUST436	171	10605000	2025-12-07	2025-12-07
CUST035	105	6755000	2025-12-07	2025-12-07
CUST181	196	12870000	2025-12-07	2025-12-07
CUST095	126	8100000	2025-12-07	2025-12-07
CUST066	161	11605000	2025-12-07	2025-12-07

Query succeeded | 1s

BankDWH (azurebank-server/BankDWH) | Query editor (preview)

Showing limited object explorer here. For full capability please click here to open Azure Data Studio.

Tables

- dbo.DimAccount
- dbo.DimBranch
- dbo.DimCustomer
- dbo.DimDate
- dbo.DimProduct
- dbo.FactCustomerActivity
- dbo.FactFraudDetection
- dbo.FactTransactions
- dbo.StgAccount
- dbo.StgCustomer

```
85 -- Customer Activity / Device & Login
86
87 SELECT
88     b.CustomerID,
89     MAX(d.FullDate) AS LastLoginDate,
90     COUNT(*) AS TotalActivityCount,
91
92
93
```

Results Messages

Query succeeded: Affected rows: 0

Query succeeded | 0s

BankDWH (azurebank-server/BankDWH) | Query editor (preview)

-- Final Customer360 View

```
CREATE| dbo.vCustomer360 AS
SELECT
    c.CustomerID,
    c.customer_name AS FullName,
    c.StartDate AS CustomerStartDate,
    c.EndDate AS CustomerEndDate,
    c.IsActive AS IsActiveCustomer,
    COALESCE(pa.NumberOfProducts, 0) AS TotalProducts,
    COALESCE(fa.FraudEventsCount, 0) AS FraudEventsCount,
    COALESCE(fa.FraudScore, 0) AS FraudScore,
    COALESCE(fa.TotalFraudLoss, 0) AS TotalFraudLoss,
    COALESCE(act.LastLoginDate, NULL) AS LastLoginDate,
    COALESCE(act.LoginCountLast30Days, 0) AS LoginCountLast30Days
FROM dbo.DimCustomer c
LEFT JOIN dbo.vCustomerProductAgg pa
    ON pa.CustomerID = c.CustomerID
LEFT JOIN dbo.vCustomerFraudAgg fa
    ON fa.CustomerID = c.CustomerID
```

Add or remove favorites by pressing **Ctrl+Shift+F**

The screenshot shows the Microsoft Azure Query editor (preview) interface. The left sidebar contains navigation links for Overview, Activity log, Tags, Diagnose and solve problems, and a expanded section for Query editor (preview) which includes Mirror database in Fabric (preview), Resource visualizer, Settings (Compute + storage, Connection strings, Maintenance, Properties, Locks), Data management (Replicas, Sync to other databases), and Add or remove favorites by pressing **Ctrl+Shift+F**. The main area displays a table titled "Query 1" with the following columns: CustomerID, FullName, CustomerStartDate, CustomerEndDate, IsActiveCustomer, and TotalProducts. The data shows 11 rows of customer information, with the last row (CUST011) highlighted.

CustomerID	FullName	CustomerStartDate	CustomerEndDate	IsActiveCustomer	TotalProducts
CUST001	CUST001	2025-12-08T20:23:34.2130000		True	2
CUST002	CUST002	2025-12-08T20:23:34.2130000		True	2
CUST003	CUST003	2025-12-08T20:23:34.2130000		True	2
CUST004	CUST004	2025-12-08T20:23:34.2130000		True	2
CUST005	CUST005	2025-12-08T20:23:34.2130000		True	2
CUST006	CUST006	2025-12-08T20:23:34.2130000		True	2
CUST007	CUST007	2025-12-08T20:23:34.2130000		True	2
CUST008	CUST008	2025-12-08T20:23:34.2130000		True	2
CUST009	CUST009	2025-12-08T20:23:34.2130000		True	2
CUST010	CUST010	2025-12-08T20:23:34.2130000		True	2
CUST011	CUST011	2025-12-08T20:23:34.2130000		True	2

The screenshot shows the Microsoft Azure Query editor (preview) interface, identical to the one above but with a different query result. The main area displays a table titled "Query 1" with the following columns: TotalProducts, FraudEventsCount, FraudScore, TotalFraudLoss, LastLoginDate, and LoginCountLast30Days. The data shows 11 rows of fraud-related metrics, with the last row (TotalProducts: 2) highlighted.

TotalProducts	FraudEventsCount	FraudScore	TotalFraudLoss	LastLoginDate	LoginCountLast30Days
2	216	0	15350000.00		0
2	123	0	8475000.00		0
2	127	0	8545000.00		0
2	251	0	15895000.00		0
2	195	0	12175000.00		0
2	67	0	4855000.00		0
2	212	0	13430000.00		0
2	134	0	9160000.00		0
2	152	0	10320000.00		0
2	103	0	7445000.00		0
2	155	0	12405000.00		0

8.2 Power BI Dashboards

Finally, you connected Power BI Desktop to the Azure SQL Database (BankDW).

You designed multiple dashboards:

8.2.1 Branch Performance Dashboard

- Metrics: total transaction amount, number of customers, etc.
- Visuals: bar charts by branch (or you described branch dimension conceptually).

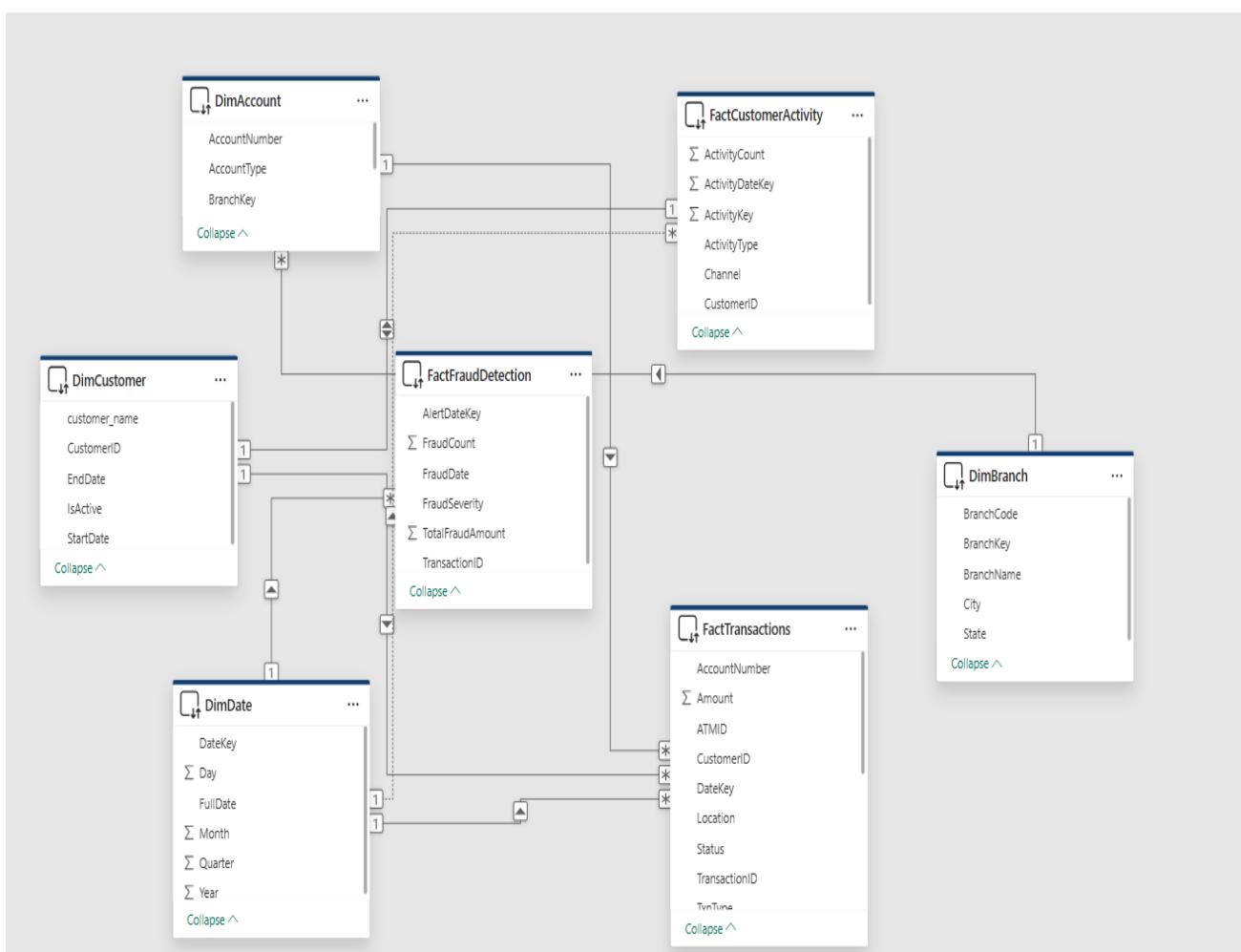
Untitled - Power BI Desktop

File Home Insert Modeling View Optimize Help

Cut Copy Format painter Paste Get data from clipboard Excel OneLake SQL Server Data Datasource Recent sources Transform data Queries New visual Text box More visual Insert New visual calculation New measure Quick Calculations Sensitivity Share Publish Prep data for Copilot AI Copilot

Build visuals with your data
Select or drag fields from the Data pane onto the report canvas.

Storage Mode: DirectQuery (click to change) 80%



Branch Performance Dashboard

Year
2023 2025



3.42bn

Total Transaction Amount

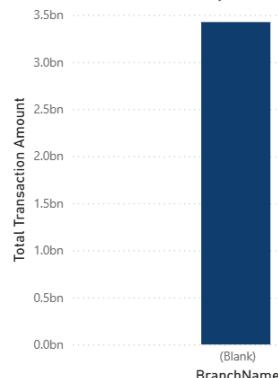
17.12M

Total Revenue

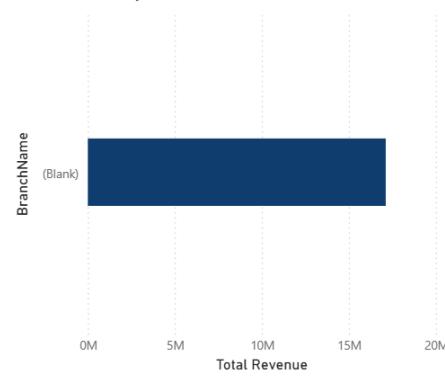
500

Active Customers

Total Transaction Amount by BranchName



Total Revenue by BranchName



Total Transaction Amount by City



8.2.2 Daily Transaction Volume Dashboard

- Shows trends over time: daily transaction counts and amounts.
- Visuals: line charts, date slicers, filters by channel (ATM / UPI).

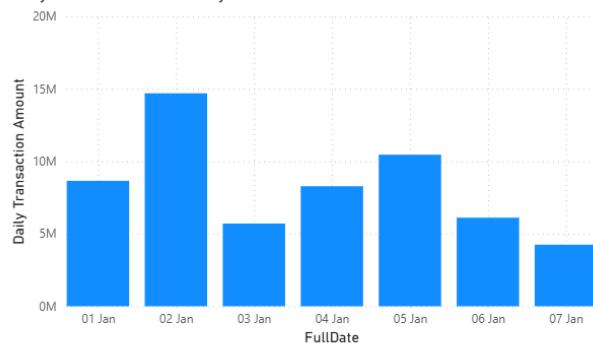
Daily Transaction Volume Dashboard

Daily Transaction Amount and Txn Amount 7D MA by FullDate

● Daily Transaction Amount ● Txn Amount 7D MA



Daily Transaction Amount by FullDate



ATMID

- ATM001
- ATM002
- ATM003
- ATM004
- ATM005
- ATM006

CustomerID

- CUST003
- CUST006
- CUST011
- CUST014
- CUST019
- CUST020

Location

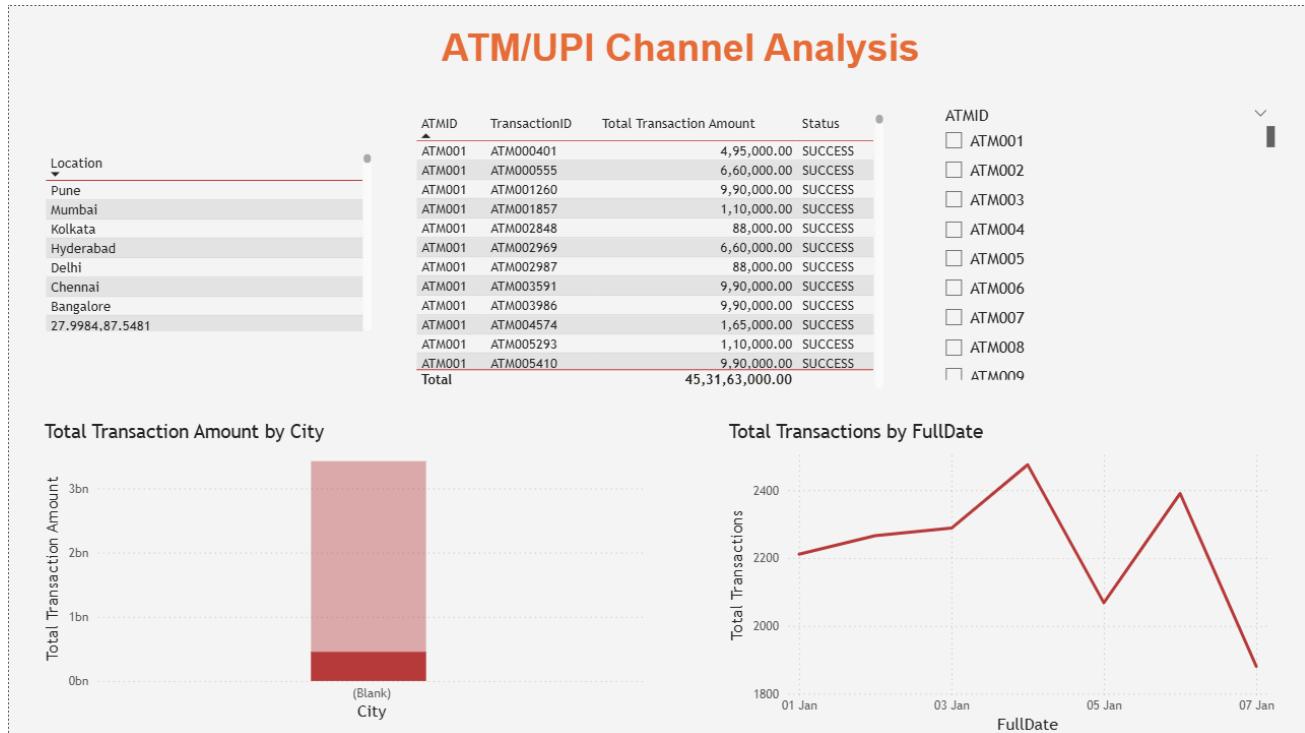
- Pune
- Mumbai
- Kolkata
- Hyderabad
- Delhi
- Chennai
- Bangalore

2395

Daily Transactions

8.2.3 ATM/UPI Channel Analysis

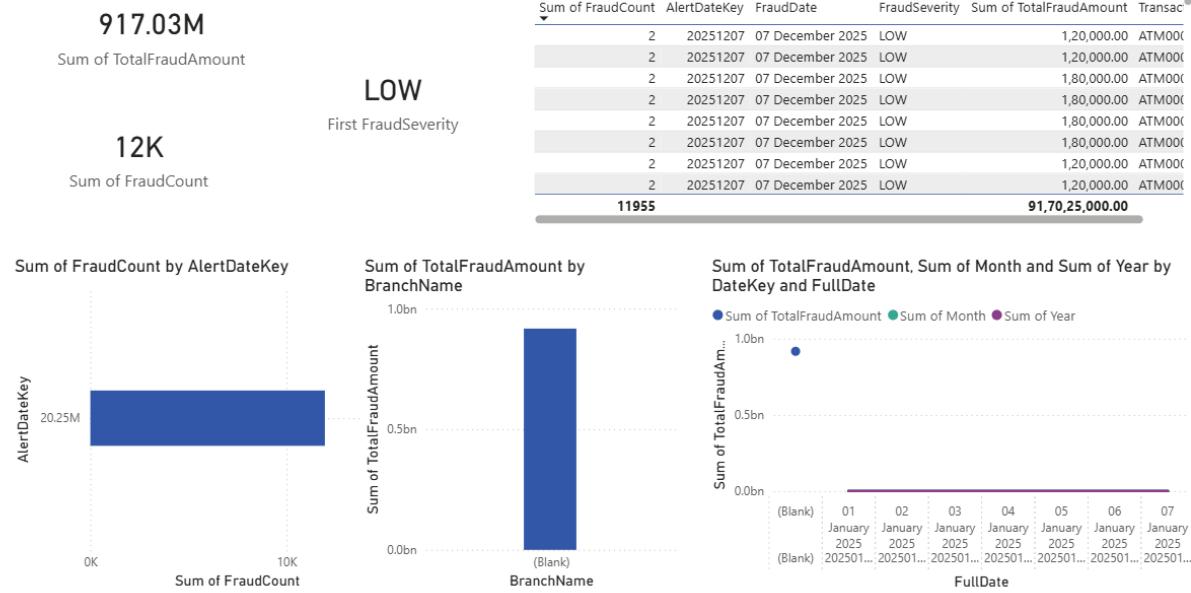
- Success vs failure ratios.
- Peak transaction hours.
- Geo-based distribution of ATM and UPI activity.



8.2.4 Fraud Analytics Dashboard

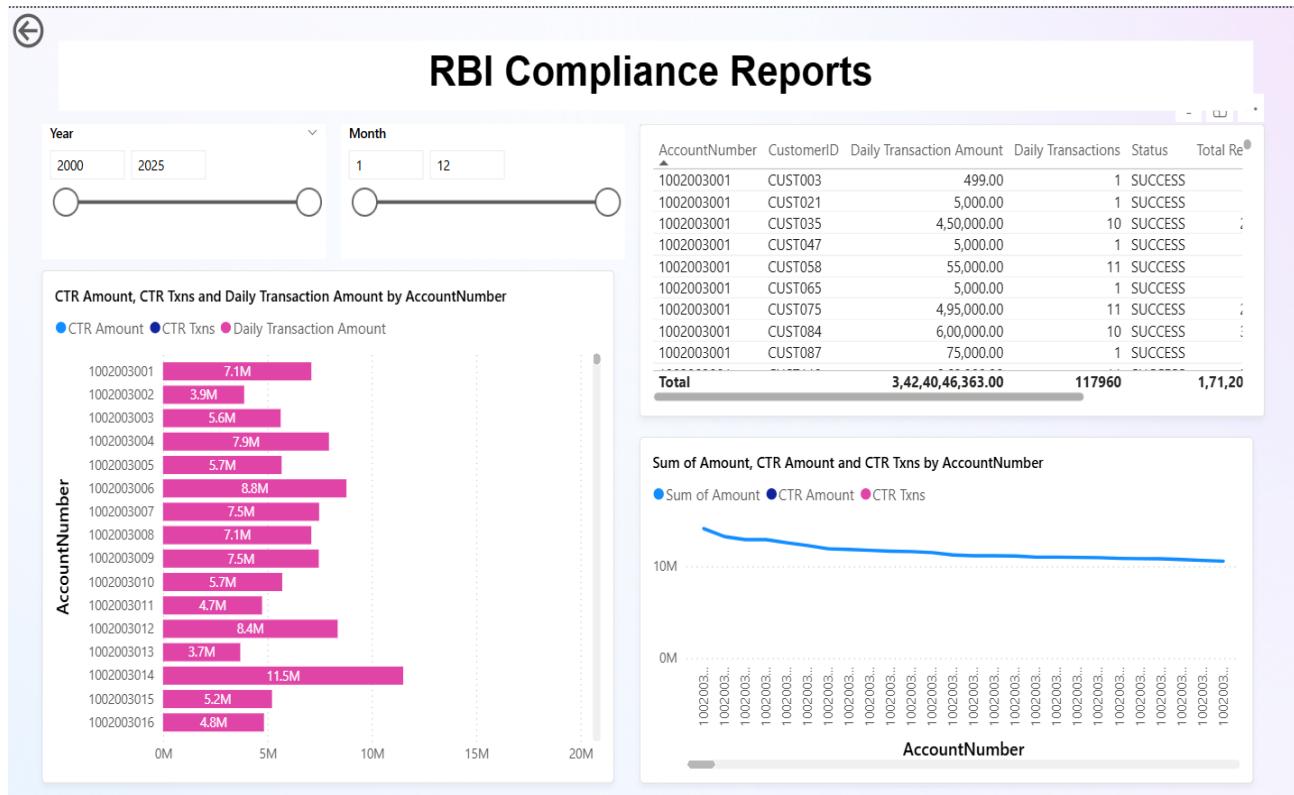
- Number of fraud alerts by type (high-value, unusual UPI, etc.).
- Total potential loss amount.
- Top risky customers/accounts.
- Timeline of alerts.

Fraud Analytics Dashboard



8.2.5 RBI Compliance / Regulatory View

- Summary tables and charts that can be used for compliance reports – e.g., large cash transactions, suspicious activity counts per period.



9. End-to-End Flow Summary

Putting it all together, your platform works like this:

1. Data Ingestion
 - Source files and systems push data into ADLS raw or Cosmos DB.
 - New file events trigger Event Grid, which notifies Azure Functions.
2. Serverless Processing & Fraud Detection
 - EventGrid/HTTP Function pushes message to Service Bus queue.
 - Queue-trigger Function downloads data, cleans it, and writes:
 - Detailed records into Cosmos DB containers (ATM, UPI, FraudAlerts).
 - Optionally raw copies into data lake.
 - Fraud rules identify high-risk transactions and log them separately.
3. Data Lake – Bronze / Silver / Gold
 - Databricks reads from Cosmos, writes raw to Bronze.
 - Silver layer applies robust cleaning and normalization.
 - Gold layer builds FactTransactions, DimCustomer, DimAccount, FactFraudDetection.
4. Data Warehouse & Sync
 - Azure SQL stores the star schema.
 - Timer-trigger Functions and Databricks jobs keep Dim and Fact tables updated daily using MERGE/UPSERT logic.
5. Security
 - Firewall rules restrict access to Cosmos DB and SQL.
 - Only trusted IPs/services or VNets can connect.
6. CI/CD
 - Function Apps, Notebooks, and SQL schema are tied to Git-based pipelines for repeatable deployments.
7. Analytics & Dashboards
 - Power BI connects to BankDW and Customer 360 View.
 - Business stakeholders get insights about:
 - Customer behaviour,
 - Channel usage (ATM/UPI),
 - Fraud trends,
 - Compliance summaries.

10. Conclusion & Learnings

10.1 Achievements

Through this project, you successfully:

- Built a cloud-native data platform on Azure.
- Implemented event-driven ingestion using Event Grid, Service Bus and Azure Functions.
- Designed and implemented a multi-layer data lake (Bronze/Silver/Gold) on ADLS.
- Integrated a NoSQL operational store (Cosmos DB) with a relational warehouse (SQL).
- Implemented fraud detection logic and a FraudAlerts store.
- Created a star schema with fact and dimension tables.
- Automated data workflows using Databricks and Functions.
- Applied network security principles via firewall/network restrictions.
- Demonstrated CI/CD practices for Functions, SQL, and notebooks.
- Developed Power BI dashboards that deliver real business value.

10.2 Key Learnings

- Importance of data modelling (star schema) for analytics performance.
- Challenges of connecting different Azure services (Cosmos ↔ Databricks ↔ SQL).
- Real-world handling of secrets, keys, and firewall rules.
- Power of serverless functions in building scalable real-time pipelines.
- Understanding how to turn raw logs into business-ready insights.

10.3 Future Enhancements

- Implement more advanced ML-based fraud detection models.
- Fully enforce private endpoints + Azure Firewall for all services.
- Introduce real SCD Type-2 logic for DimCustomer and DimAccount.
- Integrate with external notification systems (email/SMS gateways) for live customer alerts.
- Expand dashboards to include credit risk scoring, portfolio analysis, etc.