INFOCEPTS
Data & AI

FutureCart CRM

Project Report

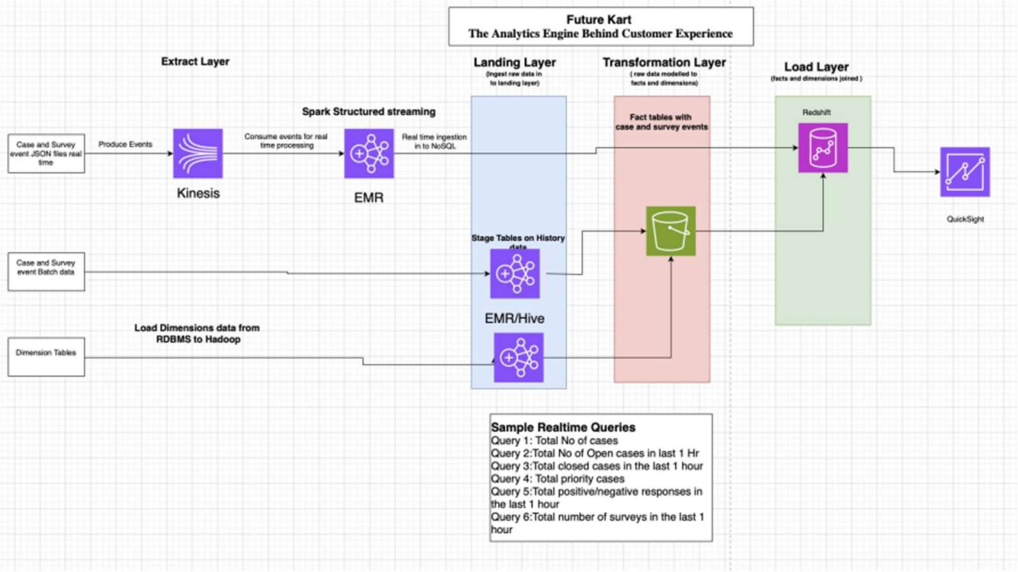
By
Aahash Kamble

Problem Statement

FutureCart Inc., a leading retail and e-commerce company in India, aims to enhance its customer retention and service quality by leveraging a modern CRM analytics platform. The company operates multiple contact centers and collects customer interaction data through various channels such as calls, chats, and emails. To transform this data into actionable insights, FutureCart needs a robust data infrastructure that supports both batch and real-time data processing.

The core challenge is to build a CRM Data Mart using a Lambda Architecture that can ingest, process, and analyze both historical and streaming data. This system should enable the generation of real-time and batch KPIs to support strategic decision-making and improve customer satisfaction.

Overview: FutureCart CRM – Lambda Architecture for Real-Time and Batch Analytics Flow Digram


Future Kart
The Analytics Engine Behind Customer Experience
Extract Layer
Landing Layer
(ingest raw data in to landing layer)
Transformation Layer
(raw data modelled to facts and dimensions)
Load Layer
(facts and dimensions joined)
Spark Structured streaming
Case and Survey event JSON files real time
Produce Events
Kinesis
Consume events for real time processing
EMR
Real time ingestion in to NoSQL
Fact tables with case and survey events
Redshift
QuickSight
Stage Tables on History data
Case and Survey event Batch data
EMR/Hive
Load Dimensions data from RDBMS to Hadoop
Dimension Tables

Sample Realtime Queries
Query 1: Total No of cases
Query 2:Total No of Open cases in last 1 Hr
Query 3:Total closed cases in the last 1 hour
Query 4: Total priority cases
Query 5:Total positive/negative responses in the last 1 hour
Query 6:Total number of surveys in the last 1 hour

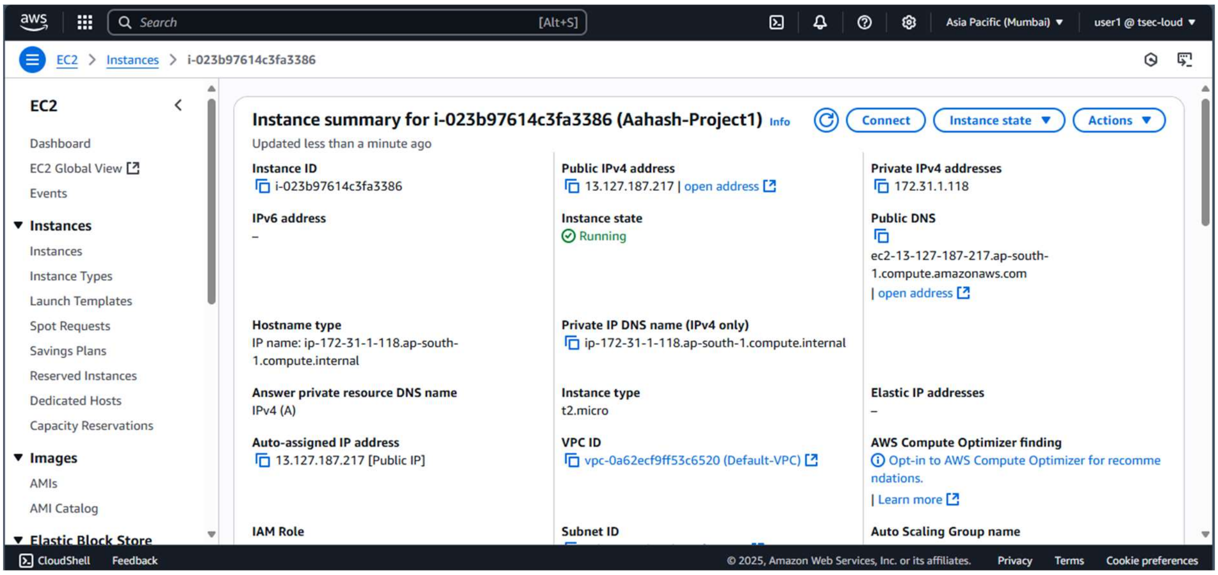
Key Technical Objectives

1. Set up MySQL on EC2 to store dimension data.
2. Ingest dimension data into Hive for batch processing.
3. Generate and load historical case and survey data into HDFS.
4. Stream real-time events (cases and surveys) using AWS Kinesis.
5. Process real-time data using Spark Structured Streaming and load it into Amazon Redshift.
6. Export historical and dimension data from Hive to S3, then load into Redshift.
7. Run analytical queries on Redshift to compute KPIs.

Phase 1: Infrastructure Setup & Initial Configuration

Tasks:

- Launched **EC2 instances** for local processing.
- Installed **MySQL 5.7** and configured schema.
- Listed all S3 buckets and verified access to aahash-project.
- Created schema and tables in **MySQL** for dimension data.



Instance summary for i-023b97614c3fa3386 (Aahash-Project1)

Updated less than a minute ago

Instance ID: i-023b97614c3fa3386

Public IPv4 address: 13.127.187.217 | open address

Private IPv4 addresses: 172.31.1.118

IPv6 address: –

Instance state: Running

Public DNS: ec2-13-127-187-217.ap-south-1.compute.amazonaws.com | open address

Hostname type: IP name: ip-172-31-1-118.ap-south-1.compute.internal

Private IP DNS name (IPv4 only): ip-172-31-1-118.ap-south-1.compute.internal

Answer private resource DNS name: IPv4 (A)

Instance type: t2.micro

Elastic IP addresses: –

Auto-assigned IP address: 13.127.187.217 [Public IP]

VPC ID: vpc-0a62ecf9ff53c6520 (Default-VPC)

AWS Compute Optimizer finding: Opt-in to AWS Compute Optimizer for recommendations. | Learn more

IAM Role

Subnet ID

Auto Scaling Group name

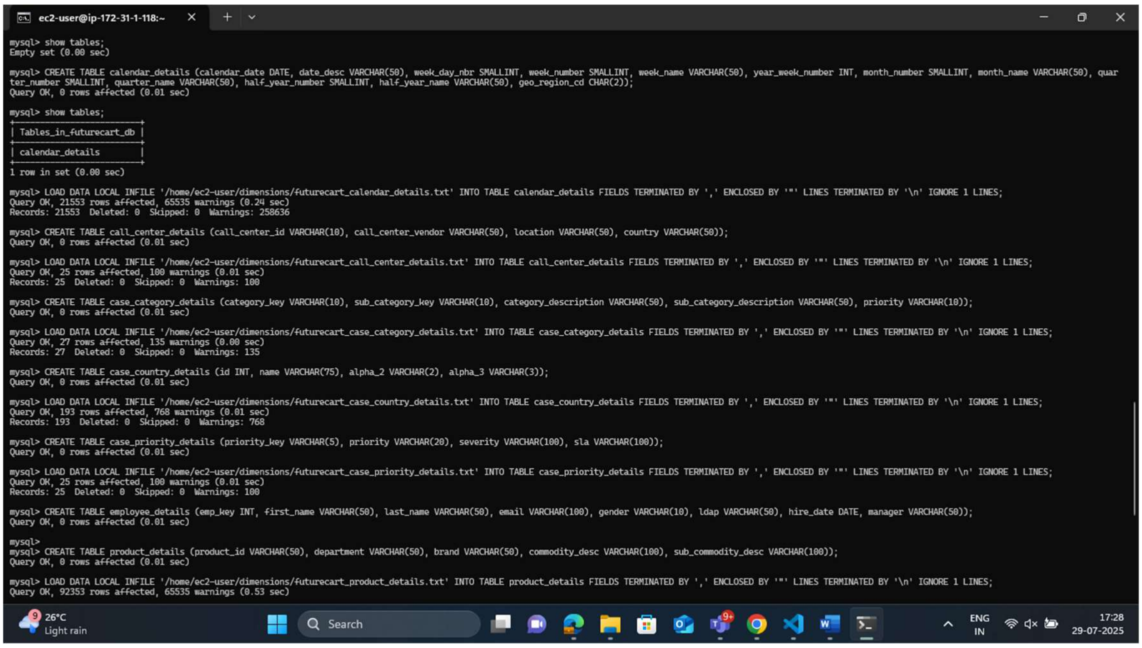
Install and setup Mysql
sudo yum update -y
sudo yum install -y mariadb-server
sudo systemctl start mariadb
sudo systemctl enable mariadb

Create MySQL Tables and Load Data

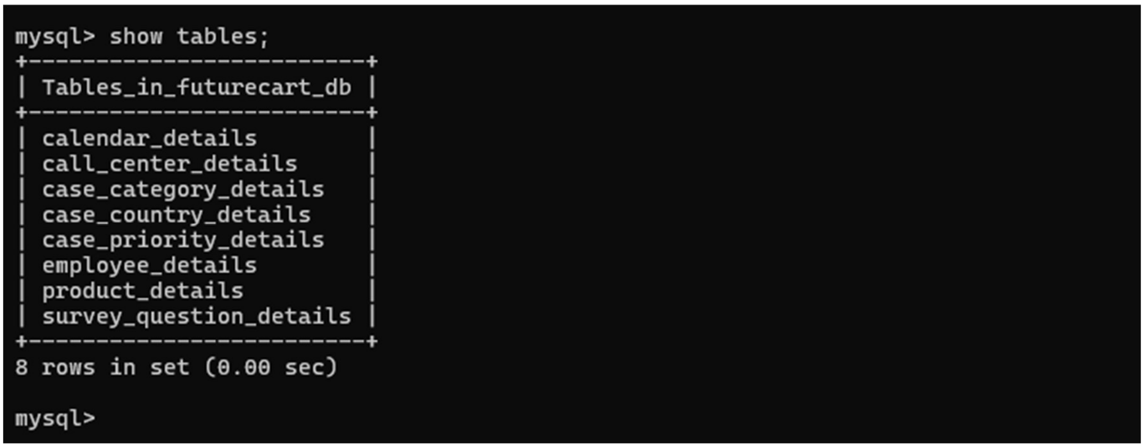
CREATE TABLE calendar_details (calendar_date DATE, date_desc VARCHAR(50), week_day_nbr SMALLINT, week_number SMALLINT, week_name VARCHAR(50), year_week_number INT, month_number SMALLINT, month_name VARCHAR(50), quarter_number SMALLINT, quarter_name VARCHAR(50), half_year_number SMALLINT, half_year_name VARCHAR(50), geo_region_cd CHAR(2));

LOAD DATA LOCAL INFILE '/home/ec2-user/dimensions/futurecart_calendar_details.txt' INTO TABLE calendar_details FIELDS TERMINATED BY ',' ENCLOSED BY '"' LINES TERMINATED BY '\n' IGNORE 1 LINES;

Tables are created :



```
mysql> show tables;
Empty set (0.00 sec)

mysql> CREATE TABLE calendar_details (calendar_date DATE, date_desc VARCHAR(50), week_day_nbr SMALLINT, week_number SMALLINT, week_name VARCHAR(50), year_week_number INT, month_number SMALLINT, month_name VARCHAR(50), quar
ter_number SMALLINT, quarter_name VARCHAR(50), half_year_number SMALLINT, half_year_name VARCHAR(50), geo_region_cd CHAR(2));
Query OK, 0 rows affected (0.01 sec)

mysql> show tables;
+-------------------------+
| Tables_in_futurecart_db |
+-------------------------+
| calendar_details        |
+-------------------------+
1 row in set (0.00 sec)

mysql> LOAD DATA LOCAL INFILE '/home/ec2-user/dimensions/futurecart_calendar_details.txt' INTO TABLE calendar_details FIELDS TERMINATED BY ',' ENCLOSED BY '"' LINES TERMINATED BY '\n' IGNORE 1 LINES;
Query OK, 21553 rows affected, 65535 warnings (0.24 sec)
Records: 21553  Deleted: 0  Skipped: 0  Warnings: 258636

mysql> CREATE TABLE call_center_details (call_center_id VARCHAR(10), call_center_vendor VARCHAR(50), location VARCHAR(50), country VARCHAR(50));
Query OK, 0 rows affected (0.01 sec)

mysql> LOAD DATA LOCAL INFILE '/home/ec2-user/dimensions/futurecart_call_center_details.txt' INTO TABLE call_center_details FIELDS TERMINATED BY ',' ENCLOSED BY '"' LINES TERMINATED BY '\n' IGNORE 1 LINES;
Query OK, 25 rows affected, 100 warnings (0.01 sec)
Records: 25  Deleted: 0  Skipped: 0  Warnings: 100

mysql> CREATE TABLE case_category_details (category_key VARCHAR(10), sub_category_key VARCHAR(10), category_description VARCHAR(50), sub_category_description VARCHAR(50), priority VARCHAR(10));
Query OK, 0 rows affected (0.01 sec)

mysql> LOAD DATA LOCAL INFILE '/home/ec2-user/dimensions/futurecart_case_category_details.txt' INTO TABLE case_category_details FIELDS TERMINATED BY ',' ENCLOSED BY '"' LINES TERMINATED BY '\n' IGNORE 1 LINES;
Query OK, 27 rows affected, 135 warnings (0.00 sec)
Records: 27  Deleted: 0  Skipped: 0  Warnings: 135

mysql> CREATE TABLE case_country_details (id INT, name VARCHAR(75), alpha_2 VARCHAR(2), alpha_3 VARCHAR(3));
Query OK, 0 rows affected (0.01 sec)

mysql> LOAD DATA LOCAL INFILE '/home/ec2-user/dimensions/futurecart_case_country_details.txt' INTO TABLE case_country_details FIELDS TERMINATED BY ',' ENCLOSED BY '"' LINES TERMINATED BY '\n' IGNORE 1 LINES;
Query OK, 193 rows affected, 768 warnings (0.01 sec)
Records: 193  Deleted: 0  Skipped: 0  Warnings: 768

mysql> CREATE TABLE case_priority_details (priority_key VARCHAR(5), priority VARCHAR(20), severity VARCHAR(100), sla VARCHAR(100));
Query OK, 0 rows affected (0.01 sec)

mysql> LOAD DATA LOCAL INFILE '/home/ec2-user/dimensions/futurecart_case_priority_details.txt' INTO TABLE case_priority_details FIELDS TERMINATED BY ',' ENCLOSED BY '"' LINES TERMINATED BY '\n' IGNORE 1 LINES;
Query OK, 25 rows affected, 100 warnings (0.01 sec)
Records: 25  Deleted: 0  Skipped: 0  Warnings: 100

mysql> CREATE TABLE employee_details (emp_key INT, first_name VARCHAR(50), last_name VARCHAR(50), email VARCHAR(100), gender VARCHAR(10), ldap VARCHAR(50), hire_date DATE, manager VARCHAR(50));
Query OK, 0 rows affected (0.01 sec)

mysql>
mysql> CREATE TABLE product_details (product_id VARCHAR(50), department VARCHAR(50), brand VARCHAR(50), commodity_desc VARCHAR(100), sub_commodity_desc VARCHAR(100));
Query OK, 0 rows affected (0.01 sec)

mysql> LOAD DATA LOCAL INFILE '/home/ec2-user/dimensions/futurecart_product_details.txt' INTO TABLE product_details FIELDS TERMINATED BY ',' ENCLOSED BY '"' LINES TERMINATED BY '\n' IGNORE 1 LINES;
Query OK, 92353 rows affected, 65535 warnings (0.53 sec)
```



```
mysql> show tables;
+-------------------------+
| Tables_in_futurecart_db |
+-------------------------+
| calendar_details        |
| call_center_details     |
| case_category_details   |
| case_country_details    |
| case_priority_details   |
| employee_details        |
| product_details         |
| survey_question_details |
+-------------------------+
8 rows in set (0.00 sec)

mysql>
```

Phase 2: Dimension Data Preparation & Upload

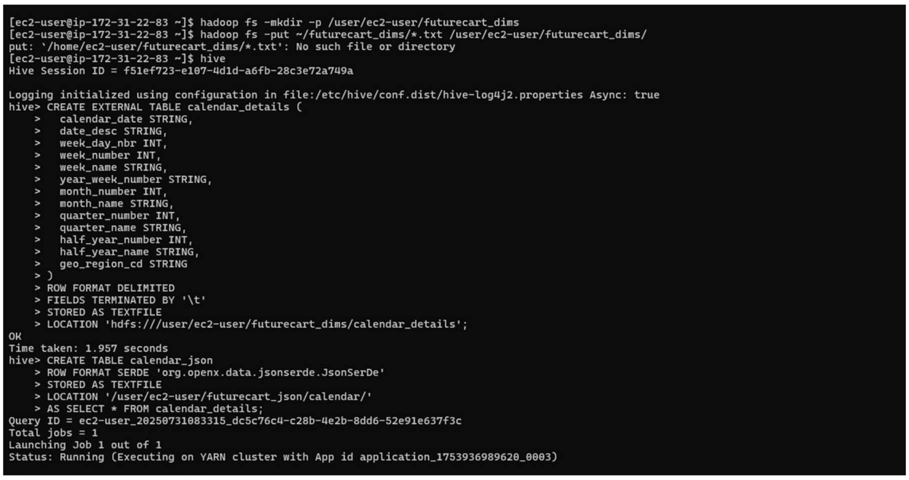
Tasks:

- Downloaded *.txt files from s3://aahash-project/dimensions/.
- Uploaded these files to EC2 instance in ~/futurecart_dims/.
- Loaded these files into **HDFS** via Hadoop command-line.
- Created **Hive tables** and loaded the data into Hive.
- Exported Hive tables as **JSON** files.
- Uploaded the JSON exports to s3://aahash-project/dimensions/.

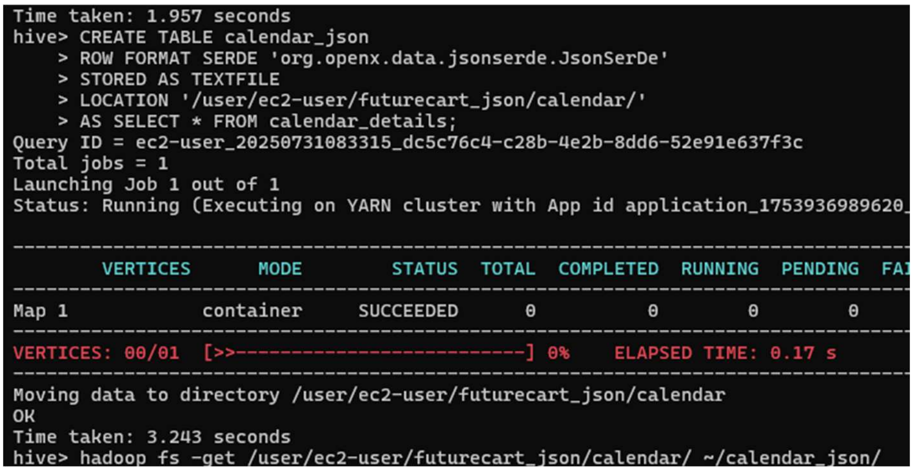
Load Data to Hadoop (EMR)

```
aws s3 cp s3://aahash-project/dimensions/ ~/futurecart_dims/ --recursive
hadoop fs -mkdir -p /user/ec2-user/futurecart_dims
hadoop fs -put ~/futurecart_dims/*.txt /user/ec2-user/futurecart_dims/
```

```
Create Hive external tables
CREATE EXTERNAL TABLE calendar_details (
  calendar_date STRING,
  date_desc STRING,
  week_day_nbr INT,
  week_number INT,
  week_name STRING,
  year_week_number STRING,
  month_number INT,
  month_name STRING,
  quarter_number INT,
  quarter_name STRING,
  half_year_number INT,
  half_year_name STRING,
  geo_region_cd STRING
)
ROW FORMAT DELIMITED
FIELDS TERMINATED BY '\t'
STORED AS TEXTFILE

LOCATION 'hdfs:///user/ec2-user/futurecart_dims/calendar_details';
```



```
[ec2-user@ip-172-31-22-83 ~]$ hadoop fs -mkdir -p /user/ec2-user/futurecart_dims
[ec2-user@ip-172-31-22-83 ~]$ hadoop fs -put ~/futurecart_dims/*.txt /user/ec2-user/futurecart_dims/
put: `/home/ec2-user/futurecart_dims/*.txt': No such file or directory
[ec2-user@ip-172-31-22-83 ~]$ hive
Hive Session ID = f51ef723-e107-4d1d-a6fb-28c3e72a749a

Logging initialized using configuration in file:/etc/hive/conf.dist/hive-log4j2.properties Async: true
hive> CREATE EXTERNAL TABLE calendar_details (
    >   calendar_date STRING,
    >   date_desc STRING,
    >   week_day_nbr INT,
    >   week_number INT,
    >   week_name STRING,
    >   year_week_number STRING,
    >   month_number INT,
    >   month_name STRING,
    >   quarter_number INT,
    >   quarter_name STRING,
    >   half_year_number INT,
    >   half_year_name STRING,
    >   geo_region_cd STRING
    > )
    > ROW FORMAT DELIMITED
    > FIELDS TERMINATED BY '\t'
    > STORED AS TEXTFILE
    > LOCATION 'hdfs:///user/ec2-user/futurecart_dims/calendar_details';
OK
Time taken: 1.957 seconds
hive> CREATE TABLE calendar_json
    > ROW FORMAT SERDE 'org.openx.data.jsonserde.JsonSerDe'
    > STORED AS TEXTFILE
    > LOCATION '/user/ec2-user/futurecart_json/calendar/'
    > AS SELECT * FROM calendar_details;
Query ID = ec2-user_20250731083315_dc5c76c4-c28b-4e2b-8dd6-52e91e637f3c
Total jobs = 1
Launching Job 1 out of 1
Status: Running (Executing on YARN cluster with App id application_1753936989620_0003)
```



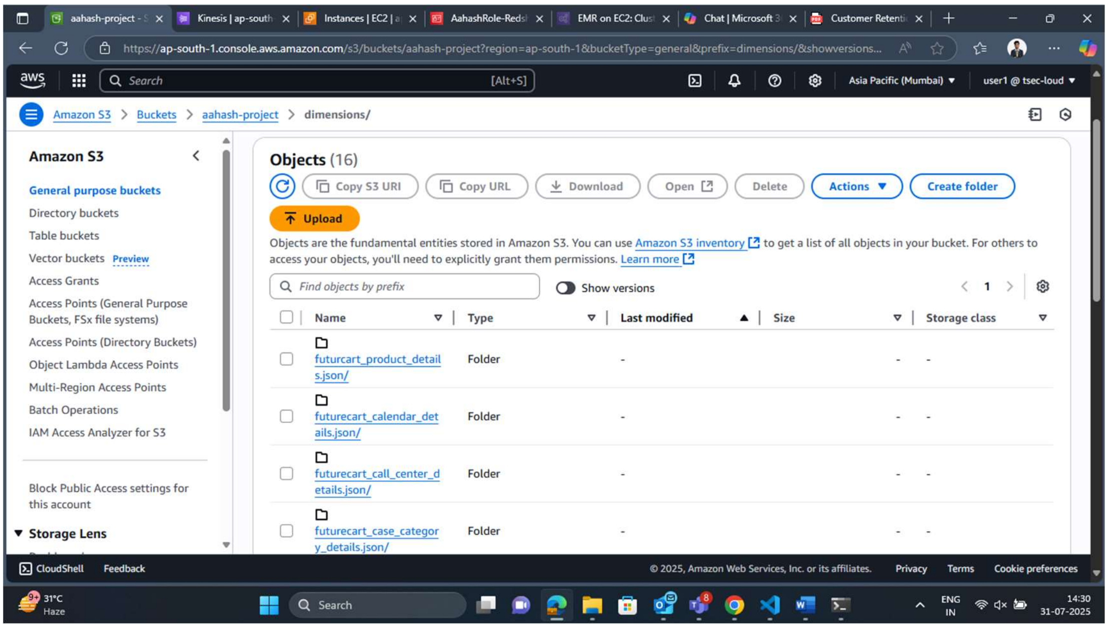
```
Time taken: 1.957 seconds
hive> CREATE TABLE calendar_json
    > ROW FORMAT SERDE 'org.openx.data.jsonserde.JsonSerDe'
    > STORED AS TEXTFILE
    > LOCATION '/user/ec2-user/futurecart_json/calendar/'
    > AS SELECT * FROM calendar_details;
Query ID = ec2-user_20250731083315_dc5c76c4-c28b-4e2b-8dd6-52e91e637f3c
Total jobs = 1
Launching Job 1 out of 1
Status: Running (Executing on YARN cluster with App id application_1753936989620_

----------------------------------------------------------------------------------
        VERTICES      MODE        STATUS  TOTAL  COMPLETED  RUNNING  PENDING  FAI
----------------------------------------------------------------------------------
Map 1            container     SUCCEEDED      0          0        0        0
----------------------------------------------------------------------------------
VERTICES: 00/01  [>>--------------------------] 0%    ELAPSED TIME: 0.17 s
----------------------------------------------------------------------------------
Moving data to directory /user/ec2-user/futurecart_json/calendar
OK
Time taken: 3.243 seconds
hive> hadoop fs -get /user/ec2-user/futurecart_json/calendar/ ~/calendar_json/
```

Copy JSON output from HDFS to local:

hadoop fs -get /user/ec2-user/futurecart_json/calendar/ ~/calendar_json/

Upload to S3:

aws s3 cp ~/calendar_json/ s3://aahash-project/dimensions/calendar_json/ --recursive



Objects (16)

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

□	Name	Type	Last modified	Size	Storage class
□	futurcart_product_details.json/	Folder	-	-	-
□	futurecart_calendar_details.json/	Folder	-	-	-
□	futurecart_call_center_details.json/	Folder	-	-	-
□	futurecart_case_category_details.json/	Folder	-	-	-

Phase 3: Historical Data Generation and Batch Processing

Tasks:

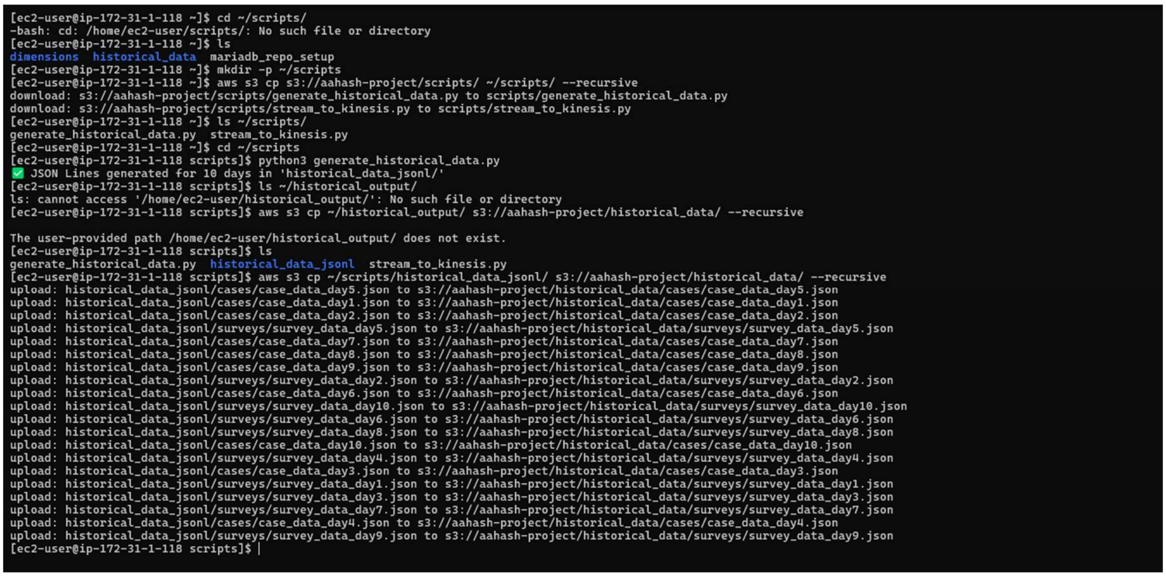
- Used a Python script to generate historical **case/survey** data in JSON.
- Uploaded the generated data to s3://aahash-project/historical-data/.

```
mkdir -p ~/scripts
copy scripts
aws s3 cp s3://aahash-project/scripts/ ~/scripts/ --recursive
cd ~/scripts
python3 generate_historical_data.py
aws s3 cp s3://aahash-project/historical_data/ ~/historical_data/ --recursive
```

Script to generate historical data

```
import os
import json
import random
from datetime import datetime, timedelta

# CONFIGURABLE PARAMETERS
NUM_DAYS = 10
CASES_PER_DAY = 100
SURVEYS_PER_DAY = 80  # must be ≤ CASES_PER_DAY
OUTPUT_DIR = "historical_data_jsonl"
def random_case_no():
   return str(random.randint(600000, 700000))
def random_timestamp(day_offset):
   date = datetime.now() - timedelta(days=day_offset)
   return date.strftime("%Y-%m-%d %H:%M:%S")
def generate_case(case_no, day_offset):
   return {
      "status": random.choice(["Open", "Closed"]),
      "category": f"CAT{random.randint(1,5)}",
      "sub_category": f"SCAT{random.randint(1,20)}",
      "last_modified_timestamp": random_timestamp(day_offset),
      "case_no": case_no,
      "create_timestamp": random_timestamp(day_offset),
      "created_employee_key": str(random.randint(200000, 300000)),
      "call_center_id": f"C-{random.randint(100,120)}",
      "product_code": str(random.randint(9000000, 9999999)),
      "country_cd": random.choice(["IN", "US", "BR", "DE", "AU"]),
      "communication_mode": random.choice(["Email", "Call", "Chat"])
   }

def generate_survey(case_no, day_offset):
   return {
      "survey_id": f"S-{random.randint(500000, 599999)}",
        "case_no": case_no,
        "survey_timestamp": random_timestamp(day_offset),
        "Q1": random.randint(1, 10),
        "Q2": random.randint(1, 10),
        "Q3": random.randint(1, 10),
        "Q4": random.choice(["Y", "N"]),
        "Q5": random.randint(1, 10)
    }

def ensure_dirs():
    os.makedirs(f"{OUTPUT_DIR}/cases", exist_ok=True)
    os.makedirs(f"{OUTPUT_DIR}/surveys", exist_ok=True)

def main():
    ensure_dirs()
    for day in range(1, NUM_DAYS + 1):
        case_nos = [random_case_no() for _ in range(CASES_PER_DAY)]
        cases = [generate_case(cn, day) for cn in case_nos]
        surveys = [generate_survey(cn, day) for cn in random.sample(case_nos, SURVEYS_PER_DAY)]

        # Write case data as JSON Lines
        with open(f"{OUTPUT_DIR}/cases/case_data_day{day}.json", "w") as f:
            for record in cases:
                f.write(json.dumps(record) + "\n")

        # Write survey data as JSON Lines
        with open(f"{OUTPUT_DIR}/surveys/survey_data_day{day}.json", "w") as f:
            for record in surveys:
                f.write(json.dumps(record) + "\n")

    print(f"✅ JSON Lines generated for {NUM_DAYS} days in '{OUTPUT_DIR}/'")

if __name__ == "__main__":
    main()
```

Loaded and generated Json case event



```
[ec2-user@ip-172-31-1-118 ~]$ cd ~/scripts/
-bash: cd: /home/ec2-user/scripts/: No such file or directory
[ec2-user@ip-172-31-1-118 ~]$ ls
dimensions  historical_data  mariadb_repo_setup
[ec2-user@ip-172-31-1-118 ~]$ mkdir -p ~/scripts
[ec2-user@ip-172-31-1-118 ~]$ aws s3 cp s3://aahash-project/scripts/ ~/scripts/ --recursive
download: s3://aahash-project/scripts/generate_historical_data.py to scripts/generate_historical_data.py
download: s3://aahash-project/scripts/stream_to_kinesis.py to scripts/stream_to_kinesis.py
[ec2-user@ip-172-31-1-118 ~]$ ls ~/scripts/
generate_historical_data.py  stream_to_kinesis.py
[ec2-user@ip-172-31-1-118 ~]$ cd ~/scripts
[ec2-user@ip-172-31-1-118 scripts]$ python3 generate_historical_data.py
✅ JSON Lines generated for 10 days in 'historical_data_jsonl/'
[ec2-user@ip-172-31-1-118 scripts]$ ls ~/historical_output/
ls: cannot access '/home/ec2-user/historical_output/': No such file or directory
[ec2-user@ip-172-31-1-118 scripts]$ aws s3 cp ~/historical_output/ s3://aahash-project/historical_data/ --recursive

The user-provided path /home/ec2-user/historical_output/ does not exist.
[ec2-user@ip-172-31-1-118 scripts]$ ls
generate_historical_data.py  historical_data_jsonl  stream_to_kinesis.py
[ec2-user@ip-172-31-1-118 scripts]$ aws s3 cp ~/scripts/historical_data_jsonl/ s3://aahash-project/historical_data/ --recursive
upload: historical_data_jsonl/cases/case_data_day5.json to s3://aahash-project/historical_data/cases/case_data_day5.json
upload: historical_data_jsonl/cases/case_data_day1.json to s3://aahash-project/historical_data/cases/case_data_day1.json
upload: historical_data_jsonl/cases/case_data_day2.json to s3://aahash-project/historical_data/cases/case_data_day2.json
upload: historical_data_jsonl/surveys/survey_data_day5.json to s3://aahash-project/historical_data/surveys/survey_data_day5.json
upload: historical_data_jsonl/cases/case_data_day7.json to s3://aahash-project/historical_data/cases/case_data_day7.json
upload: historical_data_jsonl/cases/case_data_day8.json to s3://aahash-project/historical_data/cases/case_data_day8.json
upload: historical_data_jsonl/cases/case_data_day9.json to s3://aahash-project/historical_data/cases/case_data_day9.json
upload: historical_data_jsonl/surveys/survey_data_day2.json to s3://aahash-project/historical_data/surveys/survey_data_day2.json
upload: historical_data_jsonl/cases/case_data_day6.json to s3://aahash-project/historical_data/cases/case_data_day6.json
upload: historical_data_jsonl/surveys/survey_data_day10.json to s3://aahash-project/historical_data/surveys/survey_data_day10.json
upload: historical_data_jsonl/surveys/survey_data_day6.json to s3://aahash-project/historical_data/surveys/survey_data_day6.json
upload: historical_data_jsonl/surveys/survey_data_day8.json to s3://aahash-project/historical_data/surveys/survey_data_day8.json
upload: historical_data_jsonl/cases/case_data_day10.json to s3://aahash-project/historical_data/cases/case_data_day10.json
upload: historical_data_jsonl/surveys/survey_data_day4.json to s3://aahash-project/historical_data/surveys/survey_data_day4.json
upload: historical_data_jsonl/cases/case_data_day3.json to s3://aahash-project/historical_data/cases/case_data_day3.json
upload: historical_data_jsonl/surveys/survey_data_day1.json to s3://aahash-project/historical_data/surveys/survey_data_day1.json
upload: historical_data_jsonl/surveys/survey_data_day3.json to s3://aahash-project/historical_data/surveys/survey_data_day3.json
upload: historical_data_jsonl/surveys/survey_data_day7.json to s3://aahash-project/historical_data/surveys/survey_data_day7.json
upload: historical_data_jsonl/cases/case_data_day4.json to s3://aahash-project/historical_data/cases/case_data_day4.json
upload: historical_data_jsonl/surveys/survey_data_day9.json to s3://aahash-project/historical_data/surveys/survey_data_day9.json
[ec2-user@ip-172-31-1-118 scripts]$ |
```

Phase 4: Real-Time Data Ingestion via Kinesis

Tasks:

- Created a **Kinesis Data Stream**: aahash-newstream.
- Built send_to_kinesis.py to push real-time events to the stream in JSON format.
- Validated Kinesis streaming using logs and AWS Console.

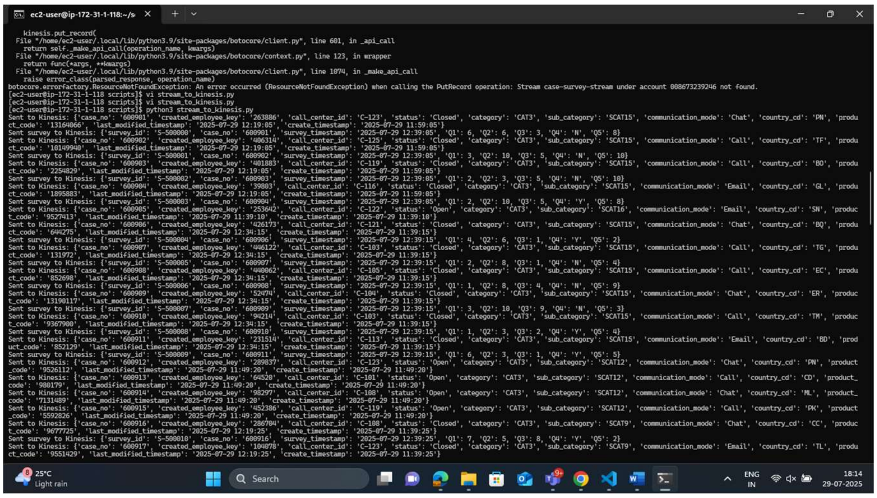
Send data to kinesis

STREAM_NAME = "aahash-data"

python3 stream_to_kinesis.py

```
import boto3
import random
import time
import calendar
import json
from datetime import datetime, timedelta

# Initialize Kinesis client
kinesis = boto3.client('kinesis', region_name='ap-south-1')
stream_name = 'aahash-stream'

# Read case data
with open('realtimedata/000000_0', 'r') as case_data_obj:
    all_case_data_lines = case_data_obj.readlines()

open_case_time_diff_mins = [40, 50, 60]
closed_case_time_diff_mins = [5, 10, 20, 30]
number_of_cases_counts = [1, 2, 3, 4, 5, 6]
scores = list(range(1, 11))
answer = ["Y", "N"]
survey_id_start = 500000
category = 'CAT3'
sub_categorys = ['SCAT8', 'SCAT9', 'SCAT10', 'SCAT11', 'SCAT12', 'SCAT13', 'SCAT14', 'SCAT15', 'SCAT16']

total_cases = len(all_case_data_lines)
i = 900

while i <= (total_cases - 1):
    sub_category = random.choice(sub_categorys)
    number_of_cases = random.choice(number_of_cases_counts)
    cases = all_case_data_lines[i:i + number_of_cases]

    current_timestamp = datetime.now()
    case_created_ts = str(current_timestamp - timedelta(minutes=random.choice(open_case_time_diff_mins)))[:19]
    case_closed_ts = str(current_timestamp - timedelta(minutes=random.choice(closed_case_time_diff_mins)))[:19]
    survey_ts = str(current_timestamp)[:19]
    file_ts = calendar.timegm(time.gmtime())

    for j in cases:
        case_no, created_employee, call_center, status, category1, sub_category1, mode, country, product = j.strip().split(',')

        case_data = {
            "case_no": case_no,
            "created_employee_key": created_employee,
            "call_center_id": call_center,
            "status": "Closed" if i % 5 == 0 else status,
            "category": category,
            "sub_category": sub_category,
            "communication_mode": mode,
            "country_cd": country,
            "product_code": product,
            "last_modified_timestamp": case_closed_ts if i % 5 == 0 else case_created_ts,
            "create_timestamp": case_created_ts
        }

        # Send to Kinesis
        kinesis.put_record(
            StreamName=stream_name,
            Data=json.dumps(case_data),
            PartitionKey=case_no
        )
        print("Sent to Kinesis:", case_data)

        if i % 5 == 0:
            survey_data = {
                "survey_id": f"S-{survey_id_start}",
                "case_no": case_no,
                "survey_timestamp": survey_ts,
                "Q1": random.choice(scores),
                "Q2": random.choice(scores),
                "Q3": random.choice(scores),
                "Q4": random.choice(answer),
                "Q5": random.choice(scores)
            }
            kinesis.put_record(
                StreamName=stream_name,
                Data=json.dumps(survey_data),
                PartitionKey=case_no
            )
            print("Sent survey to Kinesis:", survey_data)
            survey_id_start += 1

    i += number_of_cases
    time.sleep(5)
```

Verify stream data :



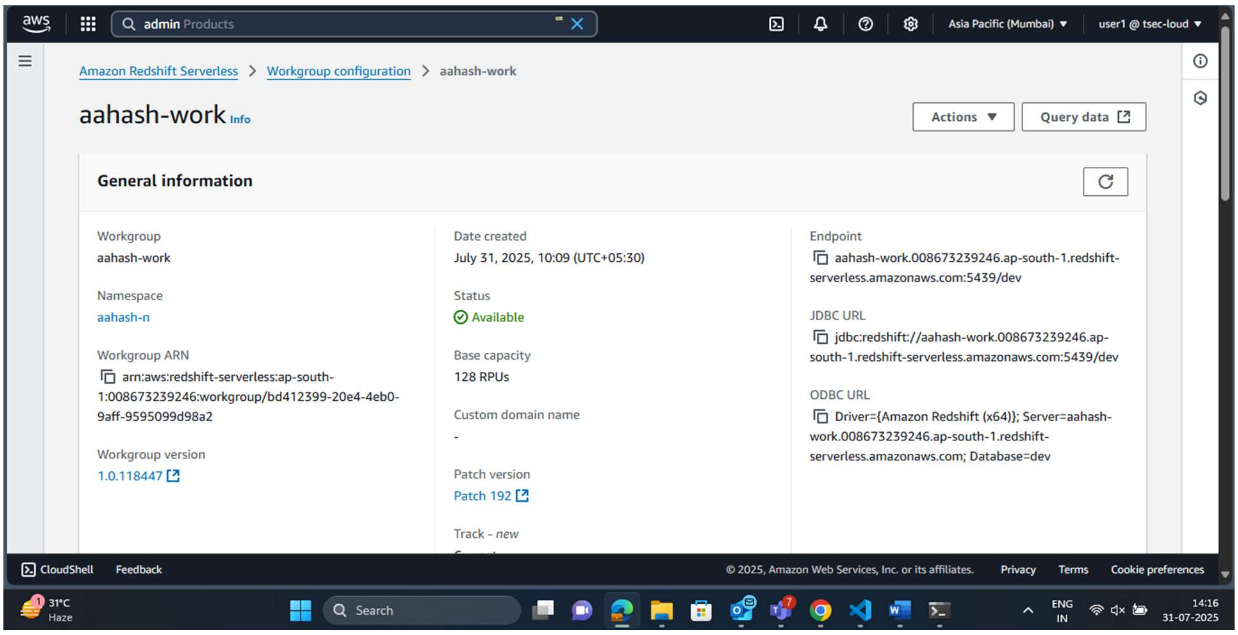
```
    kinesis.put_record(
  File "/home/ec2-user/.local/lib/python3.9/site-packages/botocore/client.py", line 601, in _api_call
    return self._make_api_call(operation_name, kwargs)
  File "/home/ec2-user/.local/lib/python3.9/site-packages/botocore/context.py", line 123, in wrapper
    return func(*args, **kwargs)
  File "/home/ec2-user/.local/lib/python3.9/site-packages/botocore/client.py", line 1074, in _make_api_call
    raise error_class(parsed_response, operation_name)
botocore.errorfactory.ResourceNotFoundException: An error occurred (ResourceNotFoundException) when calling the PutRecord operation: Stream case-survey-stream under account 008673239246 not found.
[ec2-user@ip-172-31-1-118 scripts]$ vi stream_to_kinesis.py
[ec2-user@ip-172-31-1-118 scripts]$ vi stream_to_kinesis.py
[ec2-user@ip-172-31-1-118 scripts]$ python3 stream_to_kinesis.py
Sent to Kinesis: {'case_no': '600901', 'created_employee_key': '263886', 'call_center_id': 'C-123', 'status': 'Closed', 'category': 'CAT3', 'sub_category': 'SCAT15', 'communication_mode': 'Chat', 'country_cd': 'PN', 'produ
ct_code': '13164066', 'last_modified_timestamp': '2025-07-29 12:19:05', 'create_timestamp': '2025-07-29 11:59:05'}
Sent survey to Kinesis: {'survey_id': 'S-500000', 'case_no': '600901', 'survey_timestamp': '2025-07-29 12:39:05', 'Q1': 6, 'Q2': 6, 'Q3': 3, 'Q4': 'N', 'Q5': 8}
Sent to Kinesis: {'case_no': '600902', 'created_employee_key': '406314', 'call_center_id': 'C-125', 'status': 'Closed', 'category': 'CAT3', 'sub_category': 'SCAT15', 'communication_mode': 'Call', 'country_cd': 'TF', 'produ
ct_code': '10149940', 'last_modified_timestamp': '2025-07-29 12:19:05', 'create_timestamp': '2025-07-29 11:59:05'}
Sent survey to Kinesis: {'survey_id': 'S-500001', 'case_no': '600902', 'survey_timestamp': '2025-07-29 12:39:05', 'Q1': 3, 'Q2': 10, 'Q3': 5, 'Q4': 'N', 'Q5': 10}
Sent to Kinesis: {'case_no': '600903', 'created_employee_key': '401883', 'call_center_id': 'C-119', 'status': 'Closed', 'category': 'CAT3', 'sub_category': 'SCAT15', 'communication_mode': 'Call', 'country_cd': 'BO', 'produ
ct_code': '2254829', 'last_modified_timestamp': '2025-07-29 12:19:05', 'create_timestamp': '2025-07-29 11:59:05'}
Sent survey to Kinesis: {'survey_id': 'S-500002', 'case_no': '600903', 'survey_timestamp': '2025-07-29 12:39:05', 'Q1': 2, 'Q2': 3, 'Q3': 5, 'Q4': 'N', 'Q5': 10}
Sent to Kinesis: {'case_no': '600904', 'created_employee_key': '39803', 'call_center_id': 'C-116', 'status': 'Closed', 'category': 'CAT3', 'sub_category': 'SCAT15', 'communication_mode': 'Email', 'country_cd': 'GL', 'produ
ct_code': '1895883', 'last_modified_timestamp': '2025-07-29 12:19:05', 'create_timestamp': '2025-07-29 11:59:05'}
Sent survey to Kinesis: {'survey_id': 'S-500003', 'case_no': '600904', 'survey_timestamp': '2025-07-29 12:39:05', 'Q1': 2, 'Q2': 10, 'Q3': 5, 'Q4': 'Y', 'Q5': 8}
Sent to Kinesis: {'case_no': '600905', 'created_employee_key': '253642', 'call_center_id': 'C-122', 'status': 'Open', 'category': 'CAT3', 'sub_category': 'SCAT16', 'communication_mode': 'Email', 'country_cd': 'SN', 'produc
t_code': '9527413', 'last_modified_timestamp': '2025-07-29 11:39:10', 'create_timestamp': '2025-07-29 11:39:10'}
Sent to Kinesis: {'case_no': '600906', 'created_employee_key': '426173', 'call_center_id': 'C-121', 'status': 'Closed', 'category': 'CAT3', 'sub_category': 'SCAT15', 'communication_mode': 'Chat', 'country_cd': 'BQ', 'produ
ct_code': '644275', 'last_modified_timestamp': '2025-07-29 12:34:15', 'create_timestamp': '2025-07-29 11:39:15'}
Sent survey to Kinesis: {'survey_id': 'S-500004', 'case_no': '600906', 'survey_timestamp': '2025-07-29 12:39:15', 'Q1': 4, 'Q2': 6, 'Q3': 1, 'Q4': 'Y', 'Q5': 2}
Sent to Kinesis: {'case_no': '600907', 'created_employee_key': '446122', 'call_center_id': 'C-103', 'status': 'Closed', 'category': 'CAT3', 'sub_category': 'SCAT15', 'communication_mode': 'Call', 'country_cd': 'TG', 'produ
ct_code': '131972', 'last_modified_timestamp': '2025-07-29 12:34:15', 'create_timestamp': '2025-07-29 11:39:15'}
Sent survey to Kinesis: {'survey_id': 'S-500005', 'case_no': '600907', 'survey_timestamp': '2025-07-29 12:39:15', 'Q1': 2, 'Q2': 8, 'Q3': 1, 'Q4': 'N', 'Q5': 4}
Sent to Kinesis: {'case_no': '600908', 'created_employee_key': '440062', 'call_center_id': 'C-105', 'status': 'Closed', 'category': 'CAT3', 'sub_category': 'SCAT15', 'communication_mode': 'Call', 'country_cd': 'EC', 'produ
ct_code': '852698', 'last_modified_timestamp': '2025-07-29 12:34:15', 'create_timestamp': '2025-07-29 11:39:15'}
Sent survey to Kinesis: {'survey_id': 'S-500006', 'case_no': '600908', 'survey_timestamp': '2025-07-29 12:39:15', 'Q1': 1, 'Q2': 8, 'Q3': 4, 'Q4': 'N', 'Q5': 9}
Sent to Kinesis: {'case_no': '600909', 'created_employee_key': '52474', 'call_center_id': 'C-104', 'status': 'Closed', 'category': 'CAT3', 'sub_category': 'SCAT15', 'communication_mode': 'Chat', 'country_cd': 'ER', 'produc
t_code': '13190117', 'last_modified_timestamp': '2025-07-29 12:34:15', 'create_timestamp': '2025-07-29 11:39:15'}
Sent survey to Kinesis: {'survey_id': 'S-500007', 'case_no': '600909', 'survey_timestamp': '2025-07-29 12:39:15', 'Q1': 3, 'Q2': 10, 'Q3': 9, 'Q4': 'N', 'Q5': 3}
Sent to Kinesis: {'case_no': '600910', 'created_employee_key': '94214', 'call_center_id': 'C-103', 'status': 'Closed', 'category': 'CAT3', 'sub_category': 'SCAT15', 'communication_mode': 'Call', 'country_cd': 'TM', 'produc
t_code': '9367900', 'last_modified_timestamp': '2025-07-29 12:34:15', 'create_timestamp': '2025-07-29 11:39:15'}
Sent survey to Kinesis: {'survey_id': 'S-500008', 'case_no': '600910', 'survey_timestamp': '2025-07-29 12:39:15', 'Q1': 1, 'Q2': 3, 'Q3': 2, 'Q4': 'Y', 'Q5': 4}
Sent to Kinesis: {'case_no': '600911', 'created_employee_key': '231514', 'call_center_id': 'C-113', 'status': 'Closed', 'category': 'CAT3', 'sub_category': 'SCAT15', 'communication_mode': 'Email', 'country_cd': 'BD', 'prod
uct_code': '852129', 'last_modified_timestamp': '2025-07-29 12:34:15', 'create_timestamp': '2025-07-29 11:39:15'}
Sent survey to Kinesis: {'survey_id': 'S-500009', 'case_no': '600911', 'survey_timestamp': '2025-07-29 12:39:15', 'Q1': 6, 'Q2': 3, 'Q3': 1, 'Q4': 'Y', 'Q5': 5}
Sent to Kinesis: {'case_no': '600912', 'created_employee_key': '289837', 'call_center_id': 'C-123', 'status': 'Open', 'category': 'CAT3', 'sub_category': 'SCAT12', 'communication_mode': 'Chat', 'country_cd': 'PN', 'product
_code': '9526112', 'last_modified_timestamp': '2025-07-29 11:49:20', 'create_timestamp': '2025-07-29 11:49:20'}
Sent to Kinesis: {'case_no': '600913', 'created_employee_key': '64520', 'call_center_id': 'C-101', 'status': 'Open', 'category': 'CAT3', 'sub_category': 'SCAT12', 'communication_mode': 'Call', 'country_cd': 'CD', 'product_
code': '980179', 'last_modified_timestamp': '2025-07-29 11:49:20', 'create_timestamp': '2025-07-29 11:49:20'}
Sent to Kinesis: {'case_no': '600914', 'created_employee_key': '98297', 'call_center_id': 'C-108', 'status': 'Open', 'category': 'CAT3', 'sub_category': 'SCAT12', 'communication_mode': 'Chat', 'country_cd': 'ML', 'product_
code': '7131489', 'last_modified_timestamp': '2025-07-29 11:49:20', 'create_timestamp': '2025-07-29 11:49:20'}
Sent to Kinesis: {'case_no': '600915', 'created_employee_key': '452386', 'call_center_id': 'C-119', 'status': 'Open', 'category': 'CAT3', 'sub_category': 'SCAT12', 'communication_mode': 'Call', 'country_cd': 'PK', 'product
_code': '5592826', 'last_modified_timestamp': '2025-07-29 11:49:20', 'create_timestamp': '2025-07-29 11:49:20'}
Sent to Kinesis: {'case_no': '600916', 'created_employee_key': '286704', 'call_center_id': 'C-108', 'status': 'Closed', 'category': 'CAT3', 'sub_category': 'SCAT9', 'communication_mode': 'Chat', 'country_cd': 'CC', 'produc
t_code': '9677725', 'last_modified_timestamp': '2025-07-29 12:19:25', 'create_timestamp': '2025-07-29 11:39:25'}
Sent survey to Kinesis: {'survey_id': 'S-500010', 'case_no': '600916', 'survey_timestamp': '2025-07-29 12:39:25', 'Q1': 7, 'Q2': 5, 'Q3': 8, 'Q4': 'Y', 'Q5': 2}
Sent to Kinesis: {'case_no': '600917', 'created_employee_key': '104078', 'call_center_id': 'C-123', 'status': 'Closed', 'category': 'CAT3', 'sub_category': 'SCAT9', 'communication_mode': 'Email', 'country_cd': 'TL', 'produ
ct_code': '9551429', 'last_modified_timestamp': '2025-07-29 12:19:25', 'create_timestamp': '2025-07-29 11:39:25'}
```

Phase 5: Real-Time Stream Processing with Spark

Tasks:

- Developed kinesis_to_redshift.py using Spark Structured Streaming.
- Set up Spark to ingest real-time JSON events from Kinesis stream (aahash-newstream).
- Transformed and batched incoming data for compatibility with Amazon Redshift schema.
- Wrote transformed data into Redshift tables using JDBC connector.

Created Redshift



Amazon Redshift Serverless > Workgroup configuration > aahash-work

aahash-work Info

General information

Workgroup
aahash-work

Namespace
aahash-n

Workgroup ARN
arn:aws:redshift-serverless:ap-south-1:008673239246:workgroup/bd412399-20e4-4eb0-9aff-9595099d98a2

Workgroup version
1.0.118447

Date created
July 31, 2025, 10:09 (UTC+05:30)

Status
Available

Base capacity
128 RPUs

Custom domain name
-

Patch version
Patch 192

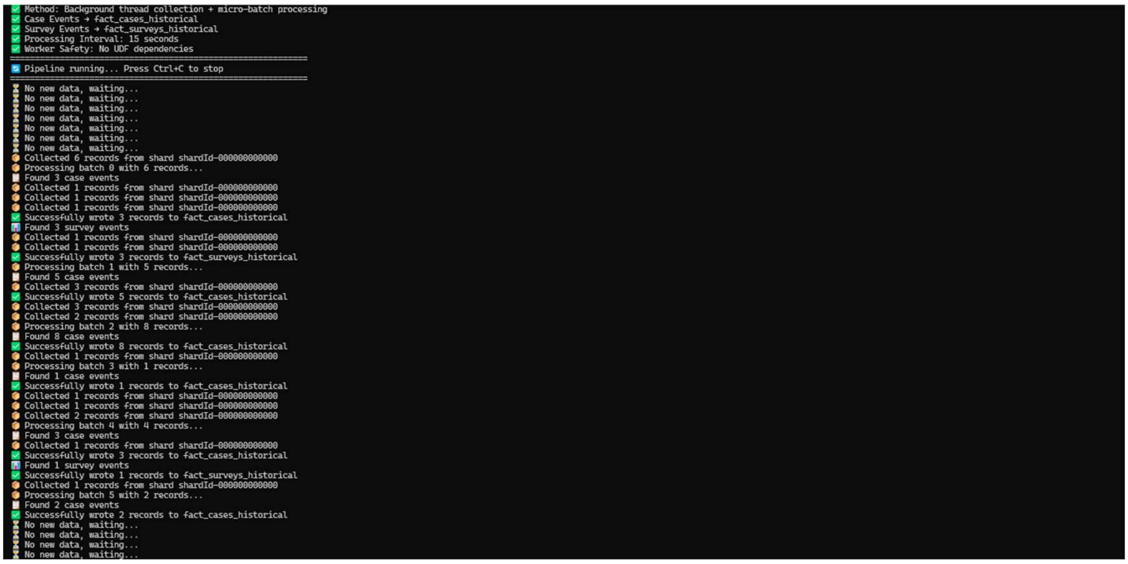
Track - *new*

Endpoint
aahash-work.008673239246.ap-south-1.redshift-serverless.amazonaws.com:5439/dev

JDBC URL
jdbc:redshift://aahash-work.008673239246.ap-south-1.redshift-serverless.amazonaws.com:5439/dev

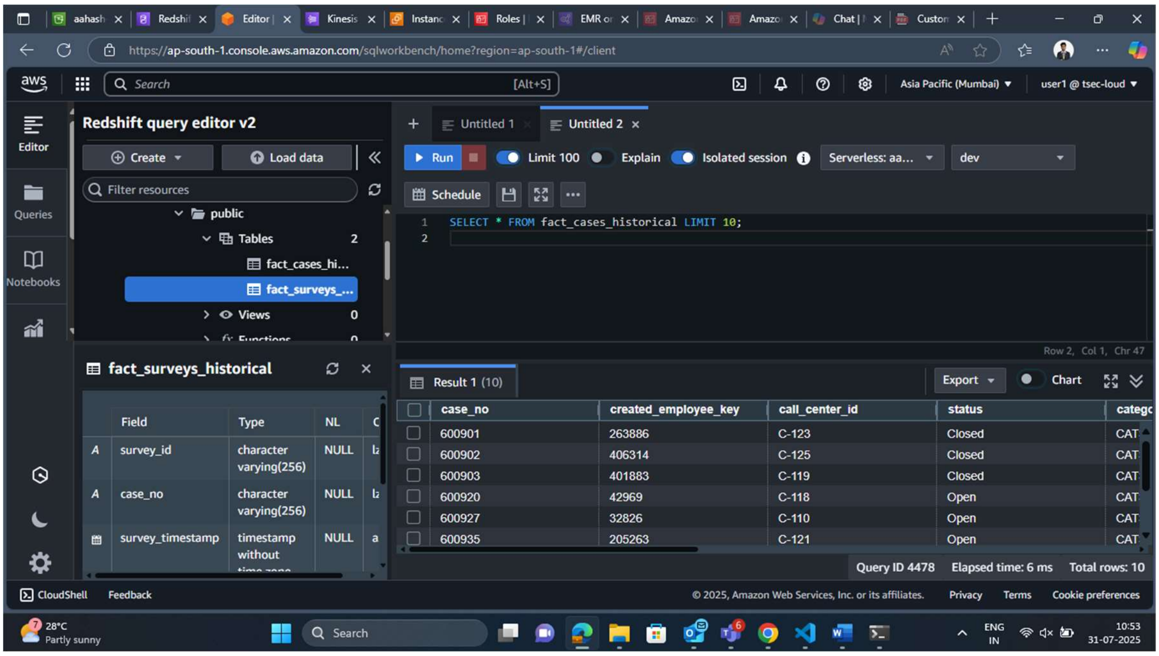
ODBC URL
Driver={Amazon Redshift (x64)}; Server=aahash-work.008673239246.ap-south-1.redshift-serverless.amazonaws.com; Database=dev

Run the job using spark-submit on the EMR cluster:



```
✅ Method: Background thread collection + micro-batch processing
✅ Case Events → fact_cases_historical
✅ Survey Events → fact_surveys_historical
✅ Processing Interval: 15 seconds
✅ Worker Safety: No UDF dependencies
============================================================
🔄 Pipeline running... Press Ctrl+C to stop
============================================================
⏳ No new data, waiting...
⏳ No new data, waiting...
⏳ No new data, waiting...
⏳ No new data, waiting...
⏳ No new data, waiting...
⏳ No new data, waiting...
⏳ No new data, waiting...
📦 Collected 6 records from shard shardId-000000000000
📦 Processing batch 0 with 6 records...
🧾 Found 3 case events
📦 Collected 1 records from shard shardId-000000000000
📦 Collected 1 records from shard shardId-000000000000
📦 Collected 1 records from shard shardId-000000000000
✅ Successfully wrote 3 records to fact_cases_historical
📊 Found 3 survey events
📦 Collected 1 records from shard shardId-000000000000
📦 Collected 1 records from shard shardId-000000000000
✅ Successfully wrote 3 records to fact_surveys_historical
📦 Processing batch 1 with 5 records...
🧾 Found 5 case events
📦 Collected 3 records from shard shardId-000000000000
✅ Successfully wrote 5 records to fact_cases_historical
📦 Collected 3 records from shard shardId-000000000000
📦 Collected 2 records from shard shardId-000000000000
📦 Processing batch 2 with 8 records...
🧾 Found 8 case events
✅ Successfully wrote 8 records to fact_cases_historical
📦 Collected 1 records from shard shardId-000000000000
📦 Processing batch 3 with 1 records...
🧾 Found 1 case events
✅ Successfully wrote 1 records to fact_cases_historical
📦 Collected 1 records from shard shardId-000000000000
📦 Collected 1 records from shard shardId-000000000000
📦 Collected 2 records from shard shardId-000000000000
📦 Processing batch 4 with 4 records...
🧾 Found 3 case events
📦 Collected 1 records from shard shardId-000000000000
✅ Successfully wrote 3 records to fact_cases_historical
📊 Found 1 survey events
✅ Successfully wrote 1 records to fact_surveys_historical
📦 Collected 1 records from shard shardId-000000000000
📦 Processing batch 5 with 2 records...
🧾 Found 2 case events
✅ Successfully wrote 2 records to fact_cases_historical
⏳ No new data, waiting...
⏳ No new data, waiting...
⏳ No new data, waiting...
⏳ No new data, waiting...
```

Data is received by Redshift:



```sql
SELECT * FROM fact_cases_historical LIMIT 10;
```

fact_surveys_historical

	Field	Type	NL	C
A	survey_id	character varying(256)	NULL	lz
A	case_no	character varying(256)	NULL	lz
	survey_timestamp	timestamp without time zone	NULL	a

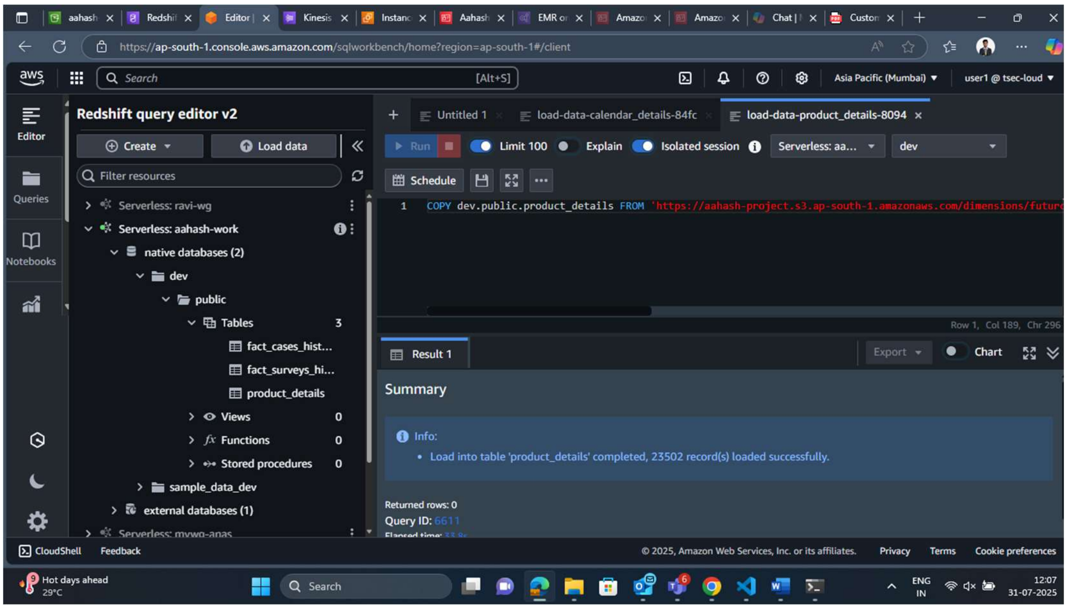
case_no	created_employee_key	call_center_id	status	catego
600901	263886	C-123	Closed	CAT
600902	406314	C-125	Closed	CAT
600903	401883	C-119	Closed	CAT
600920	42969	C-118	Open	CAT
600927	32826	C-110	Open	CAT
600935	205263	C-121	Open	CAT

Query ID 4478 Elapsed time: 6 ms Total rows: 10

Phase 6: Query Execution and Final Insights

Tasks:

- Loaded dimension JSON data from S3 to local SQL.
- Verified integrity of data using SELECT queries.
- Ran provided analytical queries on the complete dataset.



```
COPY dev.public.product_details FROM 'https://aahash-project.s3.ap-south-1.amazonaws.com/dimensions/future
```

Row 1, Col 189, Chr 296

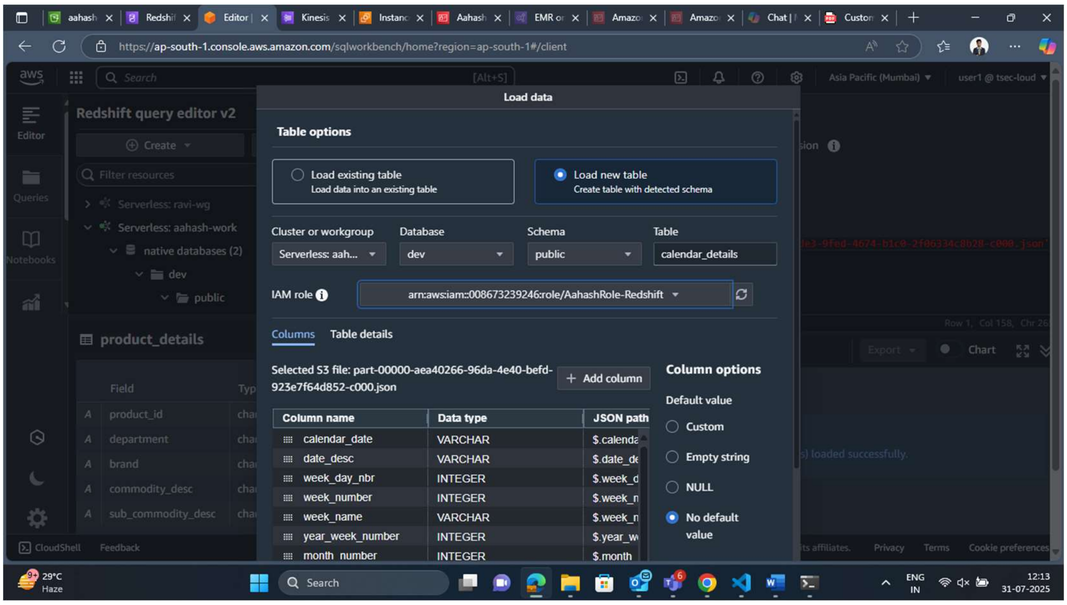
Summary

Info:

- Load into table 'product_details' completed, 23502 record(s) loaded successfully.

Returned rows: 0

Query ID: 6611



Load data

Table options

○ Load existing table
Load data into an existing table

● Load new table
Create table with detected schema

Cluster or workgroup: Serverless: aah...

Database: dev

Schema: public

Table: calendar_details

IAM role: arn:aws:iam::008673239246:role/AahashRole-Redshift

Columns | Table details

Selected S3 file: part-00000-aea40266-96da-4e40-befd-923e7f64d852-c000.json

Column name	Data type	JSON path
calendar_date	VARCHAR	$.calenda
date_desc	VARCHAR	$.date_de
week_day_nbr	INTEGER	$.week_d
week_number	INTEGER	$.week_n
week_name	VARCHAR	$.week_n
year_week_number	INTEGER	$.year_w
month_number	INTEGER	$.month

Column options

Default value

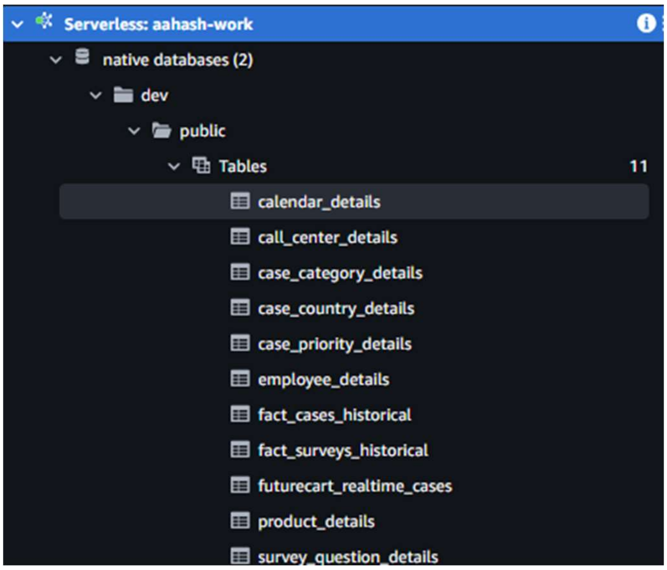
○ Custom

○ Empty string

○ NULL

● No default value

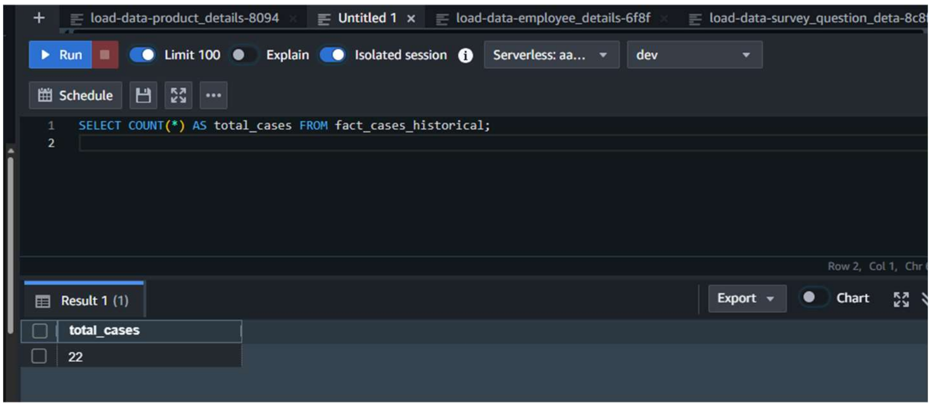
All data has been loaded:



- Serverless: aahash-work
  - native databases (2)
    - dev
      - public
        - Tables 11
          - calendar_details
          - call_center_details
          - case_category_details
          - case_country_details
          - case_priority_details
          - employee_details
          - fact_cases_historical
          - fact_surveys_historical
          - futurecart_realtime_cases
          - product_details
          - survey_question_details

Create the above queries on the Redshift tables

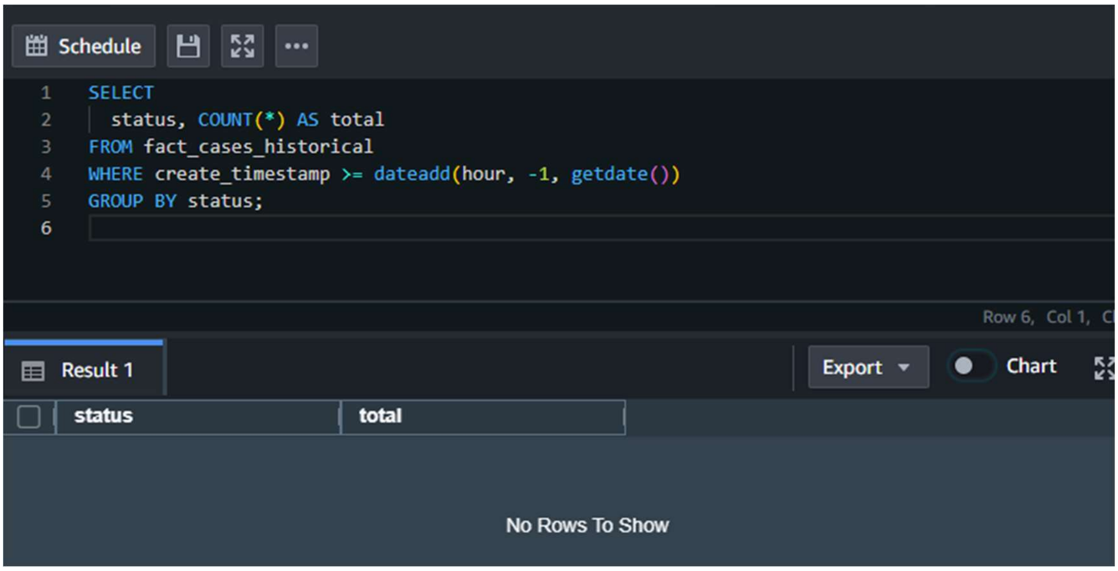
1. Total Number of Cases



```sql
SELECT COUNT(*) AS total_cases FROM fact_cases_historical;
```

total_cases
22

2. Total Open/Closed Cases in the Last 1 Hour

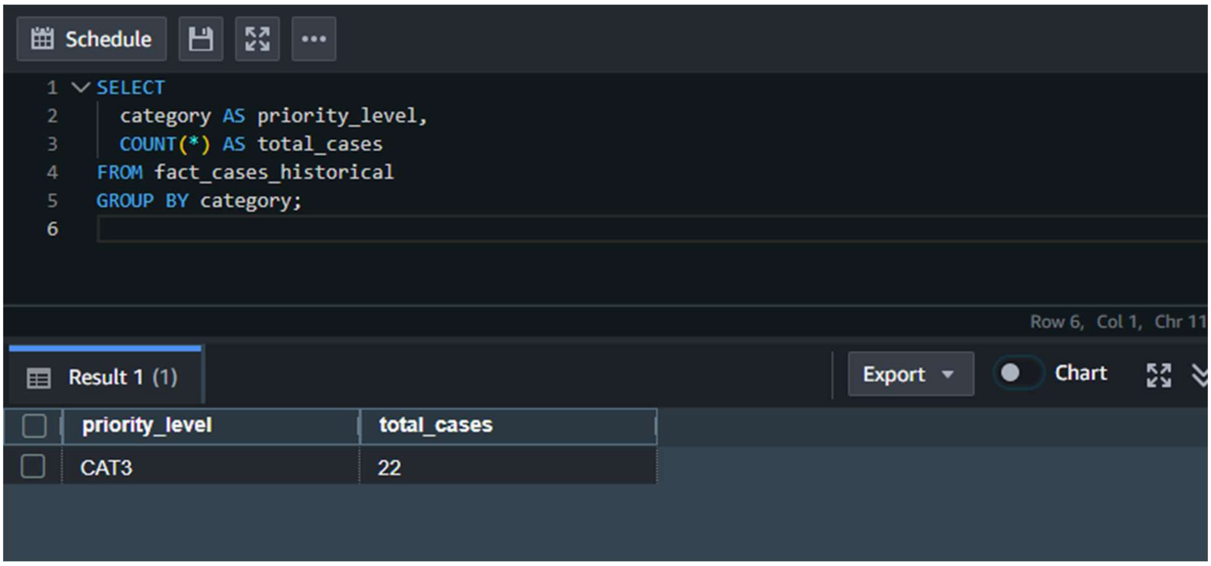


```sql
SELECT
  status, COUNT(*) AS total
FROM fact_cases_historical
WHERE create_timestamp >= dateadd(hour, -1, getdate())
GROUP BY status;
```

status	total

No Rows To Show

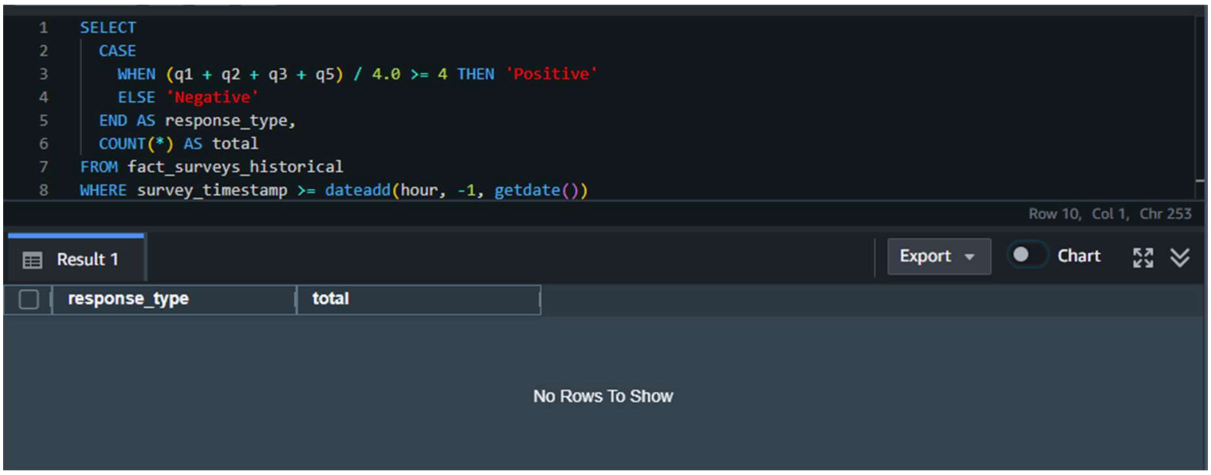
3. Total Priority Cases (Real-Time)



```sql
SELECT
    category AS priority_level,
    COUNT(*) AS total_cases
FROM fact_cases_historical
GROUP BY category;
```

priority_level	total_cases
CAT3	22

4. Total Positive/Negative Responses in the Last 1 Hour (Real-Time)

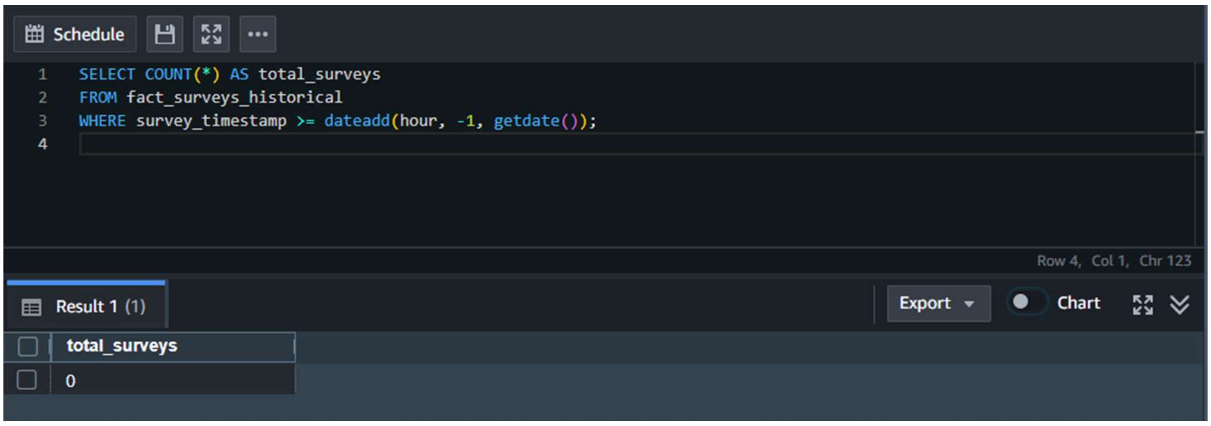


```sql
SELECT
  CASE
    WHEN (q1 + q2 + q3 + q5) / 4.0 >= 4 THEN 'Positive'
    ELSE 'Negative'
  END AS response_type,
  COUNT(*) AS total
FROM fact_surveys_historical
WHERE survey_timestamp >= dateadd(hour, -1, getdate())
```

response_type	total

No Rows To Show

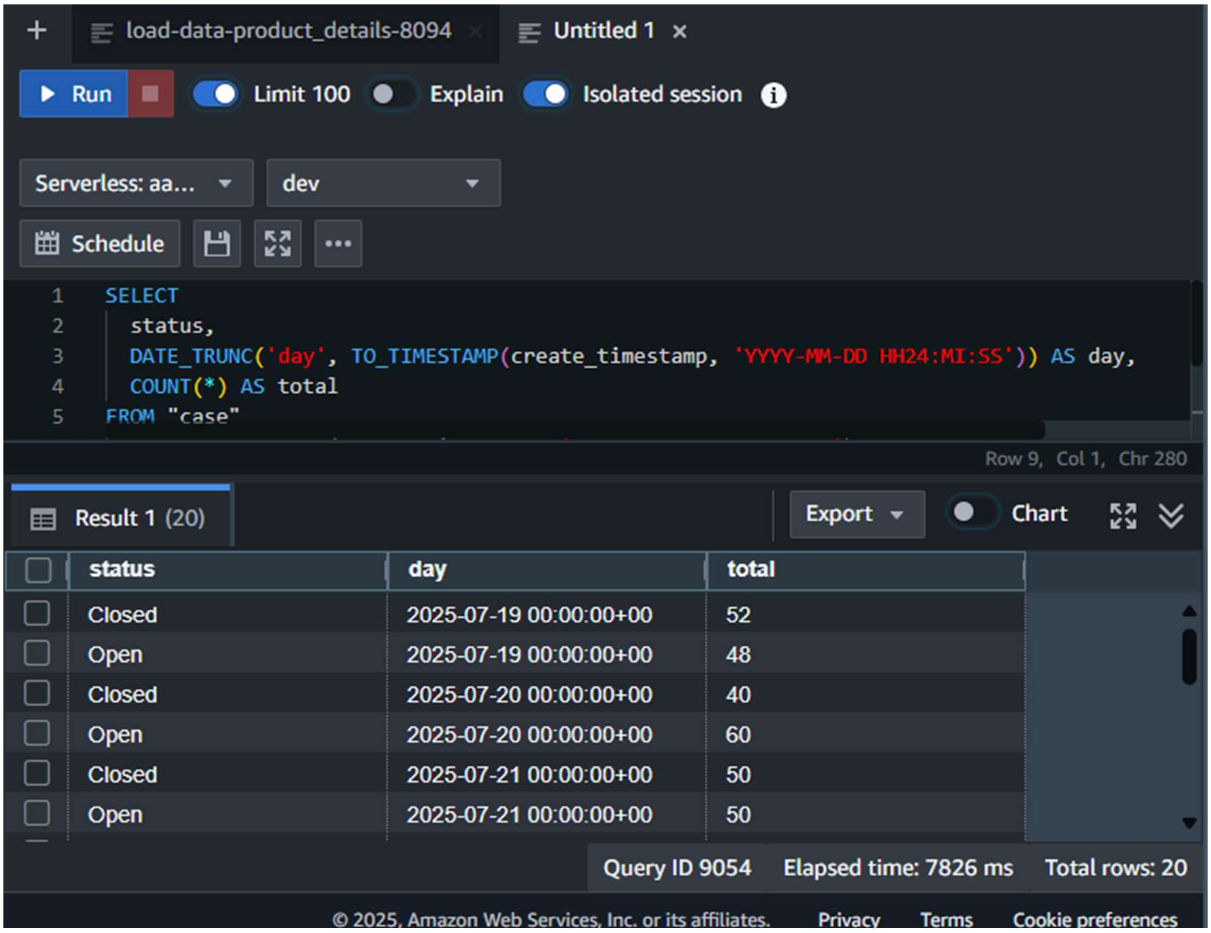
5: Total Number of Surveys in the Last 1 Hour (Real-Time)



```sql
SELECT COUNT(*) AS total_surveys
FROM fact_surveys_historical
WHERE survey_timestamp >= dateadd(hour, -1, getdate());
```

total_surveys
0

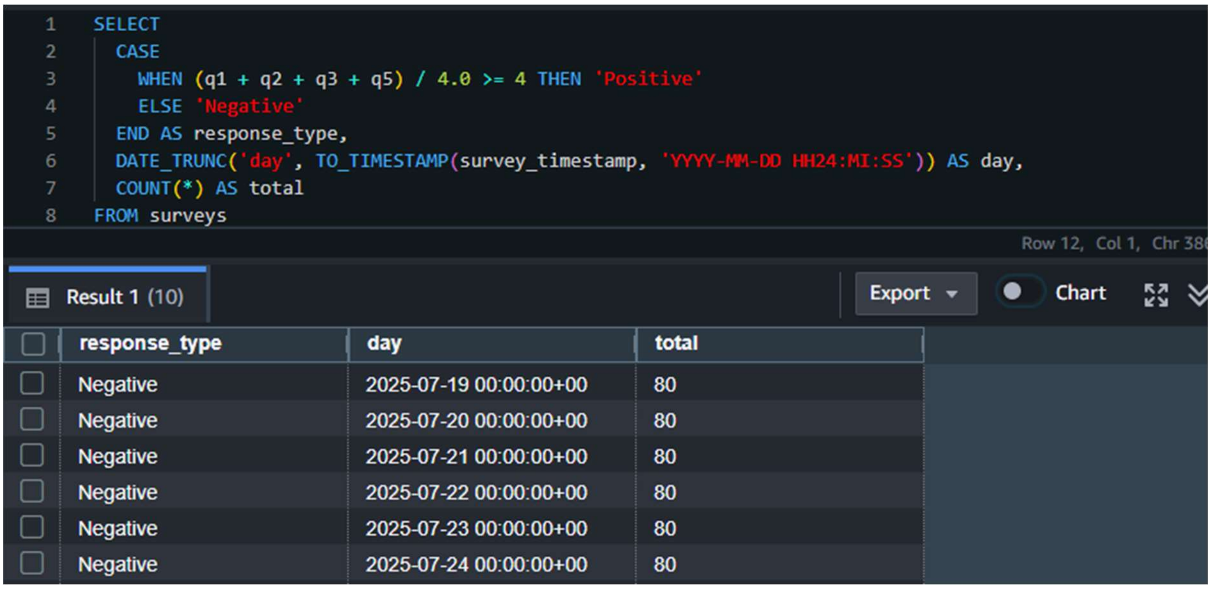
6: Total Open/Closed Cases in a Day/Week/Month (Historical)



```sql
SELECT
  status,
  DATE_TRUNC('day', TO_TIMESTAMP(create_timestamp, 'YYYY-MM-DD HH24:MI:SS')) AS day,
  COUNT(*) AS total
FROM "case"
```

□	status	day	total
□	Closed	2025-07-19 00:00:00+00	52
□	Open	2025-07-19 00:00:00+00	48
□	Closed	2025-07-20 00:00:00+00	40
□	Open	2025-07-20 00:00:00+00	60
□	Closed	2025-07-21 00:00:00+00	50
□	Open	2025-07-21 00:00:00+00	50

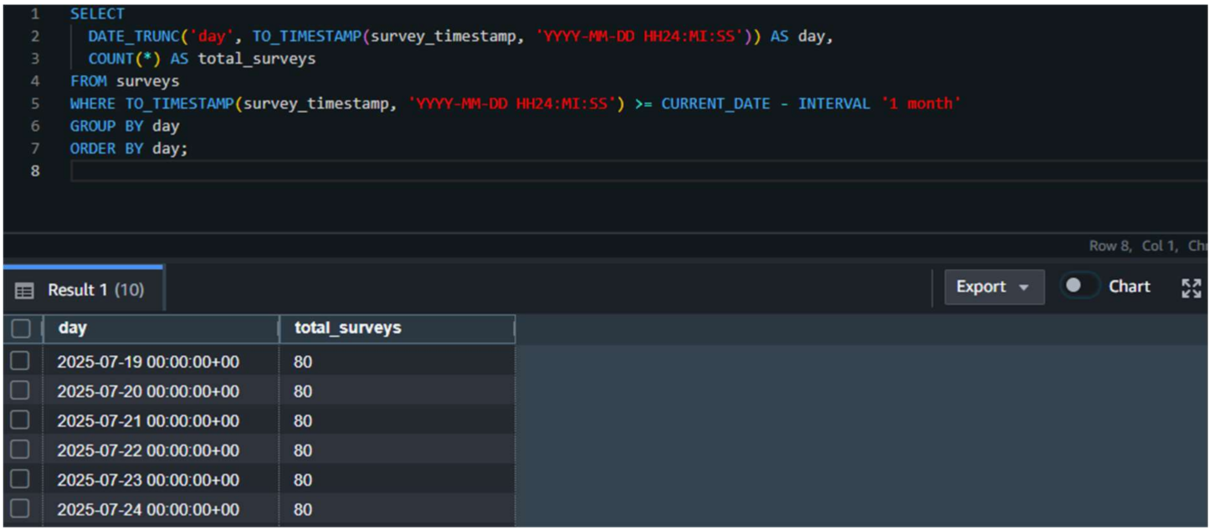
7: Total Positive/Negative Responses in a Day/Week/Month (Historical)



```sql
SELECT
  CASE
    WHEN (q1 + q2 + q3 + q5) / 4.0 >= 4 THEN 'Positive'
    ELSE 'Negative'
  END AS response_type,
  DATE_TRUNC('day', TO_TIMESTAMP(survey_timestamp, 'YYYY-MM-DD HH24:MI:SS')) AS day,
  COUNT(*) AS total
FROM surveys
```

□	response_type	day	total
□	Negative	2025-07-19 00:00:00+00	80
□	Negative	2025-07-20 00:00:00+00	80
□	Negative	2025-07-21 00:00:00+00	80
□	Negative	2025-07-22 00:00:00+00	80
□	Negative	2025-07-23 00:00:00+00	80
□	Negative	2025-07-24 00:00:00+00	80

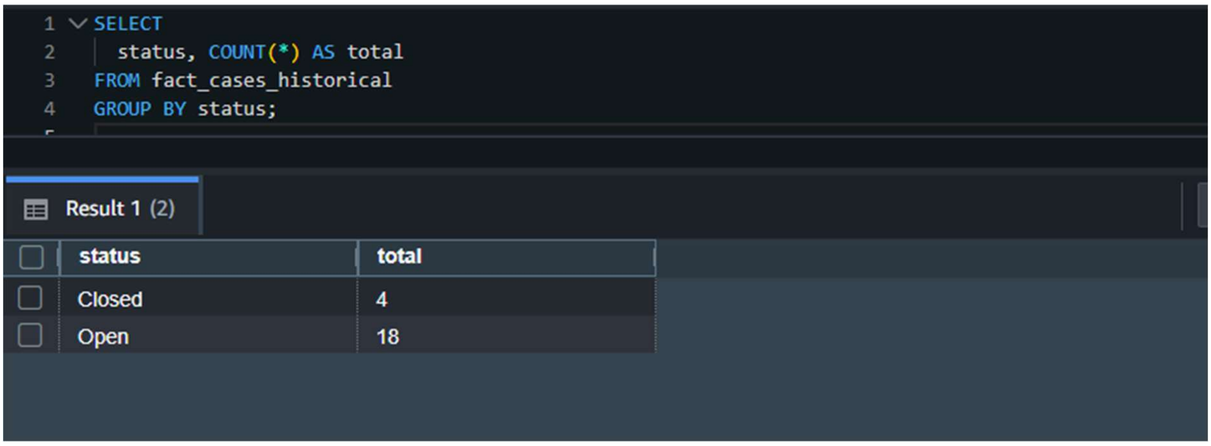
8: Total Number of Surveys in a Day/Week/Month (Historical)



```sql
SELECT
  DATE_TRUNC('day', TO_TIMESTAMP(survey_timestamp, 'YYYY-MM-DD HH24:MI:SS')) AS day,
  COUNT(*) AS total_surveys
FROM surveys
WHERE TO_TIMESTAMP(survey_timestamp, 'YYYY-MM-DD HH24:MI:SS') >= CURRENT_DATE - INTERVAL '1 month'
GROUP BY day
ORDER BY day;
```

day	total_surveys
2025-07-19 00:00:00+00	80
2025-07-20 00:00:00+00	80
2025-07-21 00:00:00+00	80
2025-07-22 00:00:00+00	80
2025-07-23 00:00:00+00	80
2025-07-24 00:00:00+00	80

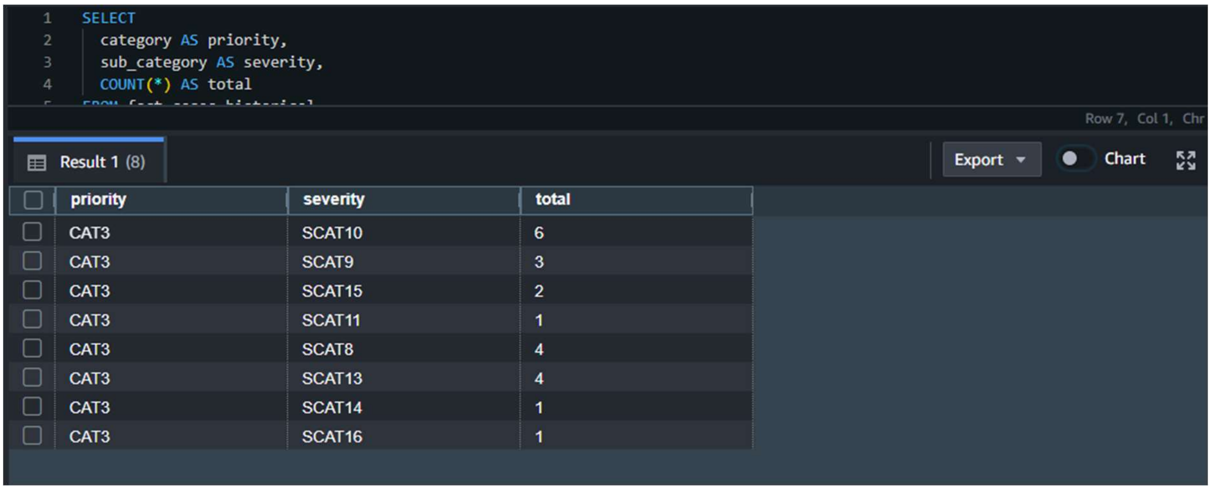
9: Real-time KPIs – Open vs Closed Cases (from Real-Time Table)



```sql
SELECT
  status, COUNT(*) AS total
FROM fact_cases_historical
GROUP BY status;
```

status	total
Closed	4
Open	18

10: Cases Received by Priority and Severity (Real-Time)



```sql
SELECT
  category AS priority,
  sub_category AS severity,
  COUNT(*) AS total
FROM fact_cases_historical
```

priority	severity	total
CAT3	SCAT10	6
CAT3	SCAT9	3
CAT3	SCAT15	2
CAT3	SCAT11	1
CAT3	SCAT8	4
CAT3	SCAT13	4
CAT3	SCAT14	1
CAT3	SCAT16	1