BIG DATA ANALYTICS
CASE STUDY

By Smrithi Venugopal

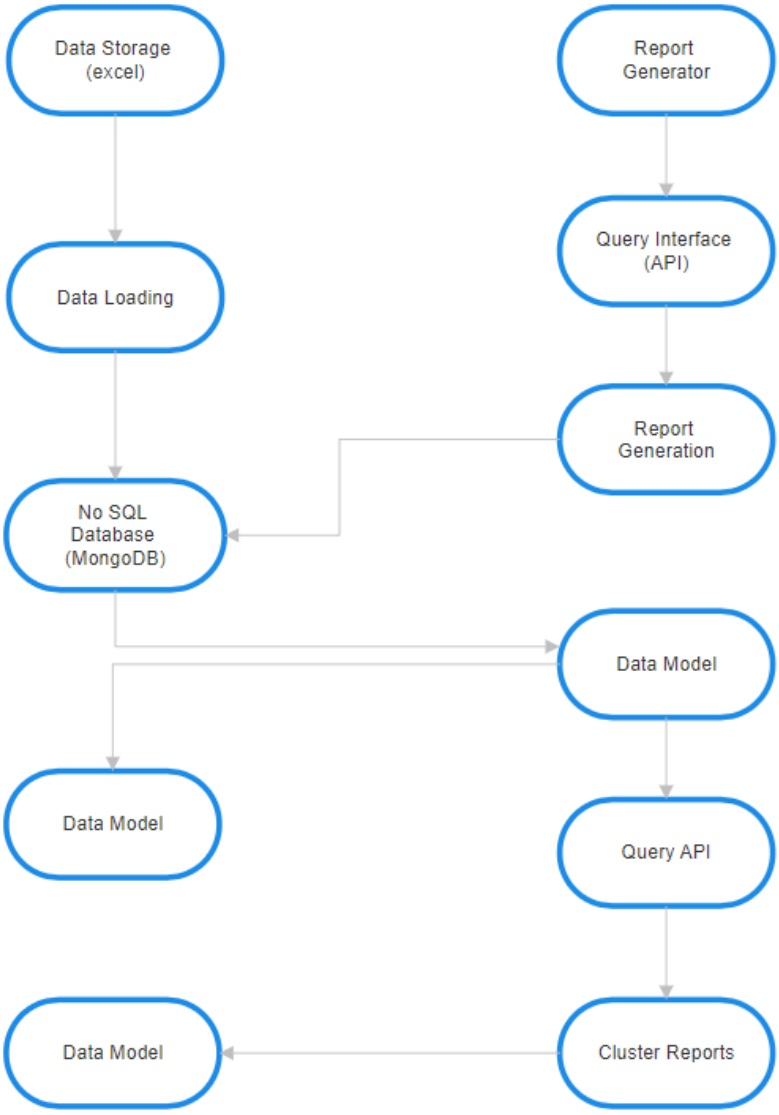
AM.EN.U4CSE21154

(CSE B)

Problem Statement : Big Data Analytics for a Telephone Company

A telephone company stores details on all phone calls made by their customers in multiple tables. They wish to generate the bill for each customer at the end of a month. A customer may make any number of phone calls in a month.

Architecture:



Task 1 : Decide on the number of tables to be used and the structure of each table:

When designing the database for a telephone company, it's important to organize the data so that we can manage customer information, call details, and billing effectively. For this, I used 3 tables namely customers, calls, and rates.

Customers:

- **Structure:**
  - *customer_id:* Each customer gets a unique ID.
  - *name:* Full name of the customer.
  - *address:* Address of the customer.
  - *phone_number:* Contact number of the customer.
- **Purpose:**
  - Stores personal details of customers.

Calls:

- **Structure:**
  - *call_id:* Unique ID for each call.
  - *customer_id:* Connects to the customers table.
  - *timestamp:* Date and time of the call.
  - *duration:* Duration of the call in seconds.
  - *cost:* Cost per minute of the call.
- **Purpose:**
  - Records details of each phone call made by customers.

Rates:

- **Structure:**
  - *rate_id:* Unique ID for each rate.
  - *description:* Description of the rate.
  - *cost_per_minute*: Cost per minute of the rate.
- **Purpose:**
  - Stores the different rates applicable to phone calls.

Task 2 : Create some dummy data in Excel:

To populate our database with realistic data, we generated dummy data using Python. This step involved creating sample information for customers, calls, and rates to simulate real-world scenarios and ensure our database design meets operational needs effectively.

Process:

Customer Data Generation:

We created 50 unique customers, each assigned a unique identifier (customer_id), along with their names, addresses, and phone numbers.

Call Data Generation:

We simulated 200 calls, each with a unique call_id. For each call, we generated a customer_id to link to the customer, timestamp, duration, and cost.

Rate Data Generation:

We created 3 different rates with unique rate_id, description, and cost_per_minute.

Code:

D: > bigdataassignment > database.py > ...

```python
import pandas as pd
import random
from faker import Faker
from datetime import datetime, timedelta

fake = Faker()


customers = []
for i in range(1, 51):
    customers.append([i, fake.name(), fake.address(), fake.phone_number()])

customers_df = pd.DataFrame(customers, columns=["customer_id", "name", "address", "phone_number"])
customers_df.to_excel("customers.xlsx", index=False)


calls = []
for i in range(1, 201):
    customer_id = random.randint(1, 50)
    timestamp = fake.date_time_this_year()
    duration = random.randint(1, 3600)
    cost = random.choice([0.05, 0.03, 0.02])
    calls.append([i, customer_id, timestamp, duration, cost])

calls_df = pd.DataFrame(calls, columns=["call_id", "customer_id", "timestamp", "duration", "cost"])
calls_df.to_excel("calls.xlsx", index=False)


rates = [
    [1, "Standard Rate", 0.05],
    [2, "Evening Rate", 0.03],
    [3, "Weekend Rate", 0.02]
]

rates_df = pd.DataFrame(rates, columns=["rate_id", "description", "cost_per_minute"])
rates_df.to_excel("rates.xlsx", index=False)

print("Data generated and saved to Excel files successfully.")
```

Task 3 : Load the data into MongoDB:

Loading the generated dummy data into MongoDB is a pivotal step in implementing our database solution for the telephone company. This process involves transferring the structured data from Excel files into MongoDB collections, ensuring that our database is populated with accurate and organized information.

Process:

Data Preparation:

We generated dummy data for customers, calls, and rates using Python and stored them in Excel files (customers.xlsx, calls.xlsx, rates.xlsx).

Connection Establishment:

Connected to MongoDB using the MongoClient in Python to interact with the MongoDB server (localhost:27017) where our database (telephone_billing) resides.

Data Loading Process:

Utilized Python to read data from Excel files and insert them into MongoDB collections.

Code:

database.py 2 ● reportgenerationqueries.py 2 mongodbdatabase.py 2 ●

D: > bigdataassignment > mongodbdatabase.py > ...

```python
from pymongo import MongoClient
import pandas as pd

client = MongoClient('localhost', 27017)
db = client.telephone_billing

customers_df = pd.read_excel('customers.xlsx')
calls_df = pd.read_excel('calls.xlsx')
rates_df = pd.read_excel('rates.xlsx')

db.customers.insert_many(customers_df.to_dict('records'))
db.calls.insert_many(calls_df.to_dict('records'))
db.rates.insert_many(rates_df.to_dict('records'))
```

Validation and Verification:

Verified successful data loading by checking the existence of collections (customers, calls, rates) within the telephone_billing database using MongoDB commands.

Task 4: Structure of the Report:

Customer Information:

- ***Name:*** Full name of the customer.
- ***Address:*** Address of the customer.
- ***Phone Number:*** Contact number of the customer.

Call History:

- ***List of Calls Made:*** Details of each call made by the customer in the past month, including:
    - ***Date:*** Date and time of the call.
    - ***Duration:*** Duration of the call in minutes.
    - ***Cost:*** Cost of the call.

Billing Summary:

- ***Total Calls:*** Number of calls made by the customer in the past month.
- ***Total Duration:*** Cumulative duration of all calls.
- ***Total Cost:*** Total cost of all calls.

Task 5 : Write queries to prepare the report:

Code:

database.py 2 | reportgenerationqueries.py 2 | mongodbdatabase.py 2

D: > bigdataassignment > reportgenerationqueries.py > generate_report

```python
import pandas as pd
from pymongo import MongoClient
from datetime import datetime

customers_df = pd.read_excel('customers.xlsx')
calls_df = pd.read_excel('calls.xlsx')
rates_df = pd.read_excel('rates.xlsx')

client = MongoClient('mongodb://localhost:27017/')
db = client['telephone_company']

db.customers.delete_many({})
db.calls.delete_many({})
db.rates.delete_many({})

db.customers.insert_many(customers_df.to_dict('records'))
db.calls.insert_many(calls_df.to_dict('records'))
db.rates.insert_many(rates_df.to_dict('records'))


def generate_report(customer_id):
    customer = db.customers.find_one({'customer_id': customer_id})
    calls = list(db.calls.find({'customer_id': customer_id}))
    rate = db.rates.find_one({'rate_id': 1})

    report = {
        'Customer Information': {
            'Name': customer['name'],
            'Address': customer['address'],
            'Phone Number': customer['phone_number']
        },
        'Call History': [],
        'Billing Summary': {
            'Total Calls': len(calls),
            'Total Duration': 0,
            'Total Cost': 0
        }
    }

```

```python
    total_duration = 0
    total_cost = 0

    for call in calls:
        duration_in_minutes = call['duration'] / 60
        cost = duration_in_minutes * rate['cost_per_minute']
        total_duration += duration_in_minutes
        total_cost += cost

        report['Call History'].append({
            'Date': call['timestamp'].strftime('%Y-%m-%d %H:%M:%S'),
            'Duration': round(duration_in_minutes, 2),
            'Cost': round(cost, 2)
        })

    report['Billing Summary']['Total Duration'] = round(total_duration, 2)
    report['Billing Summary']['Total Cost'] = round(total_cost, 2)

    return report

def print_report(report):
    print("Customer Information")
    print("====================")
    print(f"Name: {report['Customer Information']['Name']}")
    print(f"Address: {report['Customer Information']['Address']}")
    print(f"Phone Number: {report['Customer Information']['Phone Number']}")
    print("\nCall History")
    print("============")
    for call in report['Call History']:
        print(f"Date: {call['Date']}")
        print(f"Duration: {call['Duration']} minutes")
        print(f"Cost: ${call['Cost']}")
        print("------------")
    print("\nBilling Summary")
    print("================")
    print(f"Total Calls: {report['Billing Summary']['Total Calls']}")
    print(f"Total Duration: {report['Billing Summary']['Total Duration']} minutes")
    print(f"Total Cost: ${report['Billing Summary']['Total Cost']}")
    print("\n" + "="*30 + "\n")
```

```python
all_customers = db.customers.find()
reports = []

for customer in all_customers:
    customer_id = customer['customer_id']
    report = generate_report(customer_id)
    reports.append(report)

for report in reports:
    print_report(report)
```

Output:

```
================
Total Calls: 4
Total Duration: 112.67 minutes
Total Cost: $5.63

==============================

Customer Information
====================
Name: Sean Obrien
Address: 700 Sandoval Mount
Diazmouth, NE 17058
Phone Number: 724.249.9215

Call History
============
Date: 2024-05-24 18:35:08
Duration: 47.88 minutes
Cost: $2.39
------------
Date: 2024-02-03 17:53:35
Duration: 25.03 minutes
Cost: $1.25
------------
Date: 2024-03-20 13:19:11
Duration: 18.08 minutes
Cost: $0.9
------------
Date: 2024-01-19 10:37:20
Duration: 38.95 minutes
Cost: $1.95
------------

Billing Summary
================
Total Calls: 4
Total Duration: 129.95 minutes
Total Cost: $6.5

==============================

Customer Information
====================
Name: Anthony Miller
Address: 1874 Monica Lakes
New Gwendolyn, OH 17281
Phone Number: 2974753068

Call History
============

Billing Summary
================
Total Calls: 0
Total Duration: 0 minutes
Total Cost: $0

==============================

Customer Information
====================
Name: Sean Morales
Address: 309 Roberts Vista
Cohenstad, NV 96033
Phone Number: 597-781-5374x6901

Call History
============
Date: 2024-01-12 03:55:52
Duration: 25.1 minutes
Cost: $1.26
------------
Date: 2024-01-29 13:08:16
Duration: 27.43 minutes
Cost: $1.37
------------
Date: 2024-04-11 17:27:46
Duration: 42.2 minutes
Cost: $2.11
------------
Date: 2024-05-27 02:25:31
Duration: 22.13 minutes
Cost: $1.11
------------
Date: 2024-03-14 04:46:02
Duration: 59.22 minutes
Cost: $2.96
------------
Date: 2024-01-08 08:10:37
Duration: 30.4 minutes
Cost: $1.52
------------
```

```
Billing Summary
================
Total Calls: 7
Total Duration: 180.85 minutes
Total Cost: $9.04

==============================

Customer Information
====================
Name: Denise Hayes
Address: 452 Hogan Causeway
Port Christianhaven, KY 32493
Phone Number: +1-442-292-3647x386

Call History
============
Date: 2024-02-06 20:20:09
Duration: 36.98 minutes
Cost: $1.85
------------
Date: 2024-01-28 02:56:02
Duration: 3.35 minutes
Cost: $0.17
------------
Date: 2024-03-10 15:13:00
Duration: 9.75 minutes
Cost: $0.49
------------
Date: 2024-03-08 00:51:09
Duration: 30.82 minutes
Cost: $1.54
------------
Date: 2024-04-07 13:30:56
Duration: 45.23 minutes
Cost: $2.26
------------
Date: 2024-02-17 07:44:46
Duration: 17.72 minutes
Cost: $0.89
------------
Date: 2024-02-12 11:30:13
Duration: 7.13 minutes
Cost: $0.36
------------
```

```
Billing Summary
================
Total Calls: 7
Total Duration: 150.98 minutes
Total Cost: $7.55

==============================

Customer Information
====================
Name: Sarah Cooper
Address: 5087 Scott Square Apt. 299
Port Davidhaven, UT 45810
Phone Number: +1-829-649-8011x0708

Call History
============
Date: 2024-05-21 08:48:26
Duration: 57.58 minutes
Cost: $2.88
------------
Date: 2024-03-01 23:12:01
Duration: 16.7 minutes
Cost: $0.83
------------
Date: 2024-05-22 14:00:33
Duration: 2.52 minutes
Cost: $0.13
------------
Date: 2024-01-31 04:21:45
Duration: 10.07 minutes
Cost: $0.5
------------

Billing Summary
================
Total Calls: 4
Total Duration: 86.87 minutes
Total Cost: $4.34

==============================

Customer Information
====================
Name: Suzanne Mejia
Address: 16523 Snyder Viaduct Suite
```

0 ⚠ 6 0

```
Call History
============
Date: 2024-05-31 02:32:17
Duration: 6.93 minutes
Cost: $0.35
------------
Date: 2024-01-04 16:26:22
Duration: 53.42 minutes
Cost: $2.67
------------
Date: 2024-06-16 11:52:23
Duration: 53.15 minutes
Cost: $2.66
------------
Date: 2024-03-04 13:52:53
Duration: 4.45 minutes
Cost: $0.22
------------

Billing Summary
================
Total Calls: 4
Total Duration: 117.95 minutes
Total Cost: $5.9

==============================

Customer Information
====================
Name: Kenneth Rivera
Address: 70061 Mary Orchard Apt.
East Ashleyberg, MA 56869
Phone Number: (489)628-0791x44887

Call History
============
Date: 2024-06-26 15:06:17
Duration: 30.23 minutes
Cost: $1.51
------------
Date: 2024-04-21 22:26:14
Duration: 34.22 minutes
Cost: $1.71
------------
Date: 2024-04-25 19:52:41
Duration: 29.02 minutes
```

```
Date: 2024-05-23 17:06:00
Duration: 23.93 minutes
Cost: $1.2
------------
Date: 2024-03-23 22:49:43
Duration: 13.57 minutes
Cost: $0.68
------------

Billing Summary
================
Total Calls: 6
Total Duration: 144.67 minutes
Total Cost: $7.23

==============================

Customer Information
====================
Name: Terri Obrien
Address: 3317 Carter Throughway
Ginaville, IN 89223
Phone Number: 9322562826

Call History
============
Date: 2024-07-04 06:46:05
Duration: 41.3 minutes
Cost: $2.06
------------
Date: 2024-04-12 10:03:31
Duration: 49.17 minutes
Cost: $2.46
------------

Billing Summary
================
Total Calls: 2
Total Duration: 90.47 minutes
Total Cost: $4.52

==============================

(smrithi) PS D:\bigdataassignment>
```

0 ⚠ 6 0