EXPERIMENT NO.9

AIM: Develop a migration script to transfer data from MongoDB collections to corresponding tables in PostgreSQL. Handle data transformation and ensure data integrity during the migration process.

THEORY:

Transferring data from MongoDB (a NoSQL document-based database) to PostgreSQL (a relational SQL database) requires careful planning because the two systems differ significantly in data modeling, structure, and querying paradigms.

Steps for Migration:-

Step 1: Analyze MongoDB Collections
Understand the structure of documents.
Identify embedded documents, arrays, and references.
Step 2: Design PostgreSQL Schema
Convert MongoDB collections into tables.
Map embedded fields to normalized tables or JSONB if you want flexibility.
Define relationships (foreign keys) and constraints.
Step 3: Data Transformation
Convert BSON/JSON types to PostgreSQL types:
Strings → VARCHAR/TEXT
Dates → TIMESTAMP
Arrays → JSONB or normalized rows
ObjectId → UUID or TEXT
Step 4: Data Migration
Extract documents from MongoDB.
Transform them according to the SQL schema.
Insert into PostgreSQL using INSERT or COPY.
Step 5: Ensure Data Integrity
Use transactions during inserts.
Validate foreign key constraints.
Ensure unique constraints (like emails or IDs).

Best Practices for Migration:-

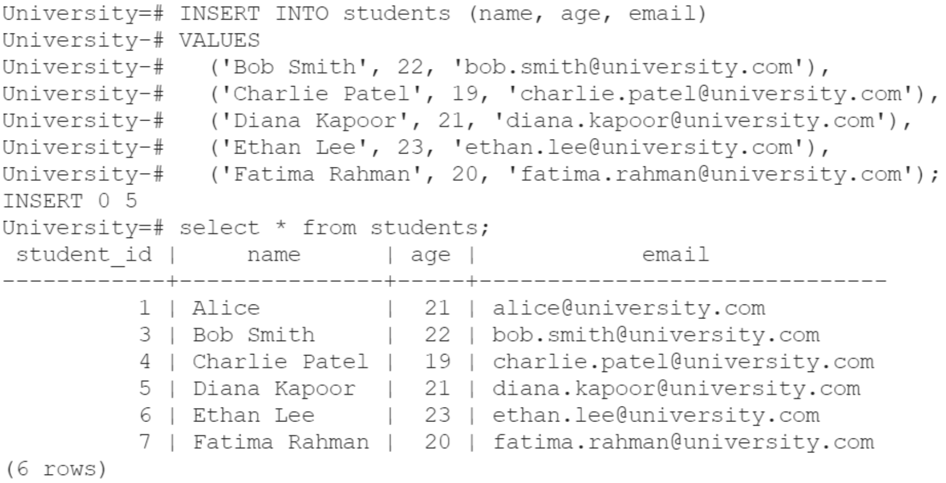
Backups First: Always back up both databases before migrating.
Batch Inserts: Use execute_batch for large data sets to speed up.
Validation Layer: Check required fields and formats before insertion.
Indexing: Create indexes in PostgreSQL for performance after migration.
Logging: Record what’s migrated and any failures for auditing.

Input

```
from pymongo import MongoClient

import psycopg2

mongo_client = MongoClient("mongodb://localhost:27017/")

mongo_db = mongo_client["University"]

mongo_students = mongo_db["students"]

pg_conn = psycopg2.connect(

    dbname="University",

    user="postgres",

    password="@1234",

    host="localhost",

    port="5432"

)

pg_cursor = pg_conn.cursor()

students_data = mongo_students.find()

for student in students_data:

    name = student.get("name", "").strip().title()

    age = int(student.get("age", 0))

    email = student.get("email", "").lower()

    try:

        pg_cursor.execute("""

            INSERT INTO students (name, age, email)

            VALUES (%s, %s, %s)

            ON CONFLICT (email) DO NOTHING;

        """, (name, age, email))

    except Exception as e:

        print(f"Error inserting {name}: {e}")

pg_conn.commit()
```

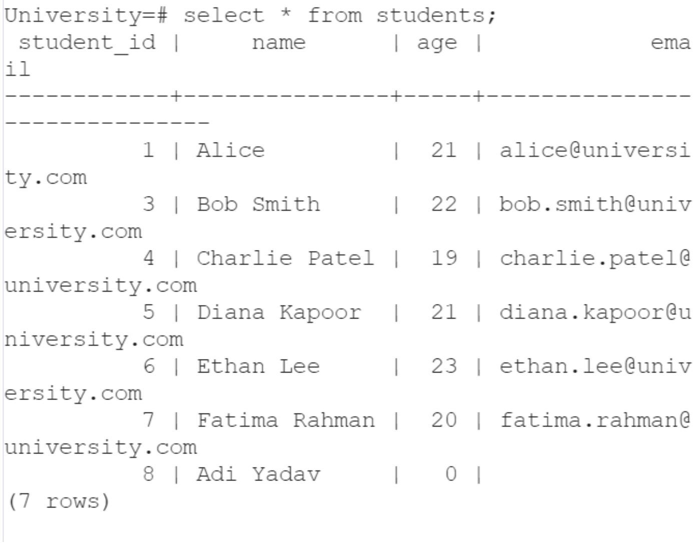
```
pg_cursor.close()

pg_conn.close()

mongo_client.close()
```

BEFORE MIGRATION IN POSTGRESQL



```
University=# INSERT INTO students (name, age, email)
University-# VALUES
University-#   ('Bob Smith', 22, 'bob.smith@university.com'),
University-#   ('Charlie Patel', 19, 'charlie.patel@university.com'),
University-#   ('Diana Kapoor', 21, 'diana.kapoor@university.com'),
University-#   ('Ethan Lee', 23, 'ethan.lee@university.com'),
University-#   ('Fatima Rahman', 20, 'fatima.rahman@university.com');
INSERT 0 5
University=# select * from students;
 student_id |     name      | age |            email
------------+---------------+-----+------------------------------
          1 | Alice         |  21 | alice@university.com
          3 | Bob Smith     |  22 | bob.smith@university.com
          4 | Charlie Patel |  19 | charlie.patel@university.com
          5 | Diana Kapoor  |  21 | diana.kapoor@university.com
          6 | Ethan Lee     |  23 | ethan.lee@university.com
          7 | Fatima Rahman |  20 | fatima.rahman@university.com
(6 rows)
```

AFTER MIGRATION IN POSTGRESQL



```
University=# select * from students;
 student_id |     name      | age |            ema
il
------------+---------------+-----+---------------
---------------
          1 | Alice         |  21 | alice@universi
ty.com
          3 | Bob Smith     |  22 | bob.smith@univ
ersity.com
          4 | Charlie Patel |  19 | charlie.patel@
university.com
          5 | Diana Kapoor  |  21 | diana.kapoor@u
niversity.com
          6 | Ethan Lee     |  23 | ethan.lee@univ
ersity.com
          7 | Fatima Rahman |  20 | fatima.rahman@
university.com
          8 | Adi Yadav     |   0 |
(7 rows)
```

CONCLUSION:

Migrating data from MongoDB to PostgreSQL is a strategic process that bridges two fundamentally different database paradigms: a flexible, document-oriented model and a structured, relational model. This kind of migration is ideal when moving from rapid prototyping or flexible development (MongoDB) to a more structured, transactional system (PostgreSQL). By following best practices, we can ensure the migrated data is clean, relationally accurate, and ready for long-term use.