Experiment - 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:

The migration of data from MongoDB to PostgreSQL involves moving information from a document-oriented NoSQL database to a relational database management system (RDBMS). This process requires addressing fundamental differences in data structure, querying mechanisms, and storage paradigms.

MongoDB

MongoDB is a NoSQL database that stores data in flexible, JSON-like documents. Key characteristics include:

- Schema flexibility (documents in the same collection can have different fields)
- Document-oriented storage (nested data structures)
- Horizontal scalability
- No support for complex joins
- Uses BSON (Binary JSON) for storage

PostgreSQL

PostgreSQL is a powerful, open-source relational database system that uses SQL. Key characteristics include:

- Rigid schema structure
- Table-based storage with rows and columns
- Strong support for ACID transactions
- Complex query capabilities with joins
- Mature ecosystem for data integrity

Schema creation in psql

```
book_store=# CREATE TABLE book (
book_store(#   id SERIAL PRIMARY KEY,
book_store(#   book_id VARCHAR(50) UNIQUE NOT NULL,
book_store(#   book_name VARCHAR(255) NOT NULL,
book_store(#   book_category VARCHAR(100) NOT NULL,
book_store(#   book_authors TEXT[] NOT NULL,
book_store(#   isbn_number VARCHAR(50) UNIQUE NOT NULL,
book_store(#   edition_number INTEGER NOT NULL,
book_store(#   year_of_publication INTEGER NOT NULL
book_store(# );
CREATE TABLE
```

Config.py

```
MONGO_URI = "mongodb://localhost:27017"
MONGO_DB = "crud"
MONGO_COLLECTION = "books"


PG_HOST = "localhost"
PG_DATABASE = "book_store"
PG_USER = "postgres"
PG_PASSWORD = "sudarshan"
PG_PORT = "5432"
PG_TABLE = "book"
```

Migration_script.py

```
def transform_data(mongo_record):
        """Transform MongoDB document to PostgreSQL record format"""
    # Handle the array of authors by converting to a comma-separated string
                                authors = ", ".join(mongo_record.get("bookAuthors", []))

    return {
            "book_id": mongo_record.get("bookId", ""),
                                        "book_name": mongo_record.get("bookName", ""),
                                    "book_category": mongo_record.get("bookCategory", ""),
        "book_authors": authors,
                                        "isbn_number": mongo_record.get("isbnNumber", ""),
                                    "edition_number": mongo_record.get("editionNumber", 1),
                                "year_of_publication": mongo_record.get("yearOfPublication", 0)
    }

def migrate_books():
    books_collection = mongo_db["books"]
    books = books_collection.find()

    # First, ensure we have the right PostgreSQL table structure
    create_table_query = """
    CREATE TABLE IF NOT EXISTS books (
        book_id VARCHAR(255) PRIMARY KEY,
        book_name VARCHAR(255) NOT NULL,
            book_category VARCHAR(255) NOT NULL,
        book_authors TEXT NOT NULL,
            isbn_number VARCHAR(255) UNIQUE NOT NULL,
        edition_number INTEGER NOT NULL,
            year_of_publication INTEGER NOT NULL
    );
    """

    try:
        pg_cursor.execute(create_table_query)
        pg_conn.commit()
    except Exception as e:
        print(f"Error creating table: {e}")
        return

    # Migrate data
    for book in books:
        transformed_data = transform_data(book)
        query = """
```

```
            INSERT INTO books (book_id, book_name, book_category, book_authors, isbn_number, edition_number, year_of_publication)
        VALUES (%s, %s, %s, %s, %s, %s, %s)
          ON CONFLICT (book_id) DO UPDATE SET
            book_name = EXCLUDED.book_name,
                                book_category = EXCLUDED.book_category,
                                book_authors = EXCLUDED.book_authors,
                                isbn_number = EXCLUDED.isbn_number,
                                edition_number = EXCLUDED.edition_number,
                                year_of_publication = EXCLUDED.year_of_publication;
        """

        try:
            pg_cursor.execute(query, (
                transformed_data["book_id"],
                transformed_data["book_name"],
                transformed_data["book_category"],
                transformed_data["book_authors"],
                transformed_data["isbn_number"],
                transformed_data["edition_number"],
                transformed_data["year_of_publication"]
            ))
        except Exception as e:
            print(f"Error inserting record for book {transformed_data['book_id']}: {e}")

    # Commit after all records are processed
    pg_conn.commit()
            print("Book migration completed successfully.")

# Run the Migration
if __name__ == "__main__":
    migrate_books()
    pg_cursor.close()
    pg_conn.close()
    client.close()
```

```
(migration_env) PS C:\Users\Sudarshan\Desktop\FSWD LAB\exp9> python migration_script.py
2025-04-08 20:19:54,253 - INFO - Connected to MongoDB database: crud
2025-04-08 20:19:54,337 - INFO - Connected to PostgreSQL database: book_store
2025-04-08 20:19:54,374 - INFO - Table book exists with columns: id, book_id, book_name, book_category, book_authors, is
2025-04-08 20:19:54,374 - INFO - Starting migration from MongoDB (books) to PostgreSQL (book)
2025-04-08 20:19:54,377 - INFO - Valid PostgreSQL columns: book_authors, book_id, isbn_number, book_name, year_of_public
2025-04-08 20:19:54,379 - INFO - Found 2 documents in books
2025-04-08 20:19:54,381 - INFO - Processing 2 documents. Progress: 2/2
```

```
    logger.info("✅ Validation successful: Record counts match")
Message: '✅ Validation successful: Record counts match'
Arguments: ()
2025-04-08 20:19:54,397 - INFO - ✅ Validation successful: Record counts match
2025-04-08 20:19:54,424 - INFO - Database connections closed
(migration_env) PS C:\Users\Sudarshan\Desktop\FSWD LAB\exp9>
```

```
_id: ObjectId('67ebec66e6046ce354f84ae3')
bookId : "B002"
bookName : "Full Stack"
bookCategory : "Programming"
▸ bookAuthors : Array (1)
isbnNumber : "978-0132350889"
editionNumber : 2
yearOfPublication : 2025
createdAt : 2025-04-01T13:38:46.332+00:00
__v : 0
```

```
▾ _id: ObjectId('67ebecb3e6046ce354f84ae5')
bookId : "B003"
bookName : "C Programming"
bookCategory : "Programming"
▾ bookAuthors : Array (1)
  0: "Reema Thareja"
isbnNumber : "978-0132350899"
editionNumber : 2
yearOfPublication : 2024
createdAt : 2025-04-01T13:40:03.169+00:00
__v : 0
```


PSQL

```
book_store=# select * from book;
 id | book_id |   book_name   | book_category |    book_authors     |  isbn_number   | edition_number | year_of_publicat
ion
----+---------+---------------+---------------+---------------------+----------------+----------------+-----------------
----
  1 | B002    | Full Stack    | Programming   | {"Sudarshan Gopal"} | 978-0132350889 |              2 |               2
025
  2 | B003    | C Programming | Programming   | {"Reema Thareja"}   | 978-0132350899 |              2 |               2
024
(2 rows)
```

Conclusion

We have successfully migrated data from MongoDB to PostgreSQL while ensuring transformations and integrity.