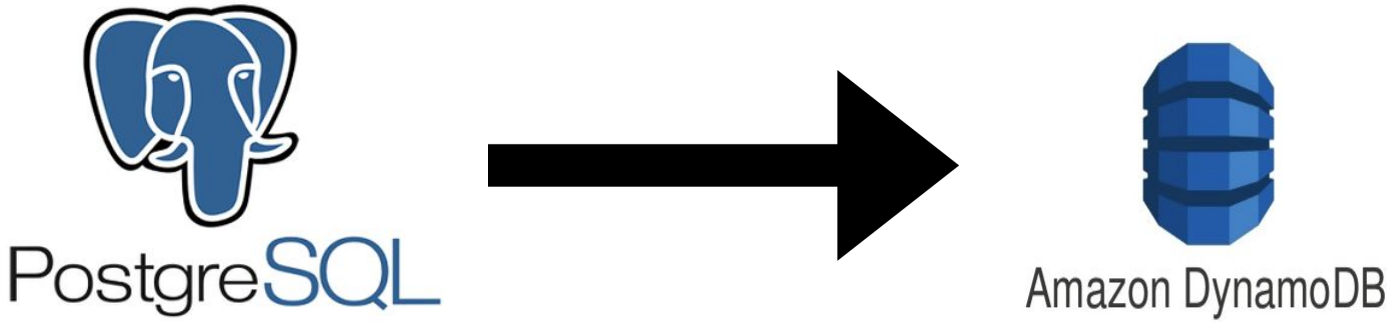
Database Systems 2021 - Assignment 2

By team Shalabi: Hasan Khadra (leader), Mahmood Darwish, Mohamad Dwik, Mohammad Shahin

Introduction

Migrating DVD rental database from PostgreSQL to DynamoDB.



DynamoDB

- Fast and flexible NoSQL database service for any scale
- Managed and developed by Amazon.
- Designing the database tables depends purely on the type of the queries.
- Pros: It is simple to set-up. Best suited for key-value type of operations. Unstructured data.
- Cons: Won't work particularly great for relational operations. Weak querying model. Harder to predict costs.

Local Installation

- Create AWS account at https://aws.amazon.com/dynamodb/.
- Download DynamoDB .tar file specific to your area from the given website.
- After you download the archive, extract the contents, and copy the extracted directory to a location of your choice.
- To start DynamoDB on your computer, open a command prompt window, navigate to the directory where you extracted DynamoDBLocal.jar, and enter the following command:

```
java -Djava.library.path=./DynamoDBLocal_lib -jar DynamoDBLocal.jar -sharedDb
```

Migration

1) Understand the structure of the data.

2) Know the purpose of the migration.

3) Choose the best way to migrate the data.

Amazon DynamoDB

Migration - Our way

- Python API - PGSQL (psycopg2) - DynamoDB (boto3)
- Retrieve table names, attributes names, and attribute types
- Retrieve the actual data
- Process the data (attribute types, data)
- Push the data to DynamoDB

Amazon DynamoDB

Migration - performance

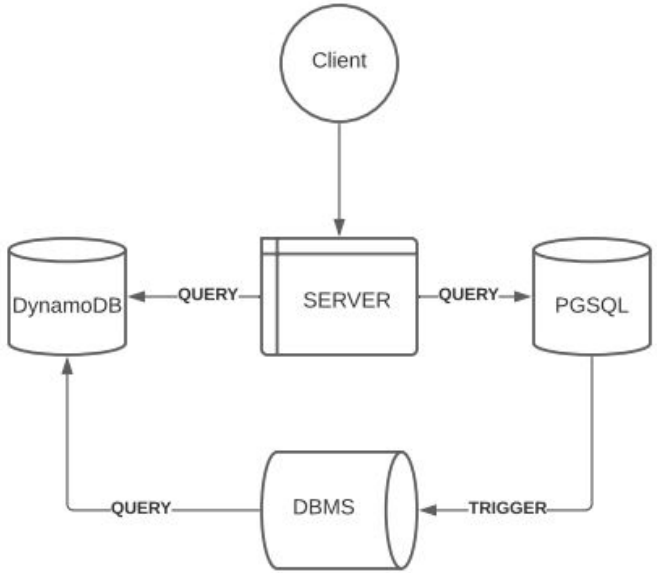
- Downloading the data from PG ~ 0.2 seconds.
- Processing the data ~ 2 seconds
- Uploading the data to Dynamodb ~ 6 minutes 23 seconds.
- Total time ~ 6 minutes 25 seconds.

Continuous Integration

1) Triggered integration.

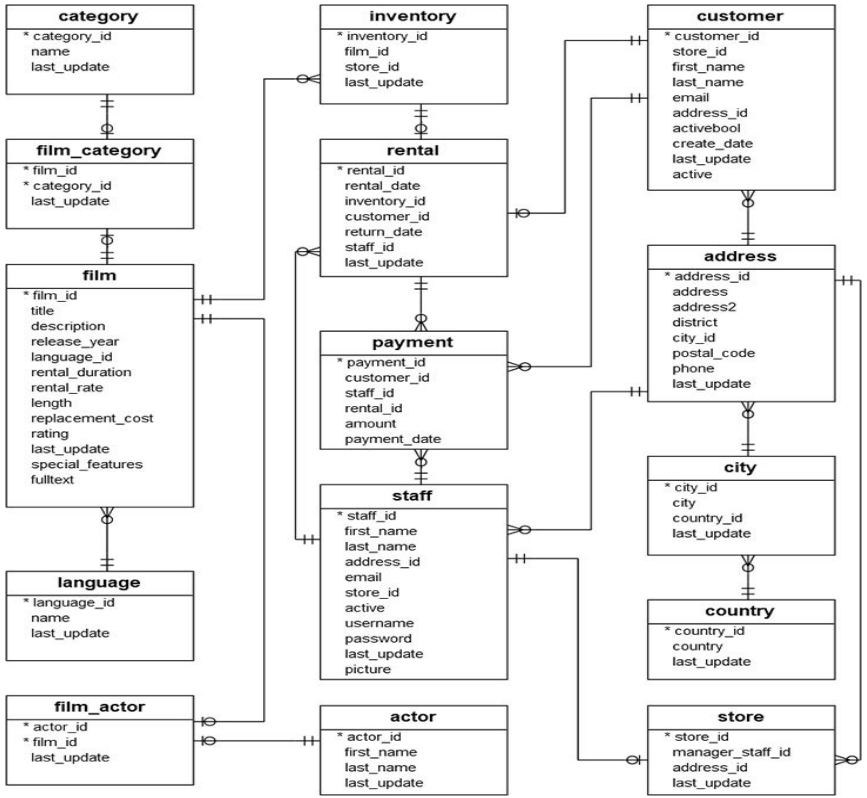
2) Timed integration.

3) Synchronize data in both DBs.



Amazon DynamoDB

Postgres UML Diagram



DynamoDB Tables Design

Design differs between RDBMS and NoSQL.

- RDBMS design

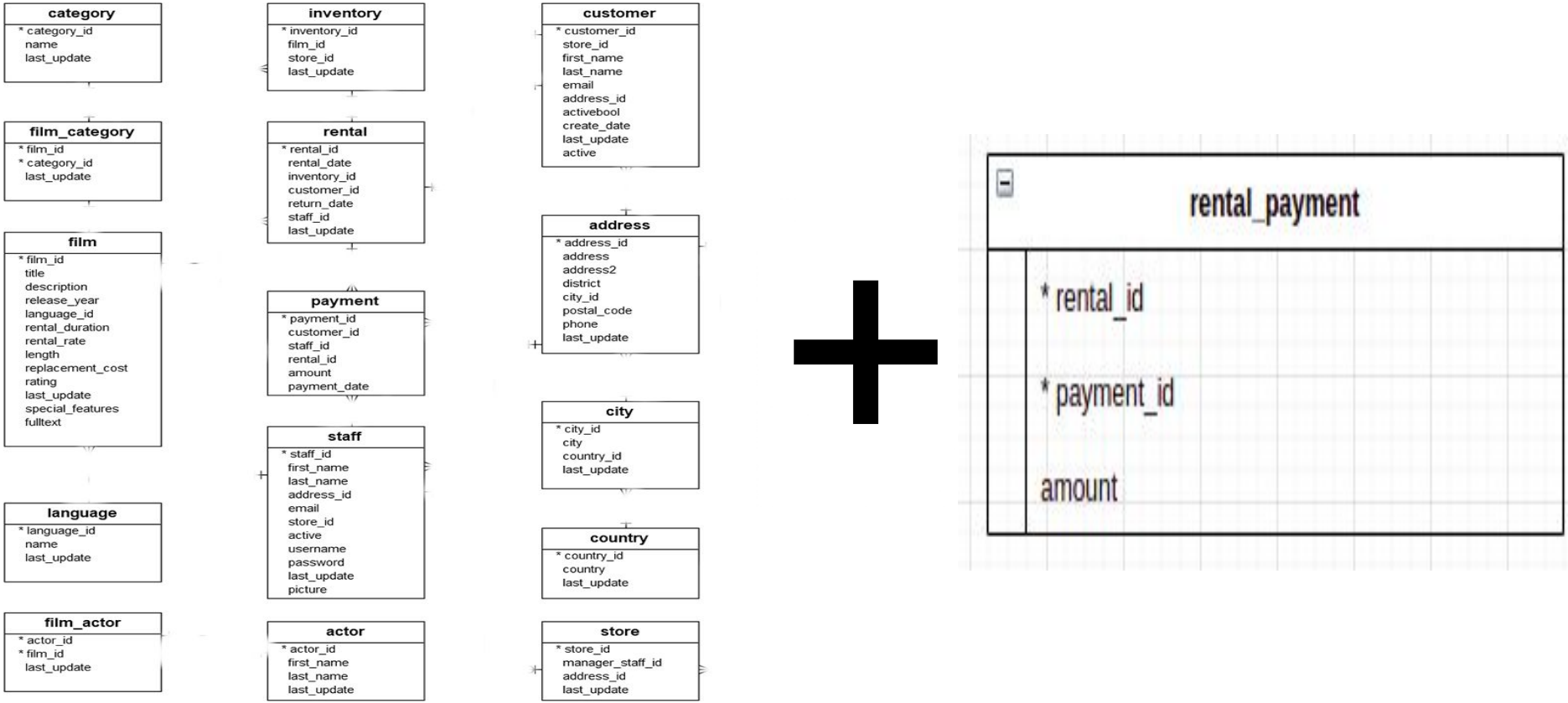
- DynamoDB design.

DynamoDB Tables Design

The tables from Postgres were moved as they are.

New table was added to handle query 1.

DynamoDB UML Diagram



Query Analysis

Query 1:

- Postgres syntax:

```
SELECT *, (SELECT COUNT(*) FROM rental r2, payment p2
WHERE r2.rental_id = p2.rental_id AND
p2.amount<p.amount) AS count_smaller_pay FROM rental
r, payment p WHERE r.rental_id = p.rental_id;
```

- Time taken to run the query: 31.84 seconds.

Query Analysis

Query 1:

- With DynamoDB, no join operation.
- We designed an additional table to make reading operation possible.

Query Analysis

Query 1:

- DynamoDB syntax
- Time taken to run the query 5 minutes 2 seconds.

```
amount = table.query(
        KeyConditionExpression=Key('rental_id').eq(rental_id)
        & Key('payment_id').eq(payment_id)
)['Items'][0]['amount']
```

```
response = table.scan(
    Select='COUNT',
    FilterExpression=Key('amount').lt(amount)
)
```

Query Analysis

Query 2:

- INSERT an actor to the actors table.
- Postgres syntax:

```
INSERT INTO actor (first_name, last_name,
last_update) VALUES ('Mohammad', 'Dwik',
datetime.datetime(2021, 4, 23, 0, 3, 20, 612992));
```

- Time taken to run the query ~0.0079 seconds.

Query Analysis

Query 2:

- INSERT an actor to the actors relation.
- DynamoDB syntax
- Time taken to run the query ~0.036 seconds

```
response = client.put_item(
    TableName='actor',
    Item={
        'first_name': {
            'S': 'Mahmood'
        },
        'last_name': {
            'S': 'Darwish'
        },
        'actor_id': {
            'N': '204'
        }
    },
    ReturnConsumedCapacity='TOTAL'
)
```

Query Analysis

Query 2:

- DELETE an actor from the actors table.
- Postgres syntax:

```
DELETE FROM actor WHERE actor_id = 201;
```

- Time taken to run the query ~0.0056 seconds.

Query Analysis

Query 2:

- DELETE an actor from the actors table.
- DynamoDB syntax
- Time taken to run the query ~0.0089 seconds.

```
response = client.delete_item(
    TableName='actor',
    Key={
        'actor_id': {'N': 201}
    }
)
```

Query Analysis

Query 2:

- UPDATE an actor in the actors table.
- Postgres syntax:

```
UPDATE actor SET first_name = 'Hasan', last_name =
'Khadra, last_update = datetime.datetime(2021, 4,
23, 0, 1, 20, 976351) WHERE actor_id = 1;
```

- Time taken to run the query ~0.0099 seconds.

Query Analysis

Query 2:

- UPDATE an actor in the actors table.
- DynamoDB syntax
- Time taken to run the query ~0.0096

```
response = client.update_item(
    ExpressionAttributeNames={
        '#AT': 'first_name',
        '#Y': 'last_name',
    },
    ExpressionAttributeValues={
        ':t': { 'S': 'Vlad',},
        ':y': { 'S': 'Yalalov',},
    },
    Key={'actor_id': { 'N': 201,}},
    TableName='actor',
    UpdateExpression='SET #Y = :y, #AT = :t'
)
```

Test Functions

- Test showing that the data was successfully migrated.

- Test showing the migration process.

Thanks for listening!