AWS

TASK 4 REPORT

CONTENTS

TASK 1: RDS MYSQL 2

TASK 2: RDS AURORA 4

TASK 3: DYNAMODB 6

TASK 1: RDS MYSQL

Creds file
```
import creds as crd host = 'dilab-mysql.cqlpneuyr8on.eu-central-1.rds.amazonaws.com' database = 'dilab_dev' port = 3306 user='admin' password='admin_mysql'
```

Scripts for establish the connection to mysql databases using own user
```
with mysql.connector.connect( user=crd.user,
                password=crd.password, host=crd.host, database=crd.database,
                                port=crd.port) as connection:
    cursor = connection.cursor()
```

Create a separate schema and granting privileges
```
    cursor.execute("CREATE schema IF NOT EXISTS Artiom_Dolzhenko")
    cursor.execute("CREATE USER IF NOT EXISTS 'Artiom_Dolzhenko'@'%' IDENTIFIED BY 'Artiom_Dolzhenko';") cursor.execute("GRANT ALL ON Artiom_Dolzhenko.* TO 'Artiom_Dolzhenko'@'%';")
```

Create several objects in your schema: Table
```
cursor.execute("CREATE TABLE  IF NOT EXISTS recipes"
            "(recipe_id INT PRIMARY KEY,"
            "recipe_name VARCHAR(30) NOT NULL);")
```

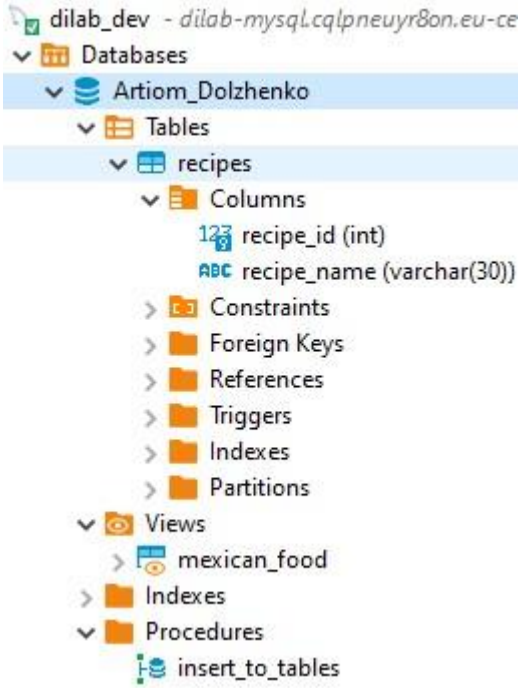
Procedure
```
cursor.execute("CREATE PROCEDURE insert_to_tables"
            "(ins_recipe_id INT, ins_recipe_name VARCHAR(50) )"
            "BEGIN "
            "INSERT INTO recipes"
            "(recipe_id, recipe_name)"
            "VALUES(ins_recipe_id,ins_recipe_name)"
            "ON DUPLICATE KEY UPDATE "
            "recipe_id = ins_recipe_id,"
            "recipe_name = ins_recipe_name;"
            "COMMIT;"
            " END  ")
```

```
cursor.execute("call insert_to_tables (1,'Tacos');")
cursor.execute("call insert_to_tables (2,'Tomato Soup');")
cursor.execute("call insert_to_tables (3,'Grilled Cheese');")
```

View
```
cursor.execute("create or replace view mexican_food as "
            "select * "
            "from recipes "
            "order by recipe_id; ")
```

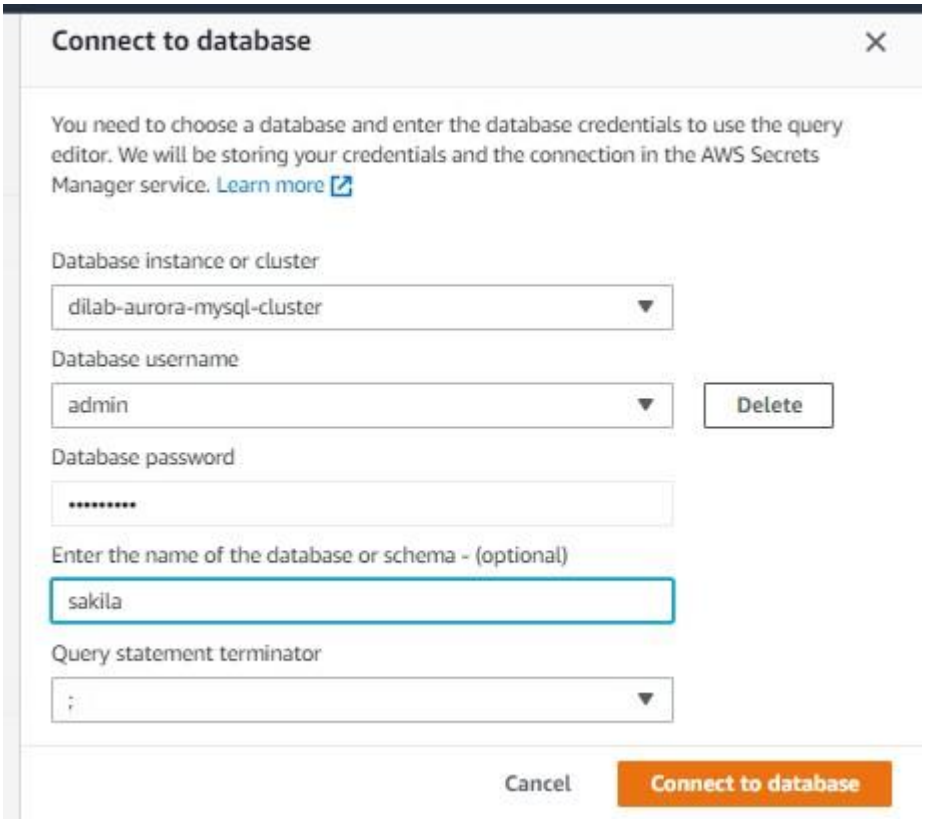
```
Done

Process finished with exit code 0
```

As a result of running current script i get



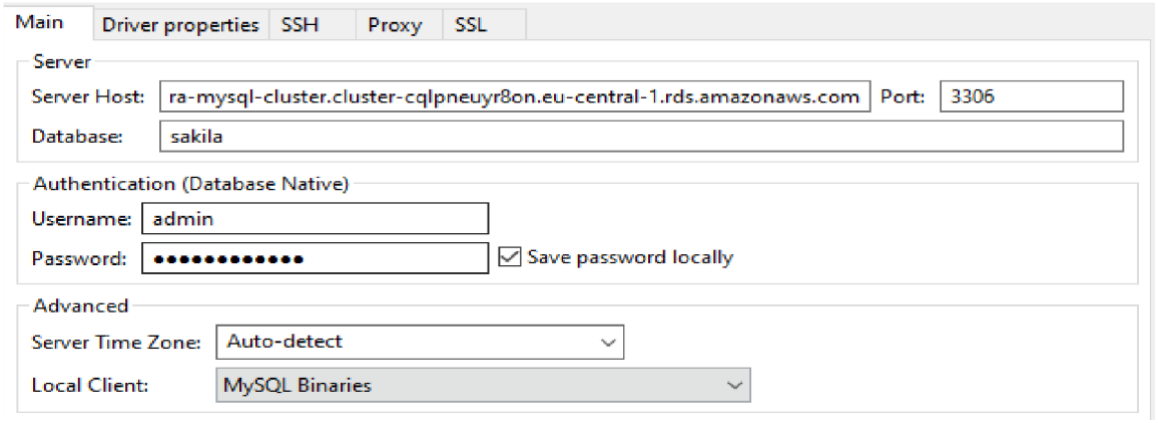
Table, view and a procedure were created

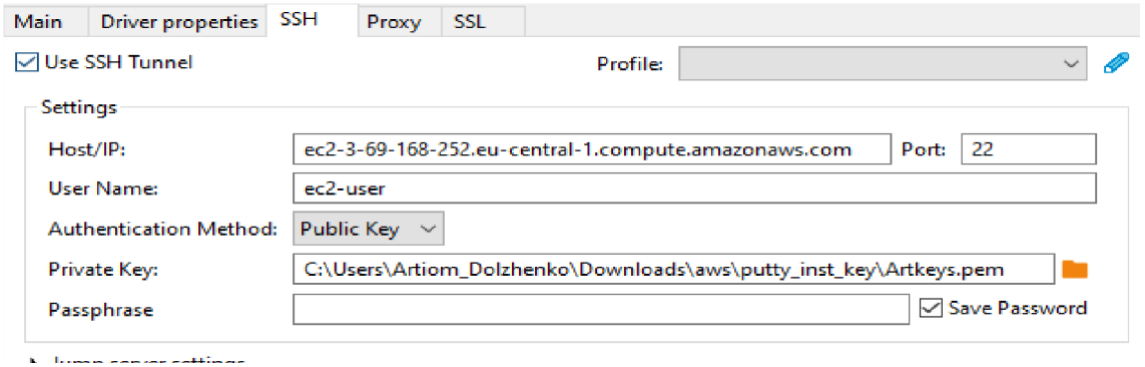
RDS AURORA



Overdue DVDs

```
SELECT CONCAT(customer.last_name, ', ', customer.first_name) AS customer
        address.phone, film.title
        FROM rental INNER JOIN customer ON rental.customer_id = customer.customer_id
        INNER JOIN address ON customer.address_id = address.address_id
        INNER JOIN inventory ON rental.inventory_id = inventory.inventory_id
        INNER JOIN film ON inventory.film_id = film.film_id
        WHERE rental.return_date IS NULL
        AND rental_date + INTERVAL film.rental_duration DAY < CURRENT_DATE()
        ORDER BY title
        LIMIT 5;
```





The whole idea behind this manipulation with an SSH key is to connect to aurora using an ec2 instance as a jump server, because they are located in one VPC.
A VPC is a virtual network that is isolated from other virtual networks in the AWS Cloud. Amazon VPC lets you launch AWS resources, such as an Amazon Aurora DB instance but not the Amazon RDS MySQL.

```
-------------------------------------
------average lenth of category-----
-------------------------------------
select category.name, avg(length)
from film join film_category using (film_id) join category using (category_id)
group by category.name
having avg(length) > (select avg(length) from film)
order by avg(length) desc;
```

ry 1

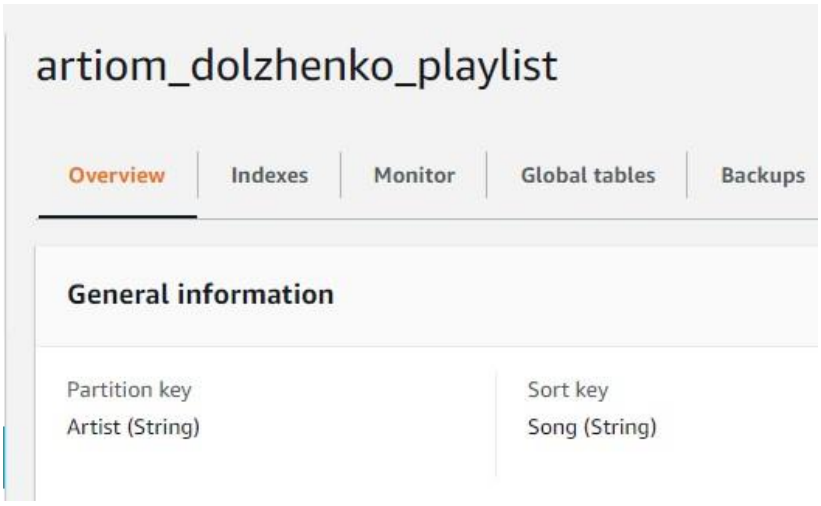
category.name, avg(length) from film join film_catego | Enter a SQL expression to filter results (use Ctr

name	avg(length)
Sports	128.2027
Games	127.8361
Foreign	121.6986
Drama	120.8387
Comedy	115.8276

TASK 3

```
C:\Users\Artiom_Dolzhenko>aws dynamodb list-tables --region eu-central-1 --profile=epam_lab_mfa
{
    "TableNames": [
        "Movies",
        "Volha_Soika1_dynDB",
        "alexandra_makhnach_table_inventory",
        "artiom_dolzhenko_playlist",
        "dilab-anastasiya_viktarovich_book_table",
        "fashion_products",
        "sophie-benko",
        "tatsiana_piytsayeva_football",
        "yekatsiaryna_meliashkevich_table"
    ]
}
```

Creating table in CLI



Idea of using this combination of keys is simple. It's impossible to have the same pair of keys twice. (Unless there is remake of old song but it usually have different name.)
Filling up table with my jsons

Retrieving 5

```
C:\Users\Artiom_Dolzhenko>aws dynamodb get-item --table-name artiom_dolzhenko_playlist
{
    "Item": {
        "Song": {
            "S": "Bacalao"
        },
        "Release year": {
            "S": "2016"
        },
        "Artist": {
            "S": "MC Ceja"
        }
    }
}
```

```
C:\Users\Artiom_Dolzhenko>aws dynamodb get-item --table-name artiom_dolzhenko_playlist
{
    "Item": {
        "Song": {
            "S": "Happy Day"
        },
        "Release year": {
            "S": ""
        },
        "Artist": {
            "S": "Acme Band"
        }
    }
}
```

```
C:\Users\Artiom_Dolzhenko>aws dynamodb get-item --table-name artiom_dolzhenko_playlist -
{
    "Item": {
        "Song": {
            "S": "Last night"
        },
        "Release year": {
            "S": ""
        },
        "Artist": {
            "S": "Bob Marley"
        }
    }
```

```
C:\Users\Artiom_Dolzhenko>
C:\Users\Artiom_Dolzhenko>aws dynamodb get-item --table-name artiom_dolzhenko_playlist
{
    "Item": {
        "Song": {
            "S": "The next Epicode"
        },
        "Release year": {
            "S": "1999"
        },
        "Artist": {
            "S": "Snoop dogg"
        }
    }
}
```

```
C:\Users\Artiom_Dolzhenko>aws dynamodb get-item --table-name artiom_dolzhenko_playlist
{
    "Item": {
        "Song": {
            "S": "Iron nattle"
        },
        "Release year": {
            "S": ""
        },
        "Artist": {
            "S": "Wu tang clan"
        }
    }
}
```

Select 1: aws dynamodb query --table-name artiom_dolzhenko_playlist --key-condition-expression "Artist = :a" --expression-attribute-values "{\":a\": {\"S\": \"Wu tang clan\"}}" --region eu-central-1 --profile=epam_lab_mfa

```
C:\Users\Artiom_Dolzhenko>aws dynamodb quer
u-central-1 --profile=epam_lab_mfa
{
    "Items": [
        {
            "Song": {
                "S": "Above The Clouds"
            },
            "Release year": {
                "S": ""
            },
            "Artist": {
                "S": "Wu tang clan"
            }
        },
        {
            "Song": {
                "S": "Iron nattle"
            },
            "Release year": {
                "S": ""
            },
            "Artist": {
                "S": "Wu tang clan"
            }
        }
    ],
    "Count": 2,
    "ScannedCount": 2,
    "ConsumedCapacity": null
}
```

SELECT 2

aws dynamodb query --table-name artiom_dolzhenko_playlist --key-condition-expression "Artist = :a and Song = :s" --expression-attribute-values "{\":a\": {\"S\": \"Wu tang clan\"}, \":s\": {\"S\": \"Iron nattle\"}}" --region eu-central-1 --profile=epam_lab_mfa

```
C:\Users\Artiom_Dolzhenko>aws dynamodb query --table-name artiom_dolzhenko_playlist
}, \":s\": {\"S\": \"Iron nattle\"}}" --region eu-central-1 --profile=epam_lab_mfa
{
    "Items": [
        {
            "Song": {
                "S": "Iron nattle"
            },
            "Release year": {
                "S": ""
            },
            "Artist": {
                "S": "Wu tang clan"
            }
        }
    ],
    "Count": 1,
    "ScannedCount": 1,
    "ConsumedCapacity": null
}
```

deleting 2 rows

```
C:\Users\Artiom_Dolzhenko>
C:\Users\Artiom_Dolzhenko>aws dynamodb delete-item --table


C:\Users\Artiom_Dolzhenko>
```