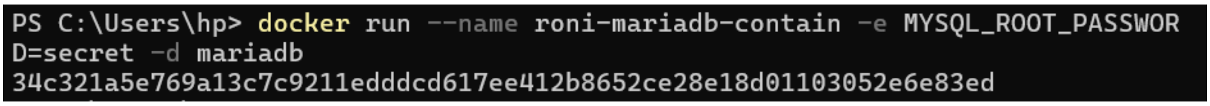
Practical Module: Type of Access Control

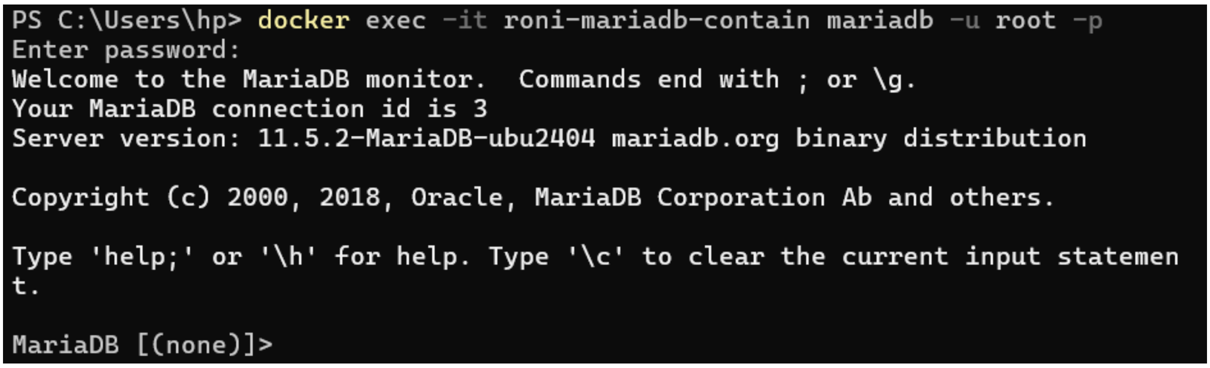
1. Penerapan dan demonstrasi Discretionary Access Control (DAC) Docker

a. Pertama buat terlebih dahulu container yang kita inginkan



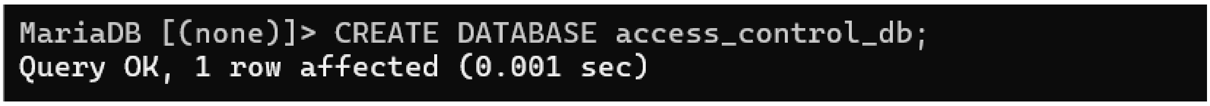
```
PS C:\Users\hp> docker run --name roni-mariadb-contain -e MYSQL_ROOT_PASSWOR
D=secret -d mariadb
34c321a5e769a13c7c9211edddcd617ee412b8652ce28e18d01103052e6e83ed
```

b. Jika sudah kita bisa masuk ke container yang sudah kita buat



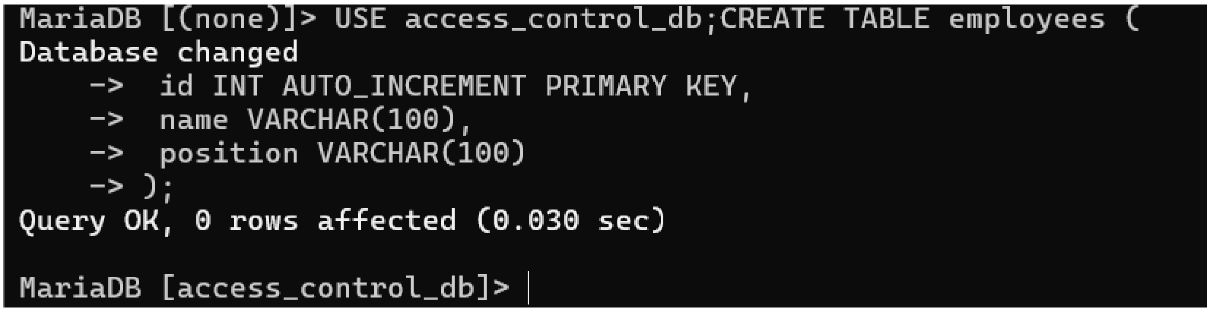
```
PS C:\Users\hp> docker exec -it roni-mariadb-contain mariadb -u root -p
Enter password:
Welcome to the MariaDB monitor.  Commands end with ; or \g.
Your MariaDB connection id is 3
Server version: 11.5.2-MariaDB-ubu2404 mariadb.org binary distribution

Copyright (c) 2000, 2018, Oracle, MariaDB Corporation Ab and others.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statemen
t.

MariaDB [(none)]>
```

c. Setelah itu buat database sesuai yang kita inginkan



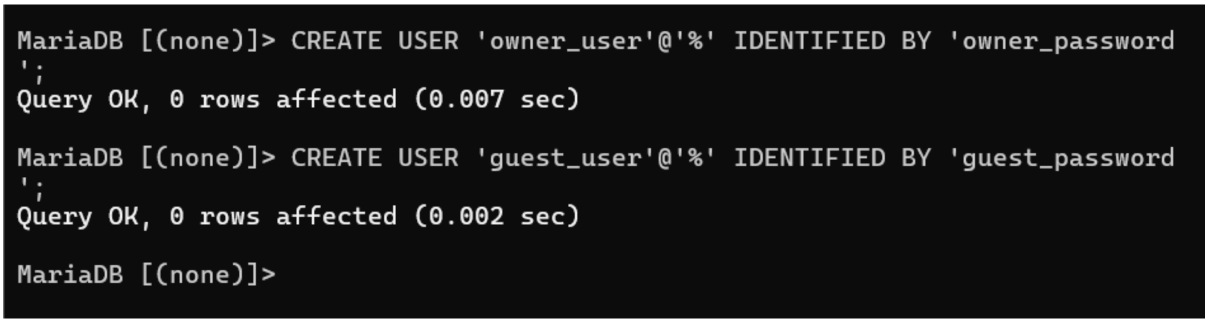
```
MariaDB [(none)]> CREATE DATABASE access_control_db;
Query OK, 1 row affected (0.001 sec)
```

d. Setelah itu gunakan database yang sudah kita buat dan buat table employees di dalamnya



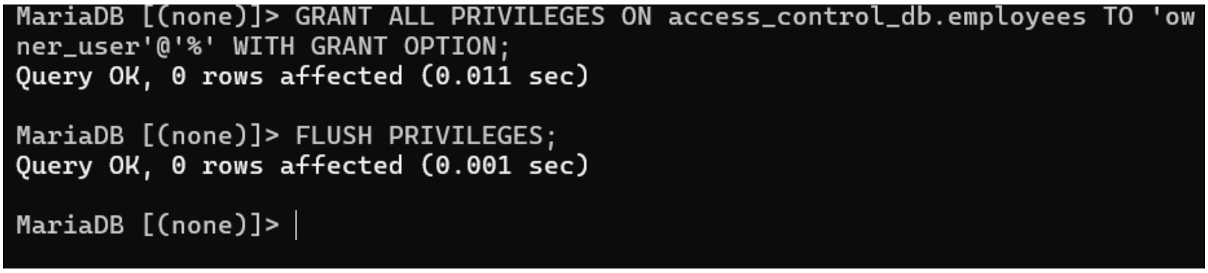
```
MariaDB [(none)]> USE access_control_db;CREATE TABLE employees (
Database changed
    ->  id INT AUTO_INCREMENT PRIMARY KEY,
    ->  name VARCHAR(100),
    ->  position VARCHAR(100)
    -> );
Query OK, 0 rows affected (0.030 sec)

MariaDB [access_control_db]>
```

e. Jika sudah kita dapat mebuat user kedalamnya



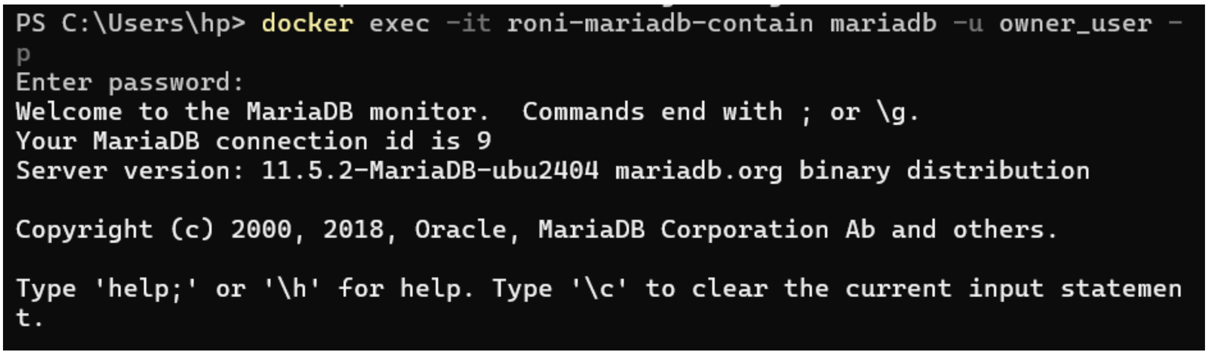
```
MariaDB [(none)]> CREATE USER 'owner_user'@'%' IDENTIFIED BY 'owner_password
';
Query OK, 0 rows affected (0.007 sec)

MariaDB [(none)]> CREATE USER 'guest_user'@'%' IDENTIFIED BY 'guest_password
';
Query OK, 0 rows affected (0.002 sec)

MariaDB [(none)]>
```

f. Setelah itu kita berikan izin kepada owner_user



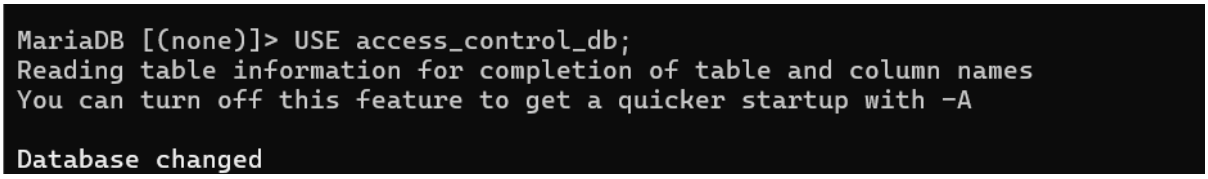
```
MariaDB [(none)]> GRANT ALL PRIVILEGES ON access_control_db.employees TO 'ow
ner_user'@'%' WITH GRANT OPTION;
Query OK, 0 rows affected (0.011 sec)

MariaDB [(none)]> FLUSH PRIVILEGES;
Query OK, 0 rows affected (0.001 sec)

MariaDB [(none)]>
```

g. Jika sudah membuat dan memberikan izin kepada user kita dapat keluar dari sesi root dan masuk ke sesi guest_user



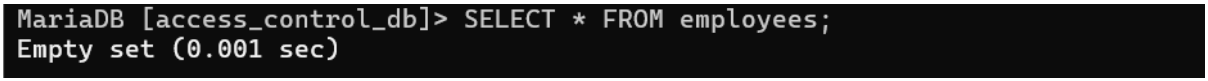
```
PS C:\Users\hp> docker exec -it roni-mariadb-contain mariadb -u owner_user -
p
Enter password:
Welcome to the MariaDB monitor.  Commands end with ; or \g.
Your MariaDB connection id is 9
Server version: 11.5.2-MariaDB-ubu2404 mariadb.org binary distribution

Copyright (c) 2000, 2018, Oracle, MariaDB Corporation Ab and others.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statemen
t.
```

h. Gunakan database yang sudah anda buat

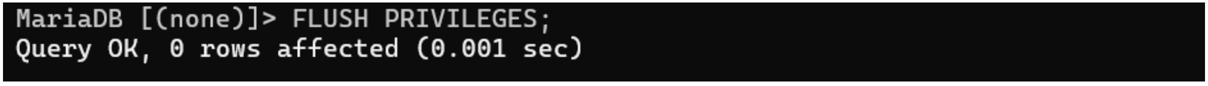


```
MariaDB [(none)]> USE access_control_db;
Reading table information for completion of table and column names
You can turn off this feature to get a quicker startup with -A

Database changed
```

i. Jika sudah kita bisa menuliskan GRANT SELECT ON employees TO 'guest_user'@'%'; dan melakukkan SELECt untuk memebrikan ijin pada guest user untuk membaca tabel database lalu ketik FLUSH untuk menerapkan perubahan

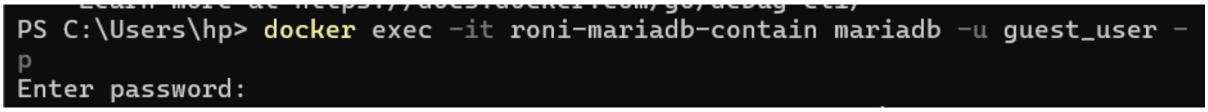


```
MariaDB [access_control_db]> SELECT * FROM employees;
Empty set (0.001 sec)
```



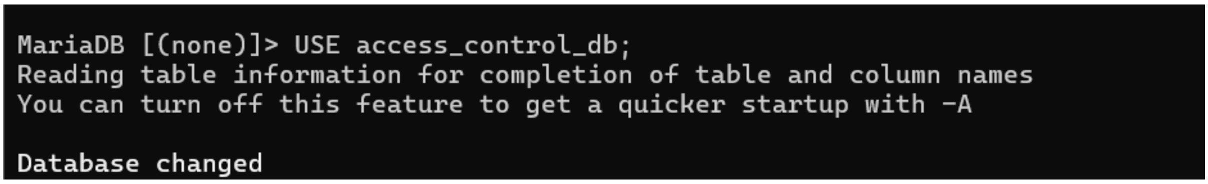
```
MariaDB [(none)]> FLUSH PRIVILEGES;
Query OK, 0 rows affected (0.001 sec)
```

j. Jika sudah kita coa untuk login ke guest_user



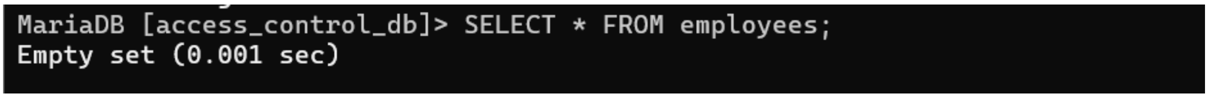
```
PS C:\Users\hp> docker exec -it roni-mariadb-contain mariadb -u guest_user -
p
Enter password:
```

k. Jika sudah gunakan database yang sudah dibuat



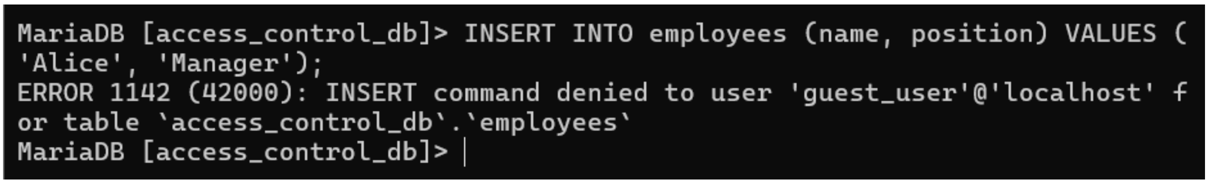
```
MariaDB [(none)]> USE access_control_db;
Reading table information for completion of table and column names
You can turn off this feature to get a quicker startup with -A

Database changed
```

l. Lalu pilih select untuk mengecek perijinan pada guest user



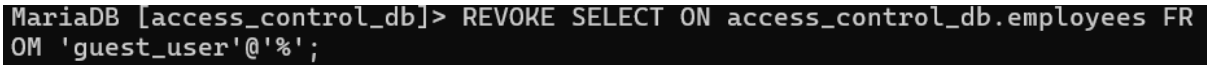
```
MariaDB [access_control_db]> SELECT * FROM employees;
Empty set (0.001 sec)
```

m. Dan uji apakah guest bisa melakukan insert, jika error maka konfigurasi sudah benar

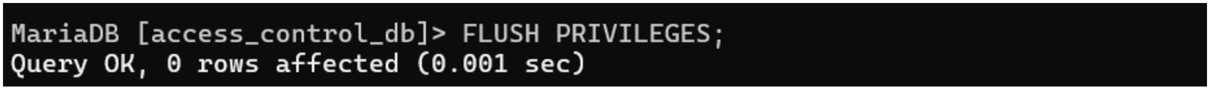


```
MariaDB [access_control_db]> INSERT INTO employees (name, position) VALUES (
'Alice', 'Manager');
ERROR 1142 (42000): INSERT command denied to user 'guest_user'@'localhost' f
or table `access_control_db`.`employees`
MariaDB [access_control_db]> |
```

n. Jika sudah keluar dari database guest dan pindah ke owner user lalu cabut ijin select pada guest user dan flush privilage untuk menetapkah perubahan perijinan.

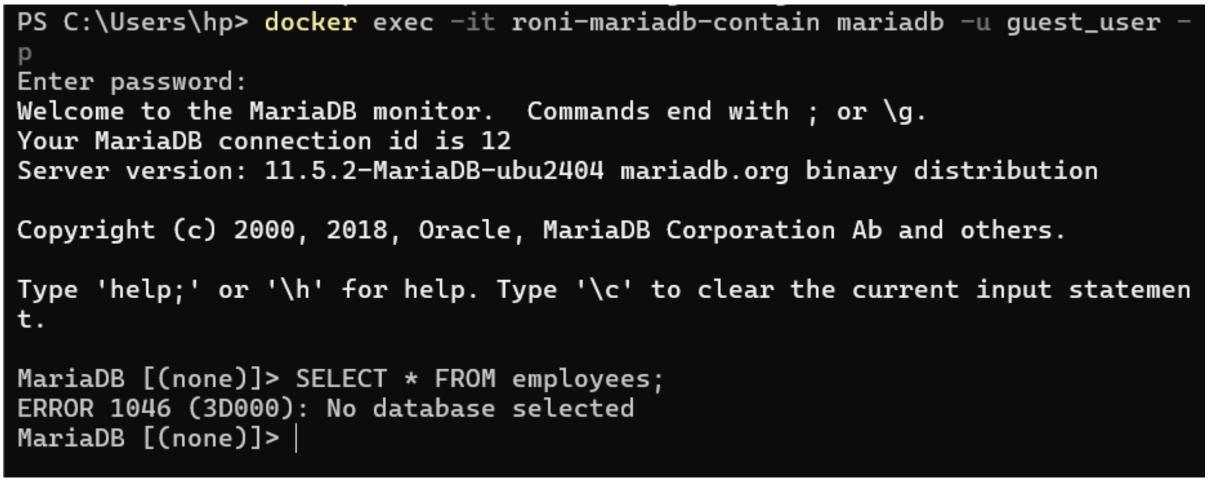


```
MariaDB [access_control_db]> REVOKE SELECT ON access_control_db.employees FR
OM 'guest_user'@'%';
```



```
MariaDB [access_control_db]> FLUSH PRIVILEGES;
Query OK, 0 rows affected (0.001 sec)
```

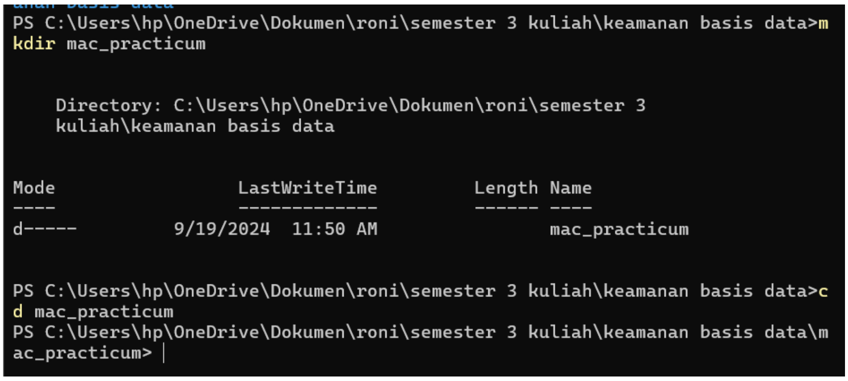
o. Jika sudah keluar dari owner dan masuk ke guest agar melihat perubahan yang terjadi, seharusnya guest tidak bisa lagi melakukan select



```
PS C:\Users\hp> docker exec -it roni-mariadb-contain mariadb -u guest_user -
p
Enter password:
Welcome to the MariaDB monitor.  Commands end with ; or \g.
Your MariaDB connection id is 12
Server version: 11.5.2-MariaDB-ubu2404 mariadb.org binary distribution

Copyright (c) 2000, 2018, Oracle, MariaDB Corporation Ab and others.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statemen
t.

MariaDB [(none)]> SELECT * FROM employees;
ERROR 1046 (3D000): No database selected
MariaDB [(none)]> |
```

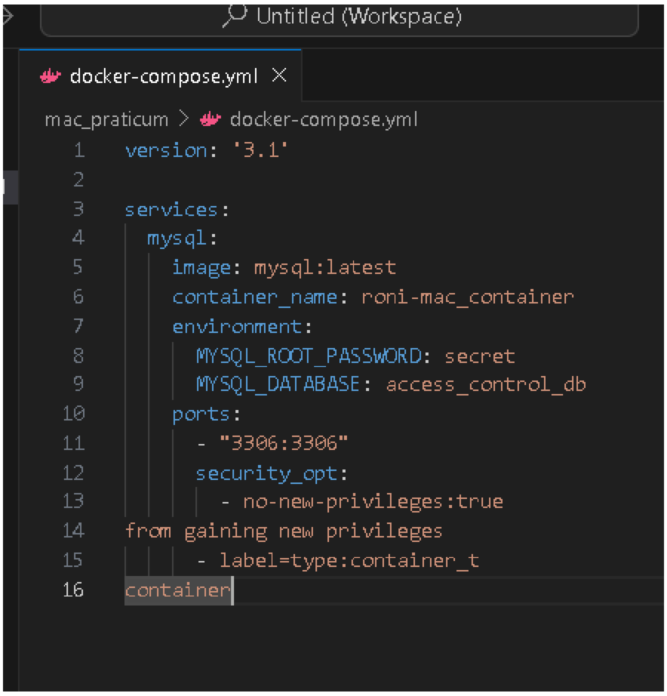
2. **Mendemonstrasikan penerapan Mandatory Access Control (MAC) pada lingkungan basis data yang di-host menggunakan Docker, dengan penekanan pada keamanan berlapis di level sistem dan database.**

   a. Buat sebuah direktori dengan nama mac_praticum lalu pindah ke direktori tersebut



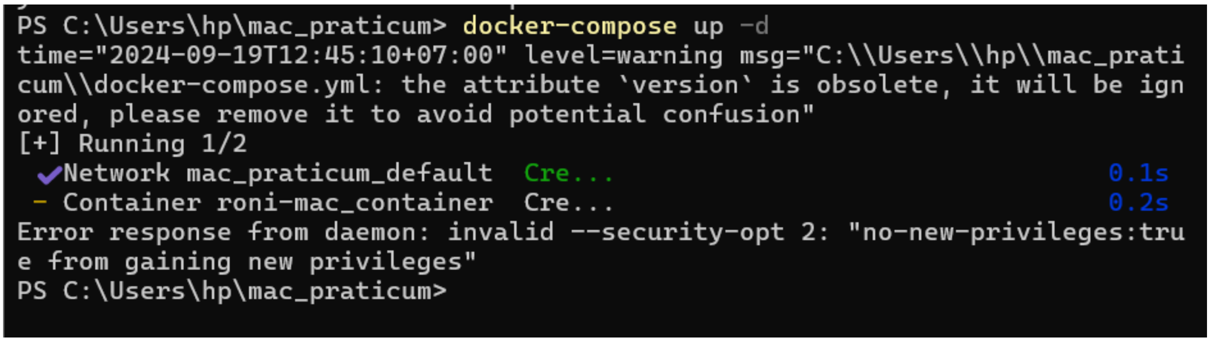
```
PS C:\Users\hp\OneDrive\Dokumen\roni\semester 3 kuliah\keamanan basis data>m
kdir mac_practicum


    Directory: C:\Users\hp\OneDrive\Dokumen\roni\semester 3
    kuliah\keamanan basis data


Mode                 LastWriteTime         Length Name
----                 -------------         ------ ----
d-----         9/19/2024  11:50 AM                mac_practicum


PS C:\Users\hp\OneDrive\Dokumen\roni\semester 3 kuliah\keamanan basis data>c
d mac_practicum
PS C:\Users\hp\OneDrive\Dokumen\roni\semester 3 kuliah\keamanan basis data\m
ac_practicum> |
```

b. Jika sudah buat file docker-compose.yml kalian bisa malakukannya dengan nano, notepad, atau dengan menggunakan code editor jika kalian mengerjakannya di windows, dengan compose ini juga kita bisa auto membuild sekaligus container dan databasenya

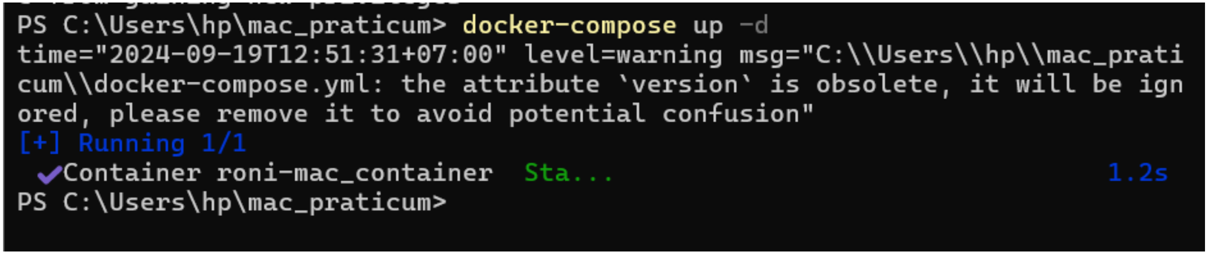


```
version: '3.1'

services:
  mysql:
    image: mysql:latest
    container_name: roni-mac_container
    environment:
      MYSQL_ROOT_PASSWORD: secret
      MYSQL_DATABASE: access_control_db
    ports:
      - "3306:3306"
      security_opt:
        - no-new-privileges:true
from gaining new privileges
      - label=type:container_t
container
```

c. Jika sudah anda bisa menjalankan compose tadi untuk membuild container serta database dengan docker-compose up -d

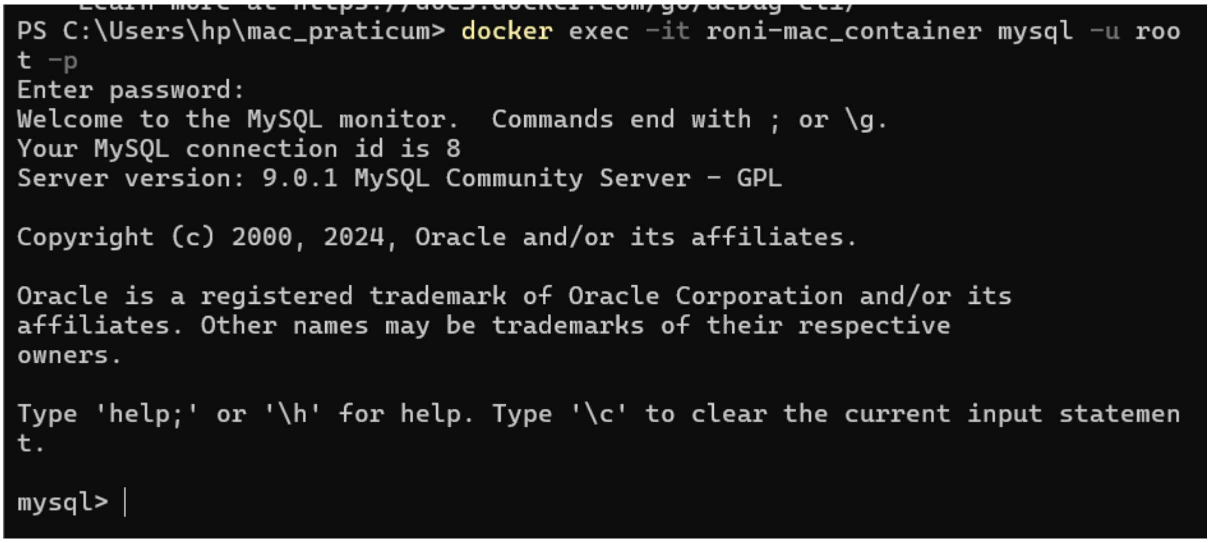


```
PS C:\Users\hp\mac_praticum> docker-compose up -d
time="2024-09-19T12:45:10+07:00" level=warning msg="C:\\Users\\hp\\mac_prati
cum\\docker-compose.yml: the attribute `version` is obsolete, it will be ign
ored, please remove it to avoid potential confusion"
[+] Running 1/2
 ✔Network mac_praticum_default  Cre...                                   0.1s
 - Container roni-mac_container  Cre...                                   0.2s
Error response from daemon: invalid --security-opt 2: "no-new-privileges:tru
e from gaining new privileges"
PS C:\Users\hp\mac_praticum>
```



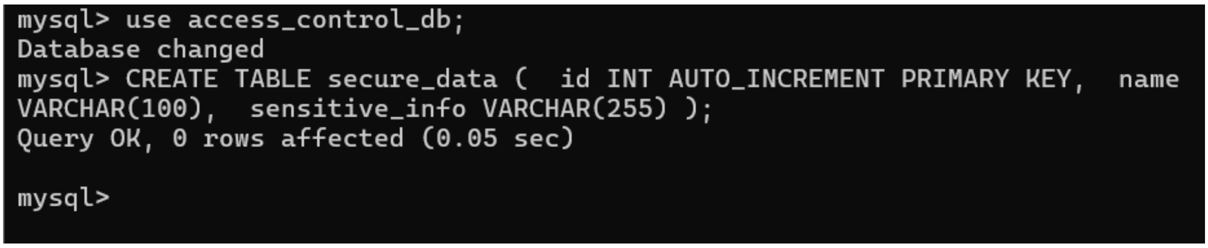
```
PS C:\Users\hp\mac_praticum> docker-compose up -d
time="2024-09-19T12:51:31+07:00" level=warning msg="C:\\Users\\hp\\mac_prati
cum\\docker-compose.yml: the attribute `version` is obsolete, it will be ign
ored, please remove it to avoid potential confusion"
[+] Running 1/1
 ✔Container roni-mac_container  Sta...                                   1.2s
PS C:\Users\hp\mac_praticum>
```

d. Setelah itu login ke container yng tadi kalian buat



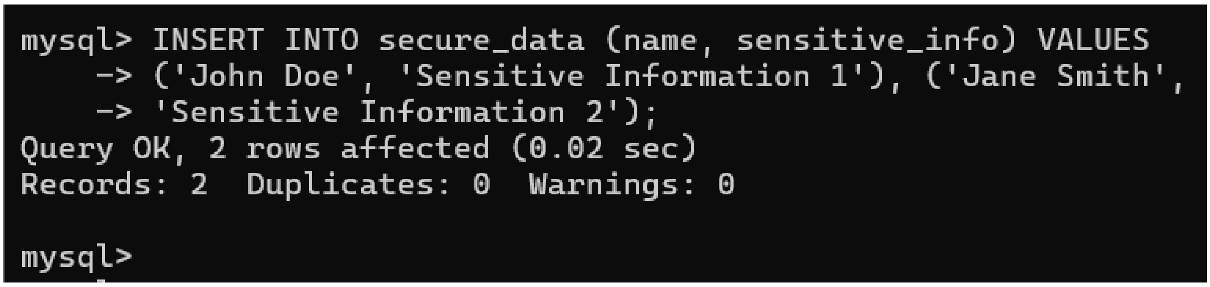
```
    Learn more at https://docs.docker.com/go/debug-cli/
PS C:\Users\hp\mac_praticum> docker exec -it roni-mac_container mysql -u roo
t -p
Enter password:
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 8
Server version: 9.0.1 MySQL Community Server - GPL

Copyright (c) 2000, 2024, Oracle and/or its affiliates.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statemen
t.

mysql>
```

e. Jika sudah masuk ke database yang sudah anda buat dan create table



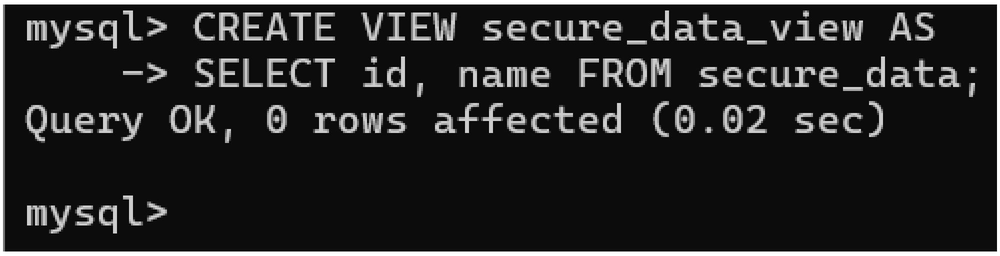
```
mysql> use access_control_db;
Database changed
mysql> CREATE TABLE secure_data (  id INT AUTO_INCREMENT PRIMARY KEY,  name
VARCHAR(100),  sensitive_info VARCHAR(255) );
Query OK, 0 rows affected (0.05 sec)

mysql>
```

f. Lalu masukkan data sesuai dengna table yang sudah kalian buat



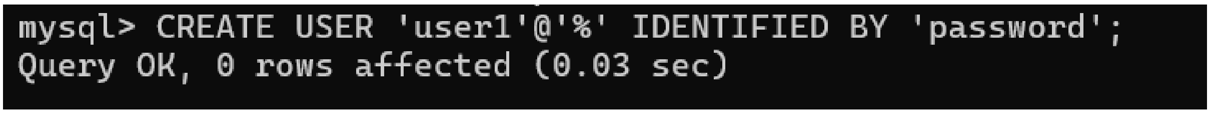
```
mysql> INSERT INTO secure_data (name, sensitive_info) VALUES
    -> ('John Doe', 'Sensitive Information 1'), ('Jane Smith',
    -> 'Sensitive Information 2');
Query OK, 2 rows affected (0.02 sec)
Records: 2  Duplicates: 0  Warnings: 0

mysql>
```

g. Jika sudah anda dapat menambahkan create view pada pengguna terntentu



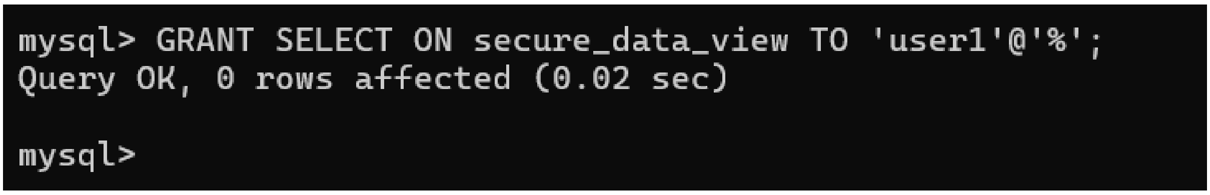
```
mysql> CREATE VIEW secure_data_view AS
    -> SELECT id, name FROM secure_data;
Query OK, 0 rows affected (0.02 sec)

mysql>
```

h. Jika sudah Create buat pengguna baru untuk mengakses view



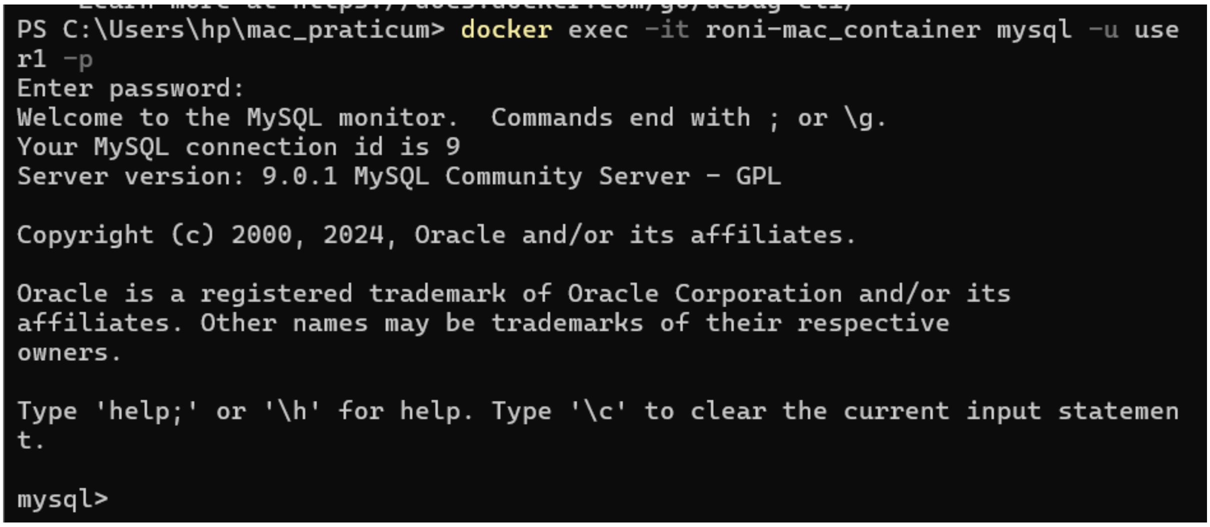
```
mysql> CREATE USER 'user1'@'%' IDENTIFIED BY 'password';
Query OK, 0 rows affected (0.03 sec)
```

i. Lalu berikan hak akes pada pengguna tersebut



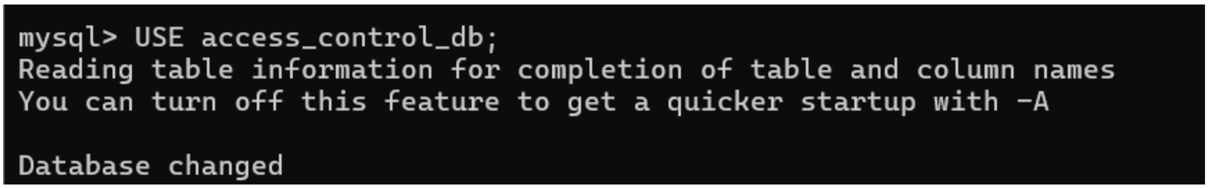
```
mysql> GRANT SELECT ON secure_data_view TO 'user1'@'%';
Query OK, 0 rows affected (0.02 sec)

mysql>
```

j. Setelah itu keluar dari container root dan uji dengan masuk ke container pengguna

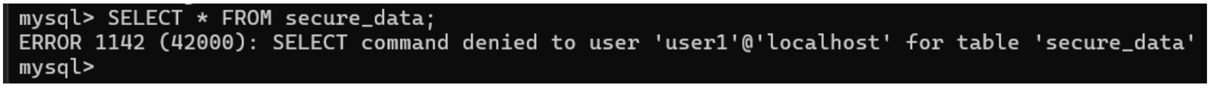


```
PS C:\Users\hp\mac_praticum> docker exec -it roni-mac_container mysql -u use
r1 -p
Enter password:
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 9
Server version: 9.0.1 MySQL Community Server - GPL

Copyright (c) 2000, 2024, Oracle and/or its affiliates.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statemen
t.

mysql>
```

k. Lalu masuk ke database anda dan ketik select secure data maka akan error karena hanya diberi akses ke data view

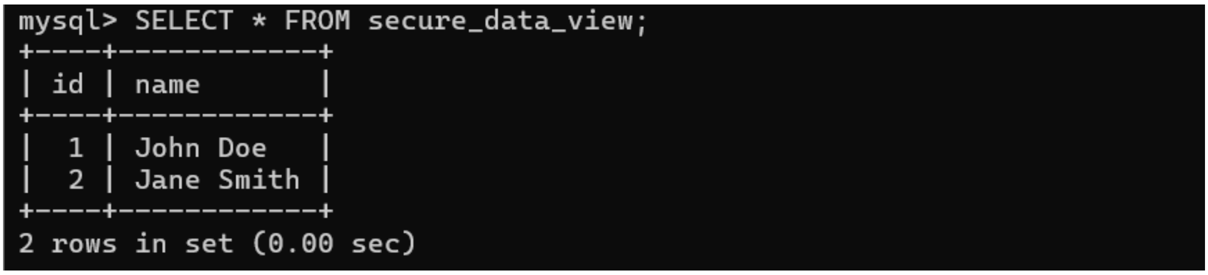


```
mysql> USE access_control_db;
Reading table information for completion of table and column names
You can turn off this feature to get a quicker startup with -A

Database changed
```



```
mysql> SELECT * FROM secure_data;
ERROR 1142 (42000): SELECT command denied to user 'user1'@'localhost' for table 'secure_data'
mysql>
```

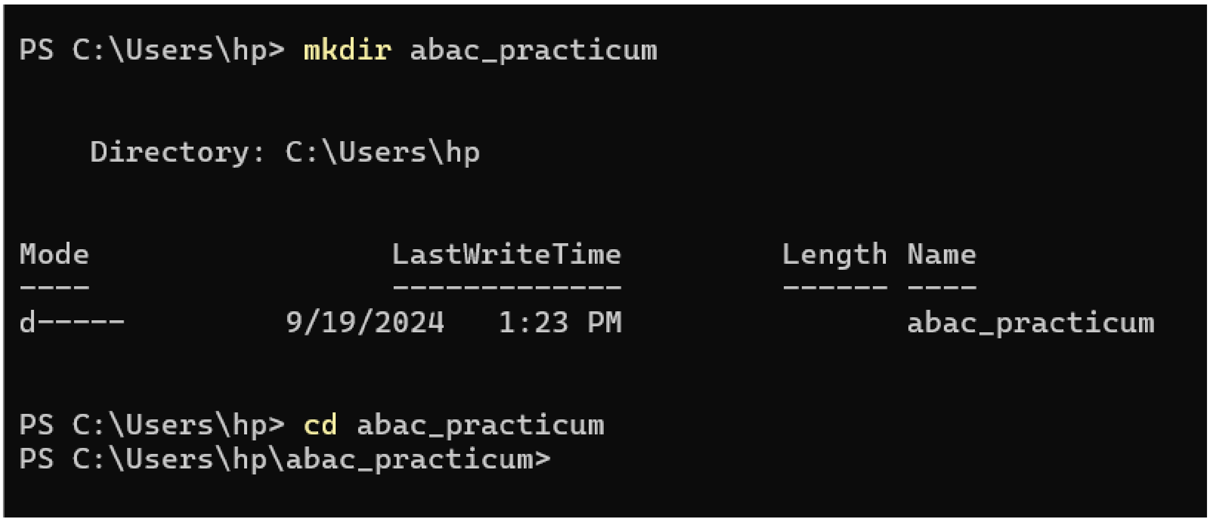
l. Jika sudah sekarang kita uji dengan secure data view maka data akan muncul



```
mysql> SELECT * FROM secure_data_view;
+----+------------+
| id | name       |
+----+------------+
|  1 | John Doe   |
|  2 | Jane Smith |
+----+------------+
2 rows in set (0.00 sec)
```

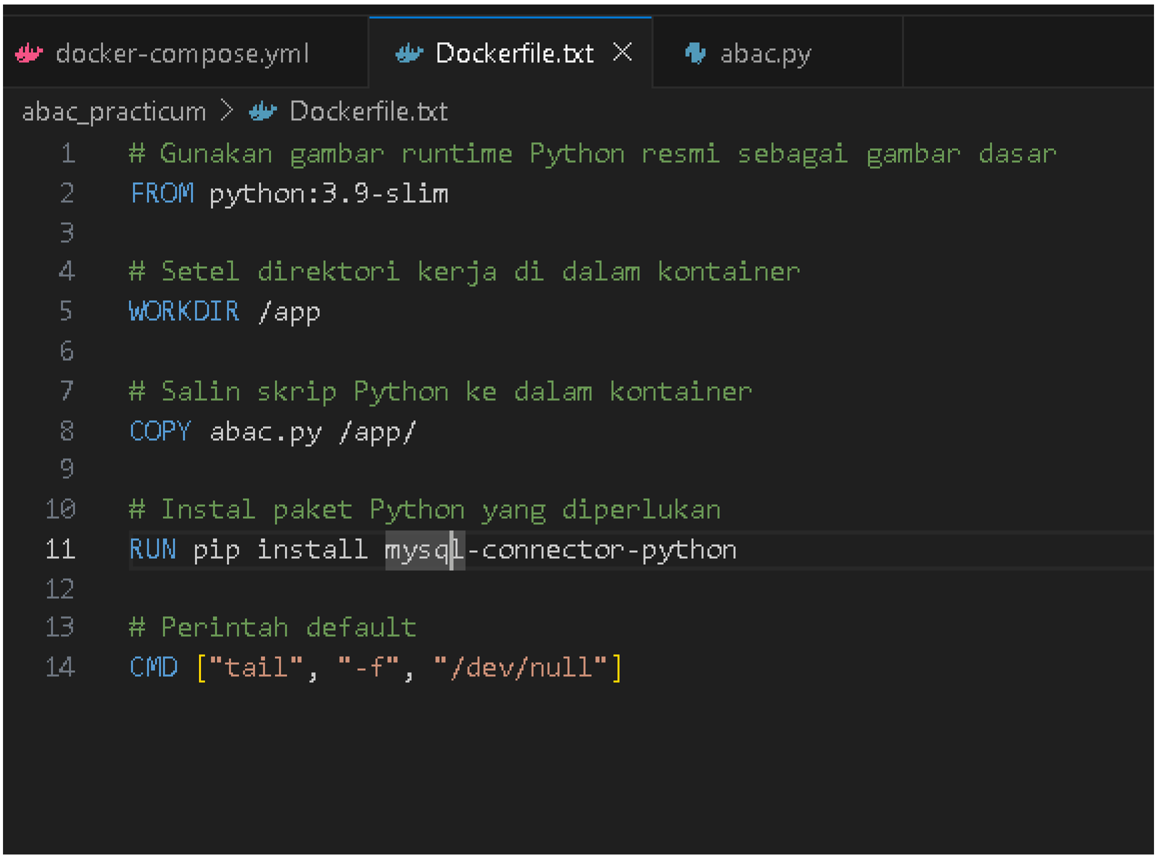
3. **Mengimplementasikan Attribute-Based Access Control (ABAC) pada lingkungan basis data MySQL dan aplikasi Python yang berjalan dalam kontainer Docker**

   a. Pertama buat terlebih dahulu direktori abac_praticum dan masuk ke dalam direktori



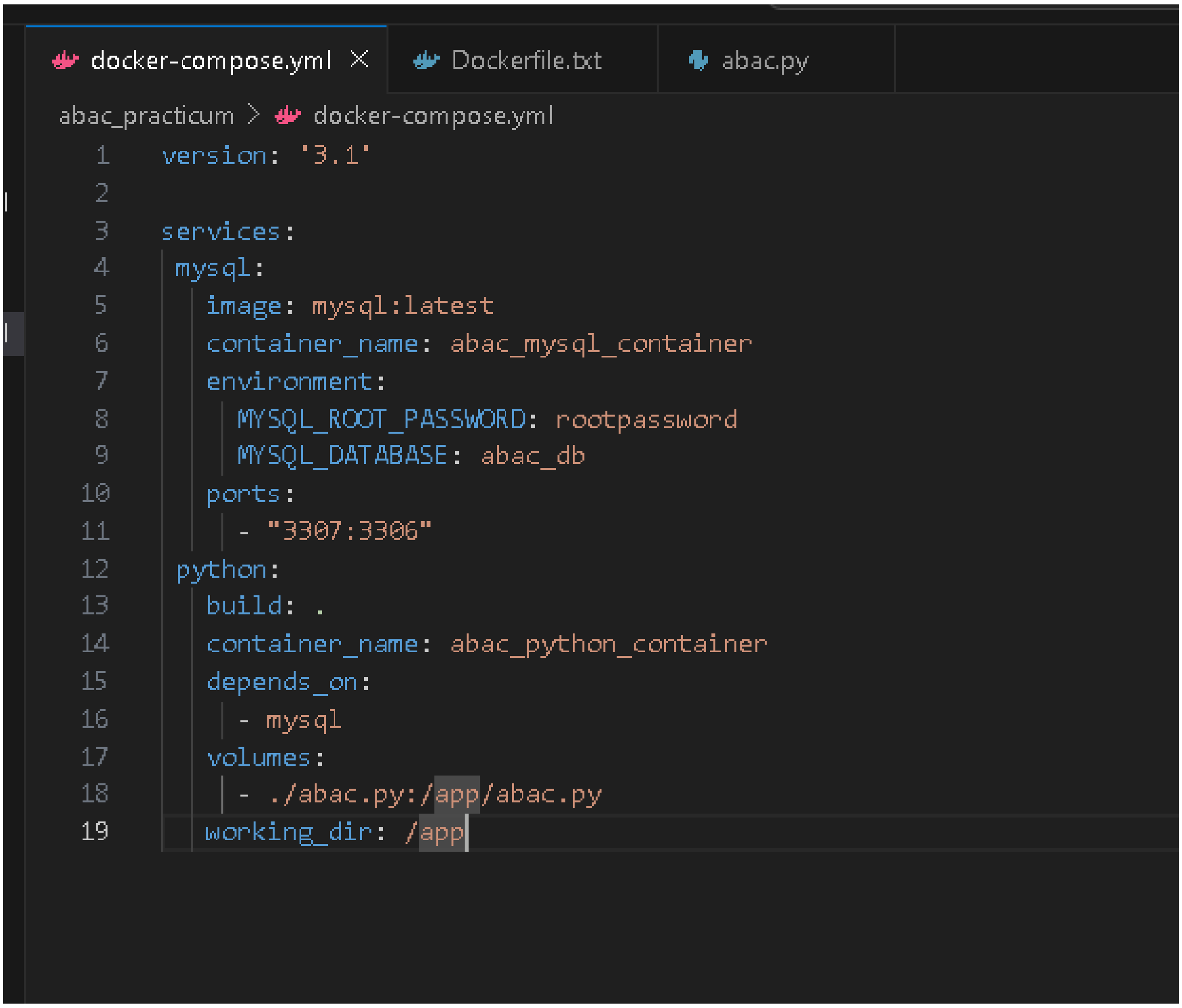
```
PS C:\Users\hp> mkdir abac_practicum


    Directory: C:\Users\hp


Mode                 LastWriteTime         Length Name
----                 -------------         ------ ----
d-----         9/19/2024   1:23 PM                abac_practicum


PS C:\Users\hp> cd abac_practicum
PS C:\Users\hp\abac_practicum>
```

   b. Mulai buat Dockerfile didalam direktori sama seperti sebelumnya kalian bisa memakai kode editor apapun



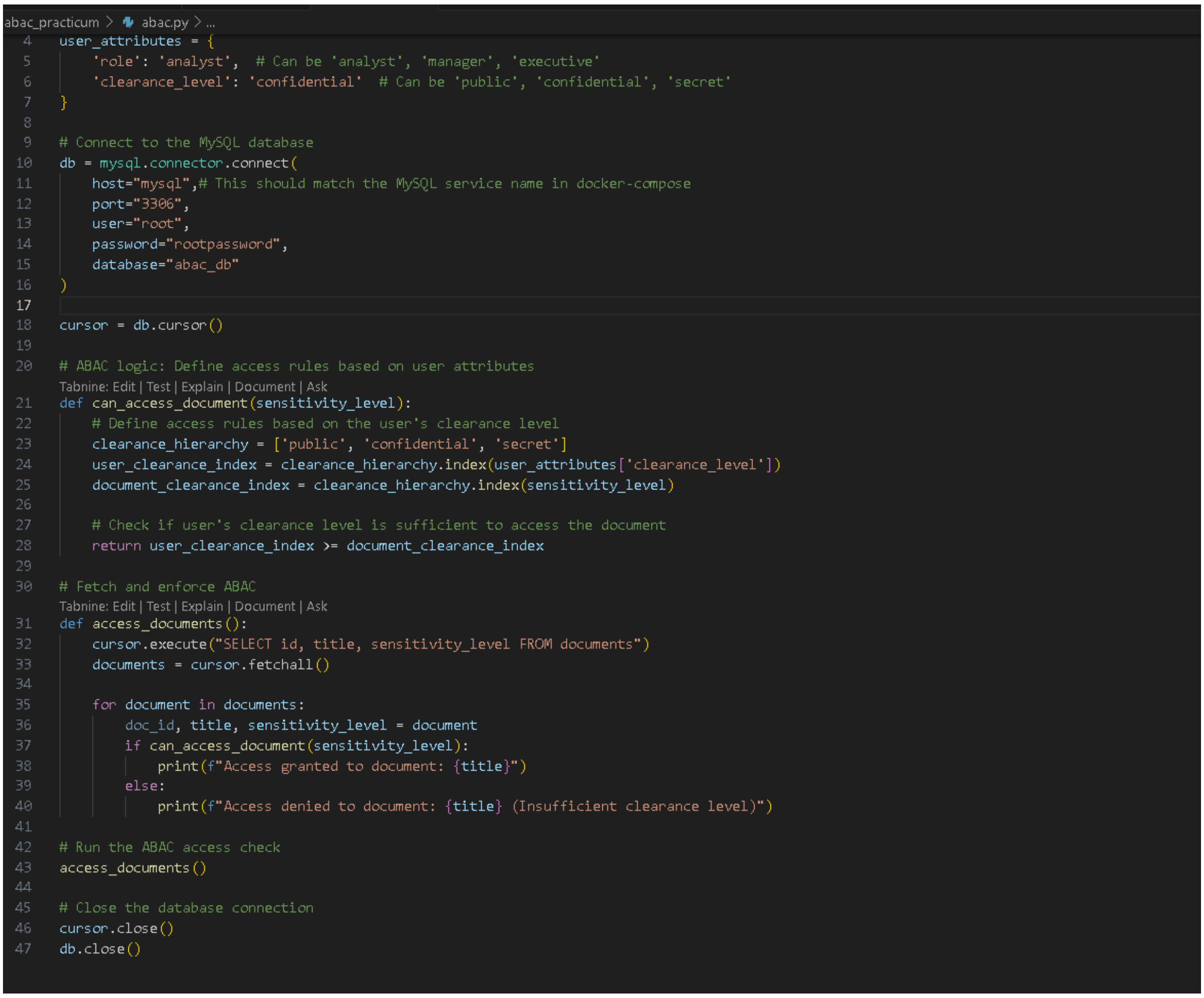
```
# Gunakan gambar runtime Python resmi sebagai gambar dasar
FROM python:3.9-slim

# Setel direktori kerja di dalam kontainer
WORKDIR /app

# Salin skrip Python ke dalam kontainer
COPY abac.py /app/

# Instal paket Python yang diperlukan
RUN pip install mysql-connector-python

# Perintah default
CMD ["tail", "-f", "/dev/null"]
```

c. Jika sudah berikutnya kalian buat file docker-compose.yml



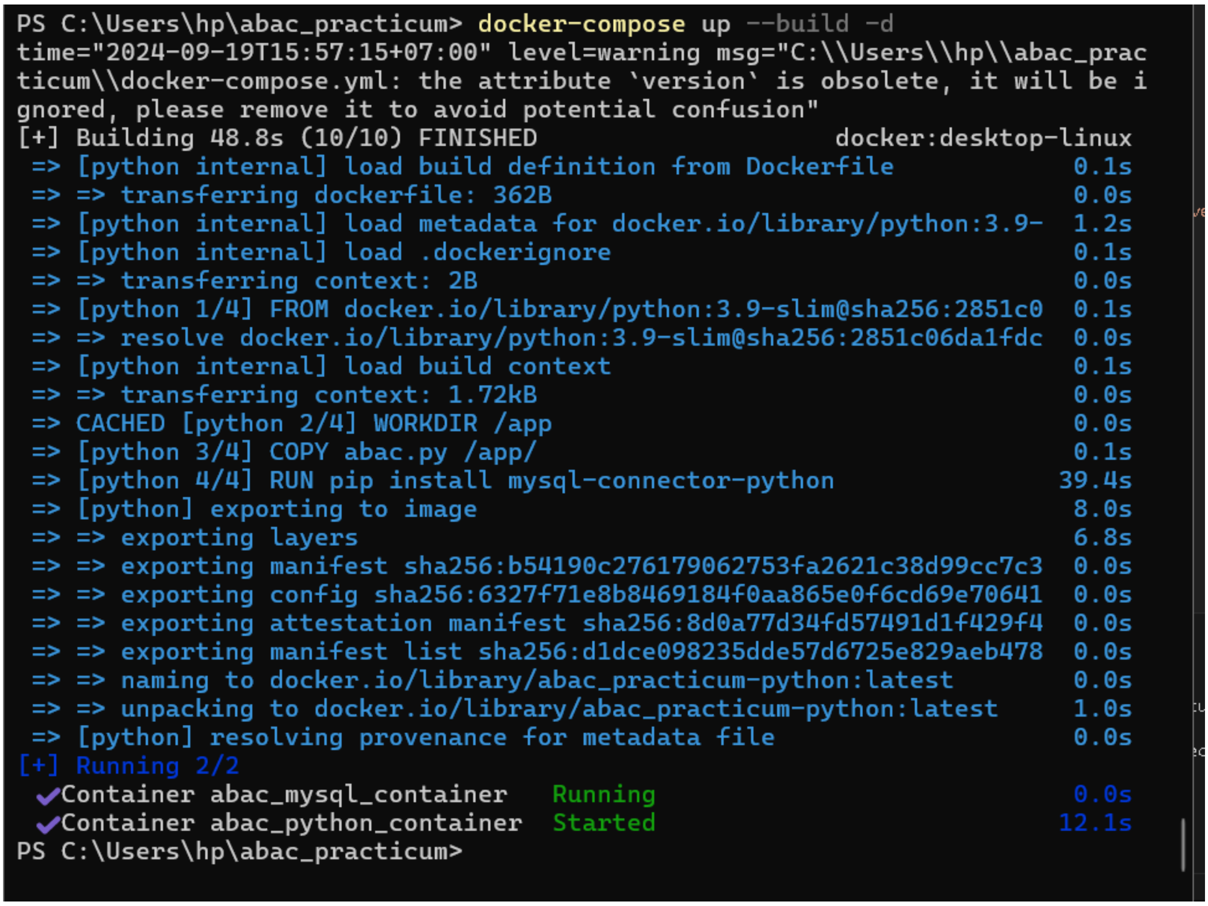
```yaml
version: '3.1'

services:
 mysql:
   image: mysql:latest
   container_name: abac_mysql_container
   environment:
     MYSQL_ROOT_PASSWORD: rootpassword
     MYSQL_DATABASE: abac_db
   ports:
     - "3307:3306"
 python:
   build: .
   container_name: abac_python_container
   depends_on:
     - mysql
   volumes:
     - ./abac.py:/app/abac.py
   working_dir: /app
```

d. Lalu buat script abac.py



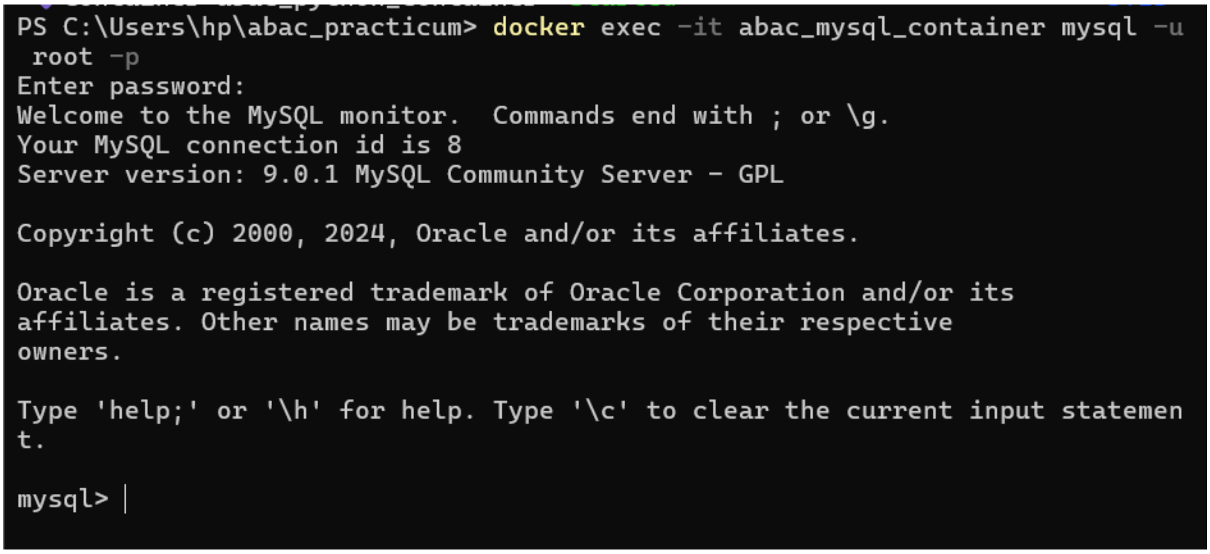
```python
user_attributes = {
    'role': 'analyst',  # Can be 'analyst', 'manager', 'executive'
    'clearance_level': 'confidential'  # Can be 'public', 'confidential', 'secret'
}

# Connect to the MySQL database
db = mysql.connector.connect(
    host="mysql",# This should match the MySQL service name in docker-compose
    port="3306",
    user="root",
    password="rootpassword",
    database="abac_db"
)

cursor = db.cursor()

# ABAC logic: Define access rules based on user attributes
Tabnine: Edit | Test | Explain | Document | Ask
def can_access_document(sensitivity_level):
    # Define access rules based on the user's clearance level
    clearance_hierarchy = ['public', 'confidential', 'secret']
    user_clearance_index = clearance_hierarchy.index(user_attributes['clearance_level'])
    document_clearance_index = clearance_hierarchy.index(sensitivity_level)

    # Check if user's clearance level is sufficient to access the document
    return user_clearance_index >= document_clearance_index

# Fetch and enforce ABAC
Tabnine: Edit | Test | Explain | Document | Ask
def access_documents():
    cursor.execute("SELECT id, title, sensitivity_level FROM documents")
    documents = cursor.fetchall()

    for document in documents:
        doc_id, title, sensitivity_level = document
        if can_access_document(sensitivity_level):
            print(f"Access granted to document: {title}")
        else:
            print(f"Access denied to document: {title} (Insufficient clearance level)")

# Run the ABAC access check
access_documents()

# Close the database connection
cursor.close()
db.close()
```

e. Jika semua sudah sekarang kita coba build ketiga file diatas



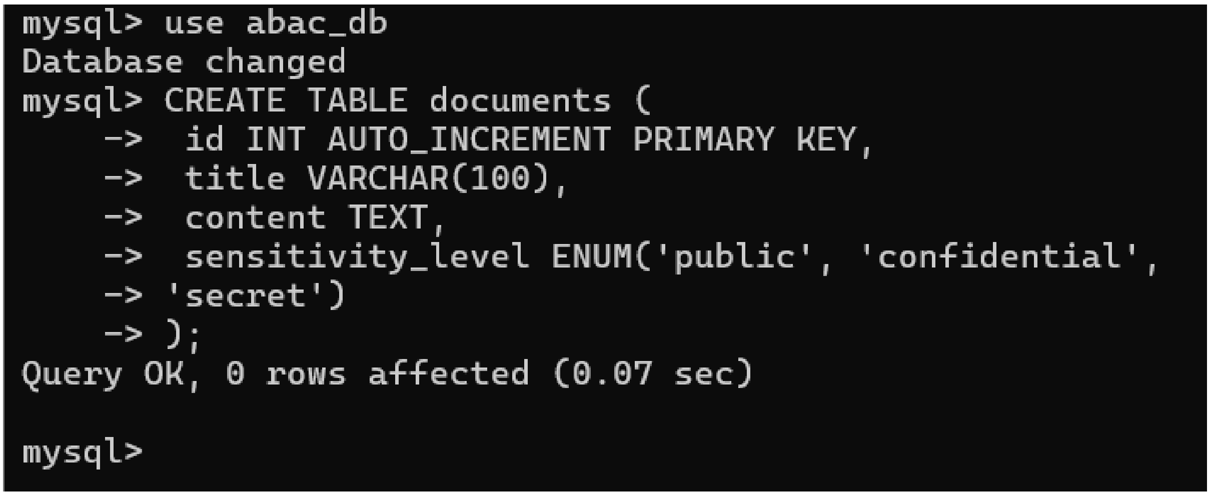
```
PS C:\Users\hp\abac_practicum> docker-compose up --build -d
time="2024-09-19T15:57:15+07:00" level=warning msg="C:\\Users\\hp\\abac_prac
ticum\\docker-compose.yml: the attribute `version` is obsolete, it will be i
gnored, please remove it to avoid potential confusion"
[+] Building 48.8s (10/10) FINISHED                    docker:desktop-linux
 => [python internal] load build definition from Dockerfile            0.1s
 => => transferring dockerfile: 362B                                    0.0s
 => [python internal] load metadata for docker.io/library/python:3.9-  1.2s
 => [python internal] load .dockerignore                                0.1s
 => => transferring context: 2B                                         0.0s
 => [python 1/4] FROM docker.io/library/python:3.9-slim@sha256:2851c0  0.1s
 => => resolve docker.io/library/python:3.9-slim@sha256:2851c06da1fdc  0.0s
 => [python internal] load build context                                0.1s
 => => transferring context: 1.72kB                                     0.0s
 => CACHED [python 2/4] WORKDIR /app                                    0.0s
 => [python 3/4] COPY abac.py /app/                                     0.1s
 => [python 4/4] RUN pip install mysql-connector-python               39.4s
 => [python] exporting to image                                         8.0s
 => => exporting layers                                                 6.8s
 => => exporting manifest sha256:b54190c276179062753fa2621c38d99cc7c3  0.0s
 => => exporting config sha256:6327f71e8b8469184f0aa865e0f6cd69e70641  0.0s
 => => exporting attestation manifest sha256:8d0a77d34fd57491d1f429f4  0.0s
 => => exporting manifest list sha256:d1dce098235dde57d6725e829aeb478  0.0s
 => => naming to docker.io/library/abac_practicum-python:latest         0.0s
 => => unpacking to docker.io/library/abac_practicum-python:latest      1.0s
 => [python] resolving provenance for metadata file                     0.0s
[+] Running 2/2
 ✔Container abac_mysql_container   Running                              0.0s
 ✔Container abac_python_container  Started                             12.1s
PS C:\Users\hp\abac_practicum>
```

f. Jika sudah masuk ke container yang sudah kita buat tadi



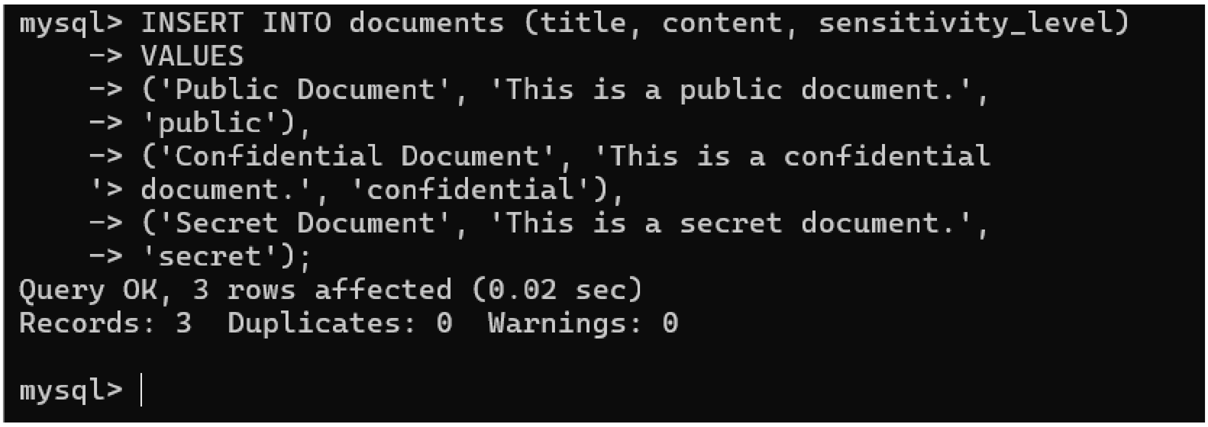
```
PS C:\Users\hp\abac_practicum> docker exec -it abac_mysql_container mysql -u
 root -p
Enter password:
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 8
Server version: 9.0.1 MySQL Community Server - GPL

Copyright (c) 2000, 2024, Oracle and/or its affiliates.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statemen
t.

mysql>
```

g. Lalu gunakan databasnya dan buat table



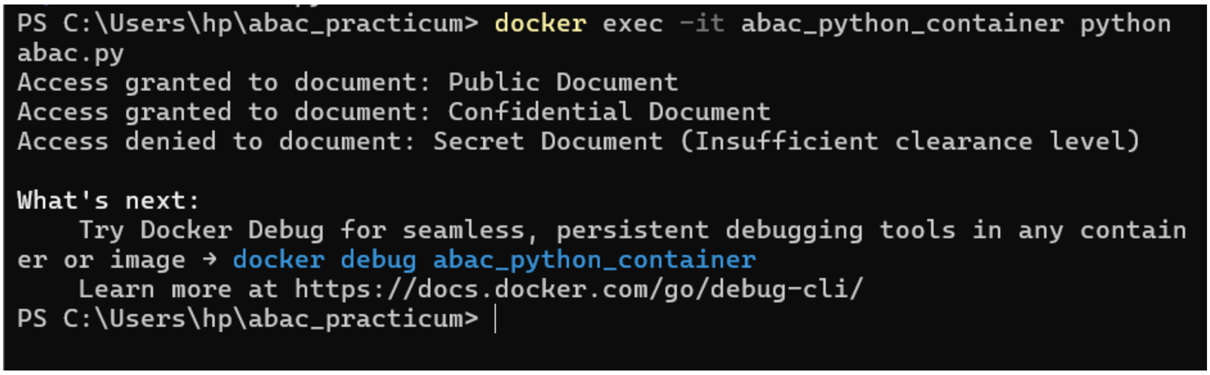
```
mysql> use abac_db
Database changed
mysql> CREATE TABLE documents (
    ->  id INT AUTO_INCREMENT PRIMARY KEY,
    ->  title VARCHAR(100),
    ->  content TEXT,
    ->  sensitivity_level ENUM('public', 'confidential',
    -> 'secret')
    -> );
Query OK, 0 rows affected (0.07 sec)

mysql>
```

h. Setelah itu masukkan data kedalam table



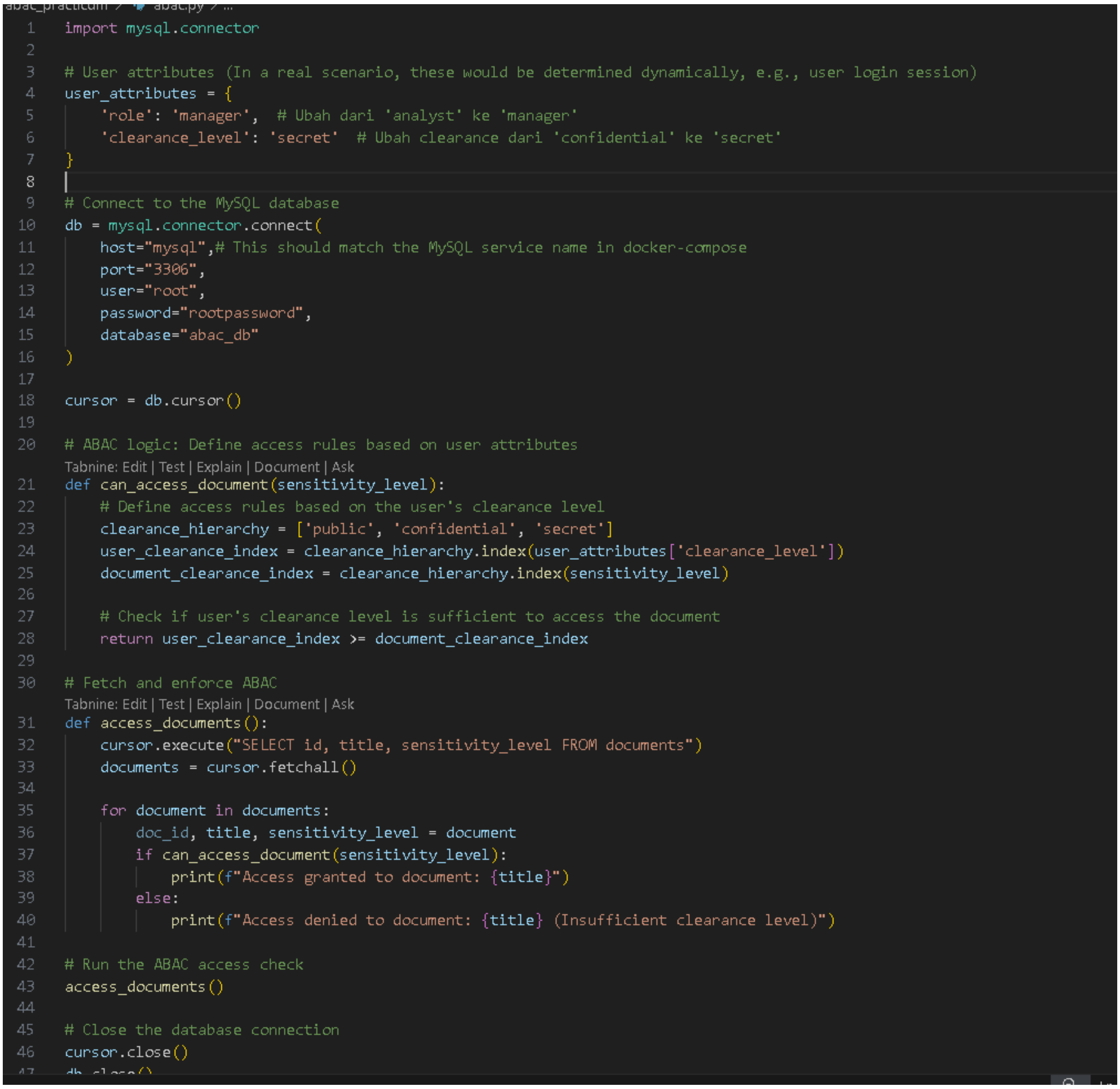
```
mysql> INSERT INTO documents (title, content, sensitivity_level)
    -> VALUES
    -> ('Public Document', 'This is a public document.',
    -> 'public'),
    -> ('Confidential Document', 'This is a confidential
    '> document.', 'confidential'),
    -> ('Secret Document', 'This is a secret document.',
    -> 'secret');
Query OK, 3 rows affected (0.02 sec)
Records: 3  Duplicates: 0  Warnings: 0

mysql> |
```

i. Jika sudah keluar dari container tersbut dan uji file abac.py



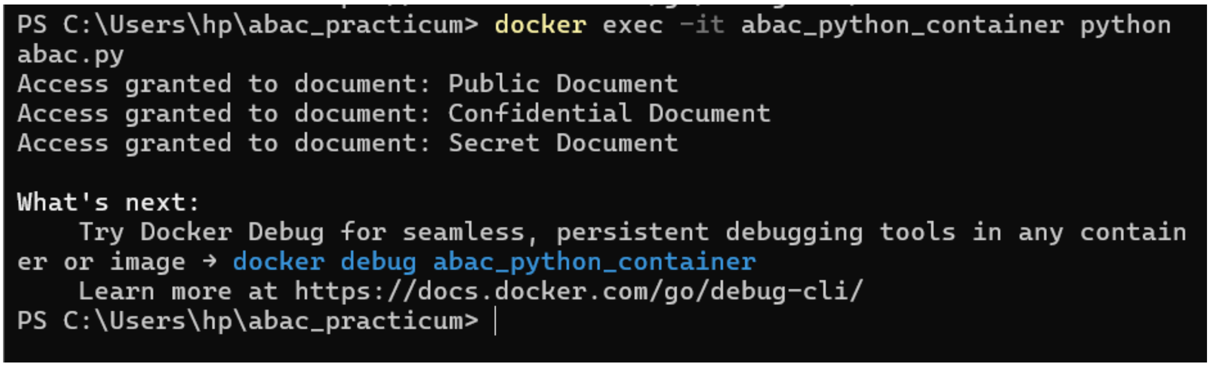
```
PS C:\Users\hp\abac_practicum> docker exec -it abac_python_container python
abac.py
Access granted to document: Public Document
Access granted to document: Confidential Document
Access denied to document: Secret Document (Insufficient clearance level)

What's next:
    Try Docker Debug for seamless, persistent debugging tools in any contain
er or image → docker debug abac_python_container
    Learn more at https://docs.docker.com/go/debug-cli/
PS C:\Users\hp\abac_practicum> |
```

j. Jika sudah pada bagian atribute di script.py dapat kalian ubah juga



```
abac_practicum > abac.py > ...
import mysql.connector

# User attributes (In a real scenario, these would be determined dynamically, e.g., user login session)
user_attributes = {
    'role': 'manager',  # Ubah dari 'analyst' ke 'manager'
    'clearance_level': 'secret'  # Ubah clearance dari 'confidential' ke 'secret'
}

# Connect to the MySQL database
db = mysql.connector.connect(
    host="mysql",# This should match the MySQL service name in docker-compose
    port="3306",
    user="root",
    password="rootpassword",
    database="abac_db"
)

cursor = db.cursor()

# ABAC logic: Define access rules based on user attributes
Tabnine: Edit | Test | Explain | Document | Ask
def can_access_document(sensitivity_level):
    # Define access rules based on the user's clearance level
    clearance_hierarchy = ['public', 'confidential', 'secret']
    user_clearance_index = clearance_hierarchy.index(user_attributes['clearance_level'])
    document_clearance_index = clearance_hierarchy.index(sensitivity_level)

    # Check if user's clearance level is sufficient to access the document
    return user_clearance_index >= document_clearance_index

# Fetch and enforce ABAC
Tabnine: Edit | Test | Explain | Document | Ask
def access_documents():
    cursor.execute("SELECT id, title, sensitivity_level FROM documents")
    documents = cursor.fetchall()

    for document in documents:
        doc_id, title, sensitivity_level = document
        if can_access_document(sensitivity_level):
            print(f"Access granted to document: {title}")
        else:
            print(f"Access denied to document: {title} (Insufficient clearance level)")

# Run the ABAC access check
access_documents()

# Close the database connection
cursor.close()
db.close()
```

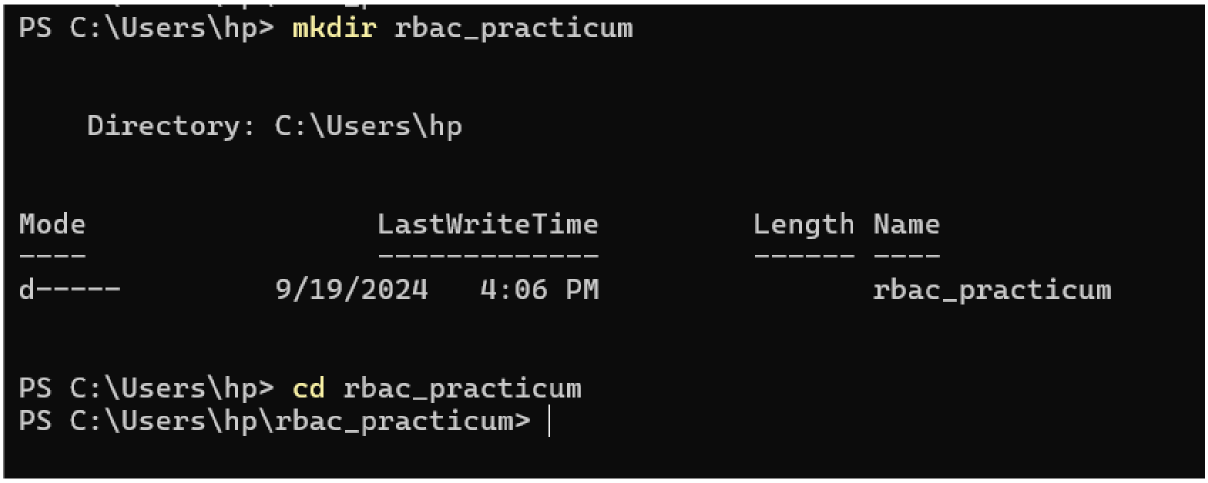
k. Dan hasilnya seperti dibawah ini



```
PS C:\Users\hp\abac_practicum> docker exec -it abac_python_container python
abac.py
Access granted to document: Public Document
Access granted to document: Confidential Document
Access granted to document: Secret Document

What's next:
    Try Docker Debug for seamless, persistent debugging tools in any contain
er or image → docker debug abac_python_container
    Learn more at https://docs.docker.com/go/debug-cli/
PS C:\Users\hp\abac_practicum> 
```

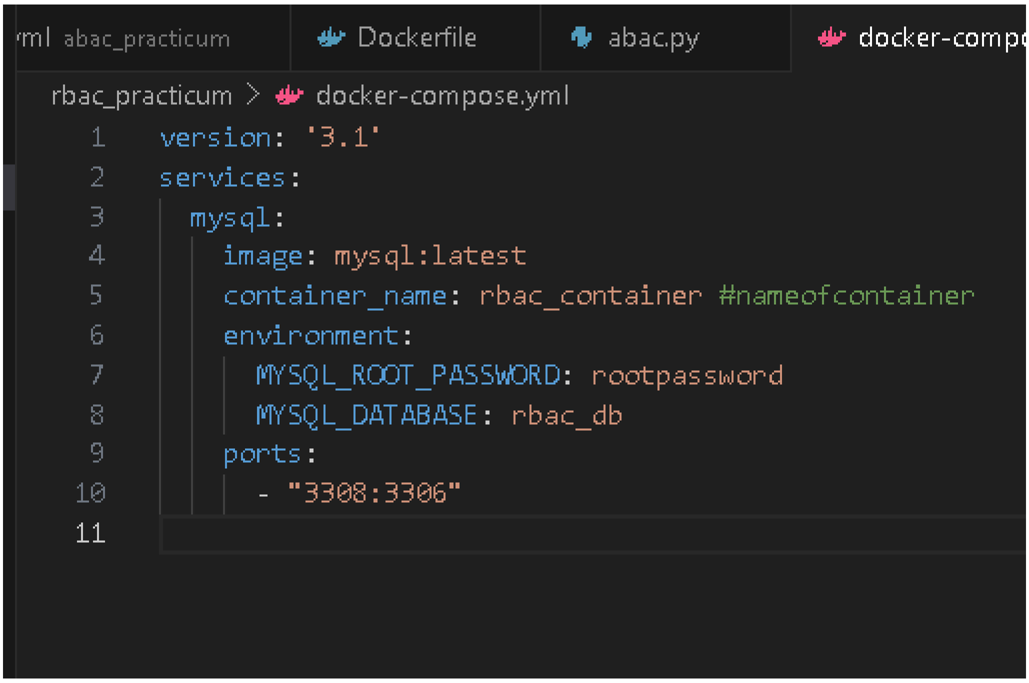
4. **Mengimplementasikan Role-Based Access Control (RBAC) menggunakan MySQL di dalam lingkungan Docker.**

   a. Buat direktori rba_practicum dan masuk ke dalam direktorinya



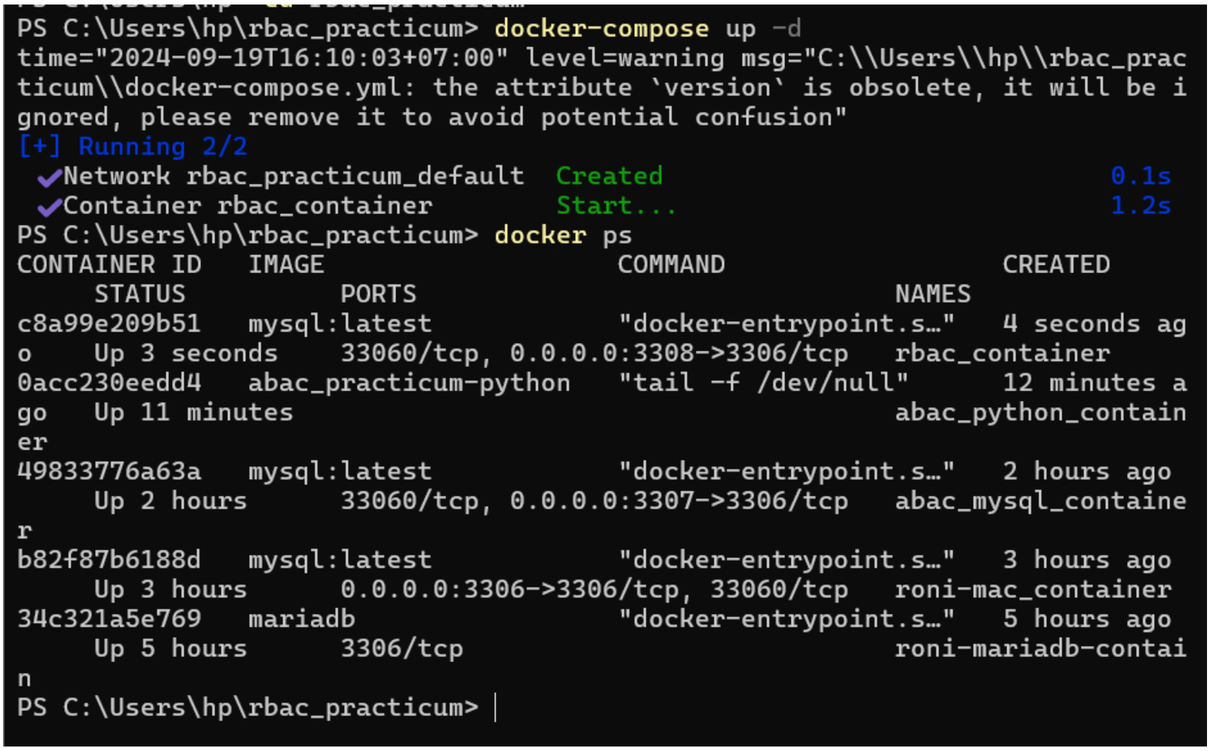
```
PS C:\Users\hp> mkdir rbac_practicum


    Directory: C:\Users\hp


Mode                 LastWriteTime         Length Name
----                 -------------         ------ ----
d-----         9/19/2024   4:06 PM                rbac_practicum


PS C:\Users\hp> cd rbac_practicum
PS C:\Users\hp\rbac_practicum> |
```

   b. Setelah itu buat file docker-compose.yml



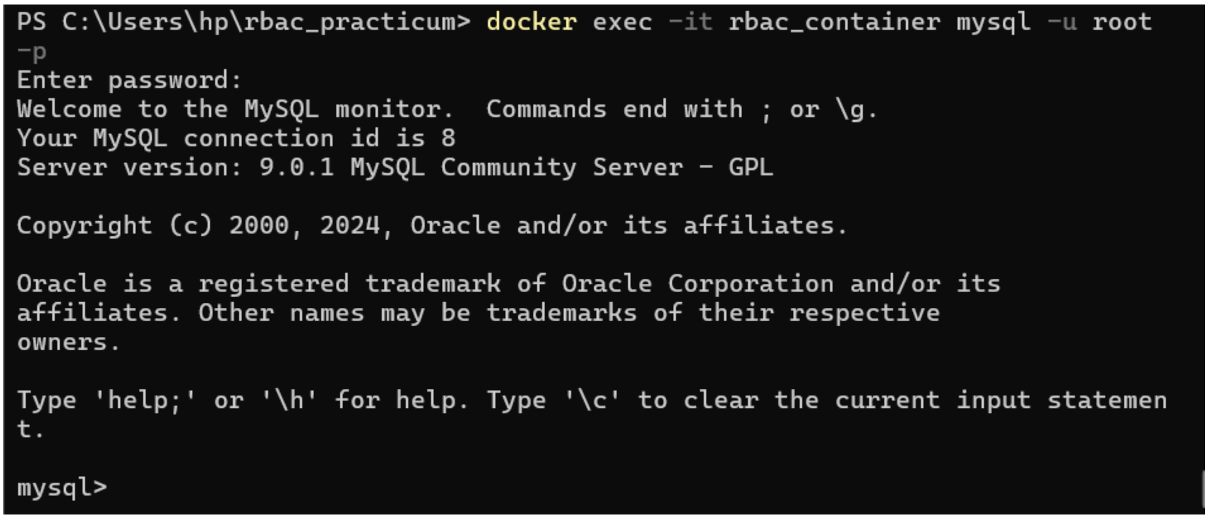
```yaml
version: '3.1'
services:
  mysql:
    image: mysql:latest
    container_name: rbac_container #nameofcontainer
    environment:
      MYSQL_ROOT_PASSWORD: rootpassword
      MYSQL_DATABASE: rbac_db
    ports:
      - "3308:3306"

```

c. Jikas sudah kita dapat build container yang sudah kita buat



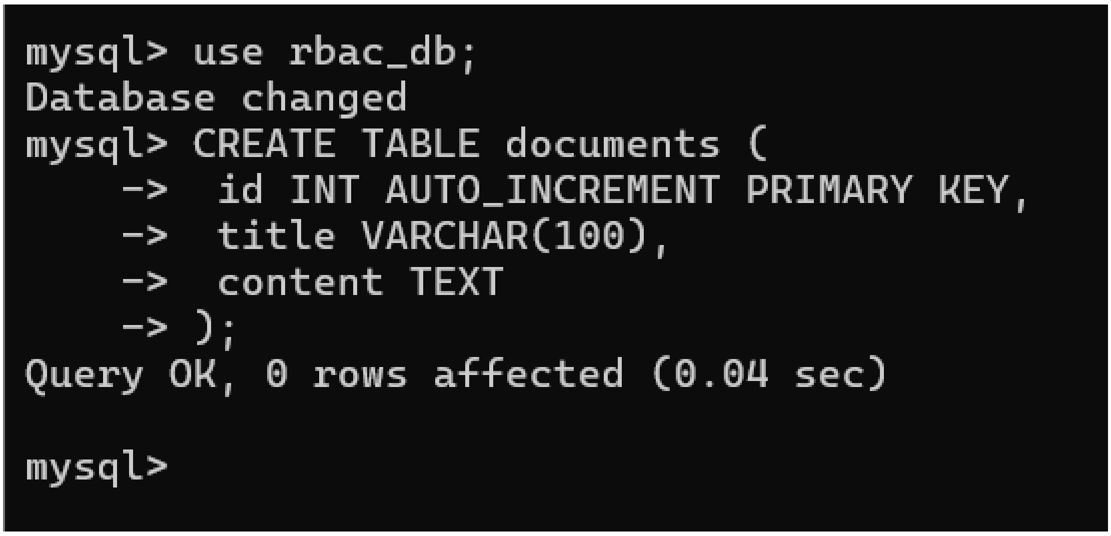
```
PS C:\Users\hp> cd rbac_practicum
PS C:\Users\hp\rbac_practicum> docker-compose up -d
time="2024-09-19T16:10:03+07:00" level=warning msg="C:\\Users\\hp\\rbac_prac
ticum\\docker-compose.yml: the attribute `version` is obsolete, it will be i
gnored, please remove it to avoid potential confusion"
[+] Running 2/2
 ✔Network rbac_practicum_default  Created                              0.1s
 ✔Container rbac_container        Start...                             1.2s
PS C:\Users\hp\rbac_practicum> docker ps
CONTAINER ID   IMAGE                    COMMAND                   CREATED
     STATUS          PORTS                               NAMES
c8a99e209b51   mysql:latest             "docker-entrypoint.s…"   4 seconds ag
o     Up 3 seconds    33060/tcp, 0.0.0.0:3308->3306/tcp   rbac_container
0acc230eedd4   abac_practicum-python    "tail -f /dev/null"       12 minutes a
go    Up 11 minutes                                       abac_python_contain
er
49833776a63a   mysql:latest             "docker-entrypoint.s…"   2 hours ago
     Up 2 hours      33060/tcp, 0.0.0.0:3307->3306/tcp   abac_mysql_containe
r
b82f87b6188d   mysql:latest             "docker-entrypoint.s…"   3 hours ago
     Up 3 hours      0.0.0.0:3306->3306/tcp, 33060/tcp   roni-mac_container
34c321a5e769   mariadb                  "docker-entrypoint.s…"   5 hours ago
     Up 5 hours      3306/tcp                            roni-mariadb-contai
n
PS C:\Users\hp\rbac_practicum> |
```

d. Setelah itu login ke container yang sudah dibuat



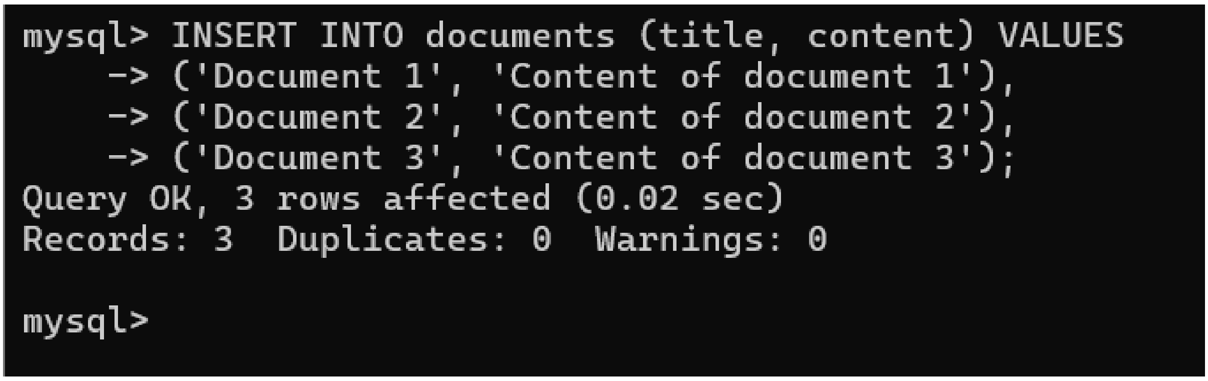
```
PS C:\Users\hp\rbac_practicum> docker exec -it rbac_container mysql -u root
-p
Enter password:
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 8
Server version: 9.0.1 MySQL Community Server - GPL

Copyright (c) 2000, 2024, Oracle and/or its affiliates.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statemen
t.

mysql>
```

e. Jika sudah masuk ke database dan buat sebuah table



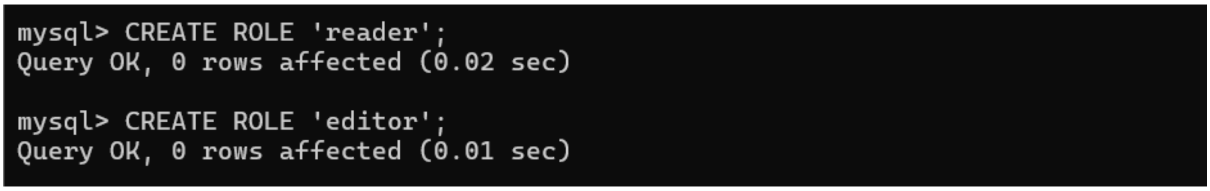
```
mysql> use rbac_db;
Database changed
mysql> CREATE TABLE documents (
    ->  id INT AUTO_INCREMENT PRIMARY KEY,
    ->  title VARCHAR(100),
    ->  content TEXT
    -> );
Query OK, 0 rows affected (0.04 sec)

mysql>
```

f. Lalu masukkan data kedalam table



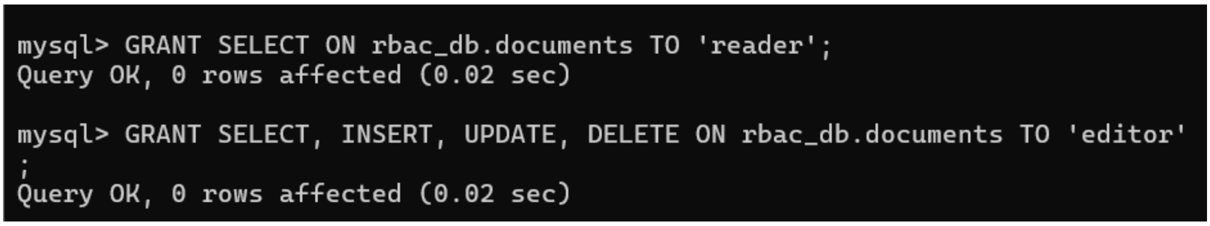
```
mysql> INSERT INTO documents (title, content) VALUES
    -> ('Document 1', 'Content of document 1'),
    -> ('Document 2', 'Content of document 2'),
    -> ('Document 3', 'Content of document 3');
Query OK, 3 rows affected (0.02 sec)
Records: 3  Duplicates: 0  Warnings: 0

mysql>
```

g. Buatlah dua peran reader dan editor



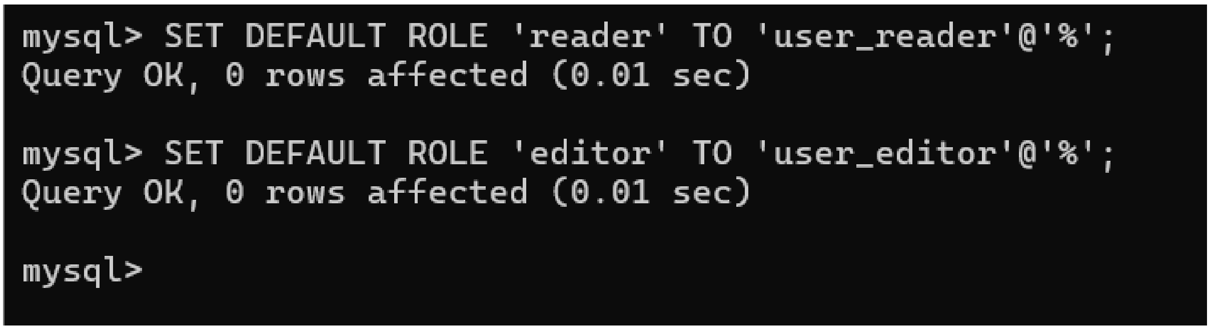
```
mysql> CREATE ROLE 'reader';
Query OK, 0 rows affected (0.02 sec)

mysql> CREATE ROLE 'editor';
Query OK, 0 rows affected (0.01 sec)
```

h. Lalu berikan hak akses periiinan ke setiap peran



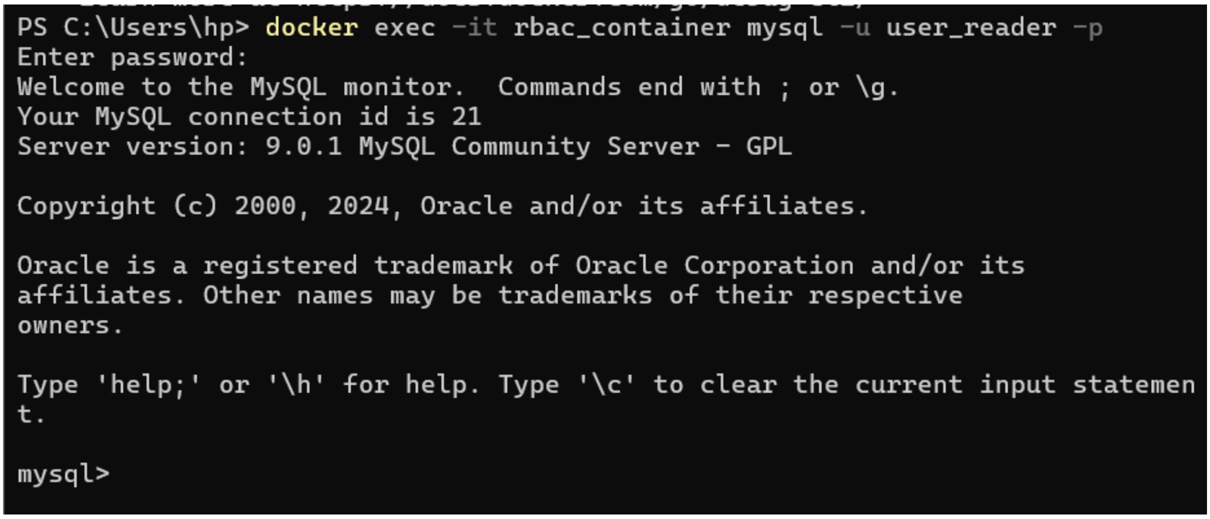
```
mysql> GRANT SELECT ON rbac_db.documents TO 'reader';
Query OK, 0 rows affected (0.02 sec)

mysql> GRANT SELECT, INSERT, UPDATE, DELETE ON rbac_db.documents TO 'editor'
;
Query OK, 0 rows affected (0.02 sec)
```

i. Jika sudah sekaran kita aktifkan peran pengguna secara default



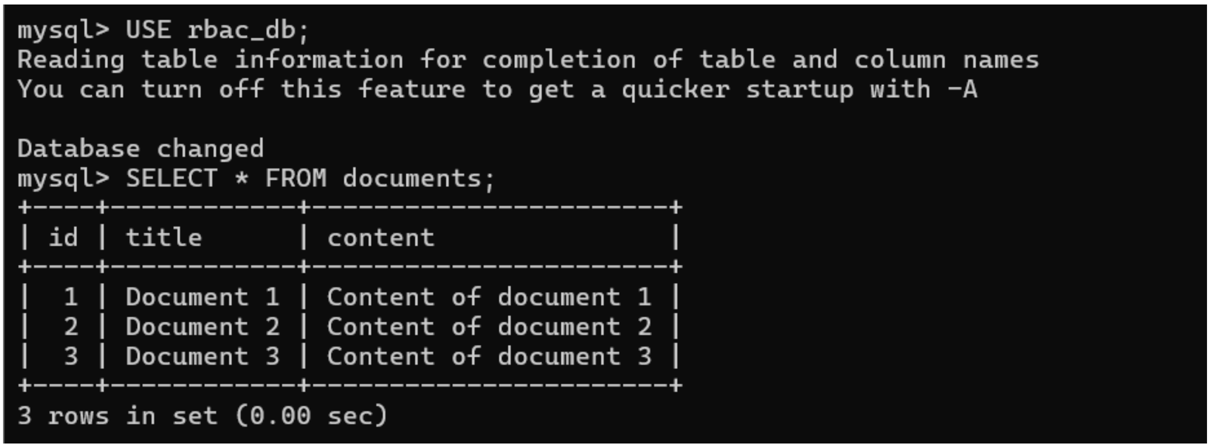
```
mysql> SET DEFAULT ROLE 'reader' TO 'user_reader'@'%';
Query OK, 0 rows affected (0.01 sec)

mysql> SET DEFAULT ROLE 'editor' TO 'user_editor'@'%';
Query OK, 0 rows affected (0.01 sec)

mysql>
```

j. Setelah itu keluar dan uji sebagai user reader



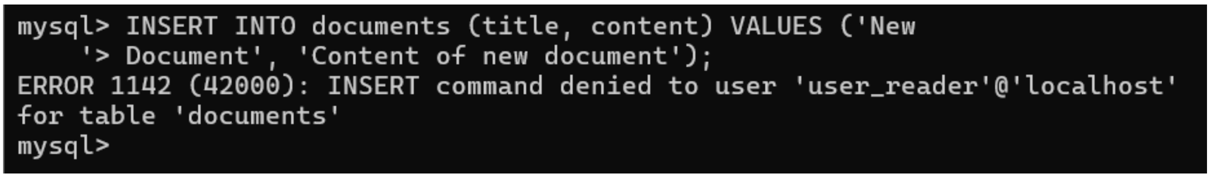
```
PS C:\Users\hp> docker exec -it rbac_container mysql -u user_reader -p
Enter password:
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 21
Server version: 9.0.1 MySQL Community Server - GPL

Copyright (c) 2000, 2024, Oracle and/or its affiliates.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statemen
t.

mysql>
```

k. Gunakan database dan coba lihat table document maka dia akan berhasil



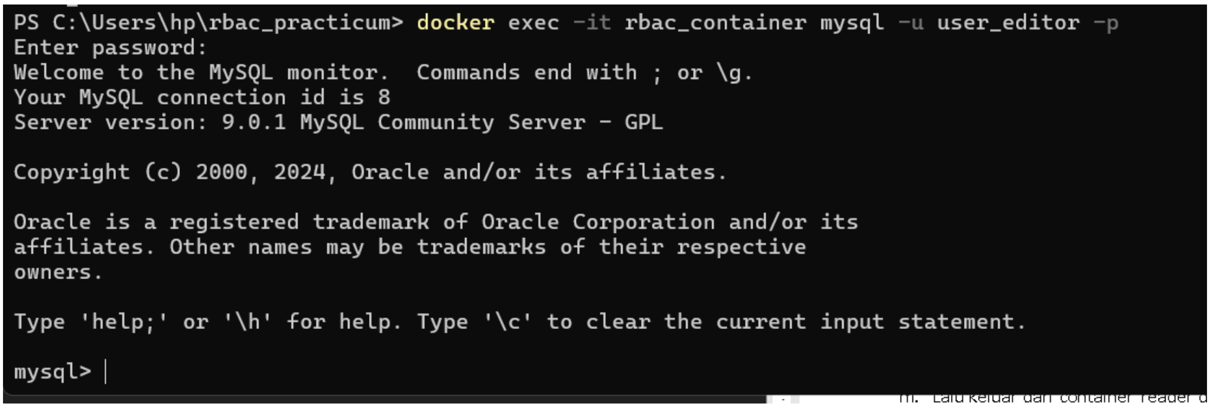
```
mysql> USE rbac_db;
Reading table information for completion of table and column names
You can turn off this feature to get a quicker startup with -A

Database changed
mysql> SELECT * FROM documents;
+----+------------+-----------------------+
| id | title      | content               |
+----+------------+-----------------------+
|  1 | Document 1 | Content of document 1 |
|  2 | Document 2 | Content of document 2 |
|  3 | Document 3 | Content of document 3 |
+----+------------+-----------------------+
3 rows in set (0.00 sec)
```

l. Setelah itu kita uji coba memasukkan data, jika hak akses yang diberikan benar seharusnya reader tidak datap memasukkan data pada table



```
mysql> INSERT INTO documents (title, content) VALUES ('New
    '> Document', 'Content of new document');
ERROR 1142 (42000): INSERT command denied to user 'user_reader'@'localhost'
for table 'documents'
mysql>
```

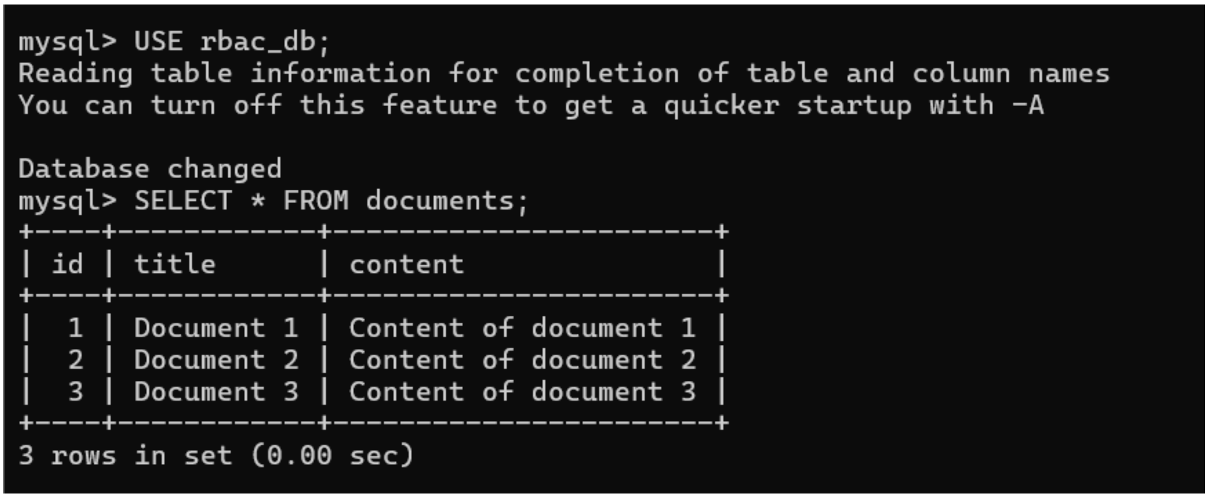
m. Lalu keluar dari container reader dan sekarang kita akan uji container editor



```
PS C:\Users\hp\rbac_practicum> docker exec -it rbac_container mysql -u user_editor -p
Enter password:
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 8
Server version: 9.0.1 MySQL Community Server - GPL

Copyright (c) 2000, 2024, Oracle and/or its affiliates.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql> |
```

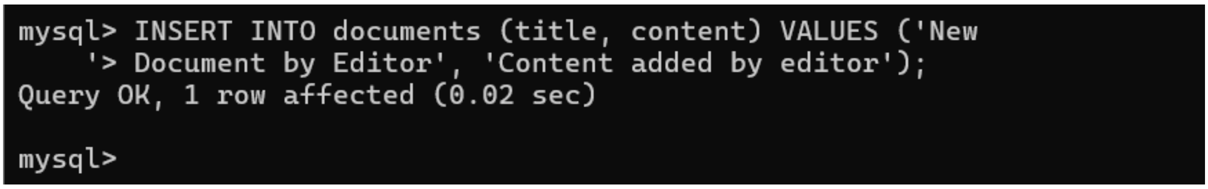
m. Lalu keluar dari container reader d

n. Lalu gunakan database yang sudah dibuat dan uji select seharusnya berhasil



```
mysql> USE rbac_db;
Reading table information for completion of table and column names
You can turn off this feature to get a quicker startup with -A

Database changed
mysql> SELECT * FROM documents;
+----+------------+----------------------+
| id | title      | content              |
+----+------------+----------------------+
|  1 | Document 1 | Content of document 1 |
|  2 | Document 2 | Content of document 2 |
|  3 | Document 3 | Content of document 3 |
+----+------------+----------------------+
3 rows in set (0.00 sec)
```

o. Seterusnya kita akan coba memasukkan data kedalamnya dan berhasil karena hak aksesnya menginjinkan untuk user editur mengatur database



```
mysql> INSERT INTO documents (title, content) VALUES ('New
    '> Document by Editor', 'Content added by editor');
Query OK, 1 row affected (0.02 sec)

mysql>
```

Sekian dari Saya Terima Kasih