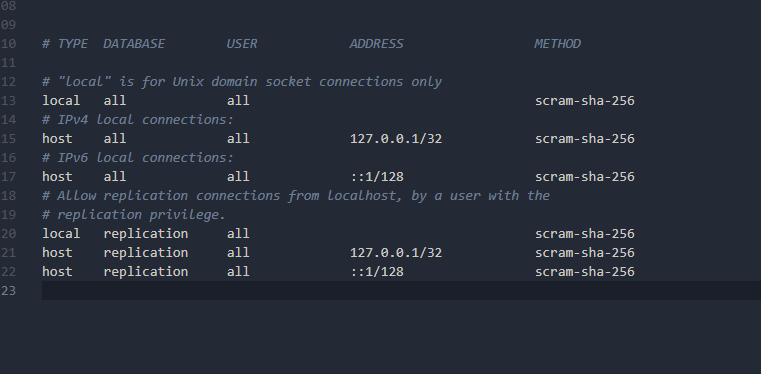
*Task 1. Figure out what security precautions are already used in your 'dvd\_rental' database. Prepare description.*

*P.S. the task was done after completing 2-3 tasks, so some roles and users are seen from previous tasks. Some codes where wrote with the help of Postgresql documentation and Ai in order to check precautions.*

Dvdrental database has many different security precautions.

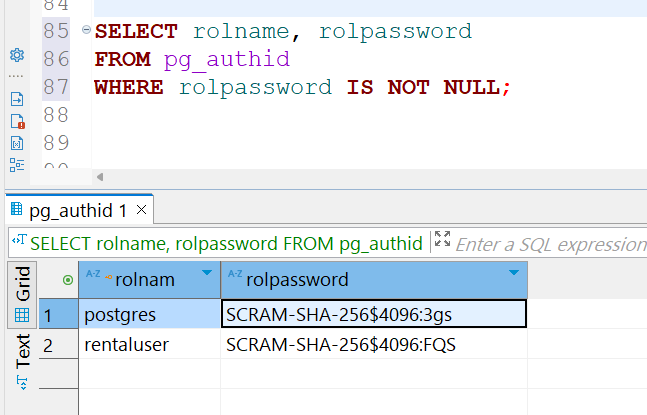
**Authentication:**

Firs of all let’s check what kind of authentication method is used in database. I found hba file in between my program file.



Method is SCRAM-SHA-256. The IP ranges (127.0.0.1, ::1) are **localhost only**—no external access.

Also, we can check in postgresql role passwords

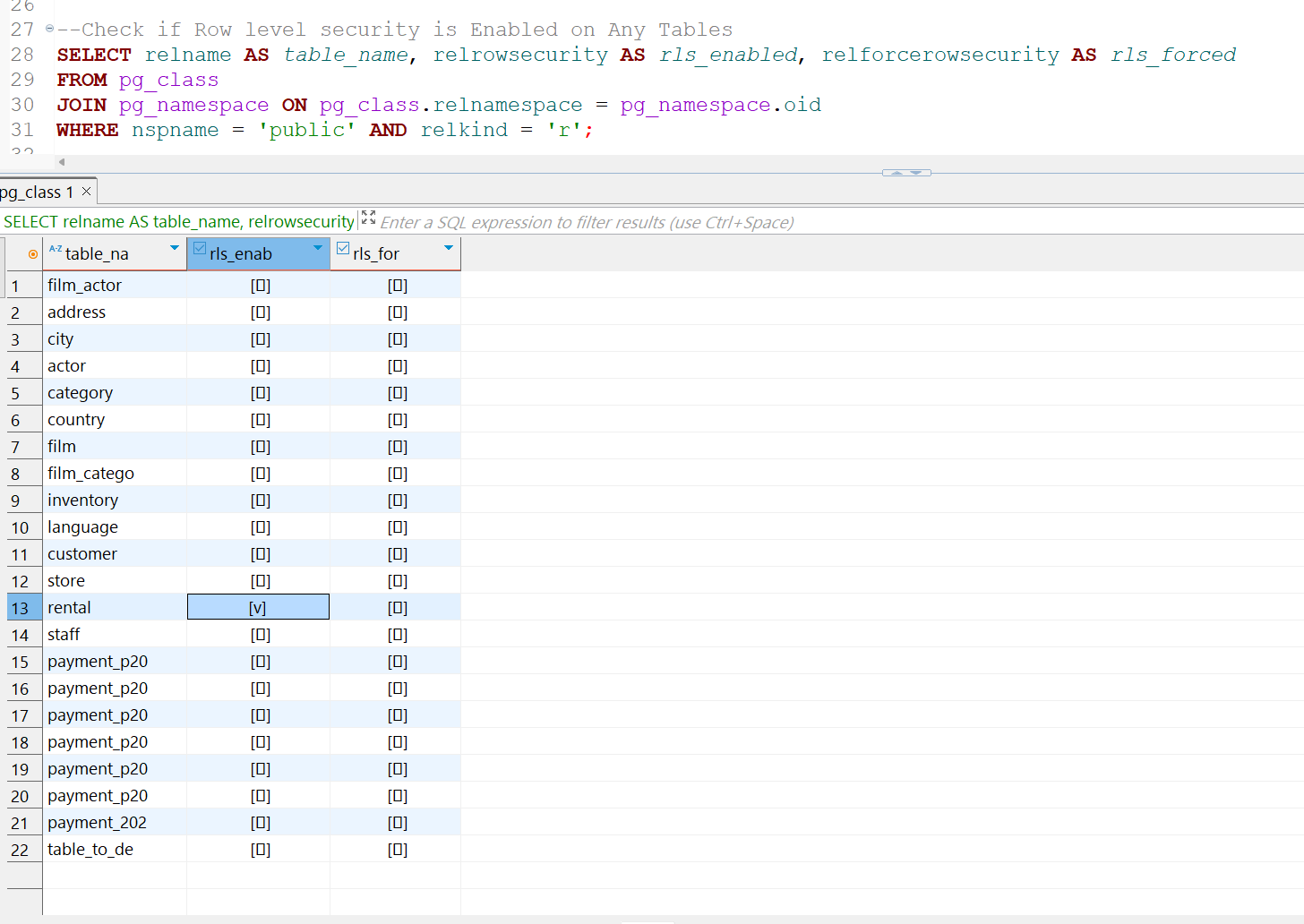
****

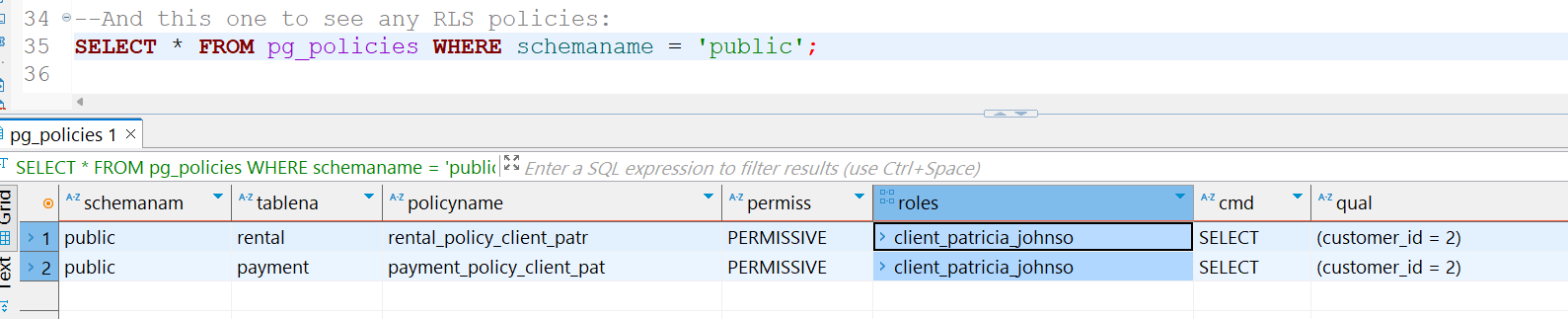
We can see that it is secured with passwords that start with **SCRAM-SHA-256** (the actual password is longer), which indicates that the most secure option is being used for password security. All connections use **SCRAM-SHA-256**, PostgreSQL's most secure authentication method.

**Authorization**

We can also check authorization in the **dvdrental** database by examining the roles and privileges, and by checking whether any row-level security (RLS) policies are applied.

By default, the **dvdrental** database has no RLS enabled and no RLS policies configured. However, for learning purposes, and after completing tasks 2 and 3, we can check whether any RLS policies have been applied.

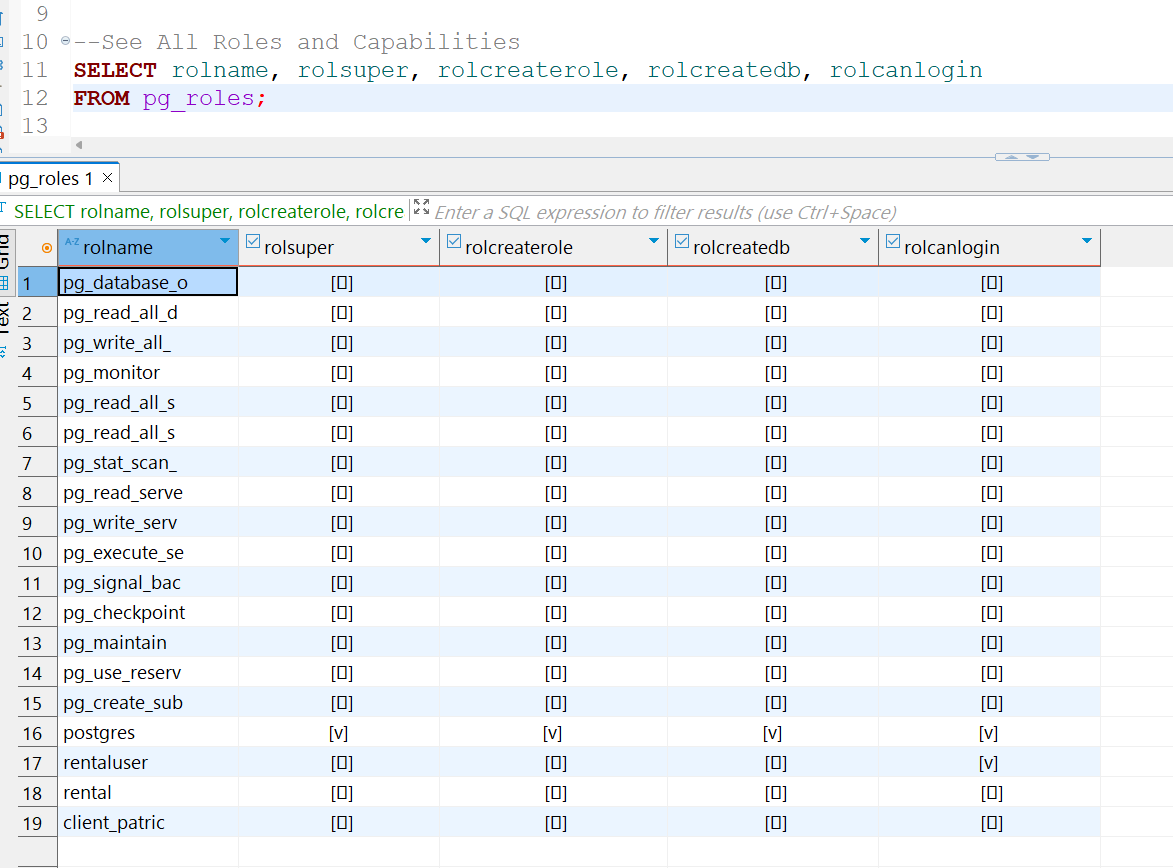




As expected, since RLS is applied for table rental, we can see that box “rls\_enabled” is checked.

Also, we find a policy name and which tables it is applied. It isnot default in dvdrental database, but it is how rsl looks like if applied.

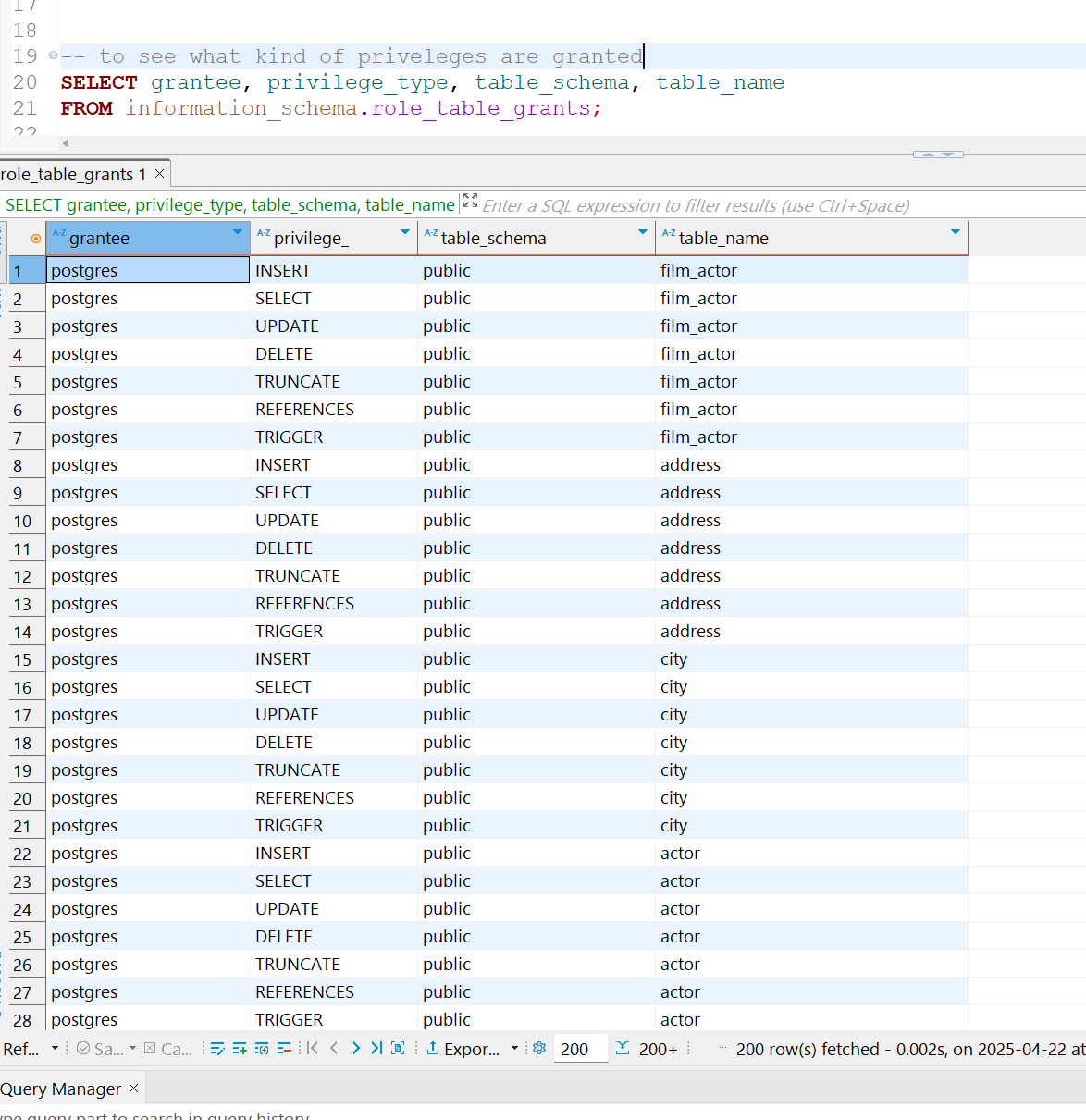
Let‘s check **roles**.



So, it means that postgres is a superuser. It's the **default admin account** created when PostgreSQL is installed. This superuser can create roles, create database, create logins, has full access to all databases, schemas, and tables. All other roles (like pg\_read\_all\_data, pg\_monitor, etc.) are **non-login** system roles and have **no superuser rights**.

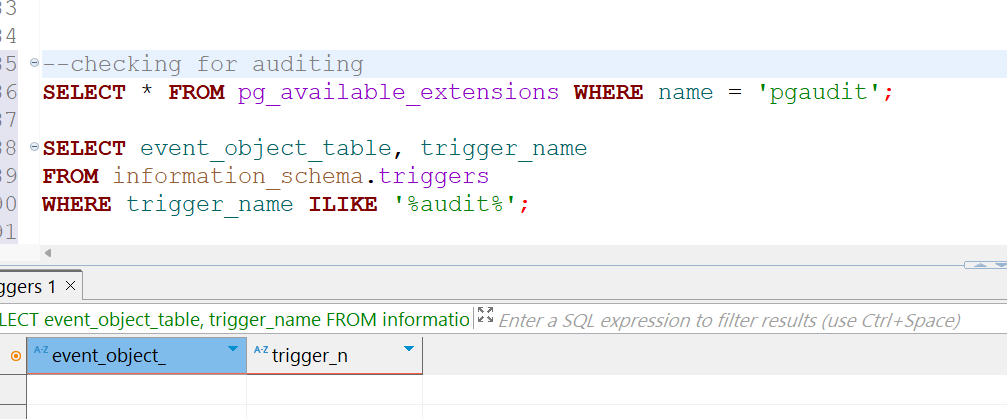
**Privilegies**

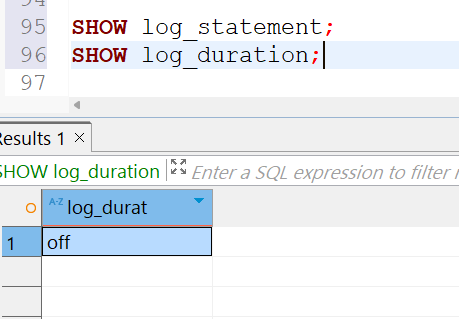
The list of privilegies is like INSERT, SELECT, UPDATE, DELETE and so on. We can check what kind of privilegies every user has. Postgres has all privilegies for all database tables as a superuser.



**Auditing**

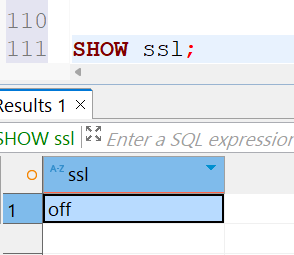
**Auditing** in PostgreSQL refers to the process of **tracking and recording database activity**, such as who has a access. PostgreSQL doesn't have a full audit trail by default, but it offers several **logging settings** and optional extensions like **pgaudit** for detailed tracking. I tried to check for pgAudit Extension but it seems it is turned off. This means PostgreSQL is not currently logging how long each query takes.





**Encryption**

PostgreSQL offers several encryption mechanisms to protect your data at different levels like requiring SSL connections or password encryption. Lets check SSL connection.



This means that SSL is currently disabled on your PostgreSQL server, so connections (including from DBeaver) are not encrypted.

**Back\_up**

Regularly backing up a database and storing backups securely are critical to ensuring that data can be restored in the event of a security breach or other disaster. DBeaver, as a program, automatically saves files, so when you close the program, there is no need to manually save the code. However, it is still recommended to save files manually.