Task 1: Analyze Security Precautions in the dvd\_rental Database

Gabriela Creţu

April 2025

The goal of this task is to investigate the existing security precautions in the dvd\_rental database. Several SQL queries were executed to analyze the privileges, row-level security, triggers, SSL encryption, and audit tables.

* Authentication

Authentication is crucial in ensuring that only authorized users can access the database. PostgreSQL sup-ports various authentication methods, including password authentication, certificate-based authentication, and external authentication through LDAP or Kerberos. By default, PostgreSQL requires users to authen-ticate via password.

show hba\_file ;

After executing the query above, I located the path to my authentication configuration: $C:/Program Files/PostgreSQL/17/data/pg\_hba.conf$. Upon reviewing the file, I discovered that the authentication method configured for my PostgreSQL instance is scram-sha-256, a secure, password-based authentication method.

* Authorization

Authorization deals with granting or restricting access to database objects, such as tables and functions. In this case, we analyze the grantee information and the privileges granted to each role.

This query retrieves information about the grantees (roles/users) and their corresponding privileges on tables in the public schema. It helps you understand who has access to what tables and what kind of privileges they have.

SELECT grantee , table\_name , p r i v i l e g e \_ t y p e

FROM i n f o r m a t i o n \_ s c h e m a . r o l e \_ t a b l e \_ g r a n t s

WHERE t a b l e \_ s c h e m a = ’public ’;

After running the SQL statement to find the privileges each grantee has, it was noticed that the only grantee is postgres. Although postgres has privileges on almost all tables and almost all privileges overall, it still does not have the DROP and ALTER permissions, among others.

The following query summarizes the types of privileges:

SELECT privilege\_type , count (\*) as counting

FROM i n f o r m a t i o n \_ s c h e m a . r o l e \_ t a b l e \_ g r a n t s

WHERE t a b l e \_ s c h e m a = ’public ’

GROUP BY p r i v i l e g e \_ t y p e ;

1

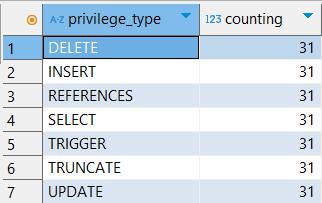


Figure 1: Privileges

The output revealed that postgres holds the following privileges on all tables:

* DELETE
* INSERT
* REFERENCES
* SELECT
* TRIGGER
* TRUNCATE
* UPDATE

However, postgres lacks the following privileges:

* CREATE
* TEMPORARY
* EXECUTE
* USAGE
* CONNECT
* Encryption

Encryption ensures that sensitive data is protected both during transmission (network encryption) and at rest (data encryption). Additionally, passwords stored in the database are encrypted.

This query checks the current SSL encryption setting for the PostgreSQL instance. SSL ensures secure connections between the database and clients.

SHOW ssl ; -- Displays the SSL setting of the P o s t g r e S Q L server

After running the query, it was found that there is no SSL encryption enabled at the database level. This may indicate a potential security risk, as connections to the database are not encrypted.

* Auditing

Auditing and triggers help monitor and maintain database integrity. Triggers can automatically track changes to the data or enforce certain constraints, which is crucial for database auditing and integrity enforcement.

2

4.1 Triggers

To identify the triggers defined on tables in the database, the following query retrieves information about the triggers, including their names, associated tables, types, functions, and arguments.

SELECT

tgname AS trigger\_name , -- Name of the trigger

relname AS table\_name , -- Name of the table a s s o c i a t e d with the

trigger

nspname AS schema\_name , -- Schema of the table

tgtype AS trigger\_type , -- Type of trigger (e.g. , BEFORE , AFTER ,

etc .)

tgfoid :: r e g p r o c e d u r e AS function\_name , -- Function that the trigger calls

tgargs AS ar gu me nts -- A rg um en ts passed to the trigger

function

FROM

p g \_ t r i g g e r t

JOIN

pg\_class c ON c . oid = t . tgrelid

JOIN

p g \_ n a m e s p a c e n ON n . oid = c . r e l n a m e s p a c e

WHERE

NOT t . t g i s i n t e r n a l -- Excludes internal triggers

AND n . nspname NOT IN ( ’p g \_ c a t a l o g ’, ’ i n f o r m a t i o n \_ s c h e m a ’) -- Excludes system

schemas

ORDER BY

schema\_name , table\_name , t r i g g e r \_ n a m e ; -- Orders by schema , table , and trigger

name

After running the query, I found the following triggers and the tables they are imposed on:

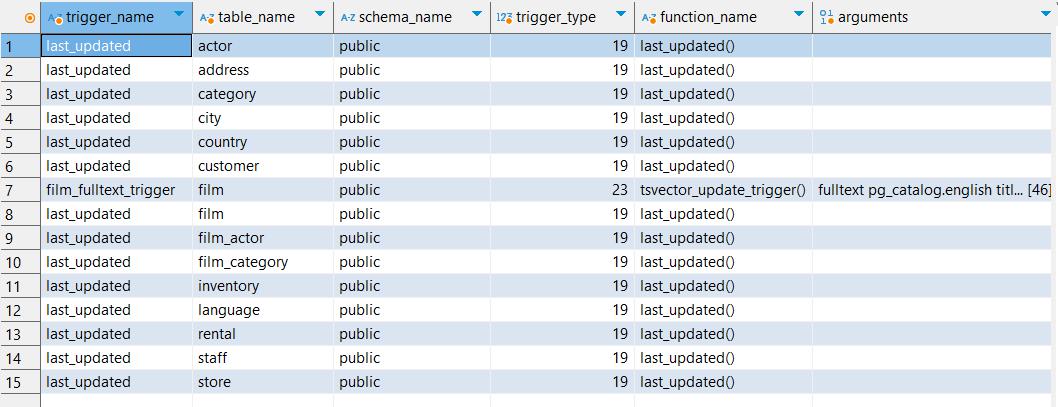


Figure 2: Triggers

4.2 Row-level security

This query checks which tables have Row-Level Security (RLS) enabled by looking at the relrowsecurity attribute in the pg\_class catalog. It will only return regular tables (not views or indexes) and exclude system schemas.

SELECT

c . relname AS table\_name , -- Name of the table

n . nspname AS schema\_name , -- Schema in which the table resides

c . r e l r o w s e c u r i t y AS r l s \_ e n a b l e d -- Boolean i n d i c a t i n g if RLS is enabled

FROM

3

pg\_class c

JOIN

p g \_ n a m e s p a c e n ON n . oid = c . r e l n a m e s p a c e

WHERE

c . r e l r o w s e c u r i t y = true -- Filters to only tables with RLS enabled

AND c . relkind = ’r’ -- Ensures we ’re checking regular tables , not

views or indexes

AND n . nspname NOT IN ( ’p g \_ c a t a l o g ’, ’ i n f o r m a t i o n \_ s c h e m a ’) -- Excludes system

schemas

ORDER BY

schema\_name , t a b l e \_ n a m e ; -- Orders by schema and table name

After running the above query, it was found that no Row-Level Security is enabled in this database.

Therefore, there is no need to search for Row-Level Security policies.

4.3 Auditing

To identify tables related to auditing (commonly named with "audit"), this query searches for all ta-bles in the database whose names contain "audit" and excludes system schemas like pg\_catalog and

information\_schema.

SELECT

s c h e m a n a m e AS schema\_name , -- Schema name

ta bl en am e AS t a b l e \_ n a m e -- Table name

FROM

pg \_t ab le s

WHERE

ta bl en am e ILIKE ’% audit %’ -- Looks for tables with " audit " in their name (

case - i n s e n s i t i v e )

AND s c h e m a n a m e NOT IN ( ’p g \_ c a t a l o g ’, ’ i n f o r m a t i o n \_ s c h e m a ’) ; -- Excludes system

schemas

After running the query, no tables related to auditing were found.

5. Backup and Recovery

Backup and recovery procedures are essential for ensuring data can be restored after a disaster or security breach. However, the database does not explicitly mention any backup and recovery configuration.

SHOW a r c h i v e \_ m o d e ;

SHOW a r c h i v e \_ c o m m a n d ;

After running the above code, I found that automate recovery or recovery is off

4