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

To begin with authentication, I verified the presence of distinct user accounts for accessing the database and found that there is only one user (postgres), who is also the superuser. This user is authenticated using a password. To ensure strong password policies are enforced, such as complexity requirements and regular updates, I executed the SHOW password\_encryption command. The result confirmed that the database uses SCRAM-SHA-256, a secure password encryption standard.

For authorization, I examined the roles and permissions defined in the database to determine if user access is appropriately restricted based on job responsibilities. The database contains roles with varying levels of privileges, which can be reviewed through queries or by checking the Roles section. One observation was that the pg\_monitor role is a child role that inherits permissions from multiple parent roles, including pg\_read\_all\_settings, pg\_read\_all\_stats, and pg\_stat\_scan\_tables. This hierarchy was analyzed using queries and verified in the Members section of the role configuration. I also audited permissions on critical tables, such as customer, payment, and staff, to ensure that privileges are appropriately restricted.

Regarding data encryption, I checked whether SSL is enabled for encrypting data transmitted between the database and clients. It was found that SSL is not enabled, meaning data in transit is not encrypted. For data at rest, I reviewed the encryption strength and concluded that the database appears to use medium-strength ciphers (typically with key lengths of 64 to 128 bits). While these are weaker than high-strength ciphers, they may be sufficient depending on the use case. Additionally, I executed a query to identify encrypted columns in tables, but no such columns were found.

Auditing and monitoring capabilities were also assessed. Logs and alerts are critical for tracking login attempts, schema changes, and other key events. I used the SHOW shared\_preload\_libraries command and discovered that no login auditing is enabled. This represents a potential area for improvement.

In the backup and recovery domain, I ran the SELECT pg\_last\_wal\_replay\_lsn() command to confirm that the backup mechanism is functioning. However, upon examining the archive settings using SHOW archive\_mode and SHOW archive\_command, it was determined that archive mode is disabled, and no archive commands are configured. This means the database does not support archival backups.

Network security was also analyzed by reviewing the pg\_hba\_file\_rules configuration. This helps verify whether the database enforces firewall restrictions to specific IP addresses and ensures private networking isolation. Finally, I inspected the PostgreSQL version and updates using the SELECT version() query. The database is running PostgreSQL 15.8, compiled with Visual C++ build 1940, a fully updated and modern version.