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

To determine the existing security precautions in the dvd\_rental database, we need to inspect the schema, roles, privileges, and configuration settings.

**Analysis and Description:**

1. **Role-Based Access Control** :
   * The database likely uses roles to manage permissions. By default, PostgreSQL creates a postgres superuser role during installation, and additional roles (such as: dvd\_rental\_user) might exist for application access.
   * The ERD shows tables with sensitive data (such as: customer.email, staff.password, payment.amount), suggesting role-based privileges are intended to restrict access.
2. **Default Privileges**:
   * The public role exists in all PostgreSQL databases and, by default, has no privileges except those explicitly granted. The dvd\_rental database likely follows this, meaning new users have no access unless granted.
   * Check current privileges with:

SELECT table\_name, privilege\_type, grantee

FROM information\_schema.role\_table\_grants

WHERE table\_schema = 'public';

(This would reveal if any broad permissions are granted to public.)

1. **Schema Isolation**:
   * The schema contains all tables, but no evidence of additional schemas suggests limited isolation. Security could be enhanced by using separate schemas for different user groups.
2. **Data Integrity and Constraints**:
   * The use of NOT NULL constraints (such as: customer.customer\_id, rental.rental\_date) ensures data integrity, indirectly supporting security by preventing invalid entries.
   * Foreign keys (such as: rental.customer\_id to customer.customer\_id) enforce referential integrity, reducing unauthorized data manipulation risks.
3. **No Row-Level Security (RLS) by Default**:
   * The ERD doesn’t indicate RLS is enabled on any table. By default, RLS is disabled (relrowsecurity = f in pg\_class), meaning all rows are accessible to users with table-level privileges.
4. **Password Protection**:
   * The staff.password column suggests password-based authentication for staff, but this is application-level (it depends on *how it's hashed/encrypted and authenticated* at the application level), not database-enforced unless encrypted and checked via authentication methods (such as pg\_hba.conf).

The **dvd\_rental** database uses standard PostgreSQL security practices, including role-based access, default privilege controls, and enforced data integrity via constraints. No row-level security or schema isolation is evident, and sensitive fields (such as: staff.password) rely on application-level protection. Further inspection of database roles, access logs, and encryption settings is recommended for a full security audit.