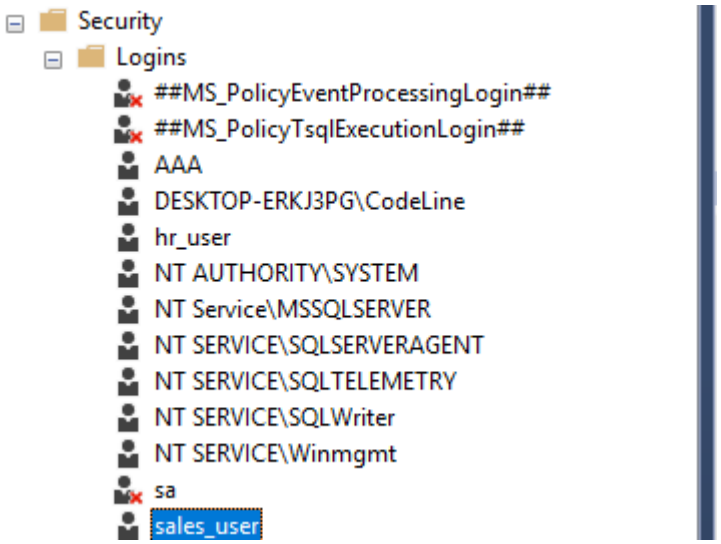
Task: Enforcing Schema-Level Access in a Company Database

Task Output Checklist

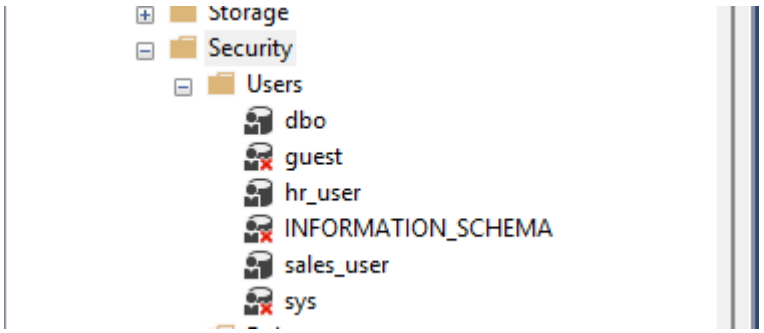
Ask the trainees to:

1. Take screenshots of:

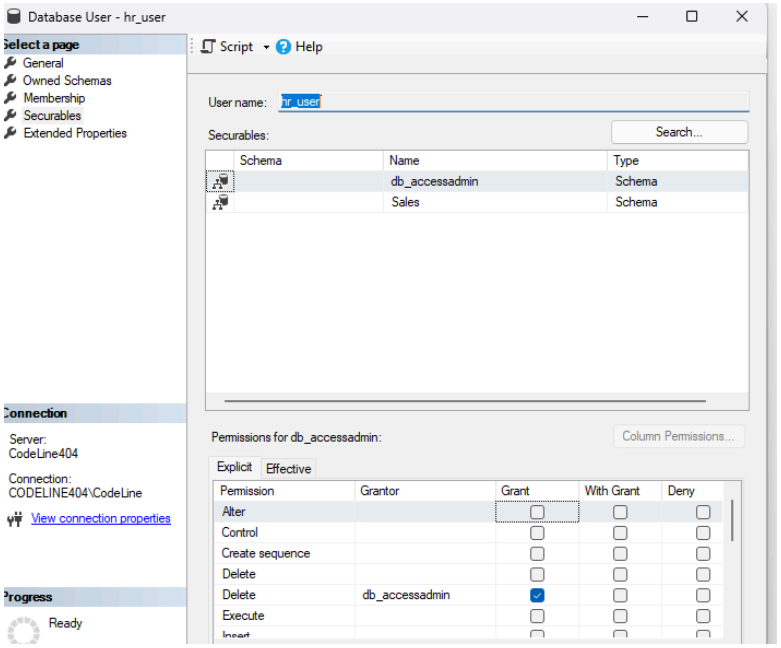
o Login creation

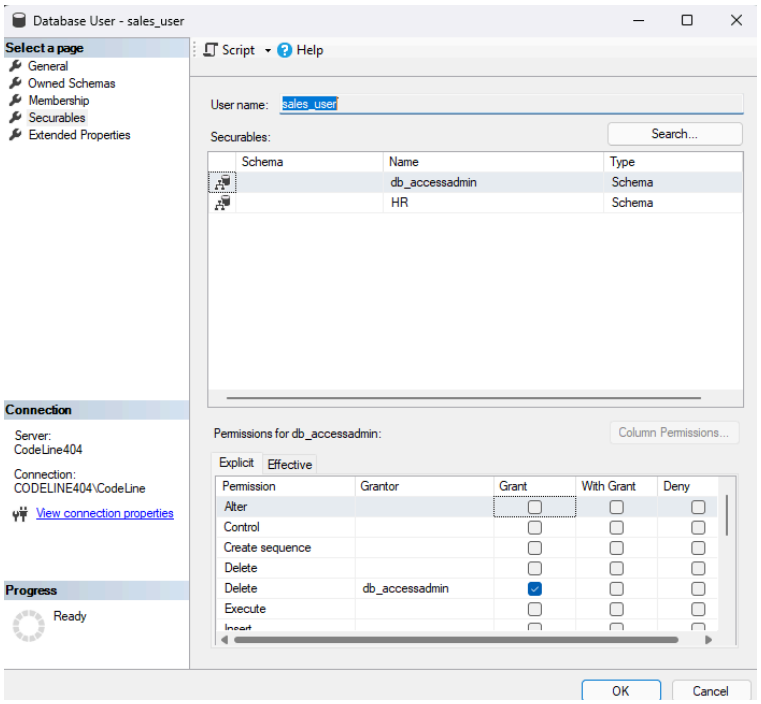


o User creation



o Schema permissions





o Query results showing access works only for their assigned schema

2. Try to:

o Connect as hr_login and access HR.Employees (should work)

o Try to access Sales.Customers (should be denied)

3. Write a short explanation:

o Why schema-level security is better than table-by-table permissions

o How this setup supports data segregation in real-world companies

1. What Are SQL Security Levels?

SQL Server uses multiple layers of security to control access:

- **Server-Level Login**
  - This is the first level of authentication.
  - A login allows access to the SQL Server instance, but **not to any specific database**.
  - Example: `hr_login` or `sales_login`.
- **Database-Level User**
  - Once logged in, a **database user** is needed to access objects inside a specific database.
  - It's linked to the login and defines **permissions inside the database**.
  - Example: `hr_user` in the `CompanyDB` database.
- **Schema-Level Permissions**
  - A schema is a logical container for tables, views, etc.
  - You can assign permissions (like SELECT, INSERT) **to the entire schema** rather than each object.
  - Example: `hr_user` gets access to everything inside the HR schema.
- **Object-Level Permissions *(Mention Briefly)***
  - These are permissions given **directly to specific tables, views, or procedures**.
  - More granular, but harder to manage on a large scale.

2. Benefits of Applying Security Levels

Implementing SQL security levels provides several benefits:

- **Restrict sensitive data:** Only HR can see salary or personal info.
- **Prevent unauthorized changes:** Only authorized roles can edit or delete records.
- **Reduce human error:** Limits what users can mistakenly do.
- **Meet compliance/audit requirements:** Helps enforce GDPR, HIPAA, etc.

3. Real-World Risks Without Security

Without proper security in place:

- **Everyone has full access:** Any employee could see or modify critical business data.
- **Developers modify production data:** Mistakes or experiments could lead to data loss or corruption.
- **Interns access HR data:** Breaches confidentiality and violates company policy or law.

4. Task Summary

- Created two **logins**: `hr_user` and `sales_user`.
- Mapped each to a **database user**: `hr_user` and `sales_user`.
- Created **schemas**: `HR` and `Sales`.
- Added **tables** under each schema.
- Assigned **schema-level permissions**:
    - HR users can only access `HR` data.
    - Sales users can only access `Sales` data.
    -
- This setup ensures each department **sees only its own data**.
- It's scalable, secure, and aligns with **real-world company needs** like protecting HR records, finance reports, or customer data.