companyDB Schema-Level Access & Security Report

Prepared by: Mohammed Yusuf Alkhusaibi

Date: 2025-05-29

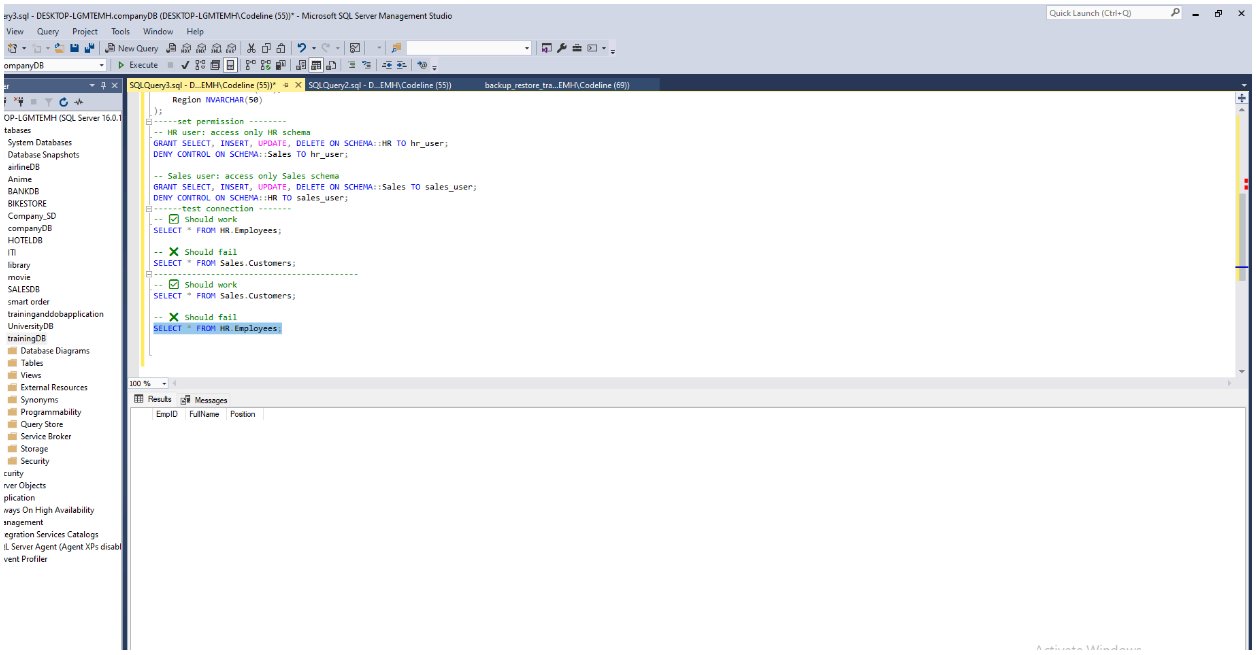
1. Task Implementation Summary

SQL logins and users were created for HR and Sales departments. Each department was assigned its own schema.
Tables were created inside each schema, and permissions were restricted so that each user could only access their own schema.

2. Schema Permissions Summary

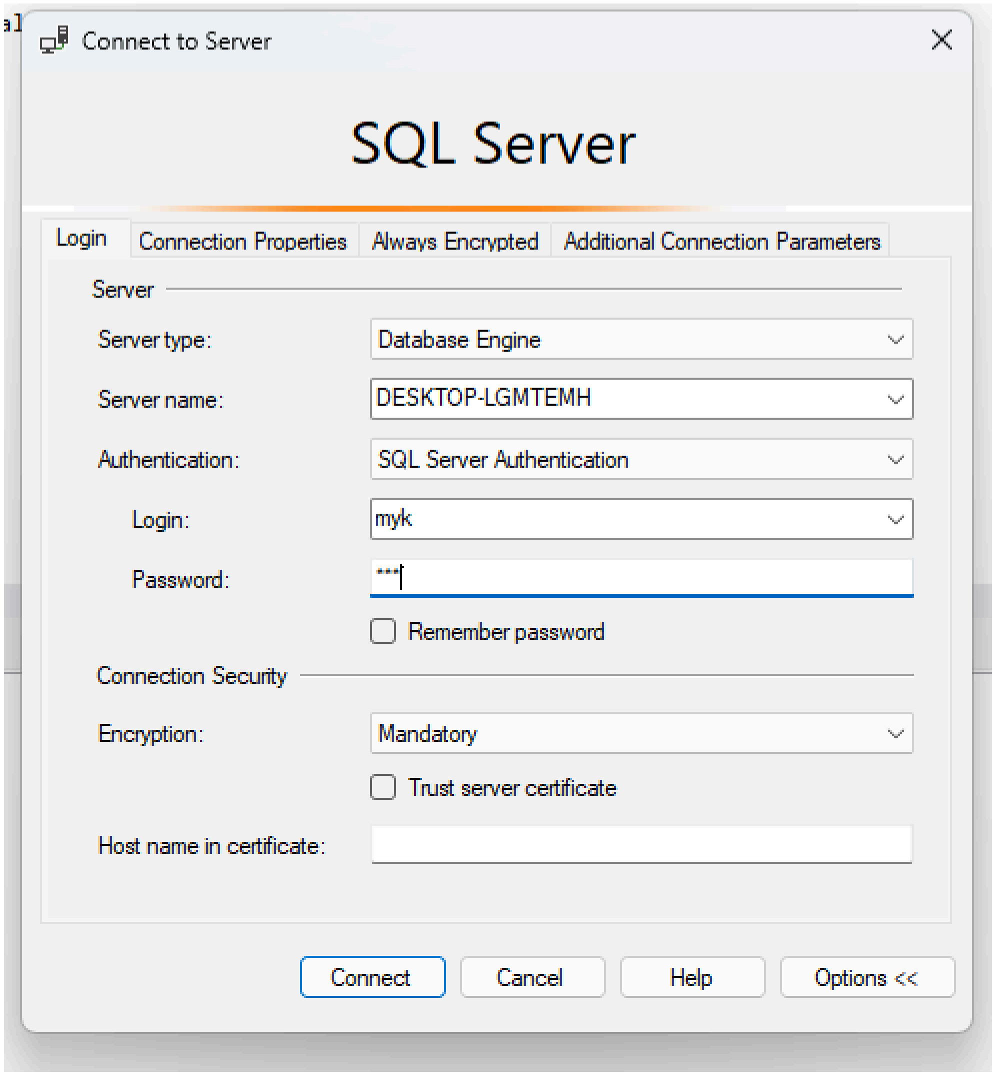
- HR user has full access to HR schema and no access to Sales.
- Sales user has full access to Sales schema and no access to HR.
- Verified by executing SELECT queries as different users.

SCREENSHOTS:



```sql
    Region NVARCHAR(50)
);
-----set permission --------
-- HR user: access only HR schema
GRANT SELECT, INSERT, UPDATE, DELETE ON SCHEMA::HR TO hr_user;
DENY CONTROL ON SCHEMA::Sales TO hr_user;

-- Sales user: access only Sales schema
GRANT SELECT, INSERT, UPDATE, DELETE ON SCHEMA::Sales TO sales_user;
DENY CONTROL ON SCHEMA::HR TO sales_user;
------test connection -------
-- ✅ Should work
SELECT * FROM HR.Employees;

-- ❌ Should fail
SELECT * FROM Sales.Customers;
----------------------------------------
-- ✅ Should work
SELECT * FROM Sales.Customers;

-- ❌ Should fail
SELECT * FROM HR.Employees;
```

	EmpID	FullName	Position



Server

Server type: Database Engine

Server name: DESKTOP-LGMTEMH

Authentication: SQL Server Authentication

Login: myk

Password: ***

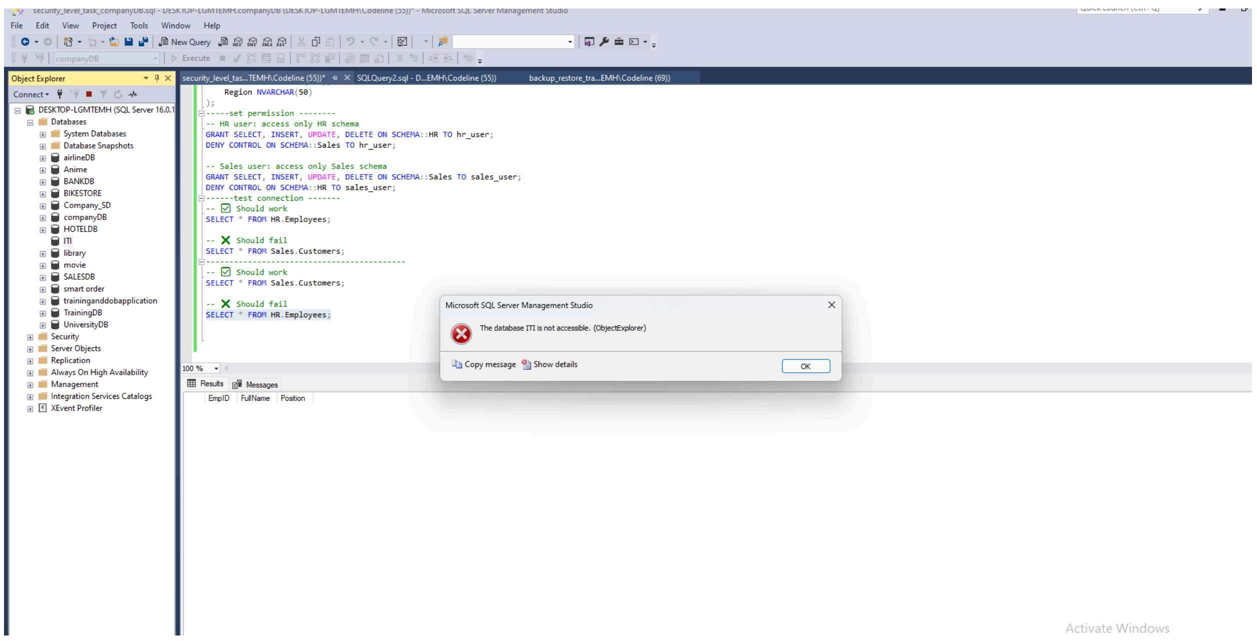
☐ Remember password

Connection Security

Encryption: Mandatory

☐ Trust server certificate

Host name in certificate:

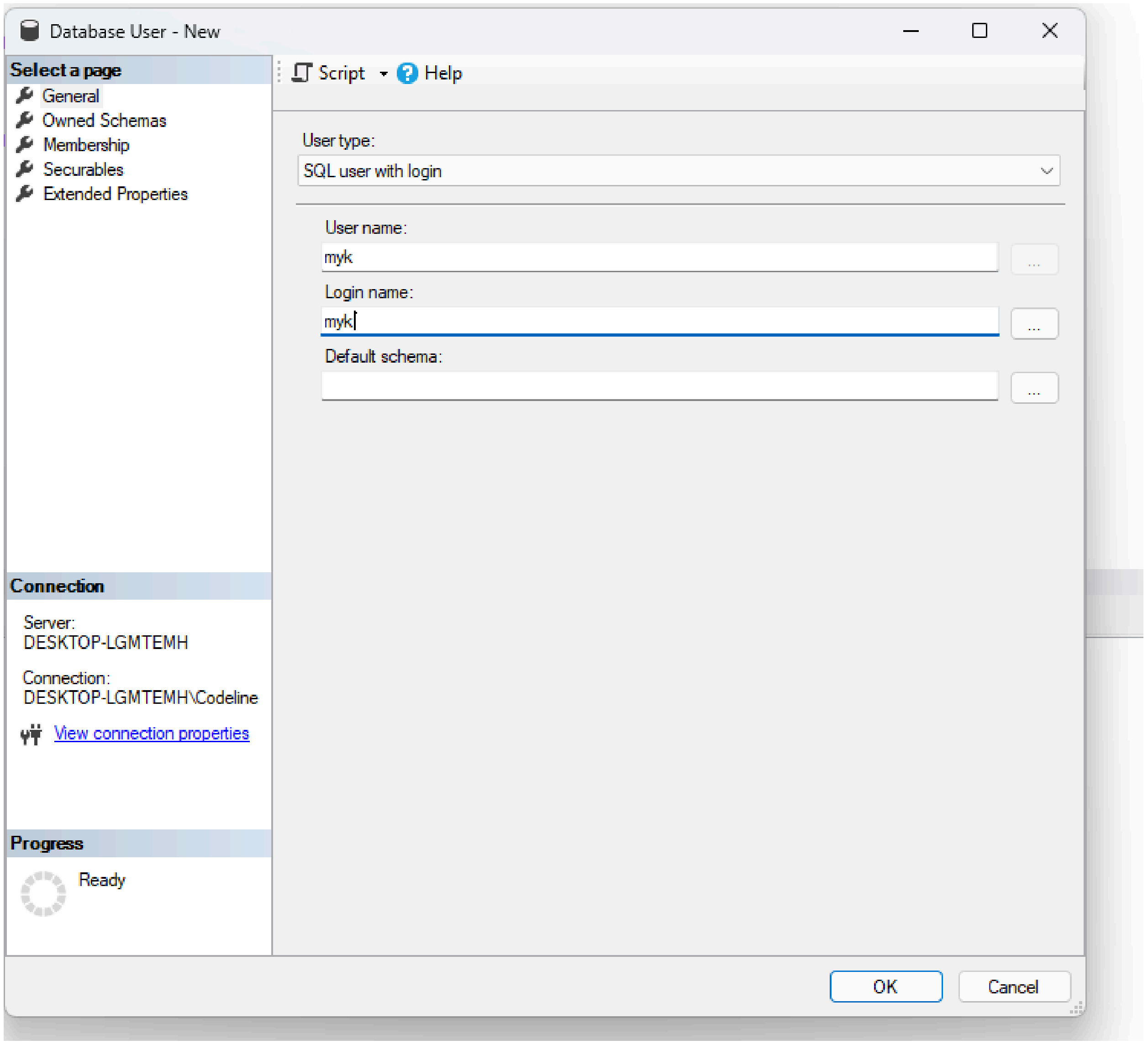

security_level_task_companyDB.sql - DESKTOP-LGMTEMH.companyDB (DESKTOP-LGMTEMH\Codeline (55))* - Microsoft SQL Server Management Studio
File
Edit
View
Project
Tools
Window
Help
New Query
companyDB
Execute
Object Explorer
Connect
DESKTOP-LGMTEMH (SQL Server 16.0.1
Databases
System Databases
Database Snapshots
airlineDB
Anime
BANKDB
BIKESTORE
Company_SD
companyDB
HOTELDB
ITI
library
movie
SALESDB
smart order
traininganddobapplication
TrainingDB
UniversityDB
Security
Server Objects
Replication
Always On High Availability
Management
Integration Services Catalogs
XEvent Profiler
security_level_tas...TEMH\Codeline (55))*
SQLQuery2.sql - D...EMH\Codeline (55))
backup_restore_tra...EMH\Codeline (69))
100 %
Results
Messages
EmpID
FullName
Position
Activate Windows

```
    Region NVARCHAR(50)
);
-----set permission --------
-- HR user: access only HR schema
GRANT SELECT, INSERT, UPDATE, DELETE ON SCHEMA::HR TO hr_user;
DENY CONTROL ON SCHEMA::Sales TO hr_user;

-- Sales user: access only Sales schema
GRANT SELECT, INSERT, UPDATE, DELETE ON SCHEMA::Sales TO sales_user;
DENY CONTROL ON SCHEMA::HR TO sales_user;
------test connection -------
-- ✅ Should work
SELECT * FROM HR.Employees;

-- ❌ Should fail
SELECT * FROM Sales.Customers;
-------------------------------------------
-- ✅ Should work
SELECT * FROM Sales.Customers;

-- ❌ Should fail
SELECT * FROM HR.Employees;
```

Microsoft SQL Server Management Studio

The database ITI is not accessible. (ObjectExplorer)

Copy message   Show details   OK



User type: SQL user with login

User name: myk

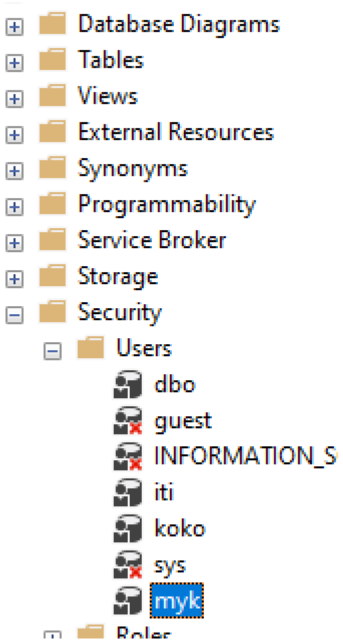
Login name: myk

Default schema:

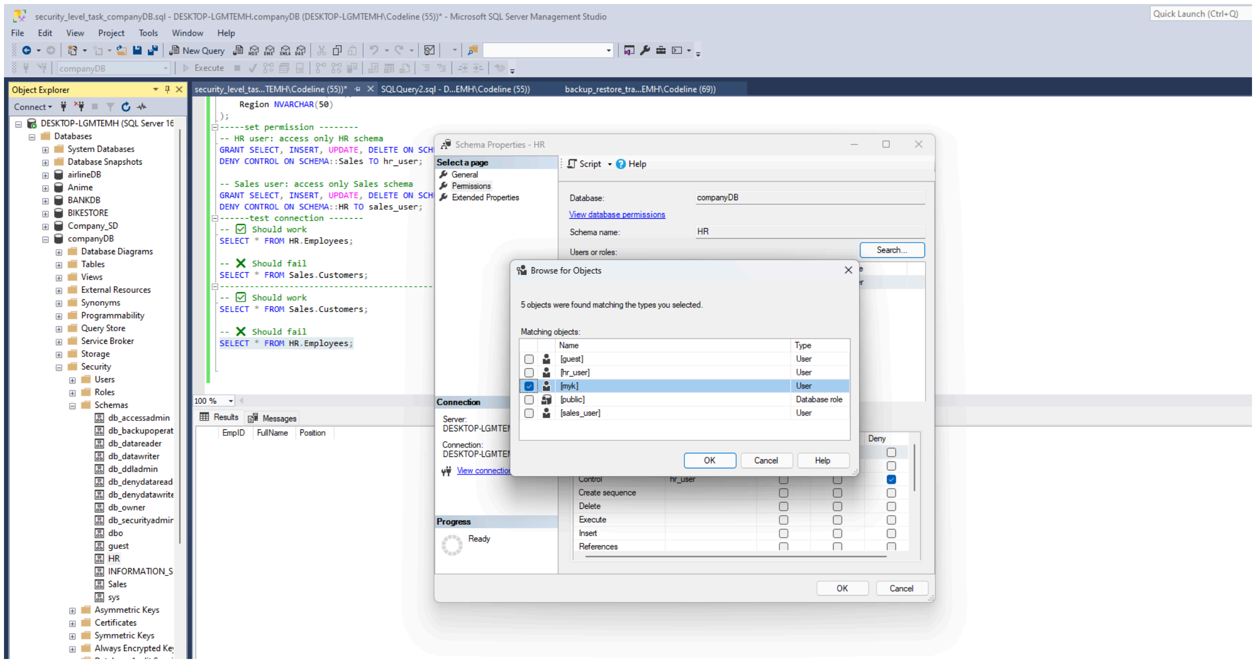
Server: DESKTOP-LGMTEMH

Connection: DESKTOP-LGMTEMH\Codeline

Ready



- Database Diagrams
- Tables
- Views
- External Resources
- Synonyms
- Programmability
- Service Broker
- Storage
- Security
  - Users
    - dbo
    - guest
    - INFORMATION_S
    - iti
    - koko
    - sys
    - myk
  - Roles



security_level_task_companyDB.sql - DESKTOP-LGMTEMH.companyDB (DESKTOP-LGMTEMH\Codeline (55))* - Microsoft SQL Server Management Studio

```sql
    Region NVARCHAR(50)
);
-----set permission --------
-- HR user: access only HR schema
GRANT SELECT, INSERT, UPDATE, DELETE ON SCH
DENY CONTROL ON SCHEMA::Sales TO hr_user;

-- Sales user: access only Sales schema
GRANT SELECT, INSERT, UPDATE, DELETE ON SCH
DENY CONTROL ON SCHEMA::HR TO sales_user;
------test connection -------
-- ✅ Should work
SELECT * FROM HR.Employees;

-- ❌ Should fail
SELECT * FROM Sales.Customers;
-----------------------------------------
-- ✅ Should work
SELECT * FROM Sales.Customers;

-- ❌ Should fail
SELECT * FROM HR.Employees;
```

Results: EmpID | FullName | Position

Schema Properties - HR

Database: companyDB

Schema name: HR

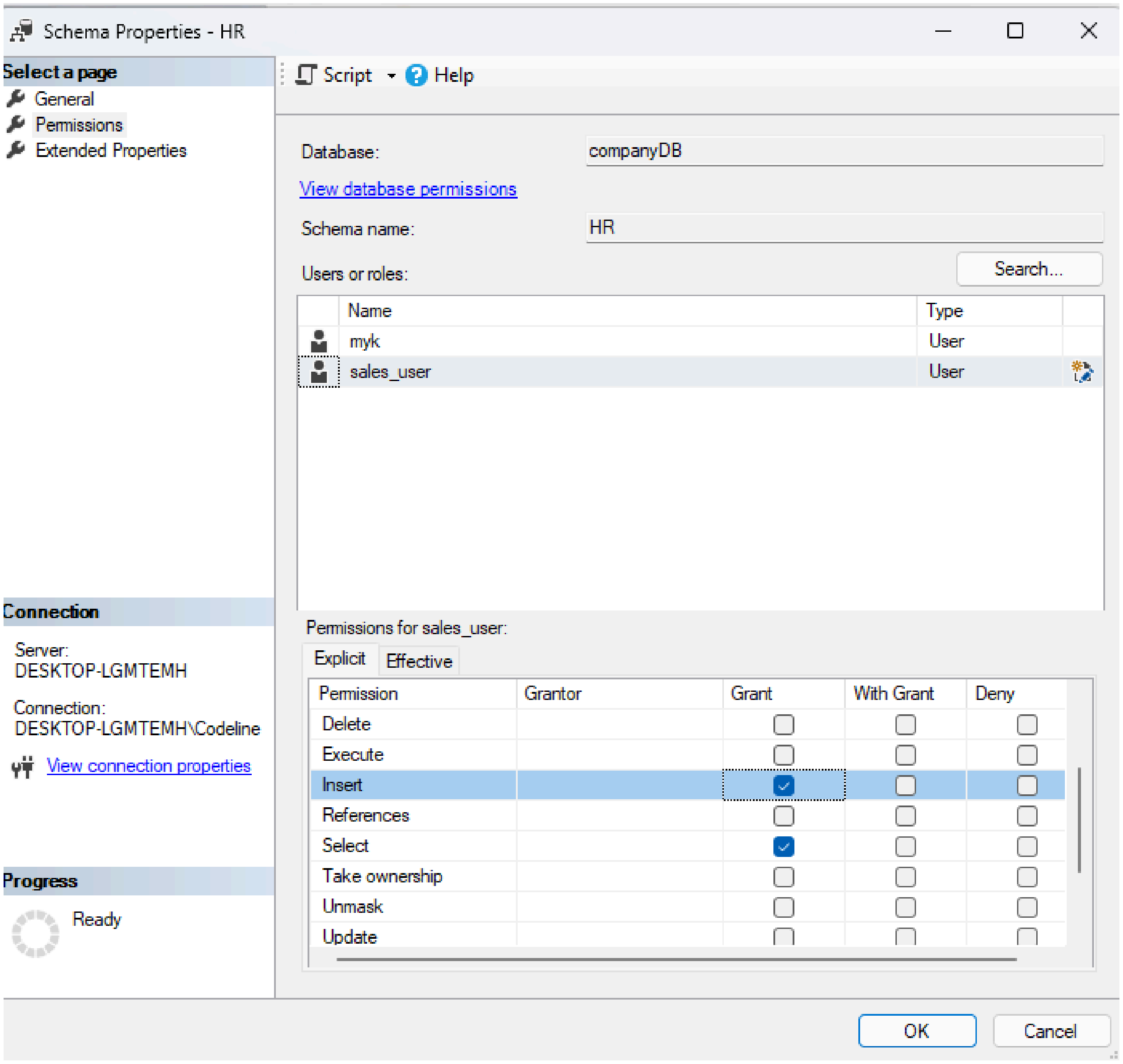
Browse for Objects

5 objects were found matching the types you selected.

Matching objects:

	Name	Type
☐	[guest]	User
☐	[hr_user]	User
✓	[myk]	User
☐	[public]	Database role
☐	[sales_user]	User

				Deny
Control	hr_user	☐	☐	✓
Create sequence		☐	☐	☐
Delete		☐	☐	☐
Execute		☐	☐	☐
Insert		☐	☐	☐
References		☐	☐	☐



Schema Properties - HR

Select a page
- General
- Permissions
- Extended Properties

Script · Help

Database: companyDB

View database permissions

Schema name: HR

Users or roles: Search...

Name	Type
myk	User
sales_user	User

Permissions for sales_user:

Explicit | Effective

Permission	Grantor	Grant	With Grant	Deny
Delete		☐	☐	☐
Execute		☐	☐	☐
Insert		✓	☐	☐
References		☐	☐	☐
Select		✓	☐	☐
Take ownership		☐	☐	☐
Unmask		☐	☐	☐
Update		☐	☐	☐

Connection

Server:
DESKTOP-LGMTEMH

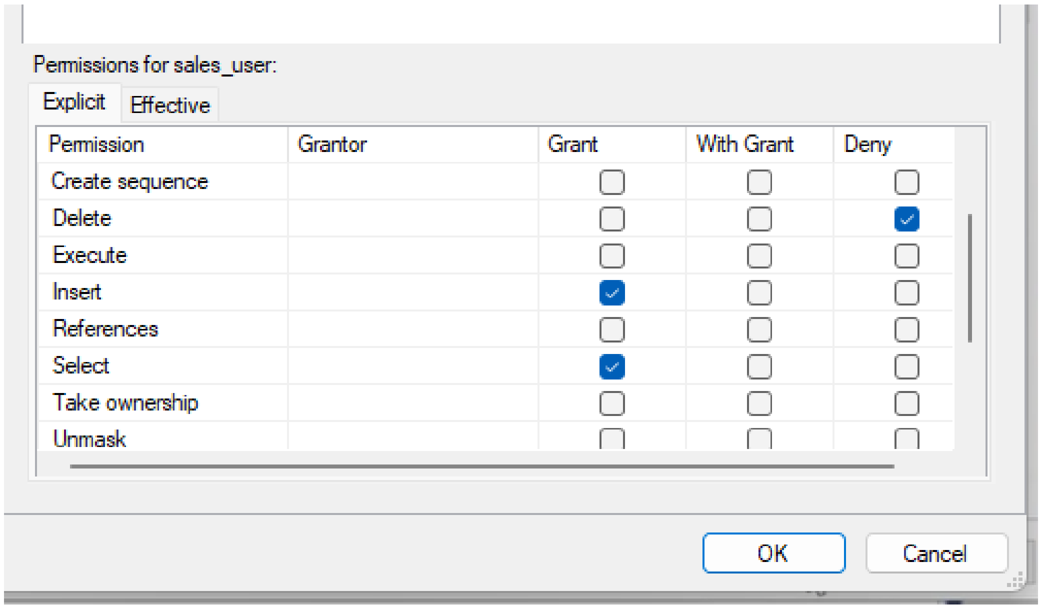
Connection:
DESKTOP-LGMTEMH\Codeline

View connection properties

Progress

Ready

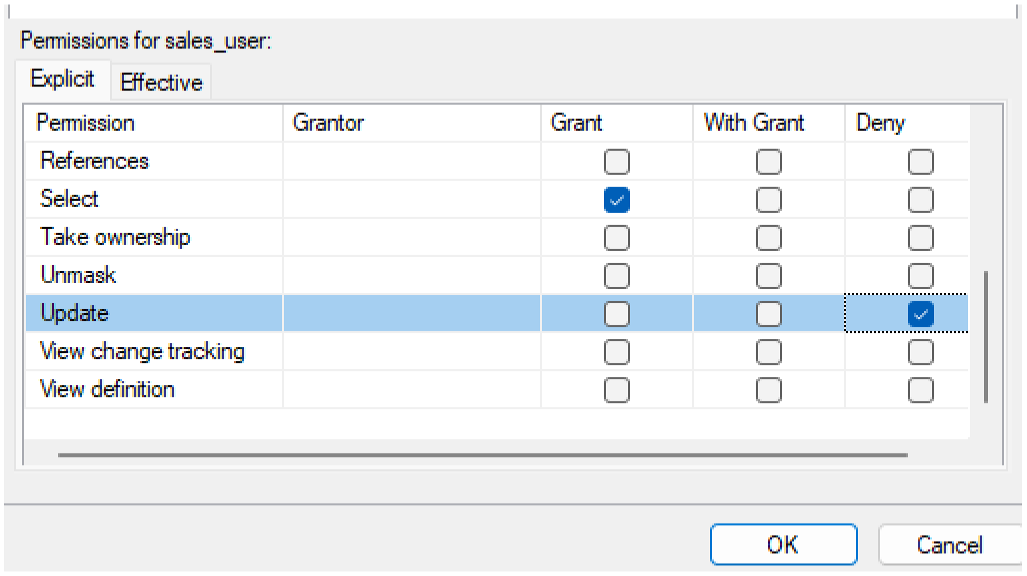
OK · Cancel



Permissions for sales_user:

Explicit | Effective

Permission	Grantor	Grant	With Grant	Deny
Create sequence		☐	☐	☐
Delete		☐	☐	✓
Execute		☐	☐	☐
Insert		✓	☐	☐
References		☐	☐	☐
Select		✓	☐	☐
Take ownership		☐	☐	☐
Unmask		☐	☐	☐



Permissions for sales_user:

Explicit | Effective

Permission	Grantor	Grant	With Grant	Deny
References		☐	☐	☐
Select		✓	☐	☐
Take ownership		☐	☐	☐
Unmask		☐	☐	☐
Update		☐	☐	✓
View change tracking		☐	☐	☐
View definition		☐	☐	☐

demonstrate:

Schema-level permission enforcement

Testing access as different users

Denial of cross-schema access

Allowing only permitted schema data

3. Why Schema-Level Security is Better

Schema-level security allows permission to be managed at a logical group level. This reduces complexity and improves maintainability,
especially when multiple tables or procedures belong to the same functional area. It also ensures clean segregation of access between departments.

4. Reflection Report: SQL Security Levels

- Server-Level Login: Grants access to the SQL Server instance.
- Database-Level User: Maps the login to a specific database.
- Schema-Level Permission: Assigns access at schema level, simplifying control.
- Object-Level Permission: Fine-grained control for individual tables or views.

5. Benefits of Security Levels

- Prevents unauthorized access to sensitive data.
- Reduces human error by limiting access scope.
- Ensures compliance with data protection regulations.
- Allows scalable security across large systems.

6. Real-World Risks Without Security

- Developers or interns may access or change sensitive data.
- Salary leaks or accidental data deletion could occur.
- Auditing and compliance may fail due to poor access control.

7. Security Scenario: The Overpowered Developer

Adil, a developer, had full access to production. He:
1. Accidentally deleted employee records.
2. Shared salary data externally.
3. Created unauthorized users.
4. Caused permission issues by using the wrong schema.

Root Causes:

- No dev/prod environment separation.
- Lack of schema enforcement.
- No role-based access control.
- Excessive permissions to non-admins.

Solutions:

- Use schema-level permissions to restrict access.
- Assign roles like ReadOnly_Dev for safer dev access.
- Enforce view-based access to sensitive data.
- Log audit actions and separate environments.

8. Lessons Learned

- Developers should have minimal, read-only access in prod.
- DBAs should handle role creation and schema ownership.
- Least-privilege principle protects data and systems.

9. Bonus: Role-Based Security Test

A role named ReadOnly_Dev was created with SELECT only access. Attempts to INSERT or DELETE were denied, confirming permission restrictions.