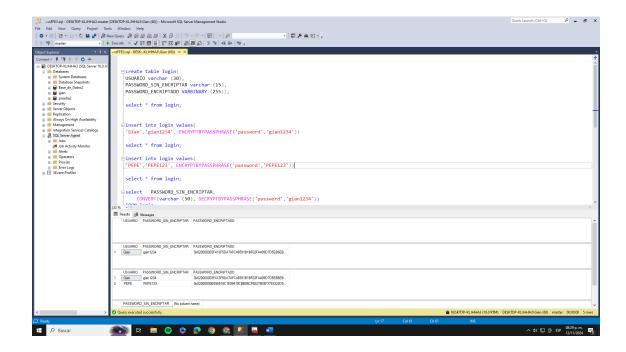
Resumen de los tipos de encriptación en SQL Server:

- 1. Encriptación a nivel de columna: Utiliza claves simétricas o certificados.
- 2. **TDE** (**Transparent Data Encryption**): Encripta toda la base de datos.
- 3. Encriptación de conexiones (SSL/TLS): Protege la transmisión de datos.
- 4. **Hashing**: Utilizado para contraseñas y verificación de integridad.
- 5. Always Encrypted: Encripta datos de forma que el servidor nunca pueda verlos.



CREAMOS LA BASE DE DATOS GIAN

```
☐ R DESKTOP-KLJHHA3 (SQL Server 16.0.1)

  Databases
    System Databases
    Database Snapshots

→ Base_de_Datos2

    🕀 📔 gian
    ⊕ prubba2
  Security
  Server Objects
  Replication
  Always On High Availability
  Management
  Integration Sarvices Catalogs

☐ 
☐ SQL Server Agent

    🔢 📕 Jobs
      Job Activity Monitor
    Alerts
    Operators
    Proxies
    Error Logs
```

Creamos la tabla

```
USUARIO varchar (30),
PASSWORD_SIN_ENCRIPTAR varchar (15),
PASSWORD_ENCRIPTADO VARBINARY (255));
```

Insertamos la información y al aves lo encriptamos

```
select * from login;

insert into login values(
    'PEPE', 'PEPE123', ENCRYPTBYPASSPHRASE('password', 'PEPE123'))

select * from login;

insert into login values(
    'Gian', 'gian1234', ENCRYPTBYPASSPHRASE('password', 'gian1234'))
```

y sale así

	USUARIO	PASSWORD_SIN_ENCRIPTAR	PASSWORD_ENCRIPTADO
1	Gian	gian1234	0x020000003F413F5DA7AFC46591B1BF02FA409D7D5EB6EB
2	PEPE	PEPE123	0x02000000E698516C1E99410CBB86CF68275E8F3793322D5

Y después lo desencriptamos

```
=select PASSWORD_SIN_ENCRIPTAR,

CONVERT(varchar (50), DECRYPTBYPASSPHRASE('password','gian1234'))
FROM login
```