Task 3:

- **What is a SQL transaction?**

A SQL transaction is a group of one or more SQL commands that are executed together. It ensures data accuracy and consistency — either all commands succeed, or none do.

- **How to Write a Transaction Block:**

```
BEGIN TRANSACTION;
UPDATE accounts SET balance = balance - 100 WHERE id = 1;
COMMIT;   —> it means that transaction is done completely.
```

in case of a failure in the transaction:

```
BEGIN TRANSACTION;
UPDATE accounts SET balance = balance - 100 WHERE id = 1;
ROLLBACK;----> it will cancel all the changes from the beginning of the transaction.
```

- **Write a script that:**

1. Starts a transaction
2. Tries to insert two new applicants
3. The second insert should have a duplicate ApplicantID (to force failure)
4. Rollback the whole transaction if any error occurs

```
BEGIN TRANSACTION;

BEGIN TRY
   INSERT INTO Applicants (ApplicantID, FullName, Email, Source, AppliedDate)
   VALUES (104, 'Majid Al Abri', 'majid.a@example.com', 'Website', '2025-05-10');
   -- This will fail due to duplicate ApplicantID

   INSERT INTO Applicants (ApplicantID, FullName, Email, Source, AppliedDate)
   VALUES (104, 'Sara Al Zadjali', 'sara.z@example.com', 'Referral', '2025-05-11');

   COMMIT TRANSACTION
END TRY
BEGIN CATCH
   ROLLBACK TRANSACTION

   PRINT 'Transaction failed and was rolled back.';
   PRINT ERROR_MESSAGE();  -- Optional: show error details
END CATCH
```

- **the error message will be shown:**

```
(0 rows affected)
Transaction failed and was rolled back.
Violation of PRIMARY KEY constraint 'PK__Applican__39AE914843E126FD'.
Cannot insert duplicate key in object 'dbo.Applicants'. The duplicate key value is
(104).
```

Note: solution attached in the SQL file.

ACID Properties Exploration

7. Research and summarize each of the ACID properties:

ACID stands for Atomicity, Consistency, Isolation, and Durability.
It defines how a transaction is processed in a predictable manner and reliably, making sure that database stays consistent, even in cases of failures or concurrent accesses.

The Four ACID Properties:

Atomicity:

- A transaction is **all or nothing**: either every part succeeds, or nothing is done.
- If any step fails, the whole transaction is rolled back.

Example: Bank Transfer, You send OMR 50 from your account to a friend. If the money doesn't leave your account, your friend won't get it either.(All or nothing)

Consistency

- A transaction must leave the database in a **valid state**, following all rules and constraints.
- It ensures the data remains correct before and after the transaction.
Example: A store has 5 phones. If you buy 1, it should show 4 left — not a wrong number like 6 or -1.

Isolation

- Transactions run **independently**, as if they were the only one in the system.
- Prevents interference between transactions that are happening at the same time.

Example: Two people use the ATM at the same time. They won't affect each other's accounts.

Durability

- Once a transaction is **committed**, it is **permanently saved**, even if the system crashes right after.
- Guarantees data persistence.

Example: You order food online. Even if the app crashes, your order is still there.