**‘AccessBank DBA Internship Program’**

**Project 18: Locking and Blocking**

At 11:00 AM, you got a phone call from the IT director. He asked if there is any problem in the database because users are not able to do their work. When you checked the SQL Server status, you noticed that there is a blocking in the system.

You must fix the issue. To simulate this scenario,

1. Create the following tables with a dummy data:

create table A (i1 int)

create table B (i1 int)

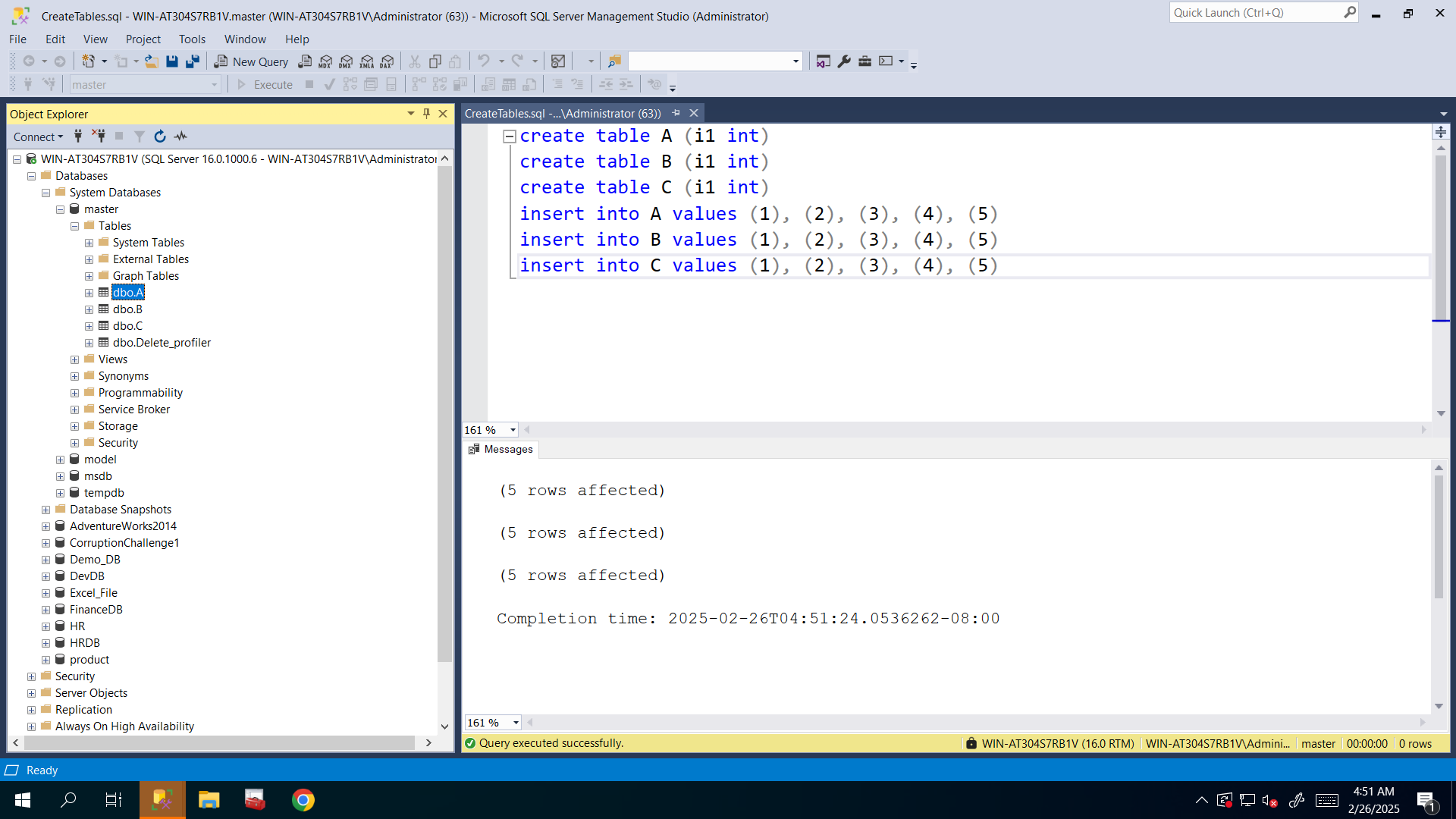
create table C (i1 int)

insert into A values (1), (2), (3), (4), (5)

insert into B values (1), (2), (3), (4), (5)

insert into C values (1), (2), (3), (4), (5)

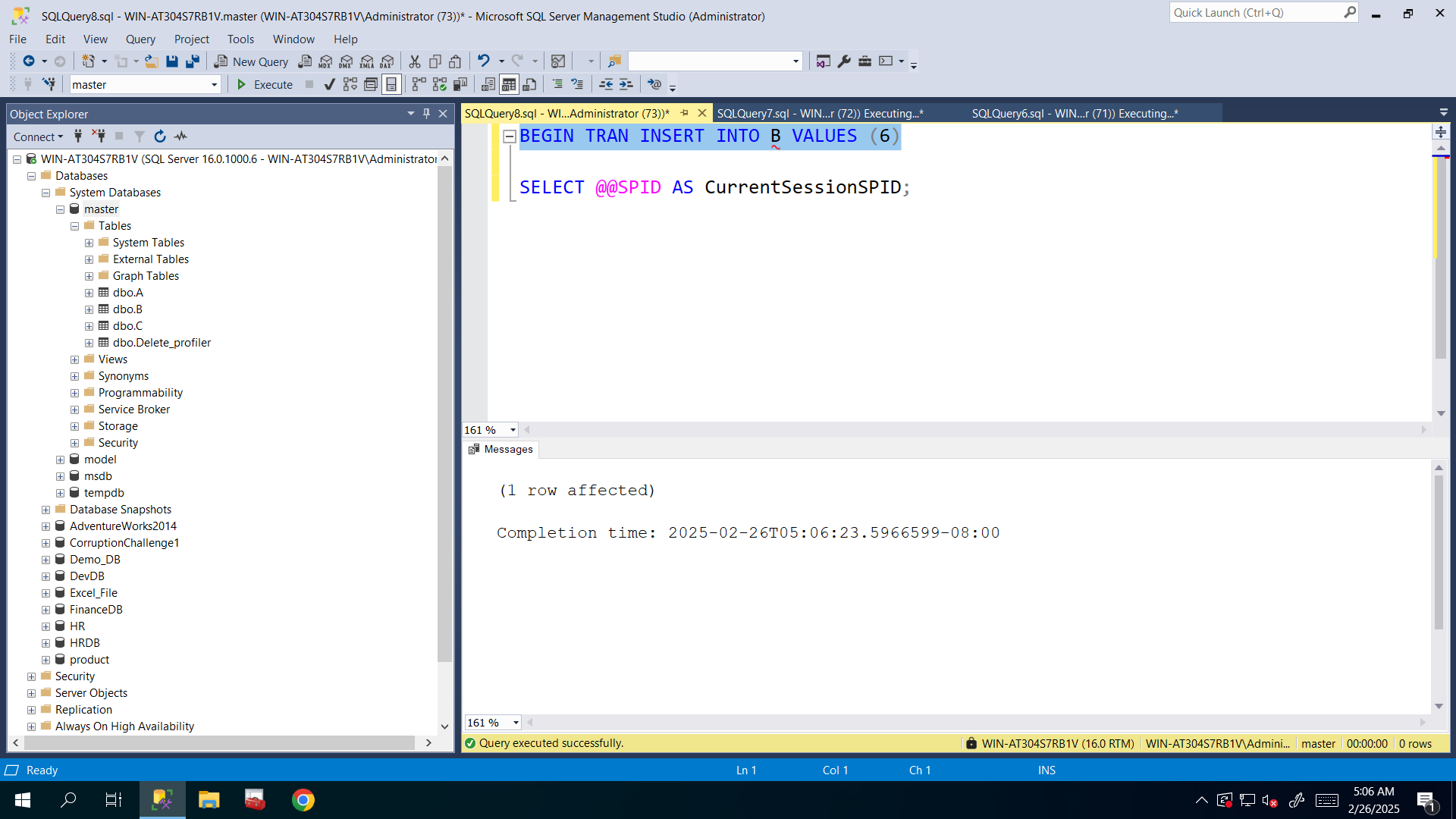
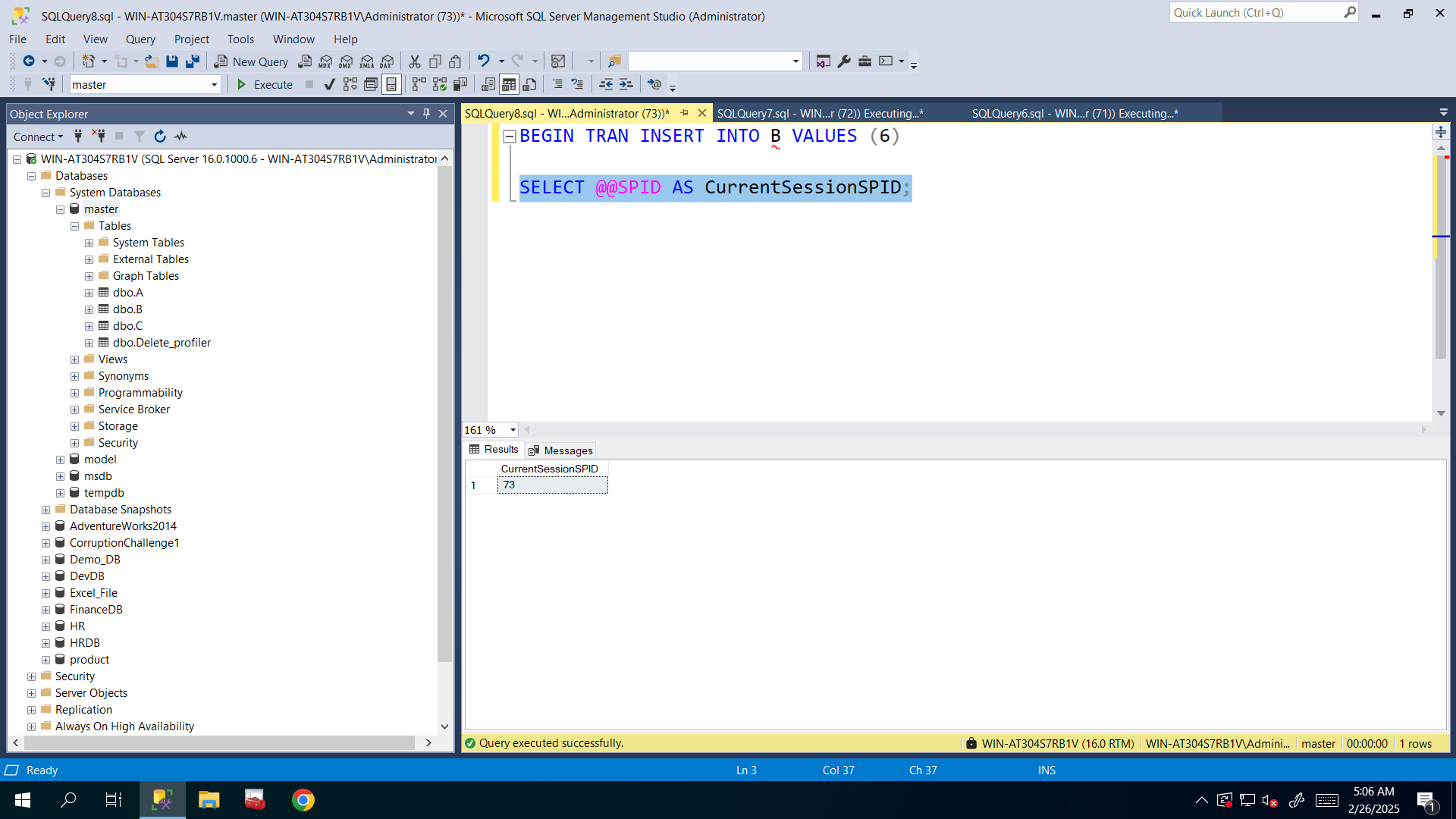
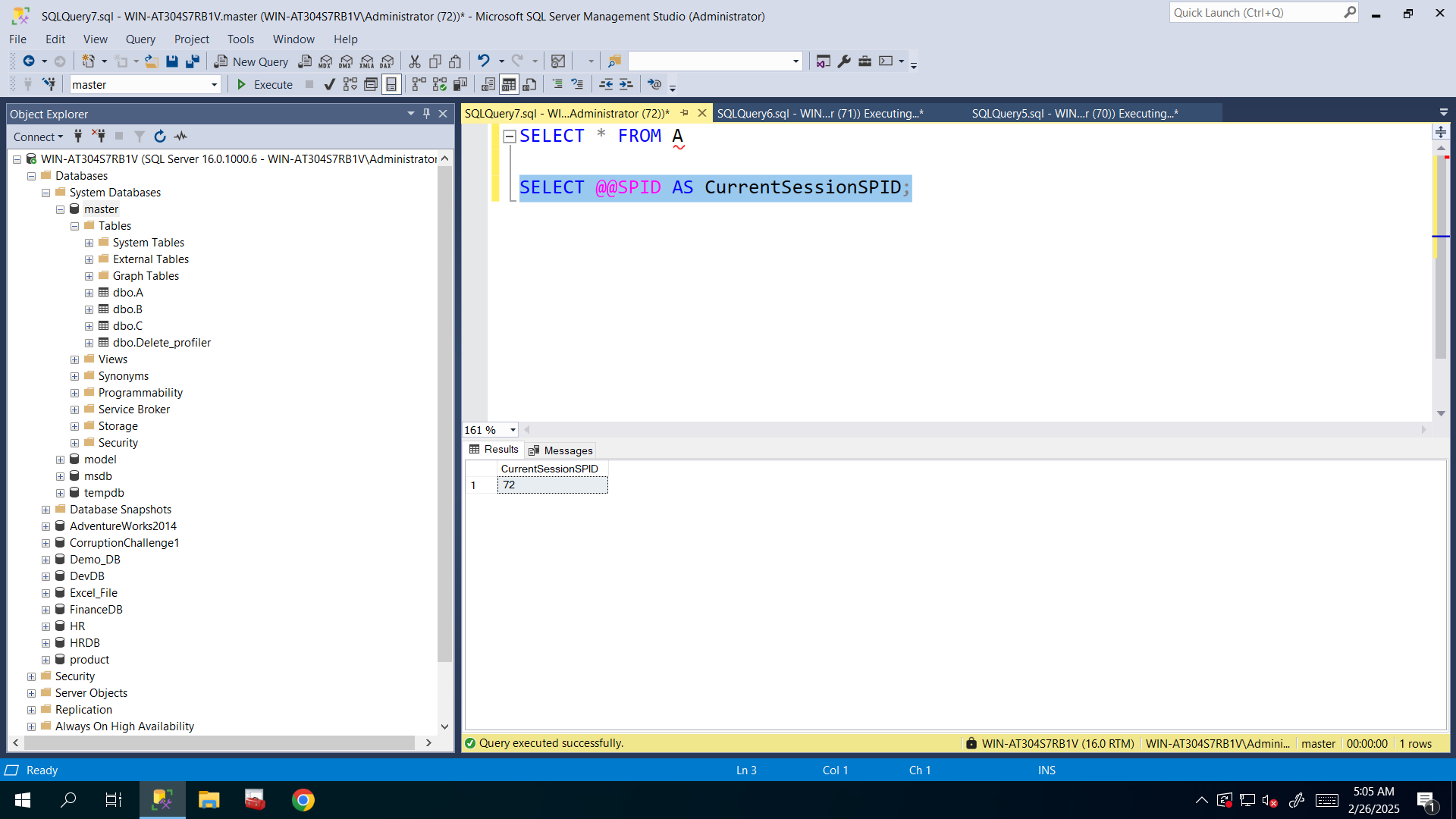
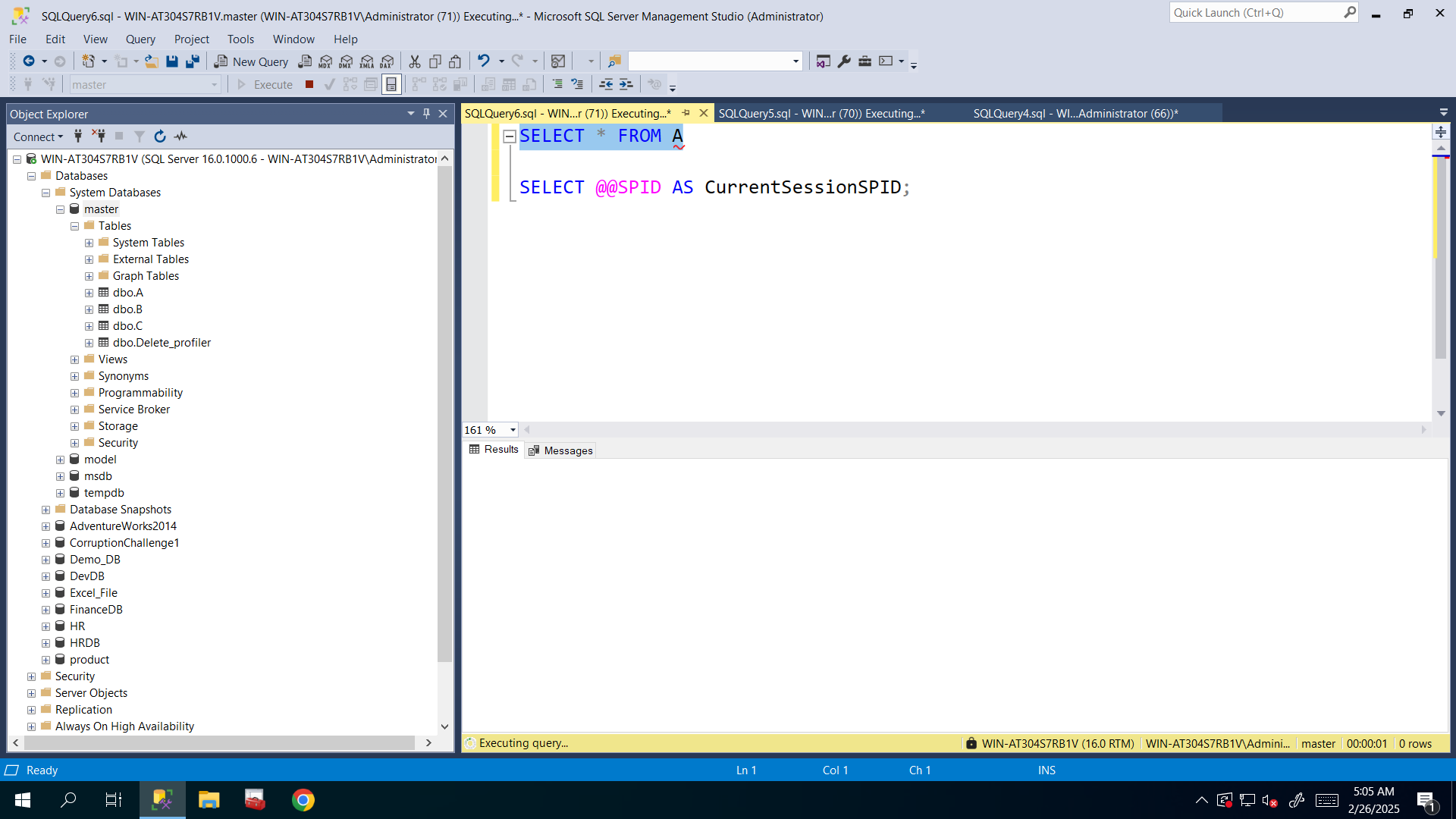
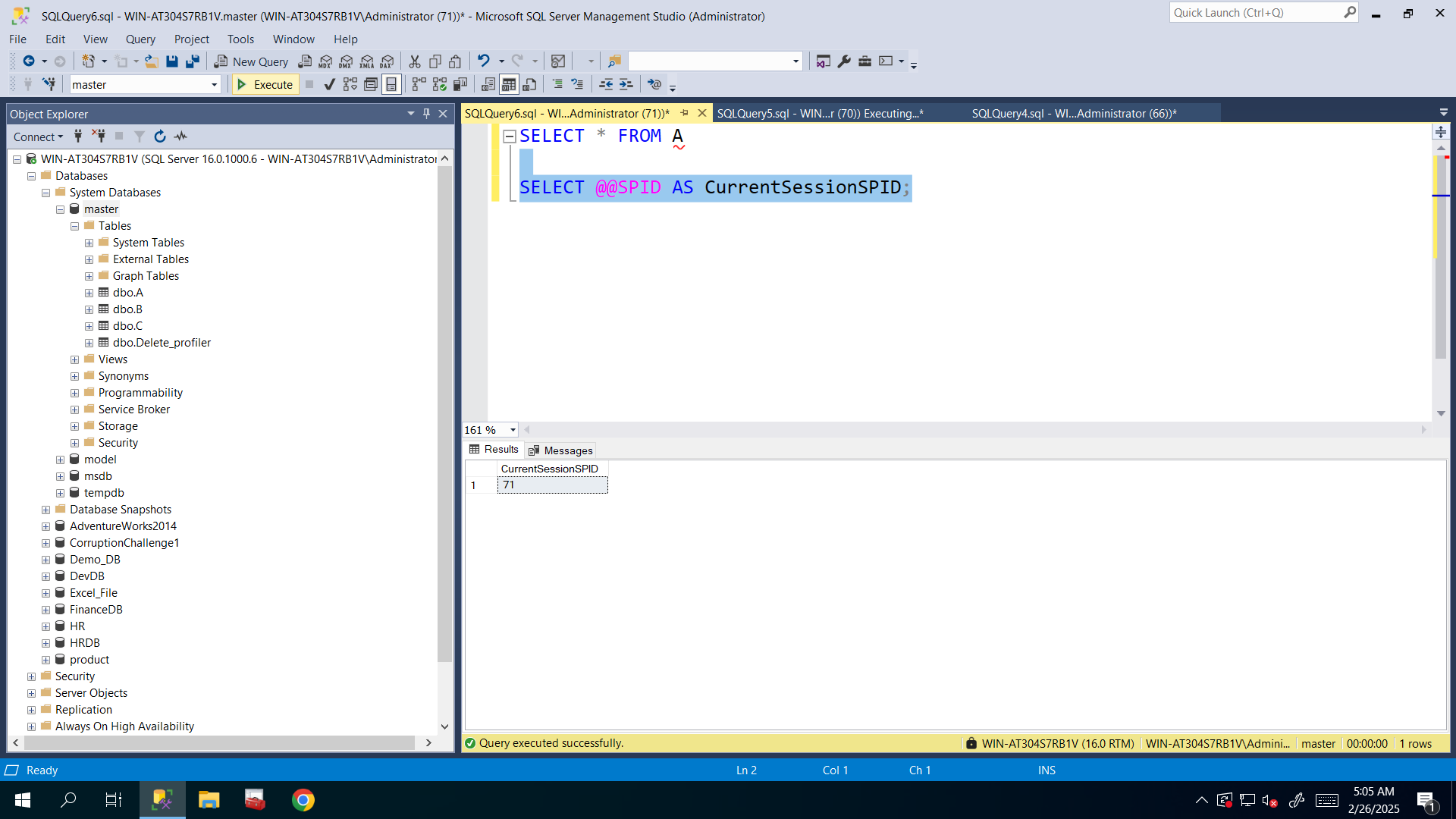
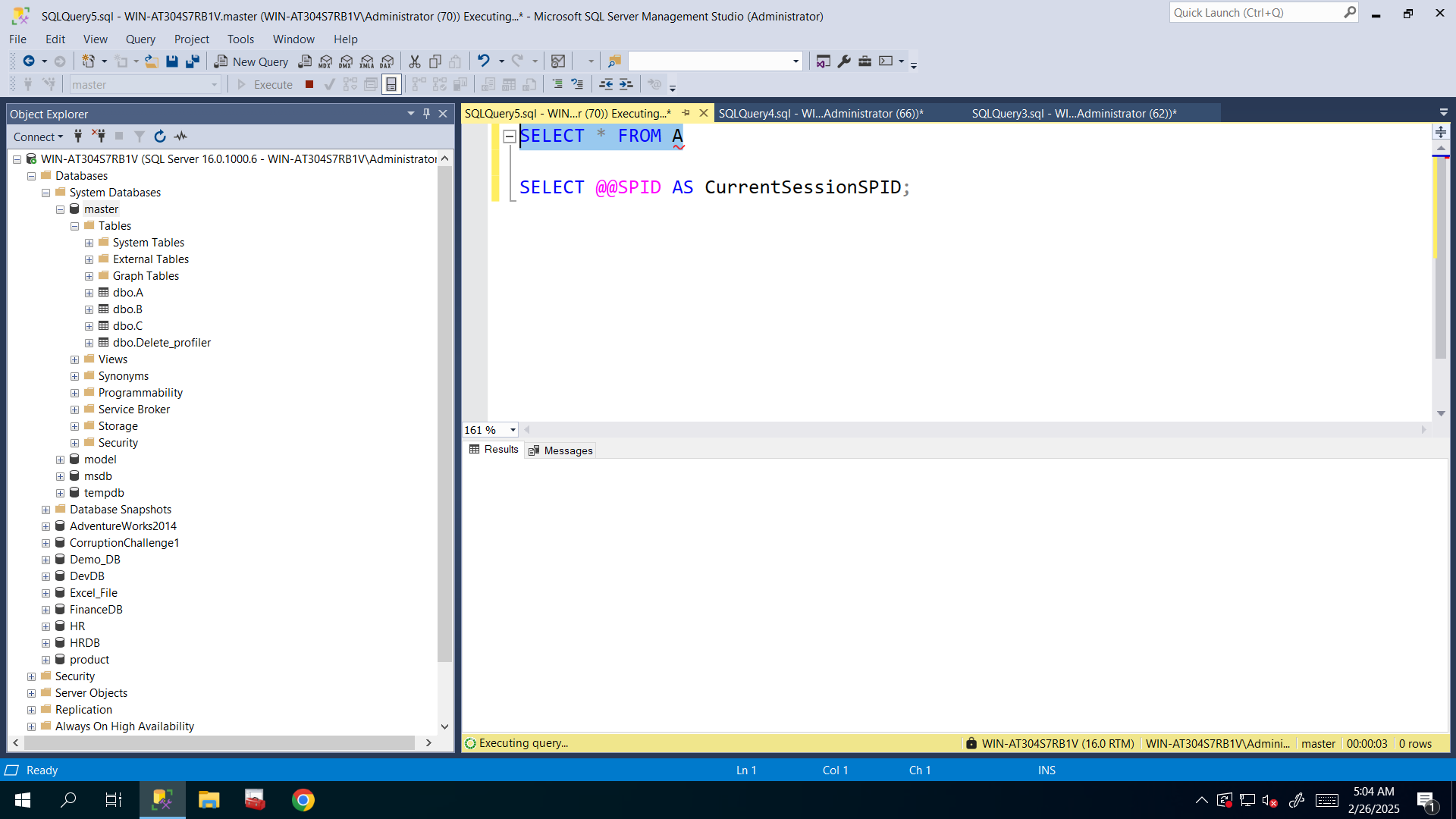
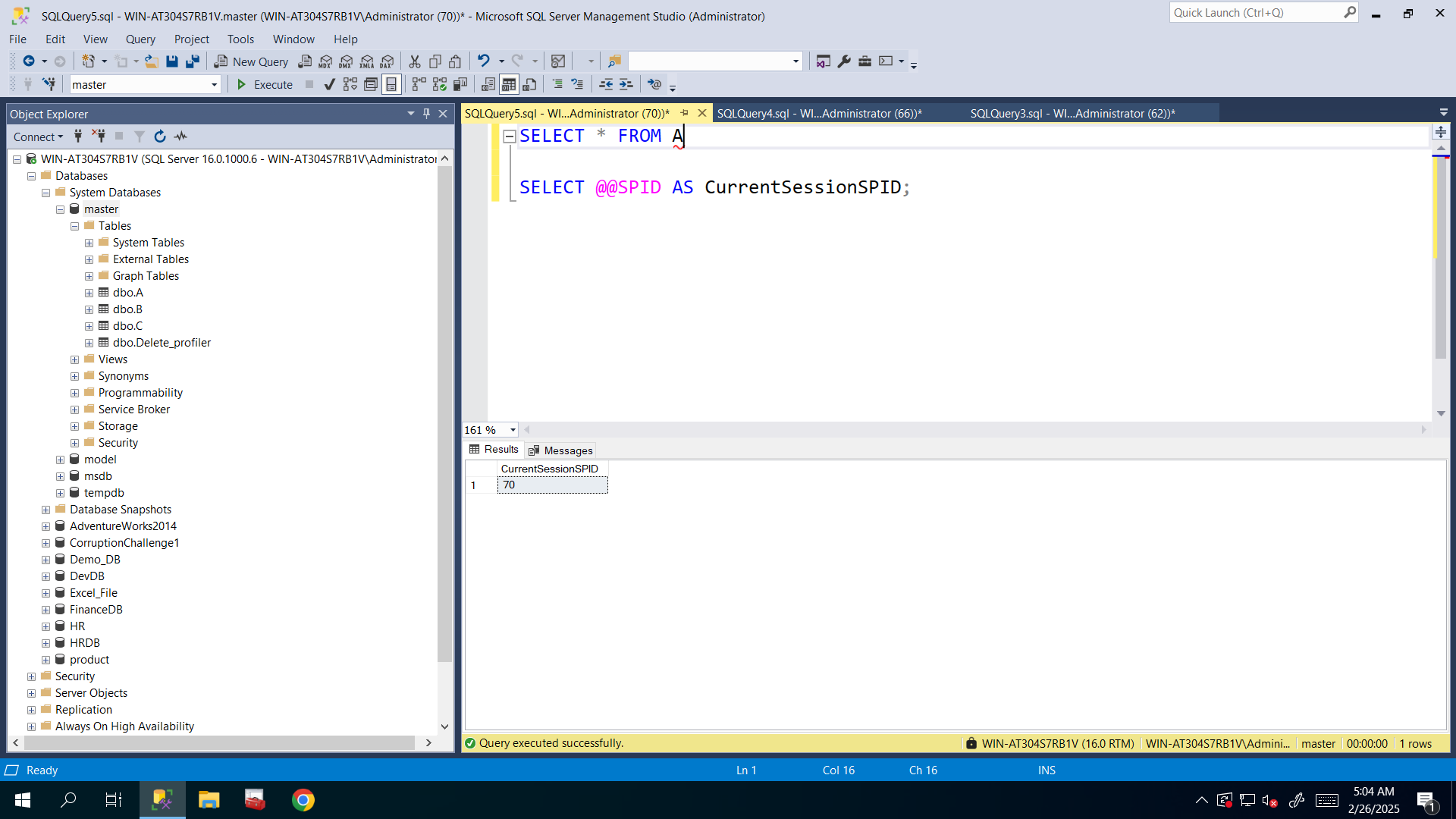
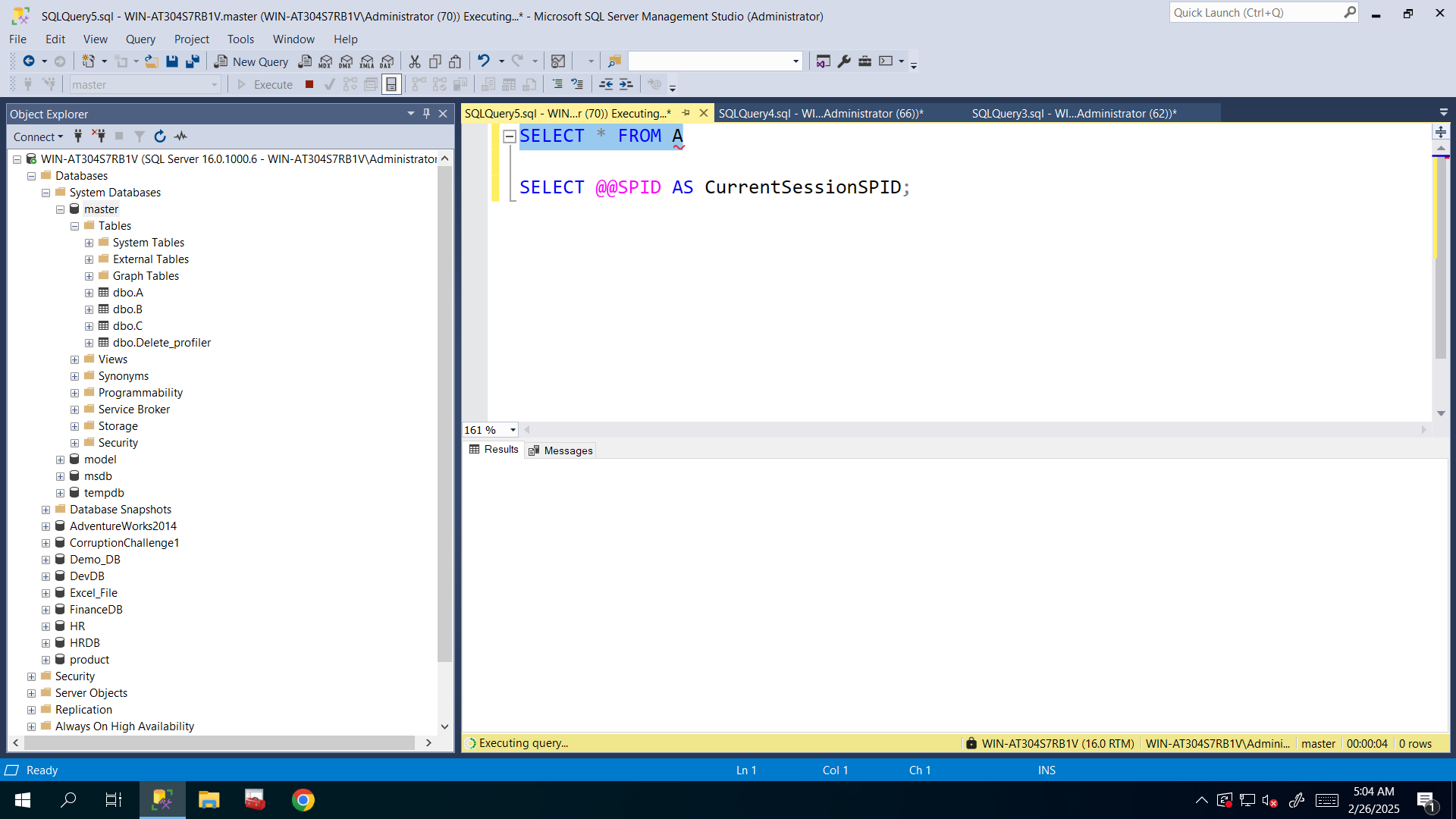
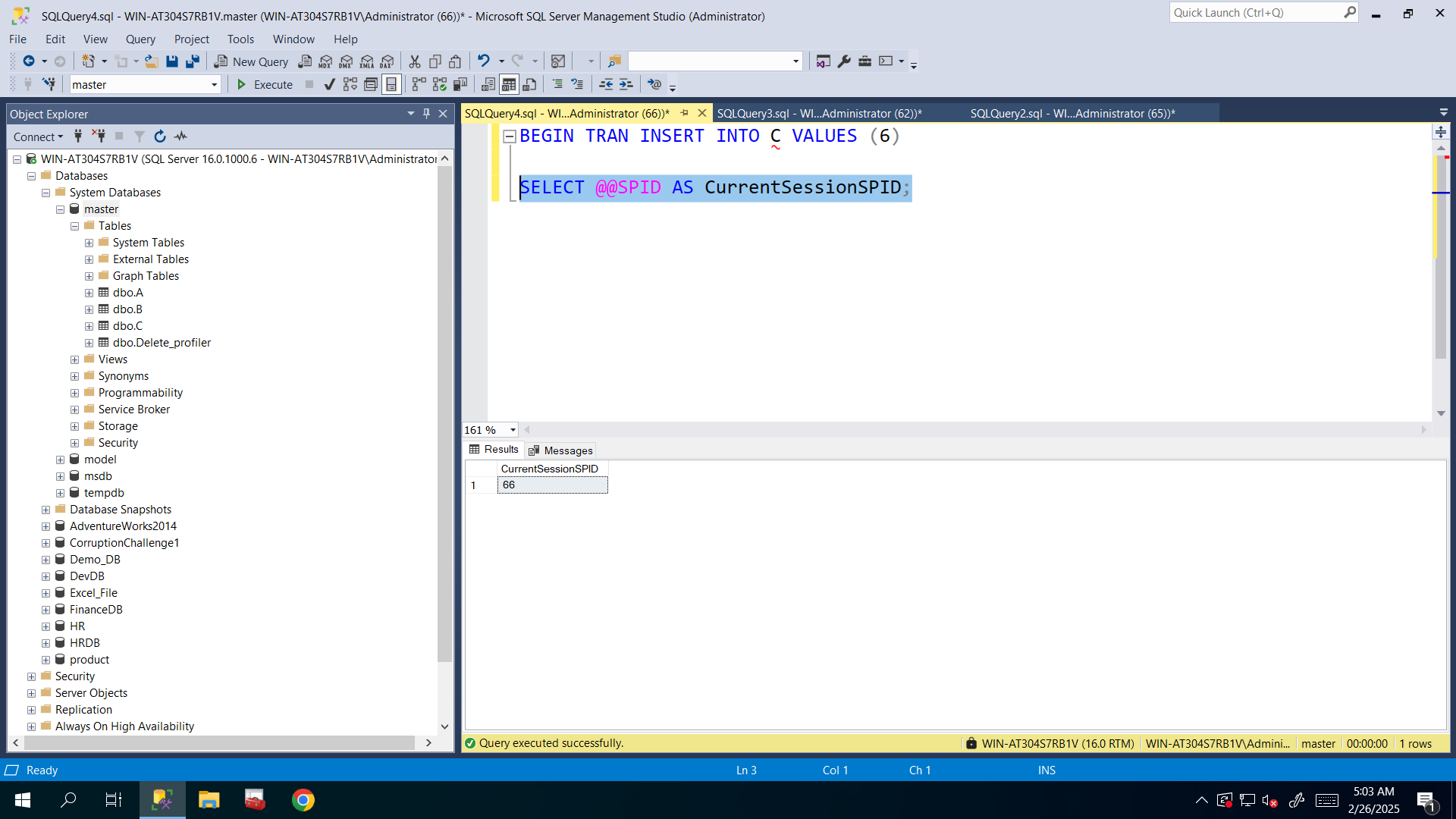
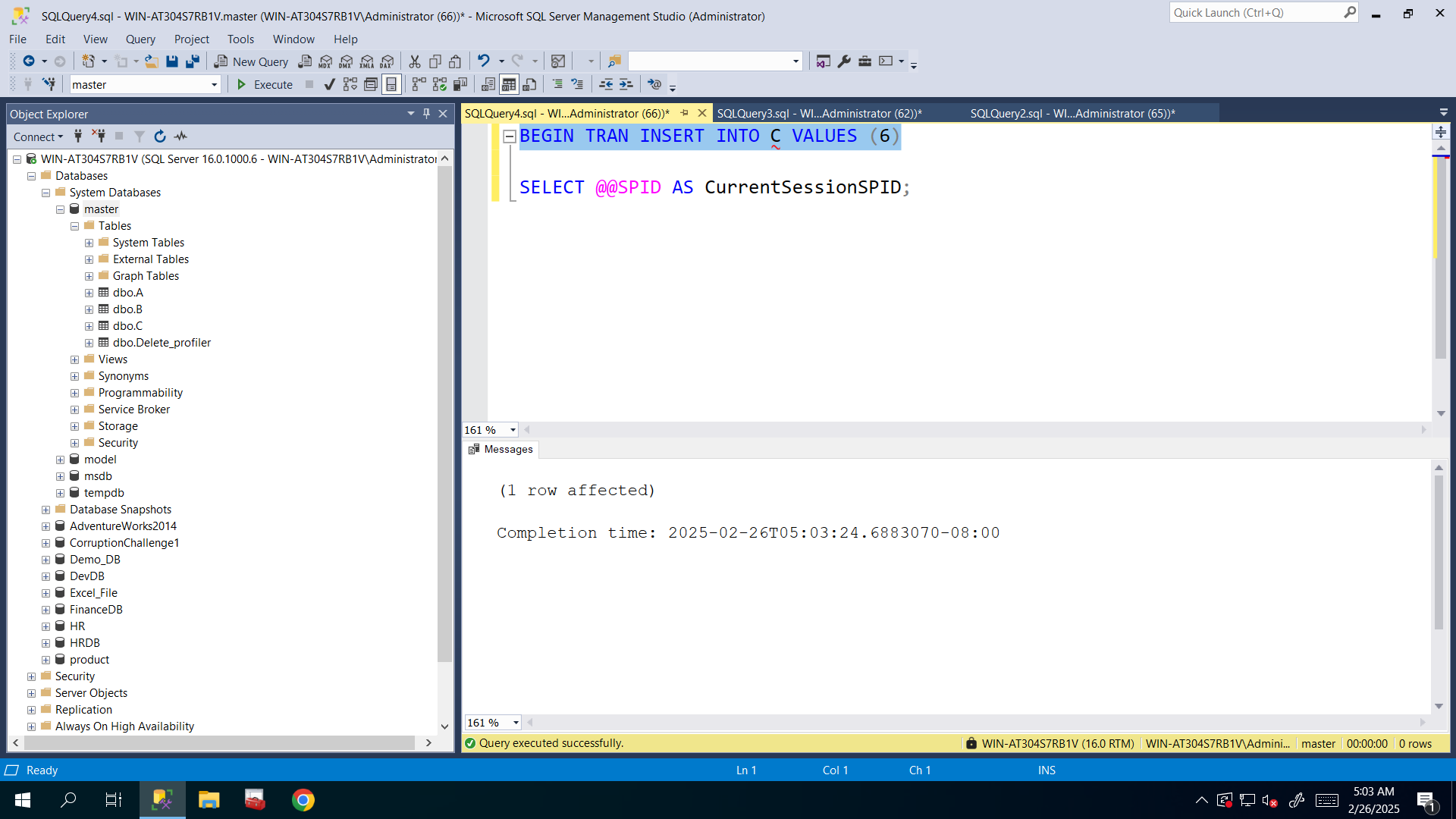
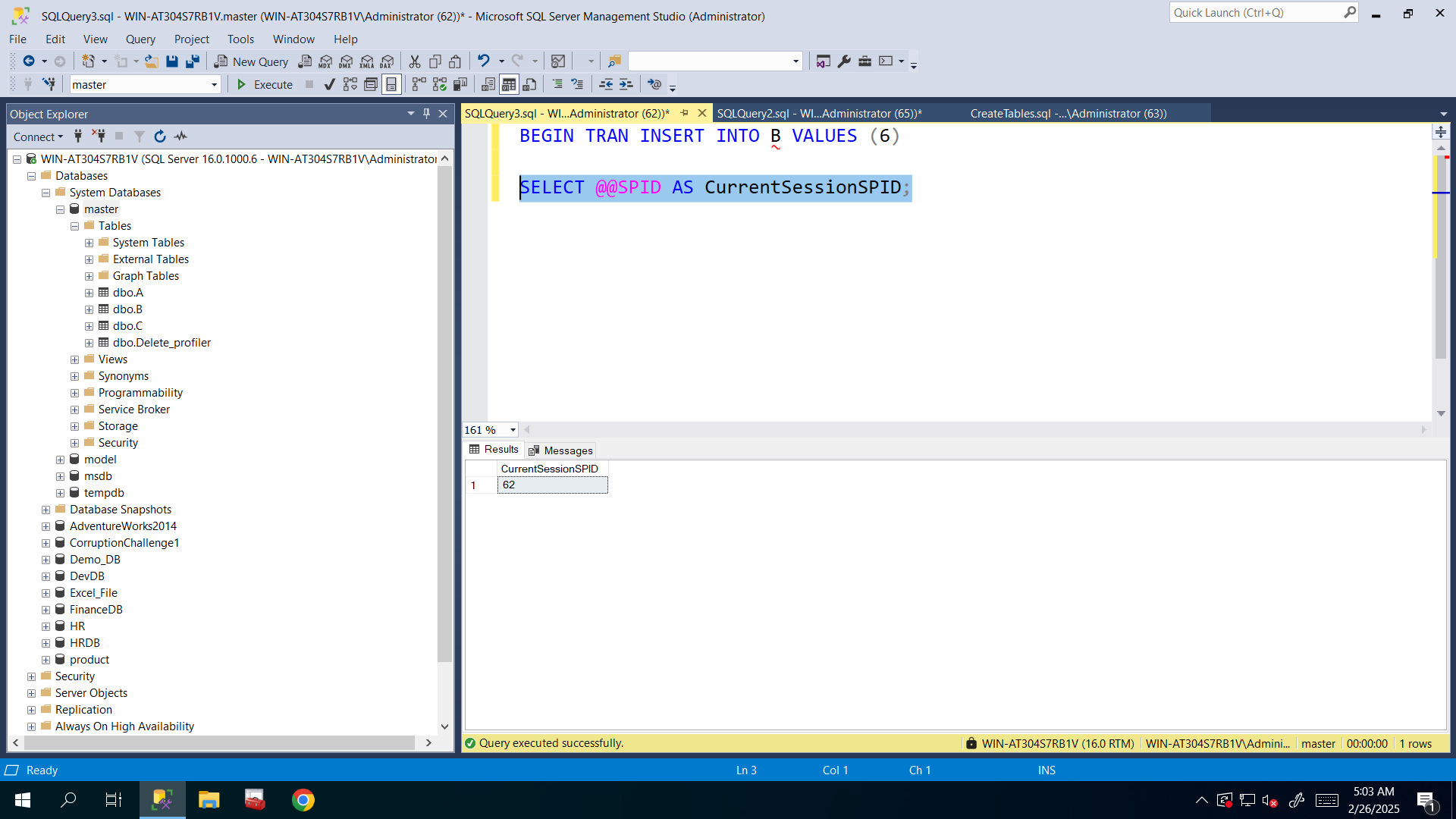
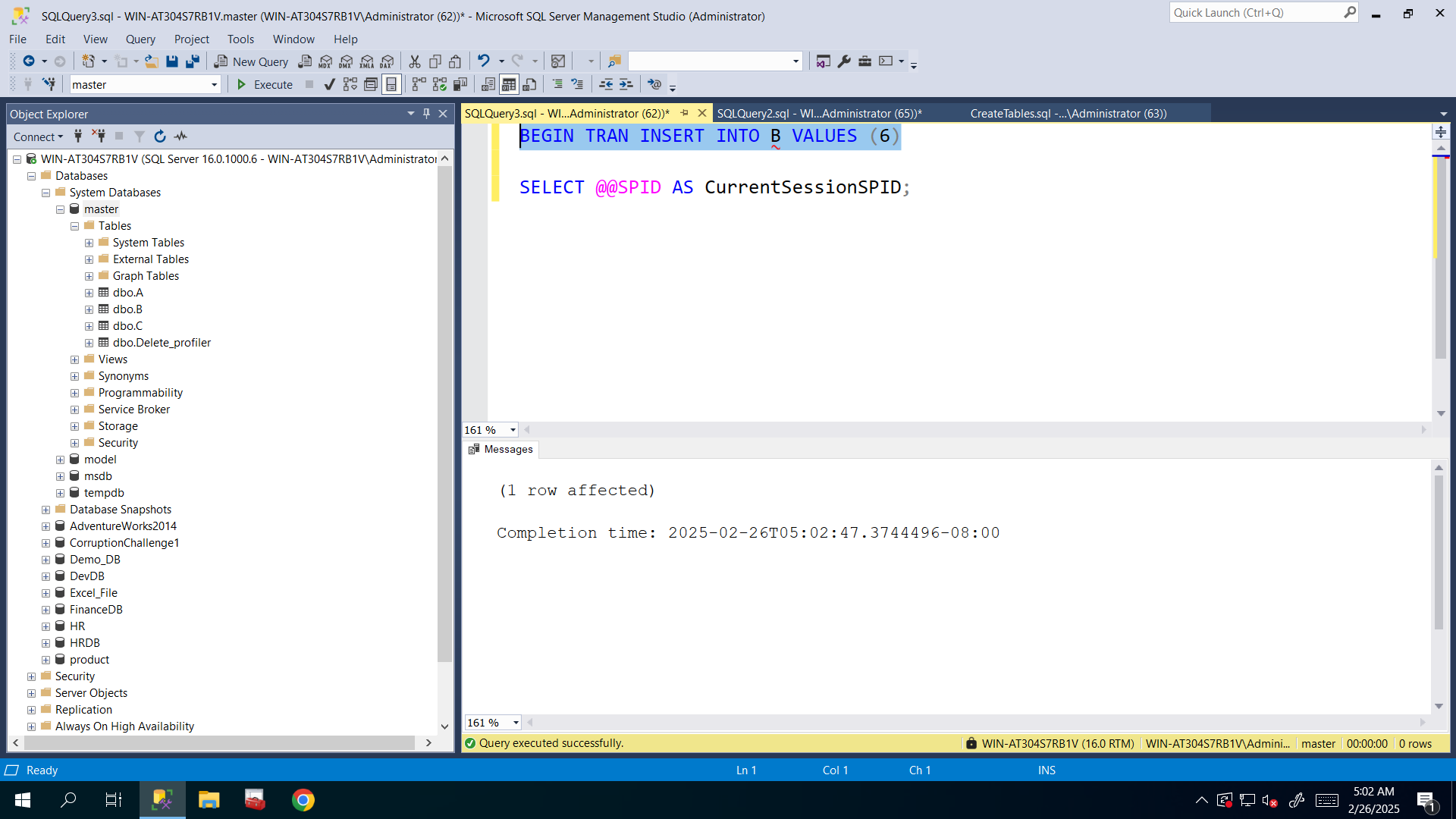
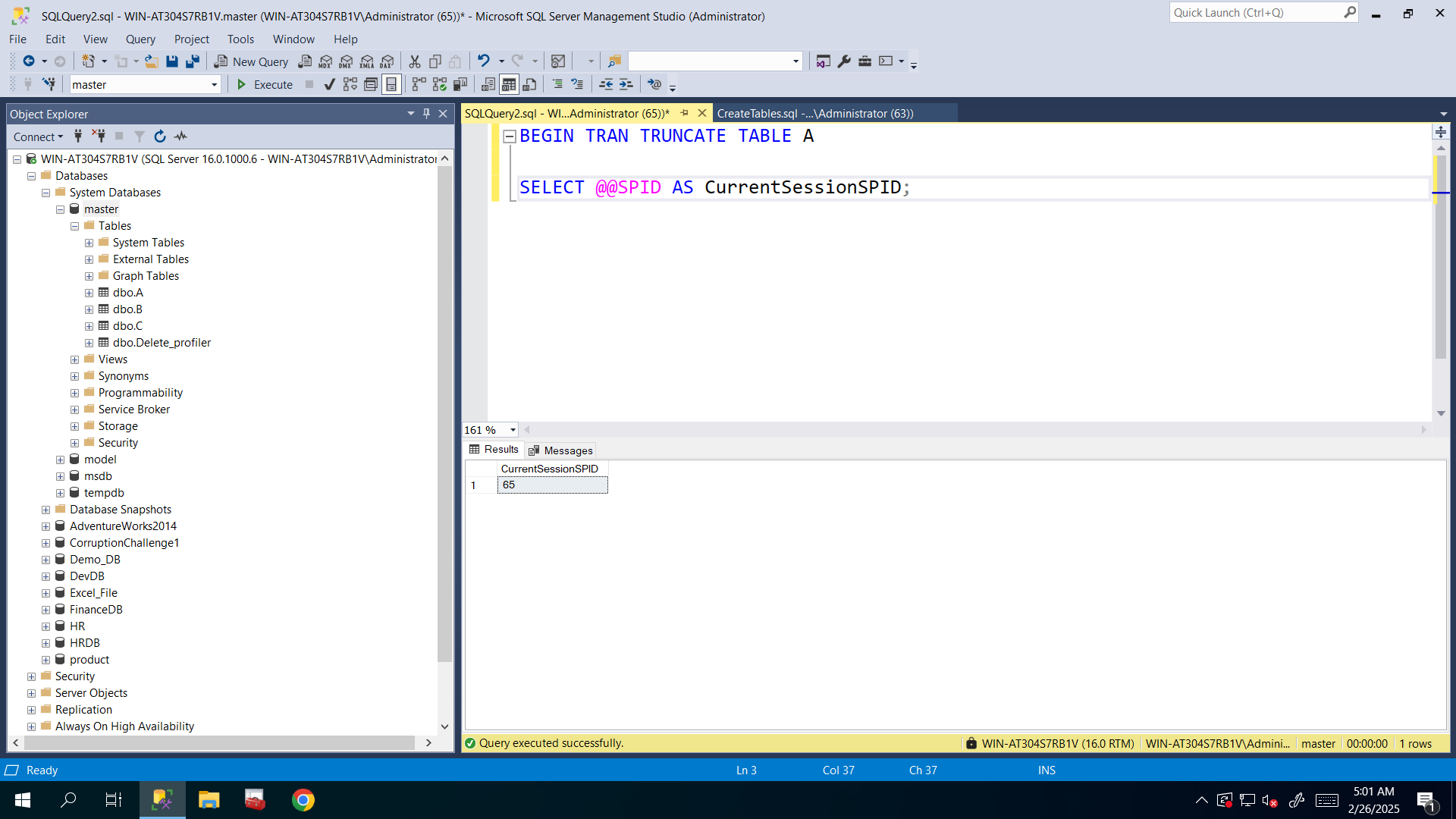
**I created these tables and inserted dummy data into them by using following query as you can see in the following screenshot.**



1. Run the following queries in the separate sessions and as you run these queries, fill up the below table:

I executed all of these queries and checked their SPID by using that following query:

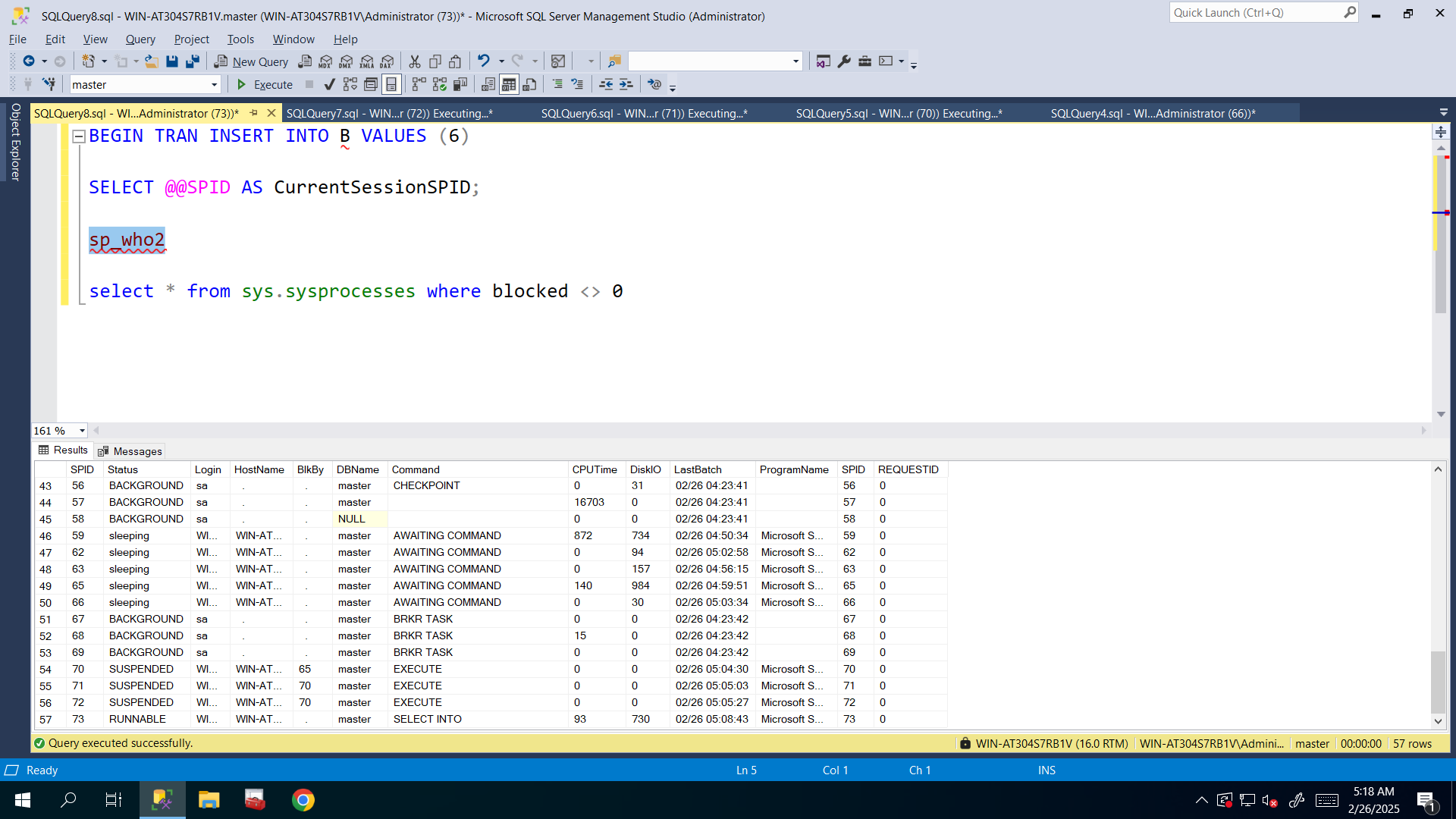
**SELECT @@SPID AS CurrentSessionSPID;**



|  |  |
| --- | --- |
| SPID | QUERY |
| **65** | BEGIN TRAN TRUNCATE TABLE A |
| **62** | BEGIN TRAN INSERT INTO B VALUES (6) |
| **66** | BEGIN TRAN INSERT INTO C VALUES (6) |
| **70** | SELECT \* FROM A |
| **71** | SELECT \* FROM A |
| **72** | SELECT \* FROM A |
| **73** | BEGIN TRAN INSERT INTO B VALUES (6) |

1. Scenario is ready. Use sp\_who2 system stored procedure to find out what sessions are currently running and which blocked. Please pay attention only to the SPIDs which are above 50 (user processes), because below 50 are usually system processes.

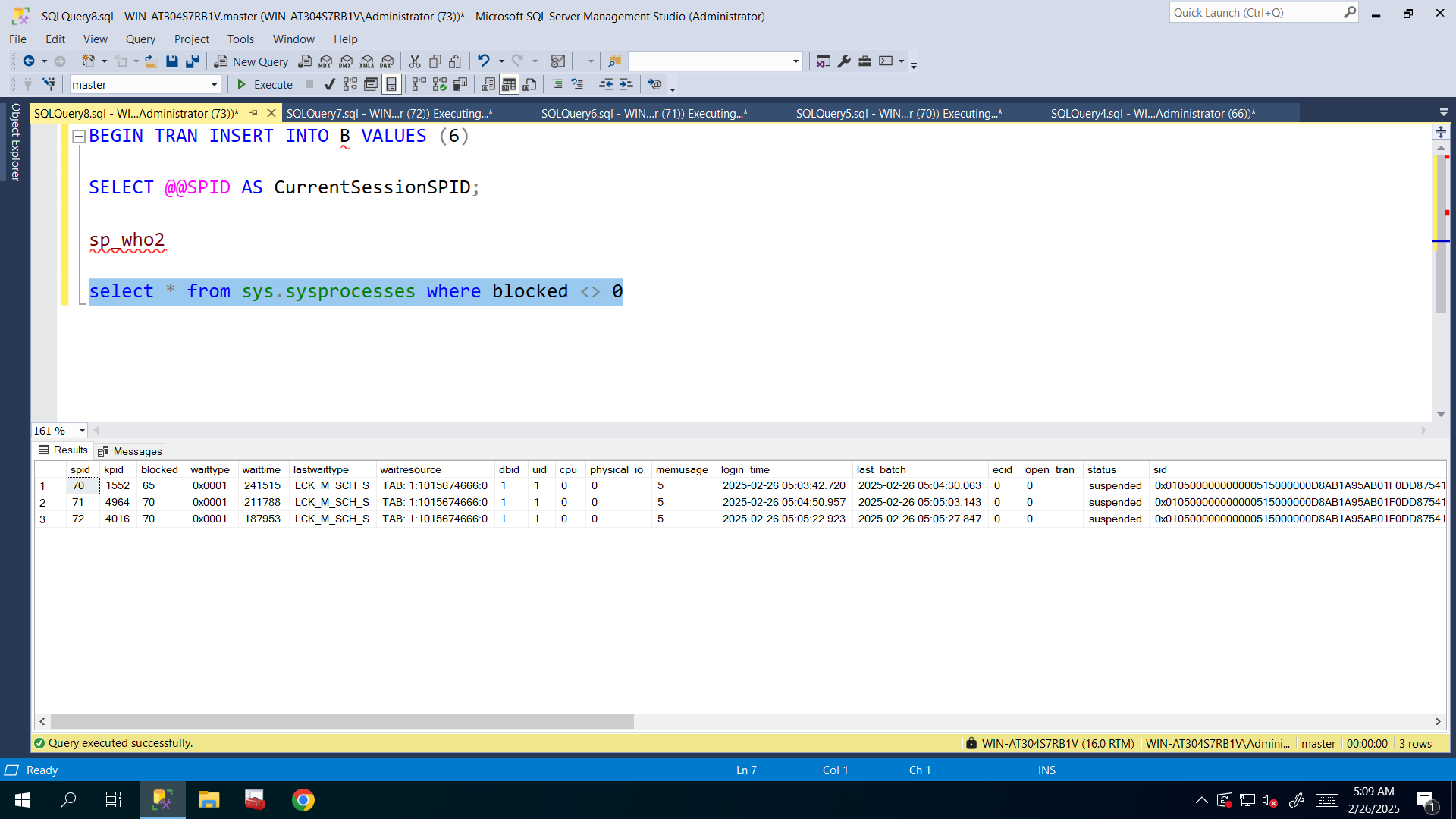
**I saw that 70,71,72 is blocked**



1. sp\_who2 does not show you which lock the blocked sessions are waiting to acquire. In order to find locks which blocked sessions are waiting to acquire, use the system view sys.sysprocesses:

select \* from sys.sysprocesses where blocked <> 0

**I executed that query to see which lock types the blocked sessions wants to get**:



1. Based on the output of 2nd, 3rd and 4th tasks fill the below gaps:

In SQL Server, LCK\_M\_SCH\_S stands for **"Lock Mode Schema Stability"**. It indicates that a session is waiting to acquire a **Schema Stability (Sch-S) lock** on an object.

**70** session is blocked by **65** session and waiting to acquire **Sch-S** lock

**71** session is blocked by **70** session and waiting to acquire **Sch-S** lock

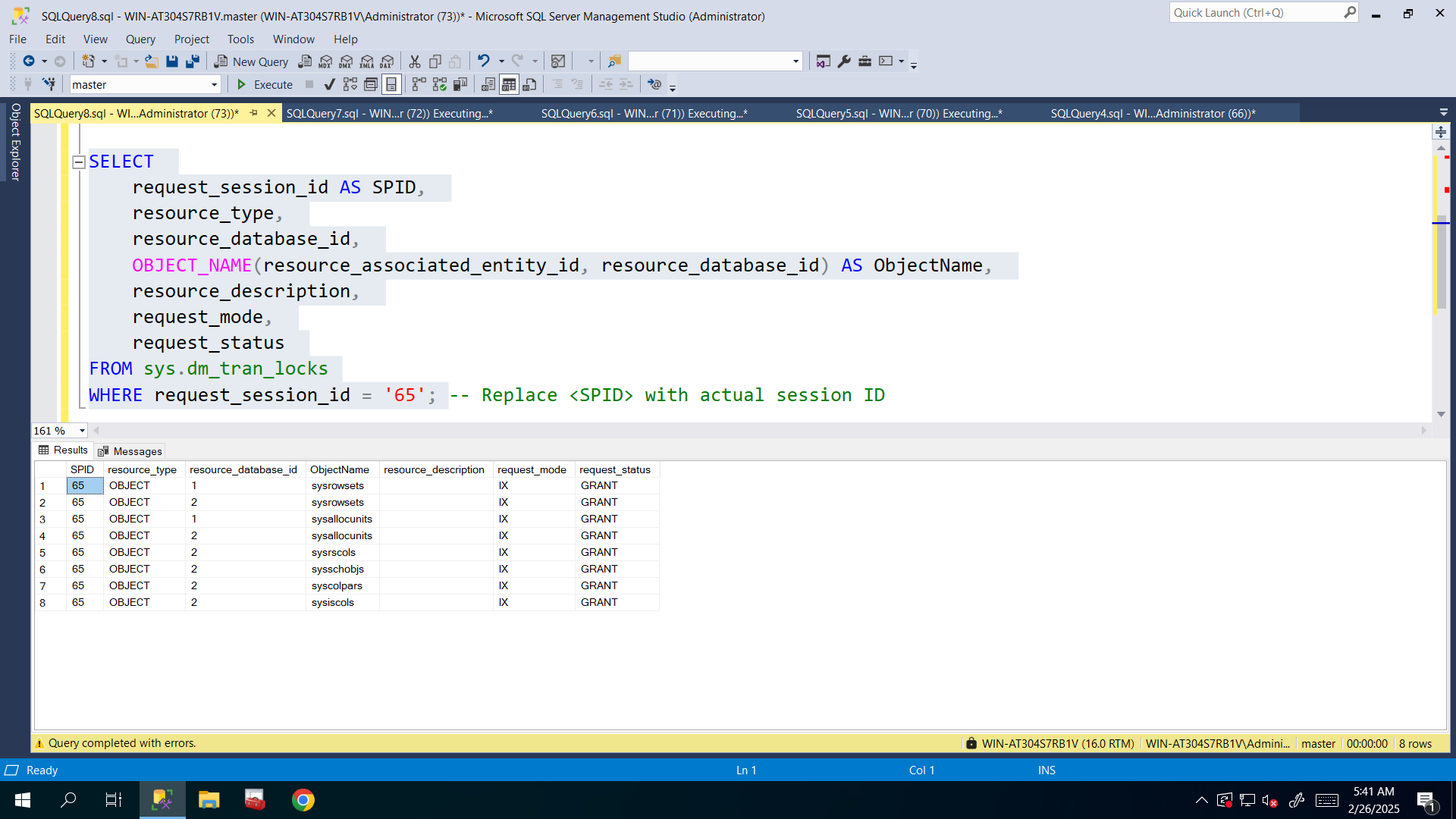
**72** session is blocked by **70** session and waiting to acquire **Sch-S** lock

\_\_\_\_ session is blocked by \_\_\_\_ session and waiting to acquire \_\_\_\_ lock

A session wants to acquire a **Schema Stability (Sch-S)** lock but is **waiting** due to a conflicting **Schema Modification (Sch-M)** lock in 65 session and others are waiting in queue.

**IX** here means **Intent Exclusive lock type for 65 SPID session.**

In that session, I truncated A table, but didn’t commit it I just began transaction, so it locks object until the transaction committed:



Bonus Topic:

**What is difference between sp\_who and sp\_who2 system stored procedures?**

**Key Differences**

1. **sp\_who**: Provides basic session details such as SPID, login name, and database.
2. **sp\_who2**: Offers more information, including **CPU time, disk IO, program name, and blocking details**, making it more useful for performance monitoring.

**Which One Should You Use?**

* Use sp\_who2 for **performance troubleshooting** since it includes CPU and IO stats.
* Use sp\_who if you need **basic session information** or are working on an older SQL Server.

**So as I understood sp\_who2 provides more and detailed information rather than sp\_who that provides basic information.**

**What is difference between TRUNCATE TABLE employees and DELETE FROM employees?**

Use TRUNCATE TABLE when you need to quickly **delete all rows** and **reset identity values**, and **no foreign keys** exist (**fails if FK exists**).

Use DELETE FROM when you need to **delete specific rows** or want **triggers to fire**.