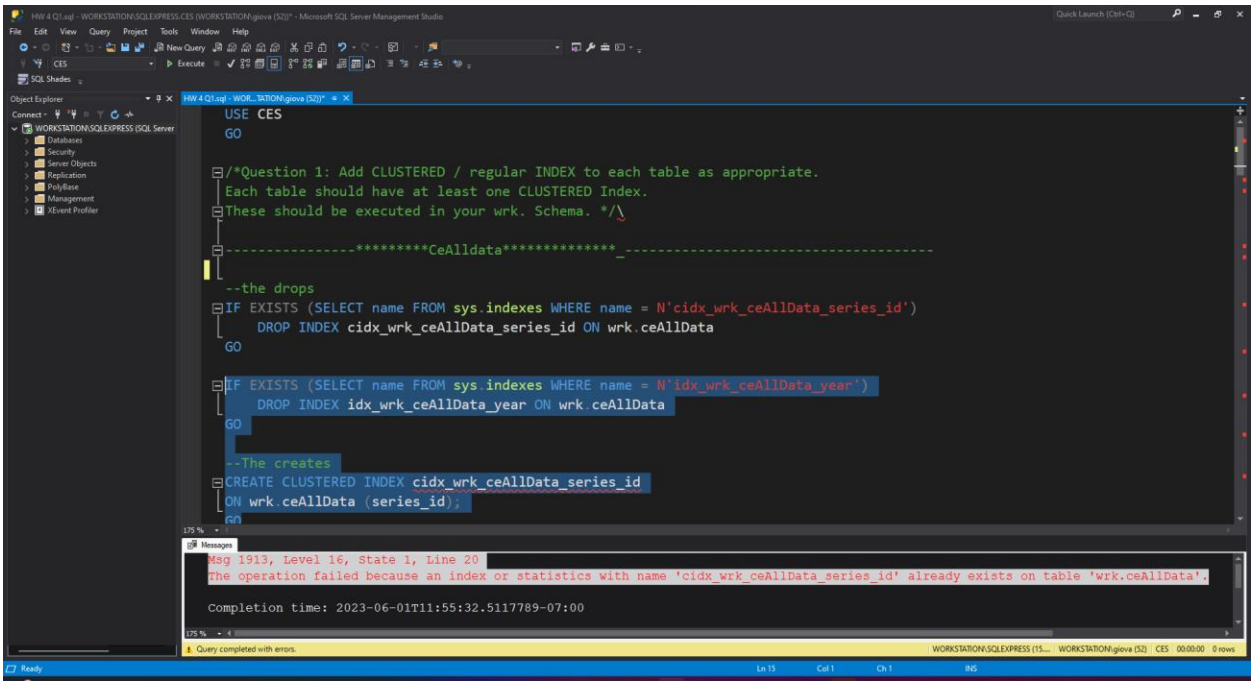
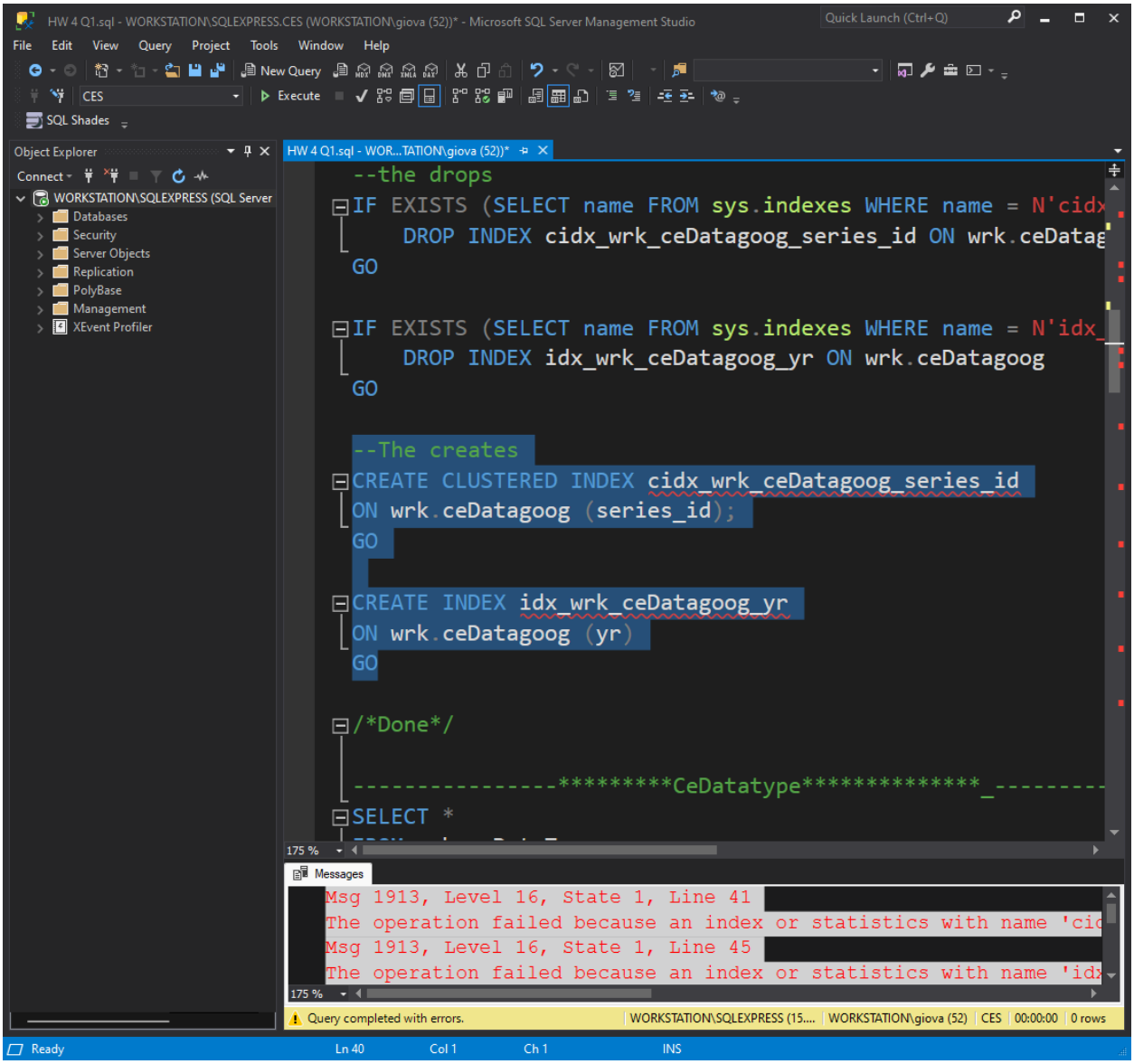
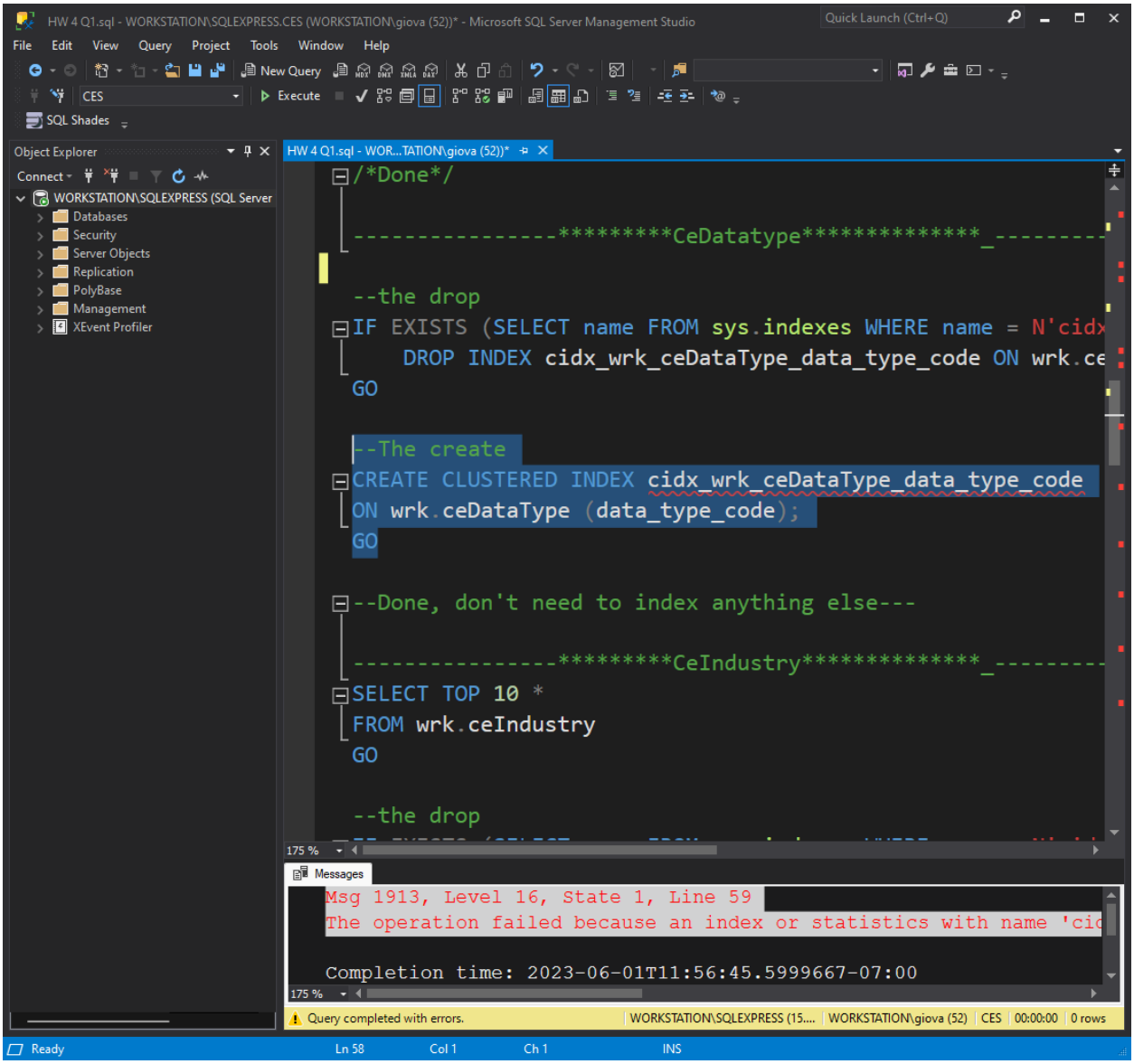
```
USE CES
GO

/*Question 1: Add CLUSTERED / regular
INDEX to each table as appropriate.
Each table should have at least one
CLUSTERED Index.
These should be executed in your wrk.
Schema. */\

----------*********CeAlldata**********_-------------------------------------

--checking
SELECT top 10*
FROM wrk.ceAllData
GO

--the drops
IF EXISTS (SELECT name FROM sys.indexes WHERE name = N'cidx_wrk_ceAllData_series_id')
    DROP INDEX cidx_wrk_ceAllData_series_id ON wrk.ceAllData
GO

IF EXISTS (SELECT name FROM sys.indexes WHERE name = N'idx_wrk_ceAllData_year')
    DROP INDEX idx_wrk_ceAllData_year ON wrk.ceAllData
GO

--The creates
CREATE CLUSTERED INDEX cidx_wrk_ceAllData_series_id
ON wrk.ceAllData (series_id);
GO

CREATE INDEX idx_wrk_ceAllData_year
ON wrk.ceAlldata (year)
GO
/*Done*/
```



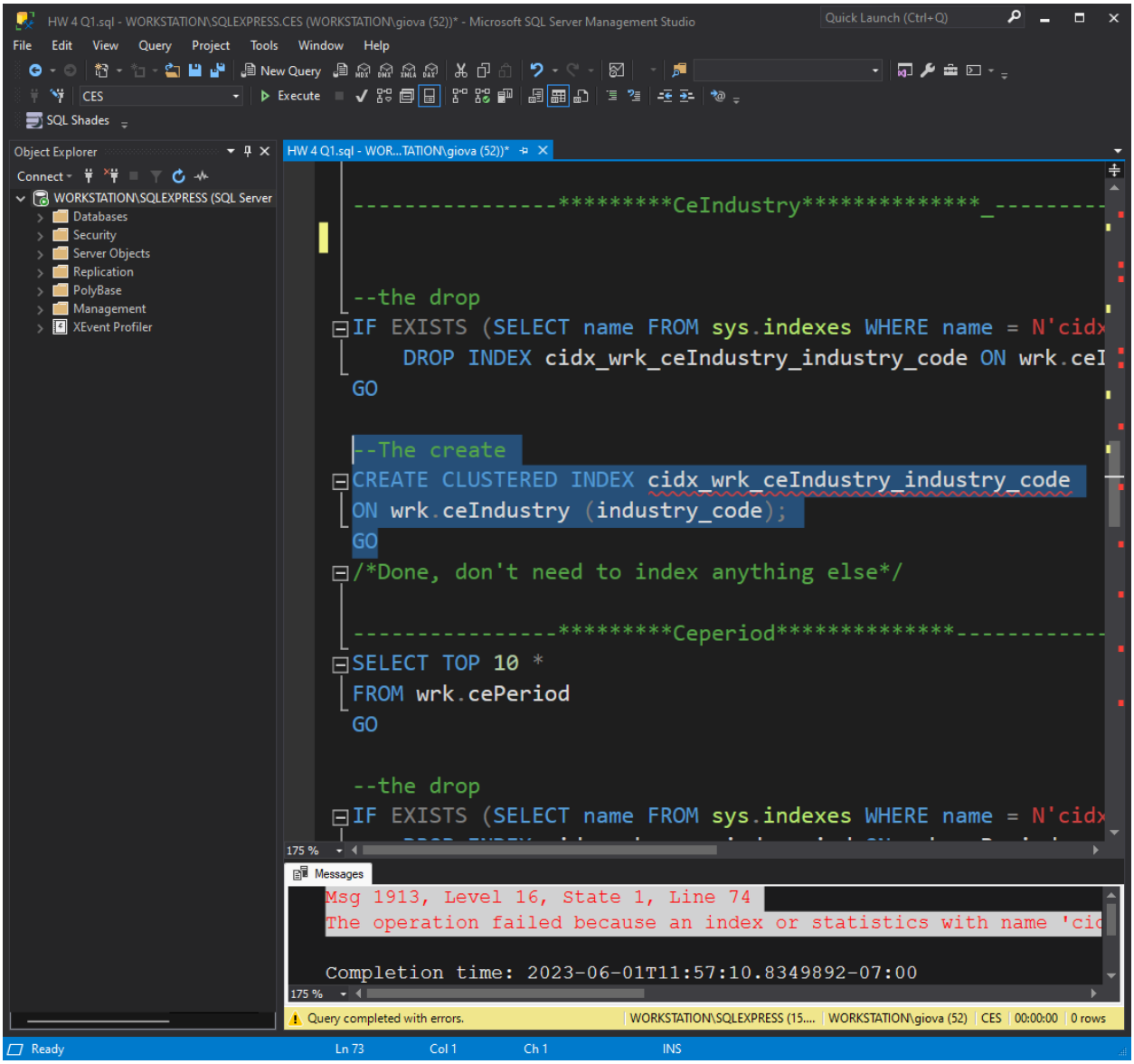

```
----------********CeDatagoog**********_----------
SELECT *
FROM wrk.ceDatagoog
GO

--the drops
IF EXISTS (SELECT name FROM sys.indexes WHERE name = N'cidx_wrk_ceDatagoog_series_id')
    DROP INDEX cidx_wrk_ceDatagoog_series_id ON wrk.ceDatagoog
GO

IF EXISTS (SELECT name FROM sys.indexes WHERE name = N'idx_wrk_ceDatagoog_yr')
    DROP INDEX idx_wrk_ceDatagoog_yr ON wrk.ceDatagoog
GO

--The creates
CREATE CLUSTERED INDEX cidx_wrk_ceDatagoog_series_id
ON wrk.ceDatagoog (series_id);
GO

CREATE INDEX idx_wrk_ceDatagoog_yr
ON wrk.ceDatagoog (yr)
GO

/*Done*/
```



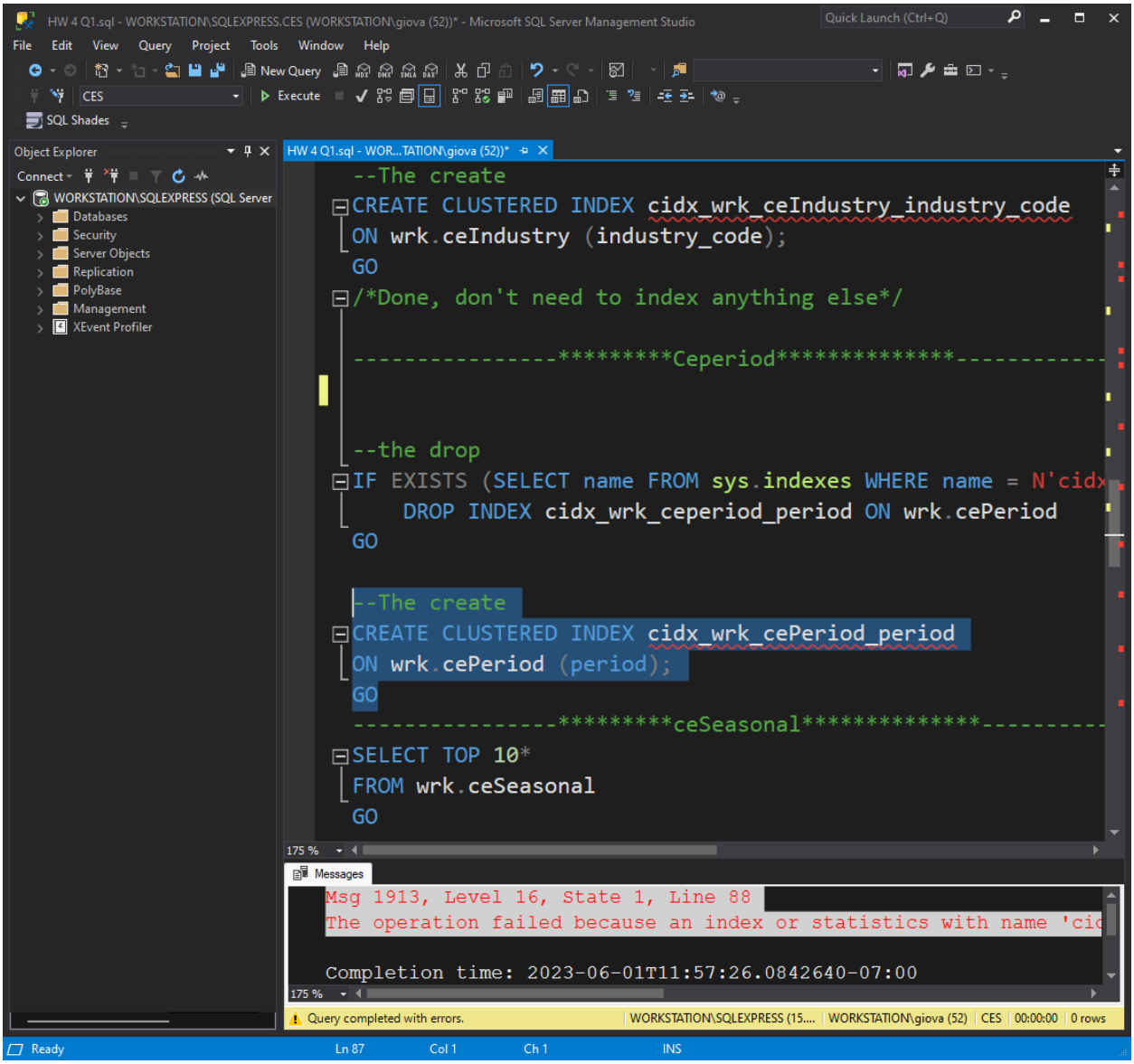

```
----------*********CeDatatype**********_-----------------------------------
SELECT *
FROM wrk.ceDataType
GO

--the drop
IF EXISTS (SELECT name FROM sys.indexes WHERE name =
N'cidx_wrk_ceDataType_data_type_code')
    DROP INDEX cidx_wrk_ceDataType_data_type_code ON wrk.ceDataType
GO

--The create
CREATE CLUSTERED INDEX cidx_wrk_ceDataType_data_type_code
ON wrk.ceDataType (data_type_code);
GO

--Done, don't need to index anything else---
```



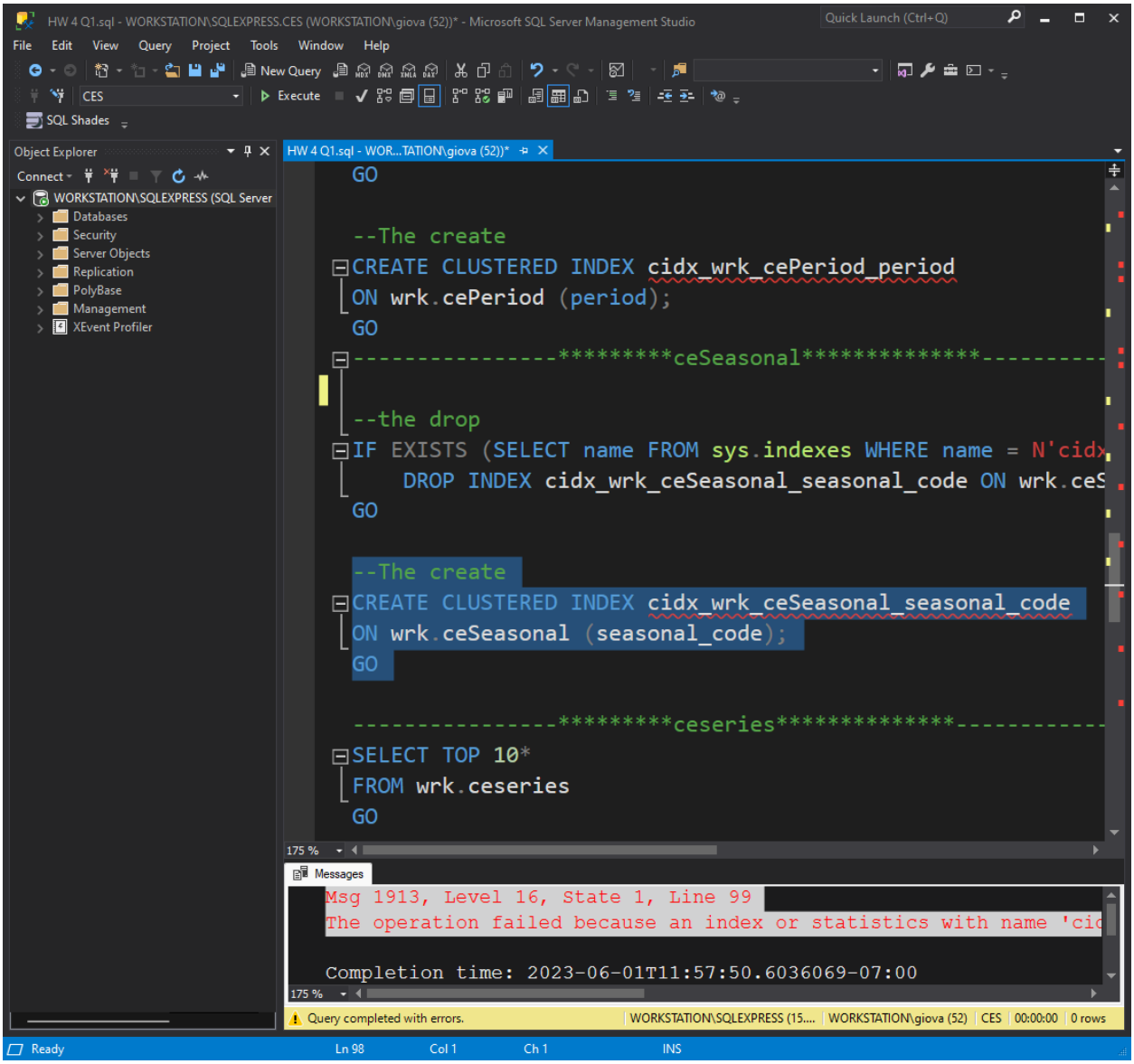

```
----------*********CeIndustry**********_-----------------------------------

--the drop
IF EXISTS (SELECT name FROM sys.indexes WHERE name =
N'cidx_wrk_ceIndustry_industry_code')
    DROP INDEX cidx_wrk_ceIndustry_industry_code ON wrk.ceIndustry
GO

--The create
CREATE CLUSTERED INDEX cidx_wrk_ceIndustry_industry_code
ON wrk.ceIndustry (industry_code);
GO
/*Done, don't need to index anything else*/
```



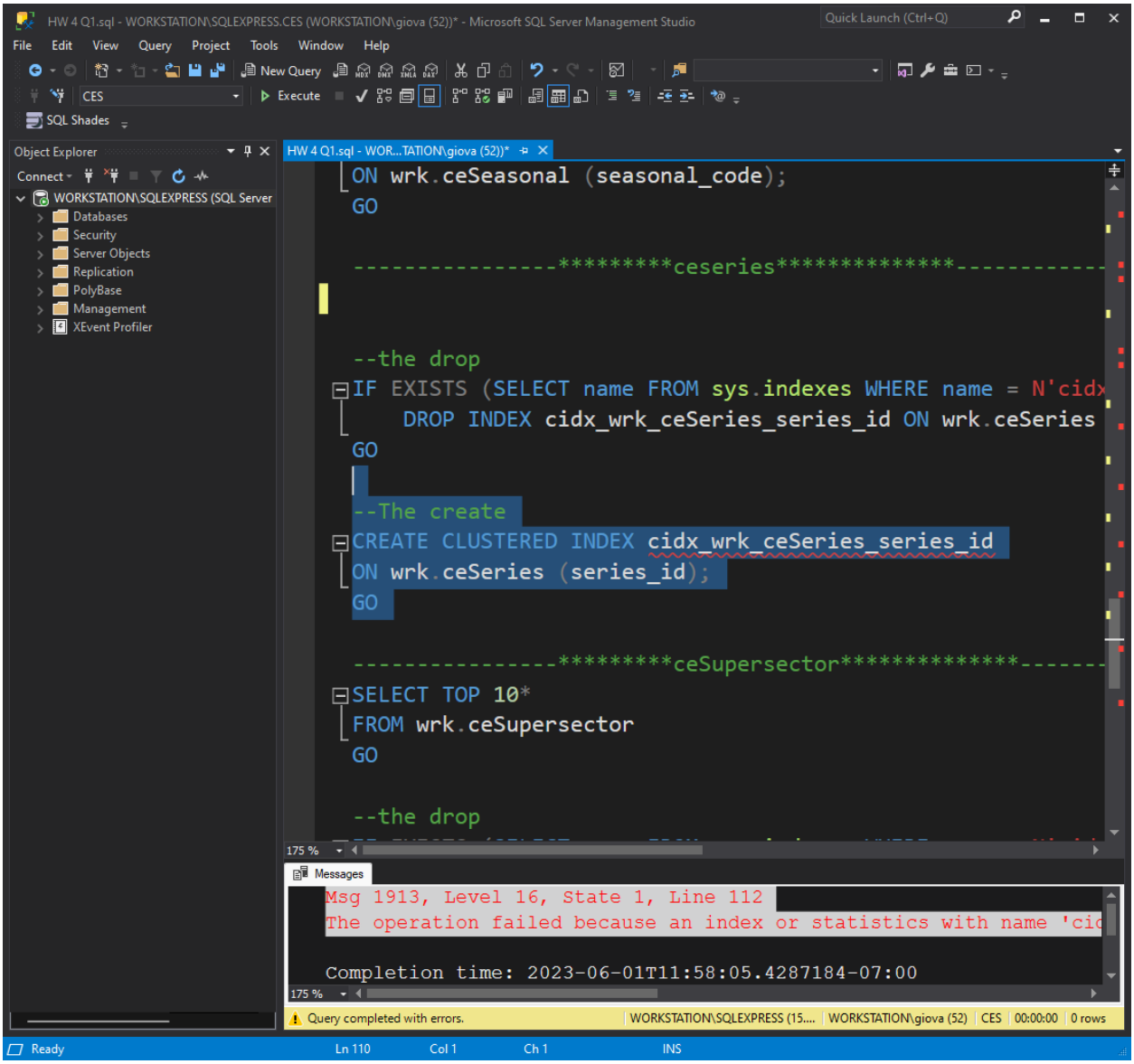

```
----------*********Ceperiod**********-----------------------------------

--the drop
IF EXISTS (SELECT name FROM sys.indexes WHERE name = N'cidx_wrk_cePeriod_period')
    DROP INDEX cidx_wrk_ceperiod_period ON wrk.cePeriod
GO

--The create
CREATE CLUSTERED INDEX cidx_wrk_cePeriod_period
ON wrk.cePeriod (period);
GO
```




```
----------*********ceSeasonal**********----------

--the drop
IF EXISTS (SELECT name FROM sys.indexes WHERE name =
N'cidx_wrk_ceSeasonal_seasonal_code')
    DROP INDEX cidx_wrk_ceSeasonal_seasonal_code ON wrk.ceSeasonal
GO

--The create
CREATE CLUSTERED INDEX cidx_wrk_ceSeasonal_seasonal_code
ON wrk.ceSeasonal (seasonal_code);
GO
```




```
----------*********ceseries**********----------

--the drop
IF EXISTS (SELECT name FROM sys.indexes WHERE name = N'cidx_wrk_ceSeries_series_id')
    DROP INDEX cidx_wrk_ceSeries_series_id ON wrk.ceSeries
GO

--The create
CREATE CLUSTERED INDEX cidx_wrk_ceSeries_series_id
ON wrk.ceSeries (series_id);
GO
```




```
----------*********ceSupersector**********----------------------------------
-
SELECT TOP 10*
FROM wrk.ceSupersector
GO

--the drop
IF EXISTS (SELECT name FROM sys.indexes WHERE name =
N'cidx_wrk_ceSupersector_supersector_code')
    DROP INDEX cidx_wrk_ceSupersector_supersector_code ON wrk.ceSupersector
GO

--The create
CREATE CLUSTERED INDEX cidx_wrk_ceSupersector_supersector_code
ON wrk.ceSupersector (supersector_code);
GO
```

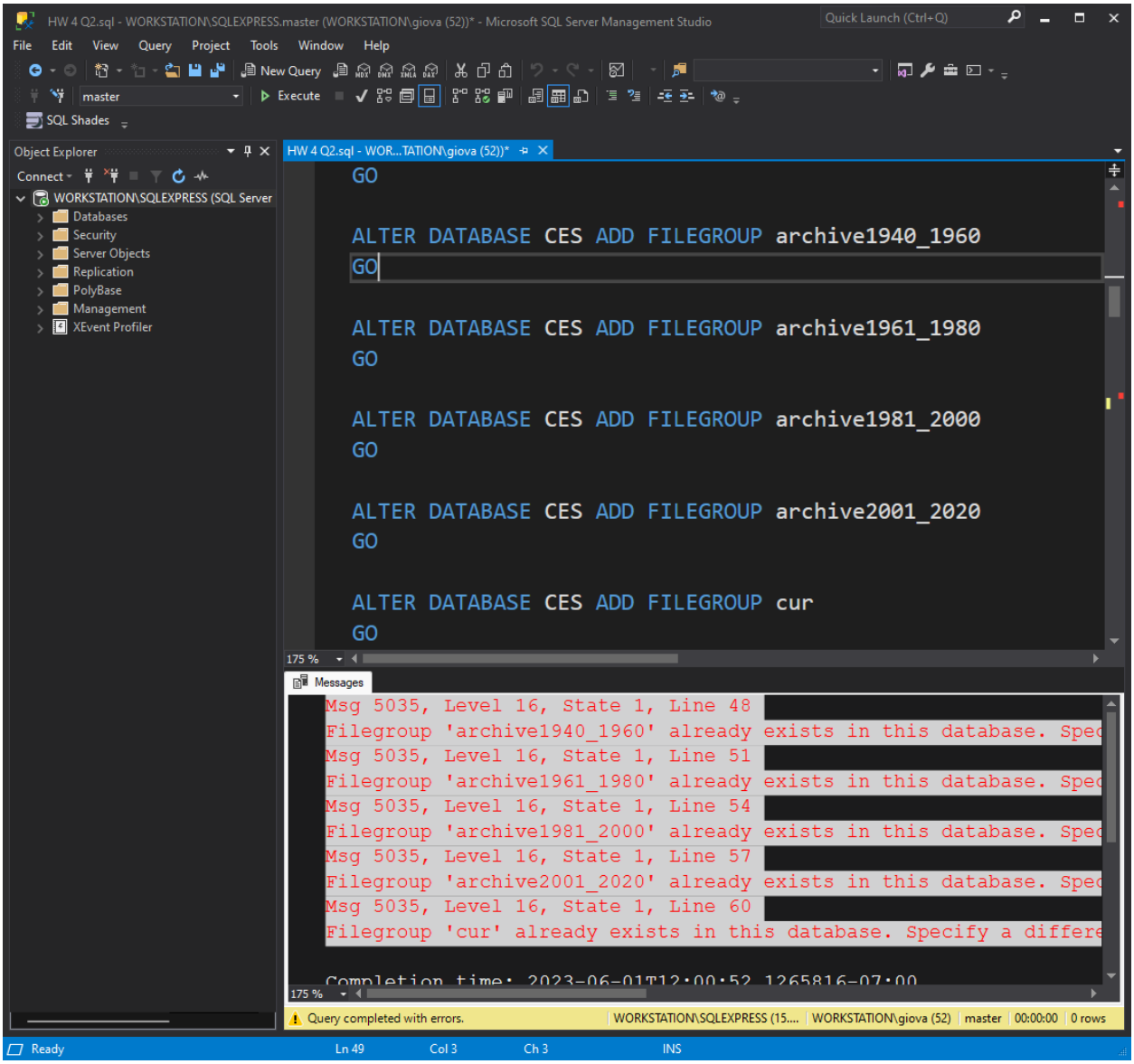
```
/*
                ****END OF QUESTION 1***
*/
```

```
USE CES
```

```
/*Question 2:Create partitions that
segment the ceAllData by 20 year periods
from 1940 to 2020.
Store your new ceAllData table into a new
schema called par that has the partitions
properly executed. */
```

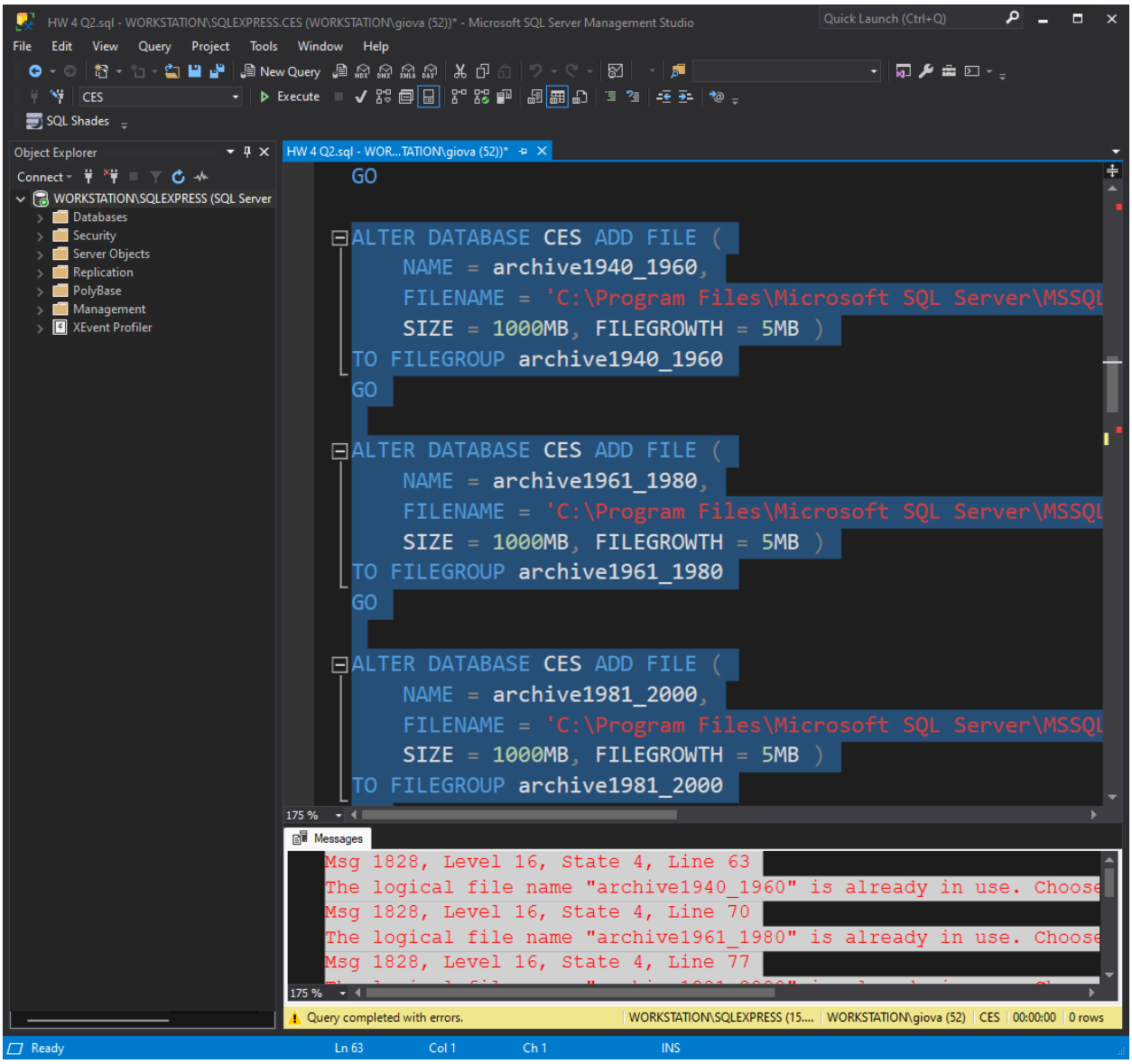
```
CREATE SCHEMA par
GO

DROP TABLE par.ceAllData
GO

DROP PARTITION SCHEME myRange
GO
```

```
DROP PARTITION FUNCTION myRangeFun
GO

ALTER DATABASE CES REMOVE FILE archive1940_1960
GO

ALTER DATABASE CES REMOVE FILE archive1961_1980
GO

ALTER DATABASE CES REMOVE FILE archive1981_2000
GO

ALTER DATABASE CES REMOVE FILE archive2001_2020
GO

ALTER DATABASE CES REMOVE FILE cur
GO

ALTER DATABASE CES REMOVE FILEGROUP archive1940_1960
GO

ALTER DATABASE CES REMOVE FILEGROUP archive1961_1980
GO

ALTER DATABASE CES REMOVE FILEGROUP archive1981_2000
GO

ALTER DATABASE CES REMOVE FILEGROUP archive2001_2020
GO

ALTER DATABASE CES REMOVE FILEGROUP cur
GO



ALTER DATABASE CES ADD FILEGROUP archive1940_1960
GO

ALTER DATABASE CES ADD FILEGROUP archive1961_1980
GO

ALTER DATABASE CES ADD FILEGROUP archive1981_2000
GO

ALTER DATABASE CES ADD FILEGROUP archive2001_2020
GO

ALTER DATABASE CES ADD FILEGROUP cur
GO
```



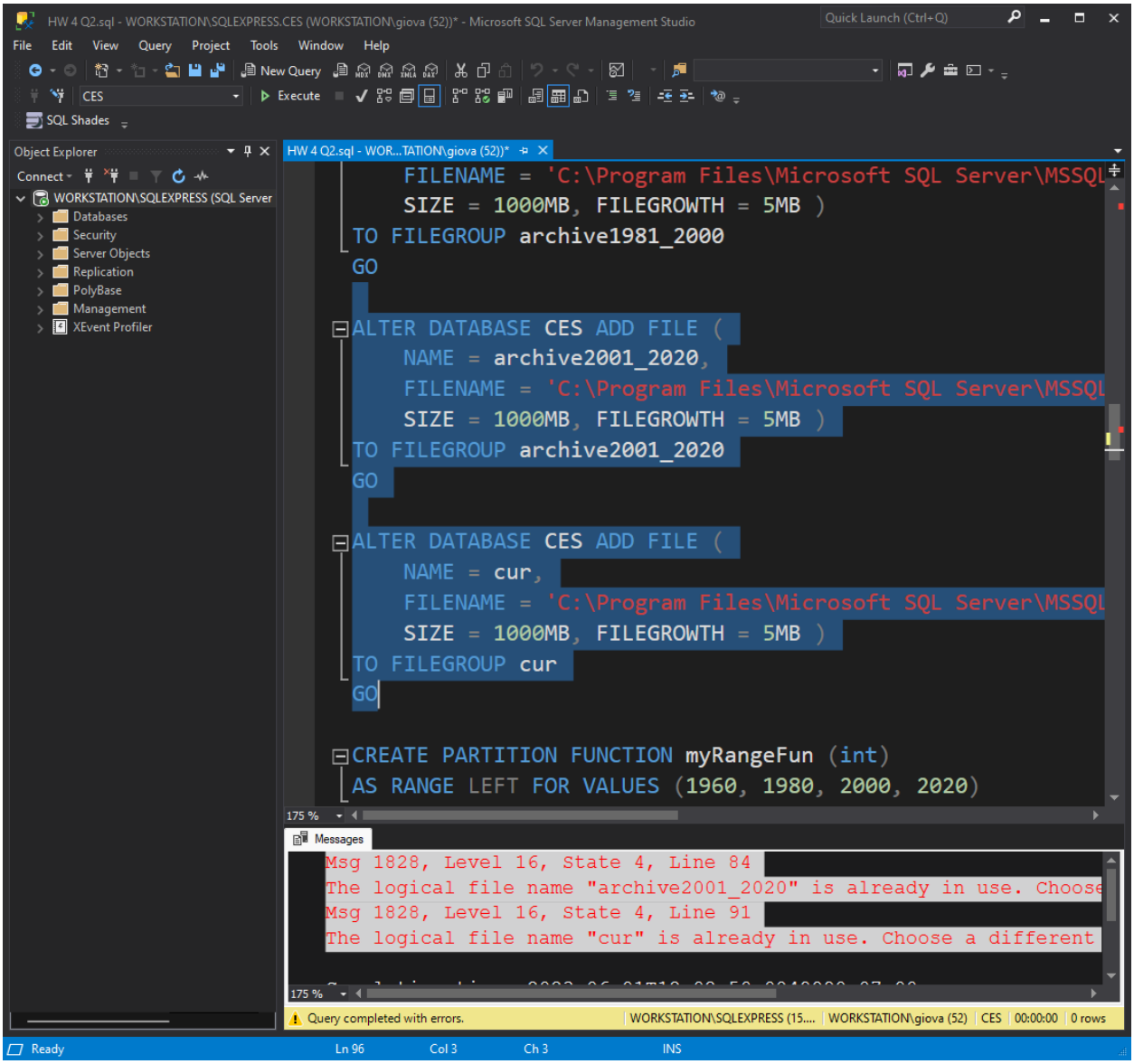

```
ALTER DATABASE CES ADD FILE (
    NAME = archive1940_1960,
    FILENAME = 'C:\Program Files\Microsoft SQL
Server\MSSQL15.SQLEXPRESS\MSSQL\DATA\archive1940_1960.ndf',
    SIZE = 1000MB, FILEGROWTH = 5MB )
TO FILEGROUP archive1940_1960
GO

ALTER DATABASE CES ADD FILE (
    NAME = archive1961_1980,
    FILENAME = 'C:\Program Files\Microsoft SQL
Server\MSSQL15.SQLEXPRESS\MSSQL\DATA\archive1961_1980.ndf',
    SIZE = 1000MB, FILEGROWTH = 5MB )
TO FILEGROUP archive1961_1980
GO

ALTER DATABASE CES ADD FILE (
```

```
    NAME = archive1981_2000,
    FILENAME = 'C:\Program Files\Microsoft SQL
Server\MSSQL15.SQLEXPRESS\MSSQL\DATA\archive1981_2000.ndf',
    SIZE = 1000MB, FILEGROWTH = 5MB )
TO FILEGROUP archive1981_2000
GO
```

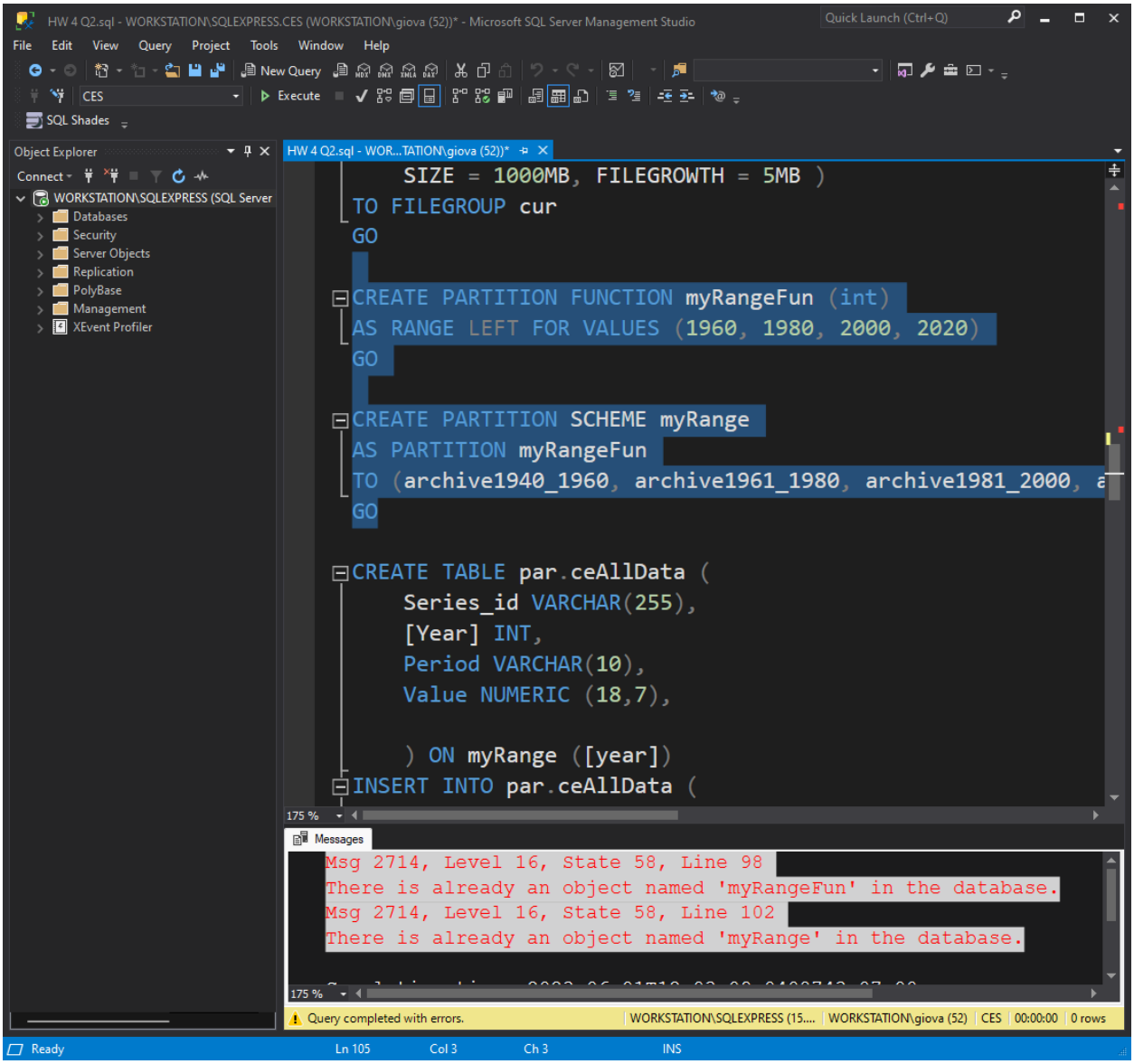



```
ALTER DATABASE CES ADD FILE (
    NAME = archive2001_2020,
    FILENAME = 'C:\Program Files\Microsoft SQL
Server\MSSQL15.SQLEXPRESS\MSSQL\DATA\archive2001_2020.ndf',
    SIZE = 1000MB, FILEGROWTH = 5MB )
TO FILEGROUP archive2001_2020
GO
```

```
ALTER DATABASE CES ADD FILE (
    NAME = cur,
    FILENAME = 'C:\Program Files\Microsoft SQL
Server\MSSQL15.SQLEXPRESS\MSSQL\DATA\cur.ndf',
    SIZE = 1000MB, FILEGROWTH = 5MB )
TO FILEGROUP cur
GO
```

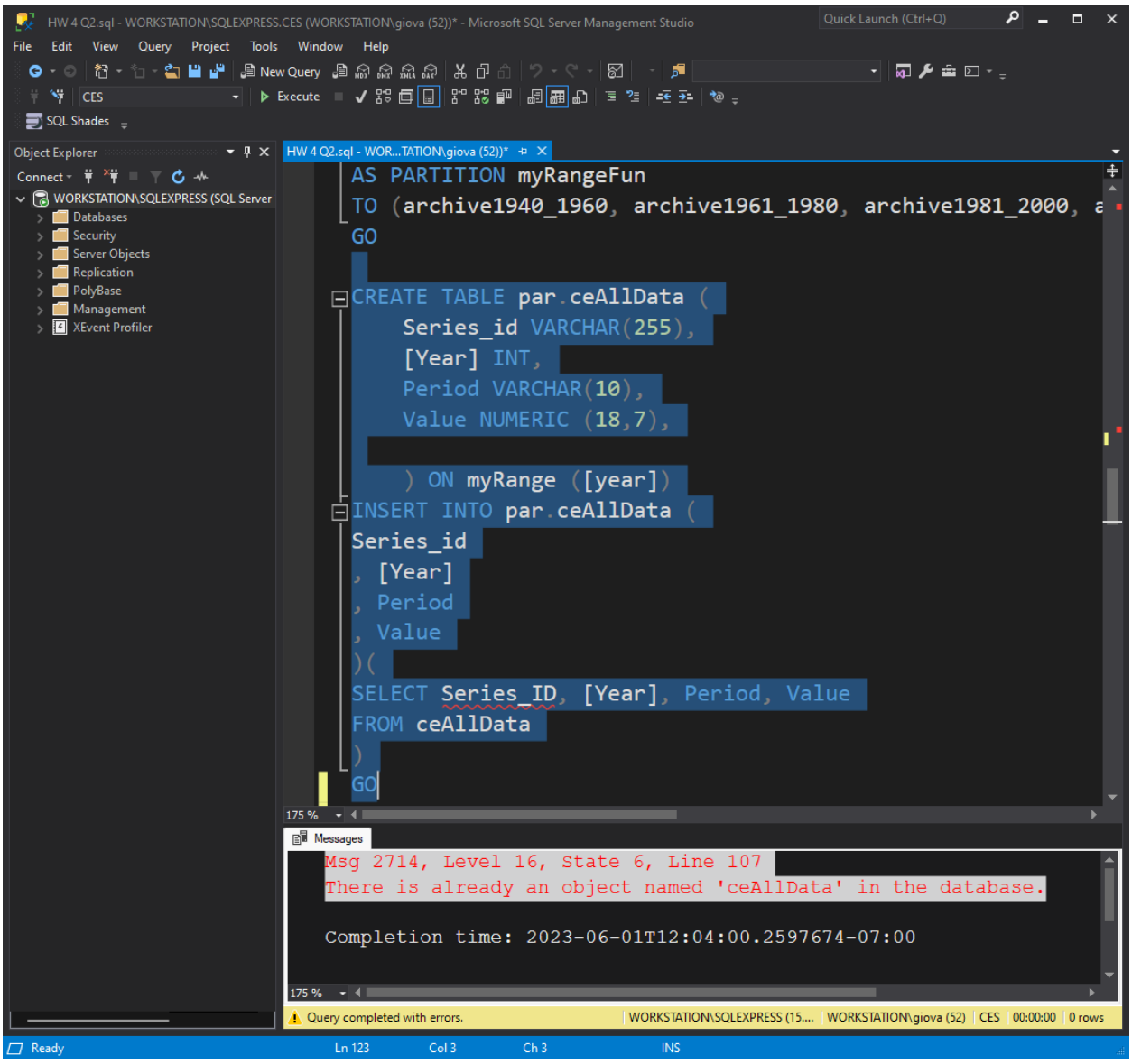



```
CREATE PARTITION FUNCTION myRangeFun (int)
AS RANGE LEFT FOR VALUES (1960, 1980, 2000, 2020)
GO

CREATE PARTITION SCHEME myRange
AS PARTITION myRangeFun
TO (archive1940_1960, archive1961_1980, archive1981_2000, archive2001_2020, cur)
GO
```

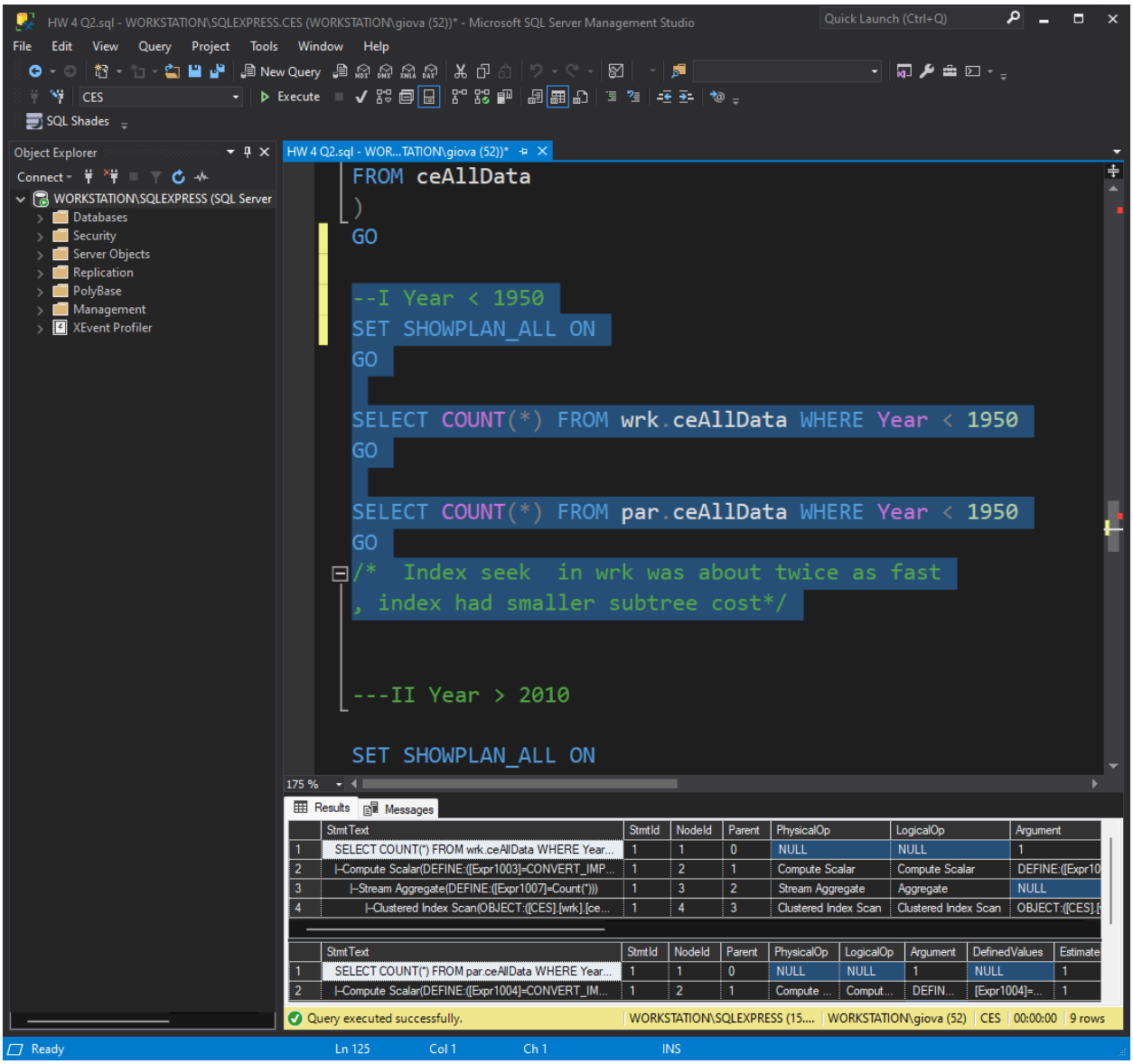



```
CREATE TABLE par.ceAllData (
    Series_id VARCHAR(255),
    [Year] INT,
    Period VARCHAR(10),
    Value NUMERIC (18,7),

      ) ON myRange ([year])
INSERT INTO par.ceAllData (
Series_id
, [Year]
, Period
, Value
)(
SELECT Series_ID, [Year], Period, Value
FROM ceAllData
)
```

GO

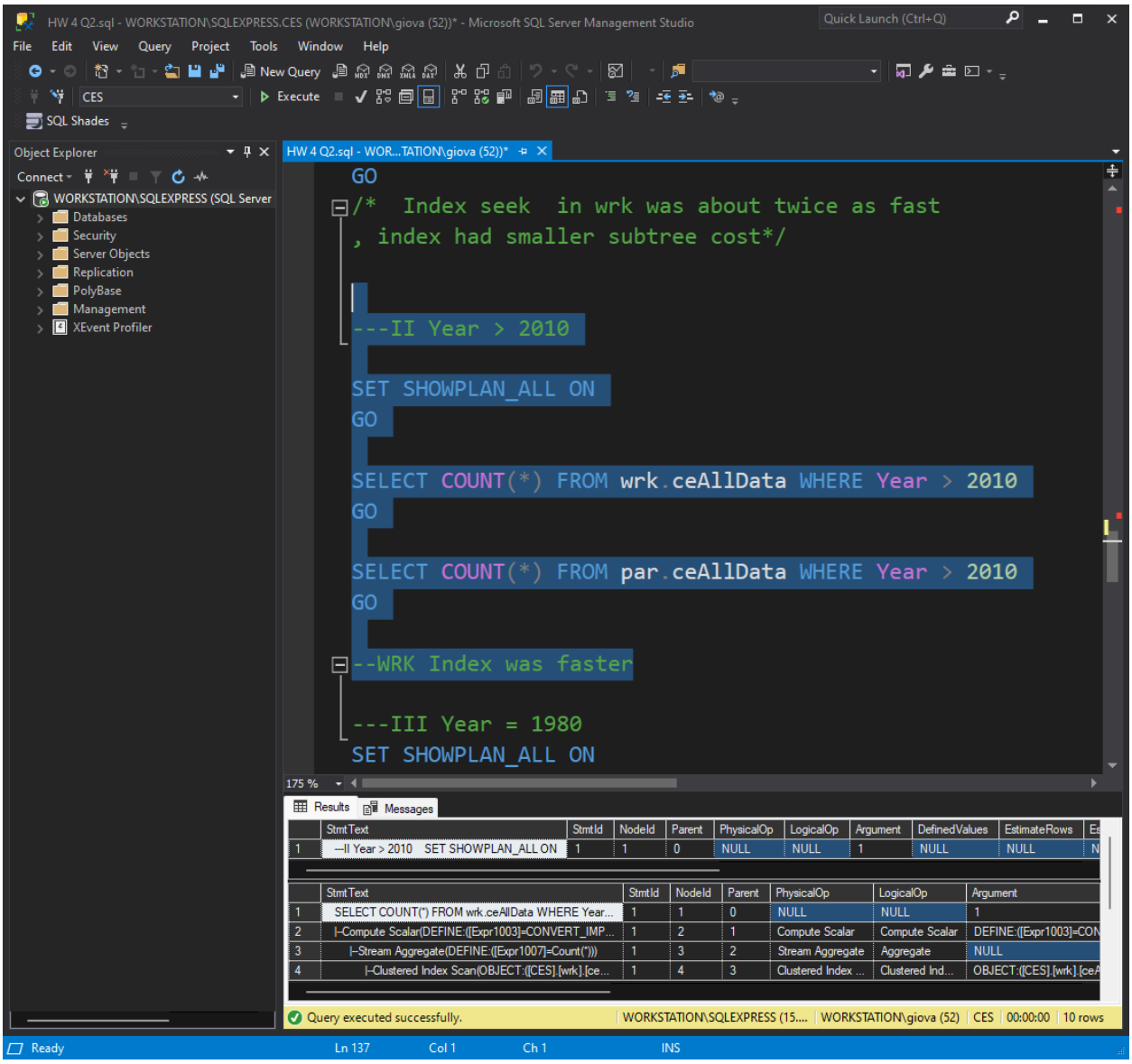

HW 4 Q2.sql - WORKSTATION\SQLEXPRESS.CES (WORKSTATION\giova (52))* - Microsoft SQL Server Management Studio
Quick Launch (Ctrl+Q)
File Edit View Query Project Tools Window Help
New Query
Execute
CES
SQL Shades
Object Explorer
Connect
WORKSTATION\SQLEXPRESS (SQL Server
Databases
Security
Server Objects
Replication
PolyBase
Management
XEvent Profiler
HW 4 Q2.sql - WOR...TATION\giova (52))*
AS PARTITION myRangeFun
TO (archive1940_1960, archive1961_1980, archive1981_2000, a
GO
CREATE TABLE par.ceAllData (
Series_id VARCHAR(255),
[Year] INT,
Period VARCHAR(10),
Value NUMERIC (18,7),
) ON myRange ([year])
INSERT INTO par.ceAllData (
Series_id
, [Year]
, Period
, Value
)(
SELECT Series_ID, [Year], Period, Value
FROM ceAllData
)
GO
175 %
Messages
Msg 2714, Level 16, State 6, Line 107
There is already an object named 'ceAllData' in the database.
Completion time: 2023-06-01T12:04:00.2597674-07:00
175 %
Query completed with errors.
WORKSTATION\SQLEXPRESS (15....
WORKSTATION\giova (52)
CES
00:00:00
0 rows
Ready
Ln 123
Col 3
Ch 3
INS

```
--I Year < 1950
SET SHOWPLAN_ALL ON
GO

SELECT COUNT(*) FROM wrk.ceAllData WHERE Year < 1950
GO

SELECT COUNT(*) FROM par.ceAllData WHERE Year < 1950
GO
/*  Index seek  in wrk was about twice as fast
, index had smaller subtree cost*/
```



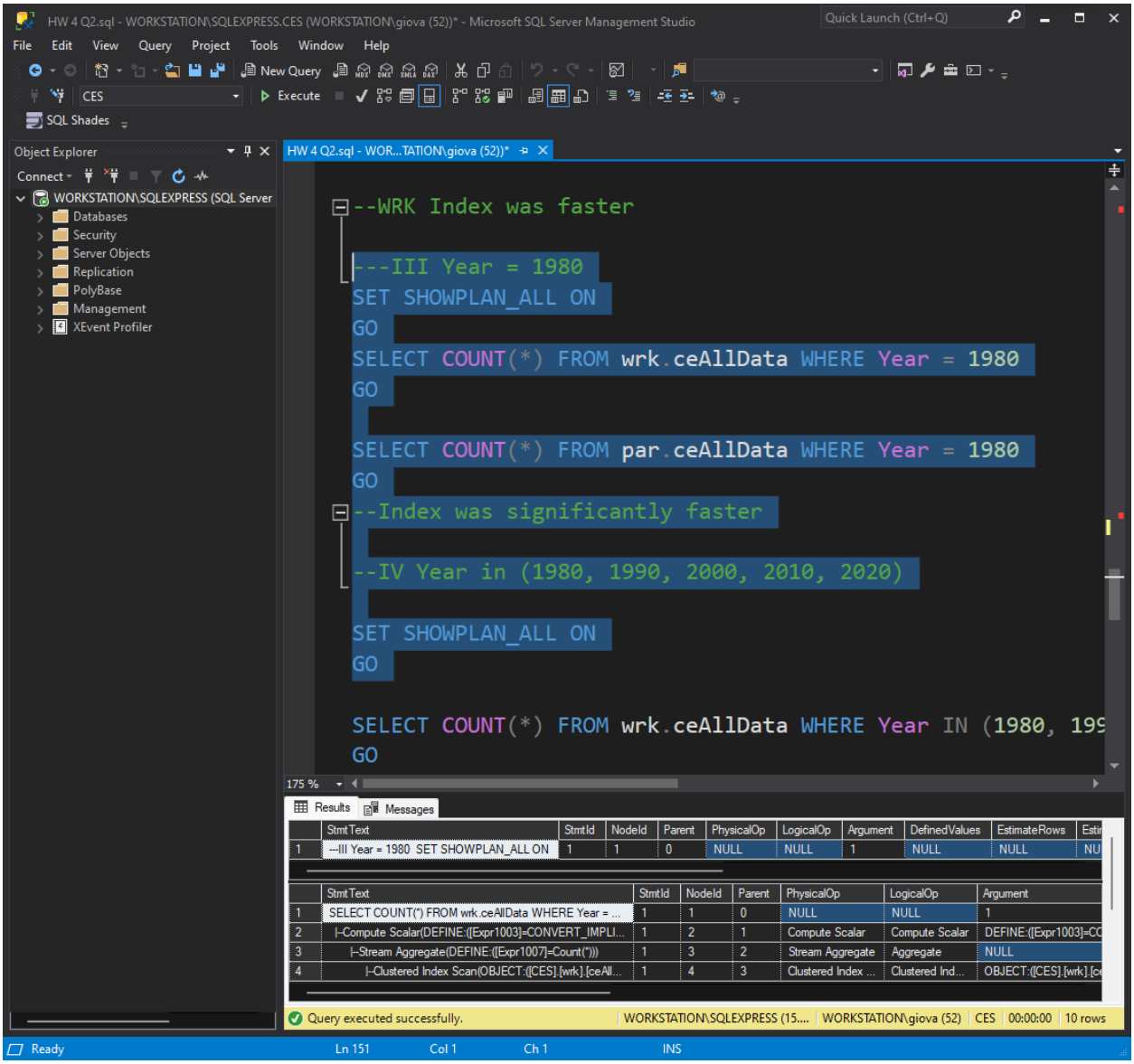

```
---II Year > 2010

SET SHOWPLAN_ALL ON
```

```
GO

SELECT COUNT(*) FROM wrk.ceAllData WHERE Year > 2010
GO

SELECT COUNT(*) FROM par.ceAllData WHERE Year > 2010
GO

--WRK Index was faster
```



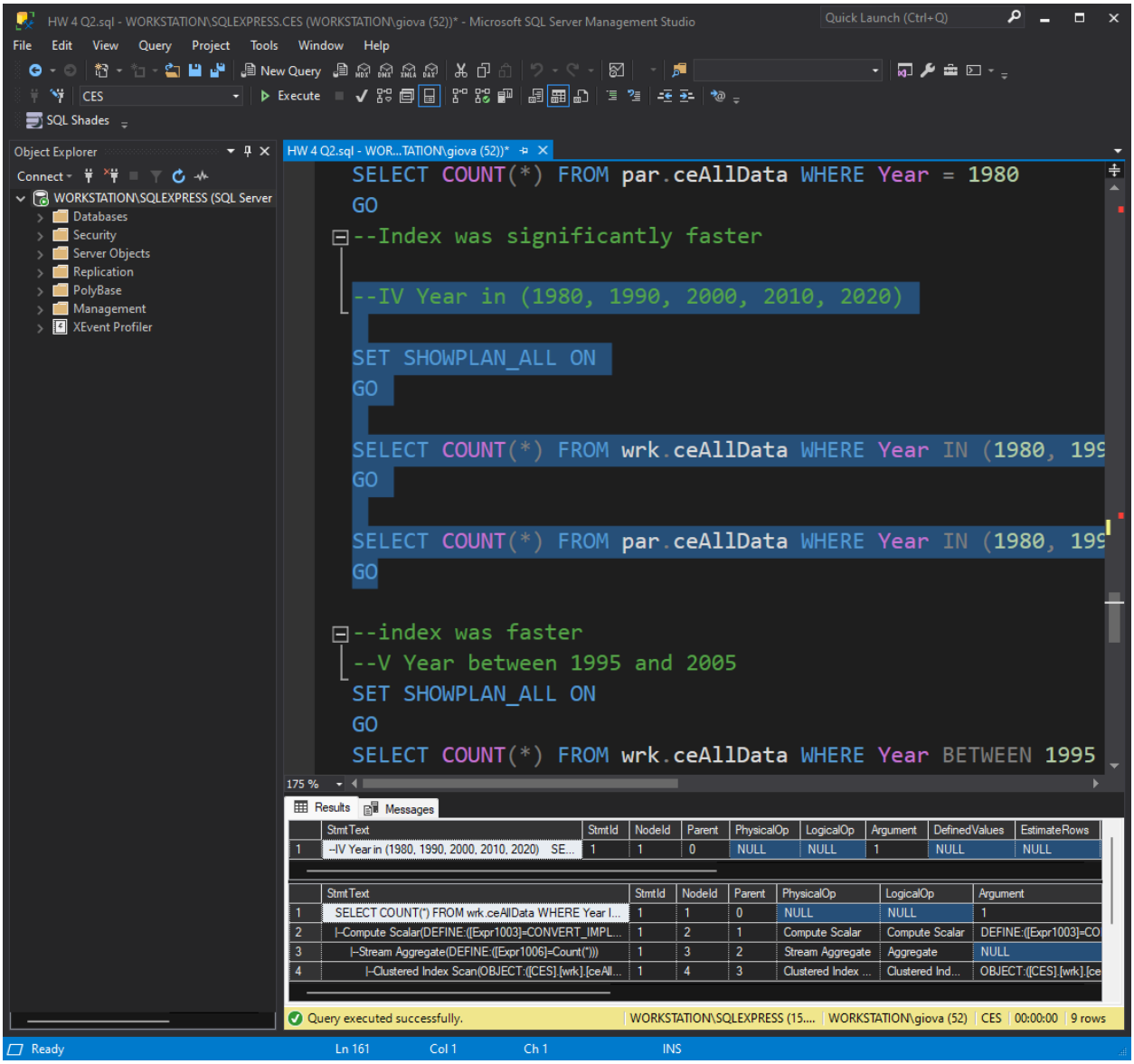

```
---III Year = 1980
SET SHOWPLAN_ALL ON
GO
SELECT COUNT(*) FROM wrk.ceAllData WHERE Year = 1980
```

```
GO

SELECT COUNT(*) FROM par.ceAllData WHERE Year = 1980
GO
--Index was significantly faster
```

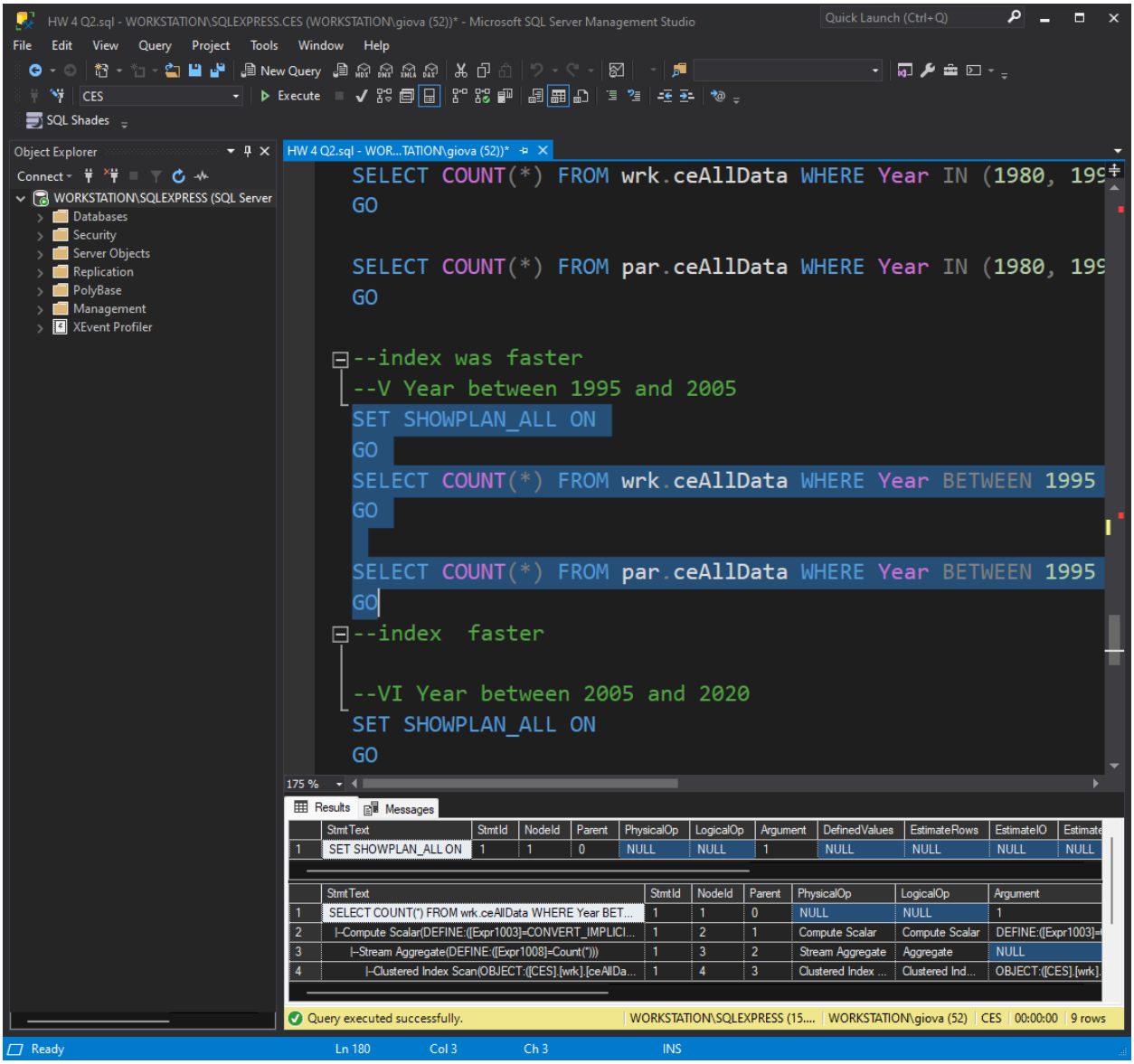



```
--IV Year in (1980, 1990, 2000, 2010, 2020)

SET SHOWPLAN_ALL ON
GO

SELECT COUNT(*) FROM wrk.ceAllData WHERE Year IN (1980, 1990, 2000, 2010, 2020)
GO

SELECT COUNT(*) FROM par.ceAllData WHERE Year IN (1980, 1990, 2000, 2010, 2020)
GO
```

```
--index was faster
```



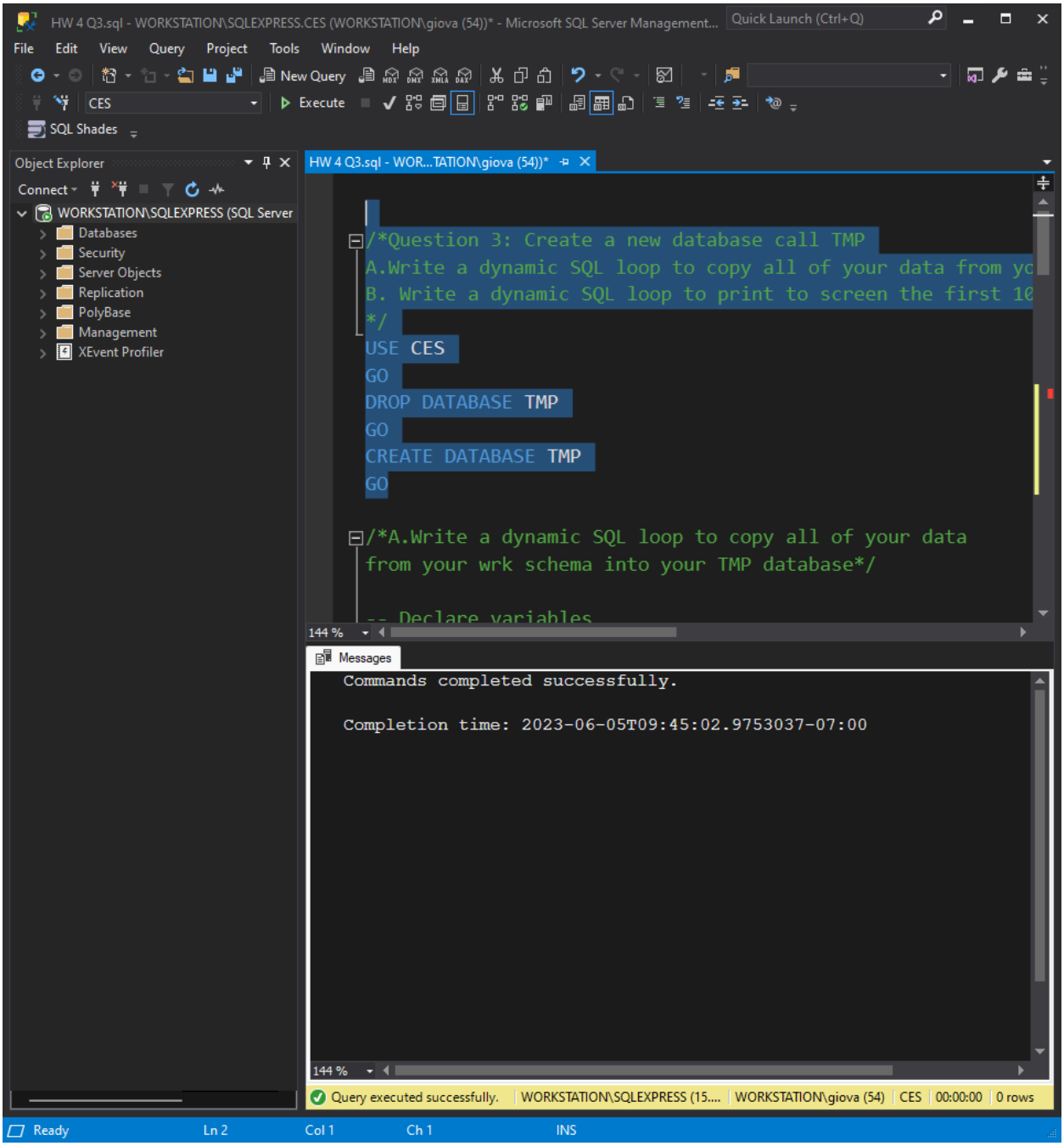

```
--V Year between 1995 and 2005
SET SHOWPLAN_ALL ON
GO
SELECT COUNT(*) FROM wrk.ceAllData WHERE Year BETWEEN 1995 AND 2005
GO

SELECT COUNT(*) FROM par.ceAllData WHERE Year BETWEEN 1995 AND 2005
GO
--index  faster
```




```
--VI Year between 2005 and 2020
SET SHOWPLAN_ALL ON
GO
SELECT COUNT(*) FROM wrk.ceAllData WHERE Year BETWEEN 2005 AND 2020
GO

SELECT COUNT(*) FROM par.ceAllData WHERE Year BETWEEN 2005 AND 2020
GO
--index faster
SET SHOWPLAN_ALL OFF
GO
```

```
/* Part B:
I'm not sure how much my PC performance affects the outcome,
```

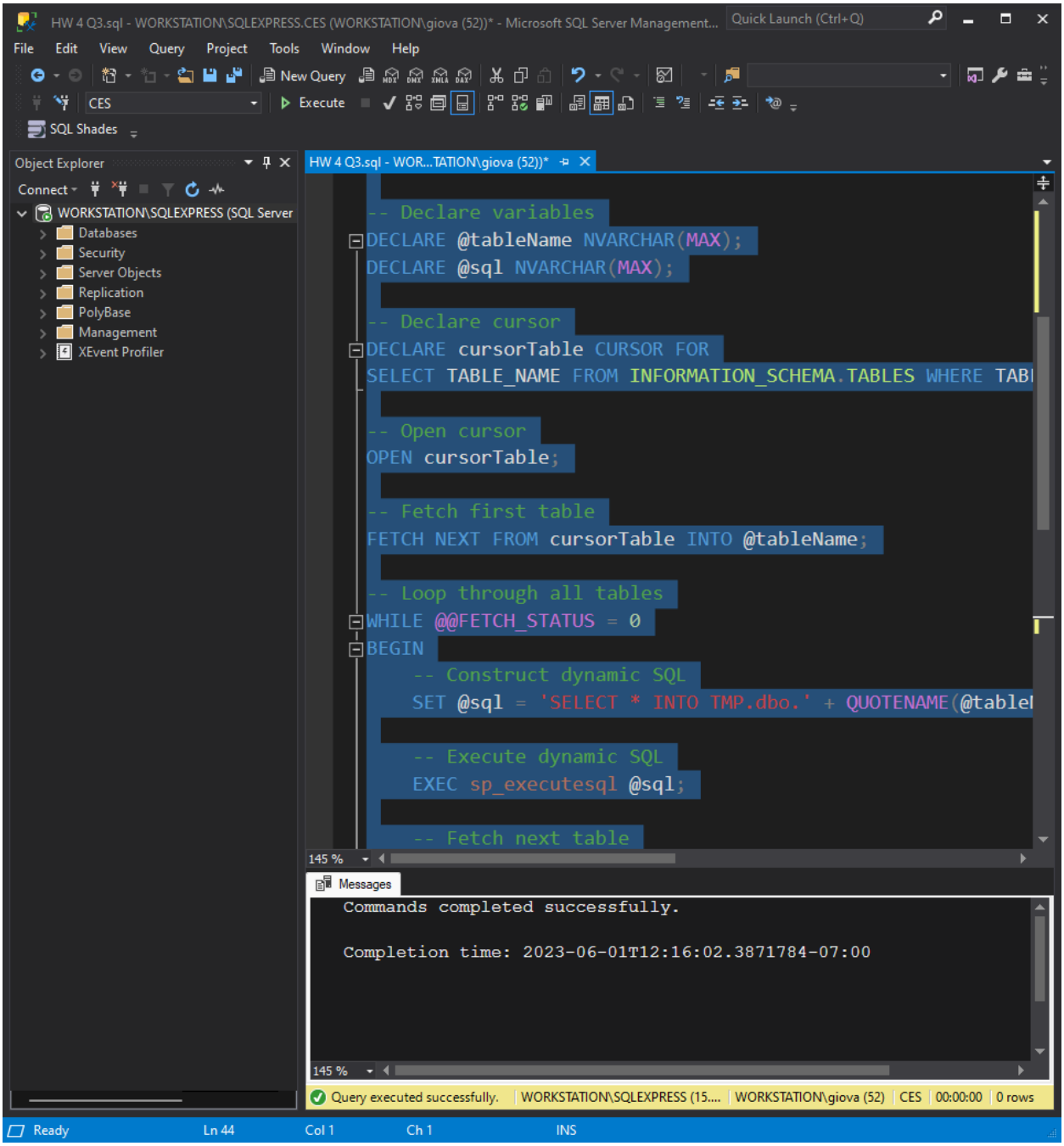
```
but my wrk.cealldata is always faster than the partitioned
query.If the total amount of data in the table is small,
the overhead of managing partitions could outweigh their
benefits. The operation starts with a Table Scan which is
usually less efficient than an index seek/scan because it
requires reading the whole table or partition. */

/*Question 3: Create a new database call TMP
USE CES
GO
DROP DATABASE TMP
GO
CREATE DATABASE TMP
GO
```

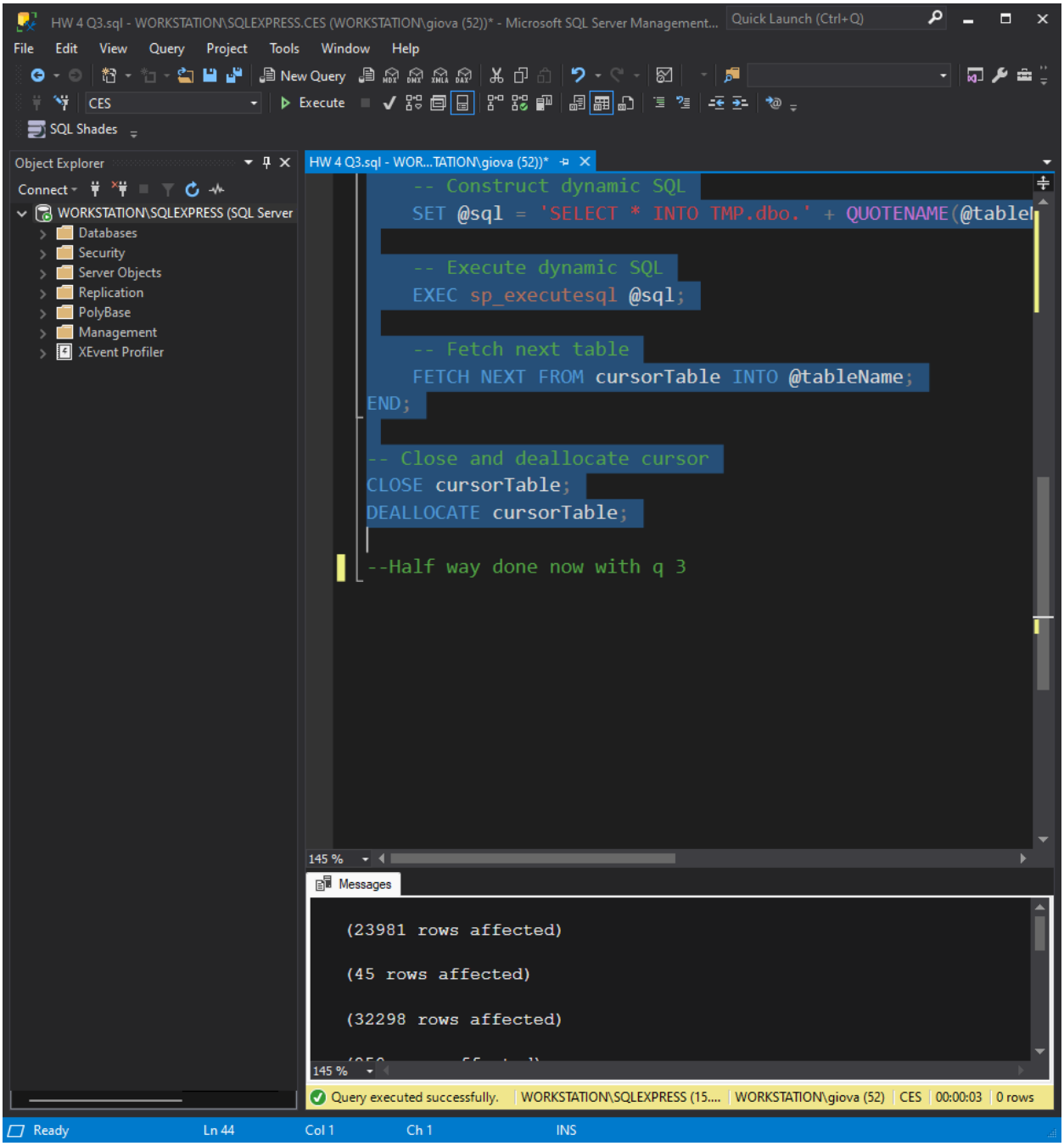


A.Write a dynamic SQL loop to copy all of your data from your wrk schema into your TMP database

```
-- Declare variables
DECLARE @tableName NVARCHAR(MAX);
```

```
DECLARE @sql NVARCHAR(MAX);

-- Declare cursor
DECLARE cursorTable CURSOR FOR
SELECT TABLE_NAME FROM INFORMATION_SCHEMA.TABLES WHERE TABLE_SCHEMA = 'wrk';

-- Open cursor
OPEN cursorTable;

-- Fetch first table
FETCH NEXT FROM cursorTable INTO @tableName;

-- Loop through all tables
WHILE @@FETCH_STATUS = 0
BEGIN
    -- Construct dynamic SQL
    SET @sql = 'SELECT * INTO TMP.dbo.' + QUOTENAME(@tableName) + ' FROM ' +
QUOTENAME(@tableName);

    -- Execute dynamic SQL
    EXEC sp_executesql @sql;

    -- Fetch next table
    FETCH NEXT FROM cursorTable INTO @tableName;
END;

-- Close and deallocate cursor
CLOSE cursorTable;
DEALLOCATE cursorTable;

--Half way done now with q 3
```

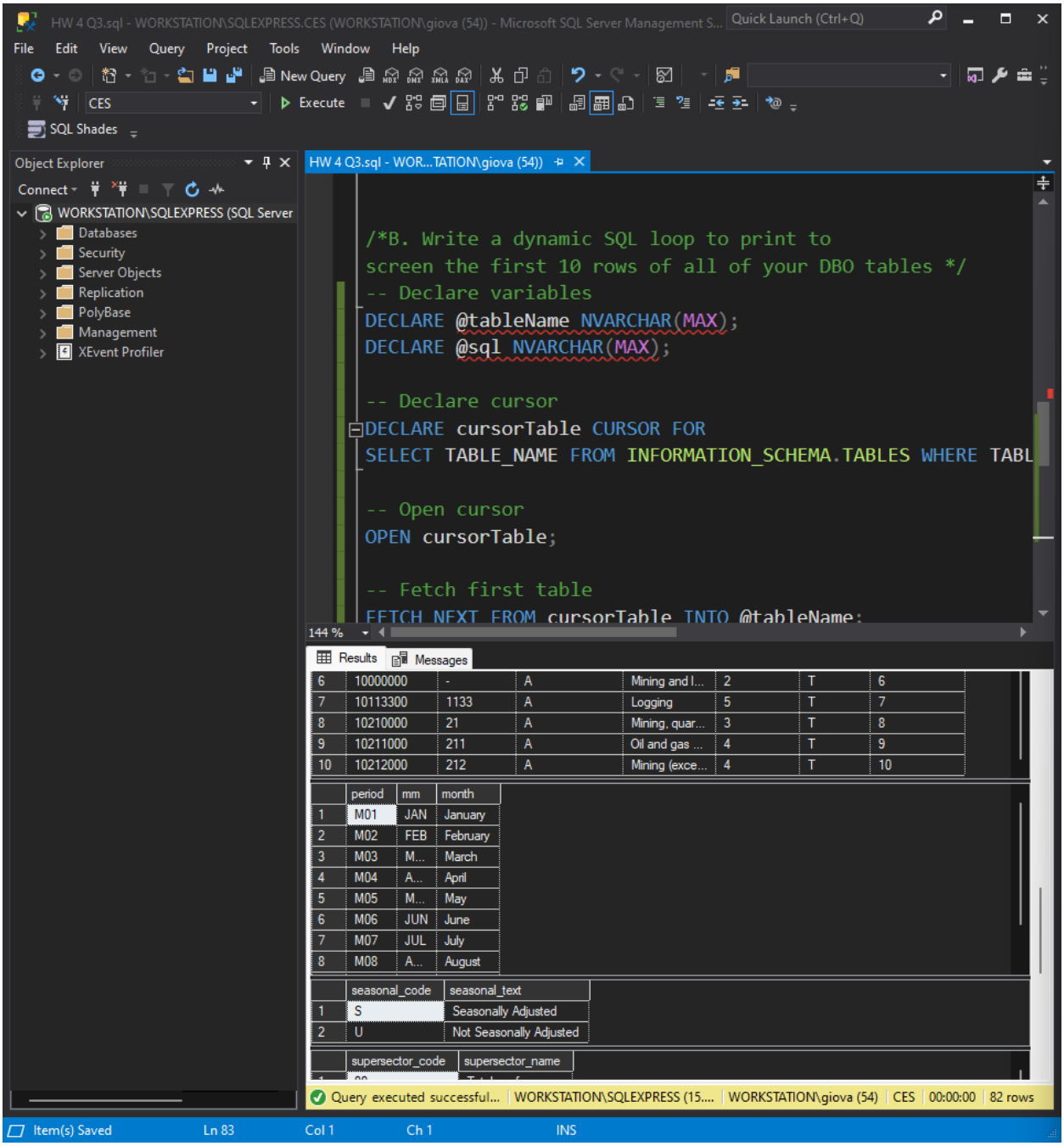

HW 4 Q3.sql - WORKSTATION\SQLEXPRESS.CES (WORKSTATION\giova (52))* - Microsoft SQL Server Management...
Quick Launch (Ctrl+Q)
File Edit View Query Project Tools Window Help
New Query
CES
Execute
SQL Shades
Object Explorer
Connect
WORKSTATION\SQLEXPRESS (SQL Server
Databases
Security
Server Objects
Replication
PolyBase
Management
XEvent Profiler
HW 4 Q3.sql - WOR...TATION\giova (52))*
-- Declare variables
DECLARE @tableName NVARCHAR(MAX);
DECLARE @sql NVARCHAR(MAX);
-- Declare cursor
DECLARE cursorTable CURSOR FOR
SELECT TABLE_NAME FROM INFORMATION_SCHEMA.TABLES WHERE TABL
-- Open cursor
OPEN cursorTable;
-- Fetch first table
FETCH NEXT FROM cursorTable INTO @tableName;
-- Loop through all tables
WHILE @@FETCH_STATUS = 0
BEGIN
-- Construct dynamic SQL
SET @sql = 'SELECT * INTO TMP.dbo.' + QUOTENAME(@tablel
-- Execute dynamic SQL
EXEC sp_executesql @sql;
-- Fetch next table
145 %
Messages
Commands completed successfully.
Completion time: 2023-06-01T12:16:02.3871784-07:00
145 %
Query executed successfully.
WORKSTATION\SQLEXPRESS (15....
WORKSTATION\giova (52)
CES
00:00:00
0 rows
Ready
Ln 44
Col 1
Ch 1
INS

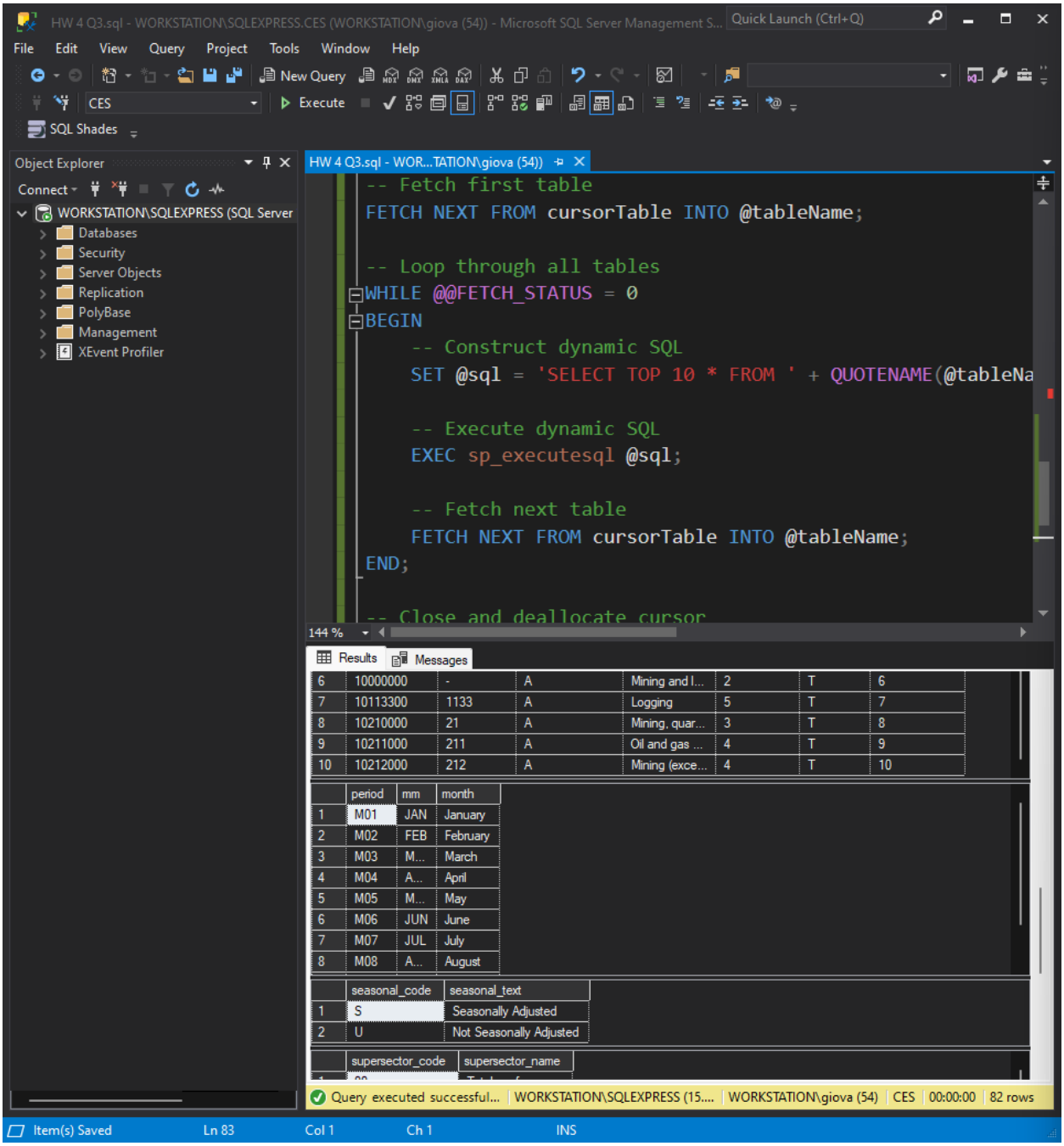


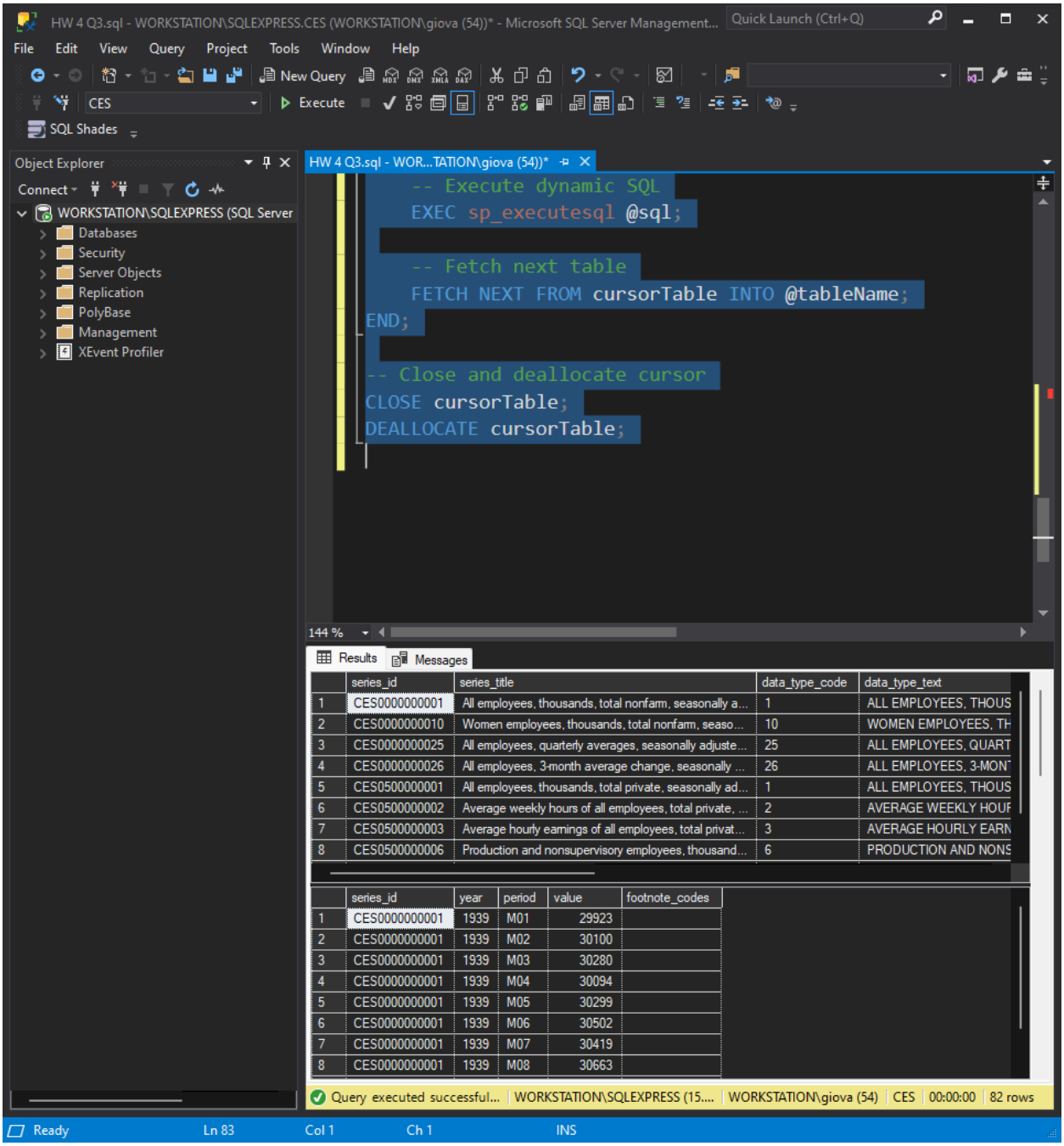
B. Write a dynamic SQL loop to print to screen the first 10 rows of all of your DBO tables */

```
-- Declare variables
```

```sql
DECLARE @tableName NVARCHAR(MAX);
DECLARE @sql NVARCHAR(MAX);

-- Declare cursor
DECLARE cursorTable CURSOR FOR
SELECT TABLE_NAME FROM INFORMATION_SCHEMA.TABLES WHERE TABLE_SCHEMA = 'dbo';

-- Open cursor
OPEN cursorTable;

-- Fetch first table
FETCH NEXT FROM cursorTable INTO @tableName;

-- Loop through all tables
WHILE @@FETCH_STATUS = 0
BEGIN
    -- Construct dynamic SQL
    SET @sql = 'SELECT TOP 10 * FROM ' + QUOTENAME(@tableName);

    -- Execute dynamic SQL
    EXEC sp_executesql @sql;

    -- Fetch next table
    FETCH NEXT FROM cursorTable INTO @tableName;
END;

-- Close and deallocate cursor
CLOSE cursorTable;
DEALLOCATE cursorTable;
```


HW 4 Q3.sql - WORKSTATION\SQLEXPRESS.CES (WORKSTATION\giova (54)) - Microsoft SQL Server Management S...
Quick Launch (Ctrl+Q)
File Edit View Query Project Tools Window Help
New Query
Execute
CES
SQL Shades
Object Explorer
Connect
WORKSTATION\SQLEXPRESS (SQL Server
Databases
Security
Server Objects
Replication
PolyBase
Management
XEvent Profiler
HW 4 Q3.sql - WOR...TATION\giova (54))
/*B. Write a dynamic SQL loop to print to
screen the first 10 rows of all of your DBO tables */
-- Declare variables
DECLARE @tableName NVARCHAR(MAX);
DECLARE @sql NVARCHAR(MAX);
-- Declare cursor
DECLARE cursorTable CURSOR FOR
SELECT TABLE_NAME FROM INFORMATION_SCHEMA.TABLES WHERE TABL
-- Open cursor
OPEN cursorTable;
-- Fetch first table
FETCH NEXT FROM cursorTable INTO @tableName;
144 %
Results
Messages
6 10000000 - A Mining and l... 2 T 6
7 10113300 1133 A Logging 5 T 7
8 10210000 21 A Mining, quar... 3 T 8
9 10211000 211 A Oil and gas ... 4 T 9
10 10212000 212 A Mining (exce... 4 T 10
period mm month
1 M01 JAN January
2 M02 FEB February
3 M03 M... March
4 M04 A... April
5 M05 M... May
6 M06 JUN June
7 M07 JUL July
8 M08 A... August
seasonal_code seasonal_text
1 S Seasonally Adjusted
2 U Not Seasonally Adjusted
supersector_code supersector_name
Query executed successful...
WORKSTATION\SQLEXPRESS (15....
WORKSTATION\giova (54)
CES
00:00:00
82 rows
Item(s) Saved
Ln 83
Col 1
Ch 1
INS

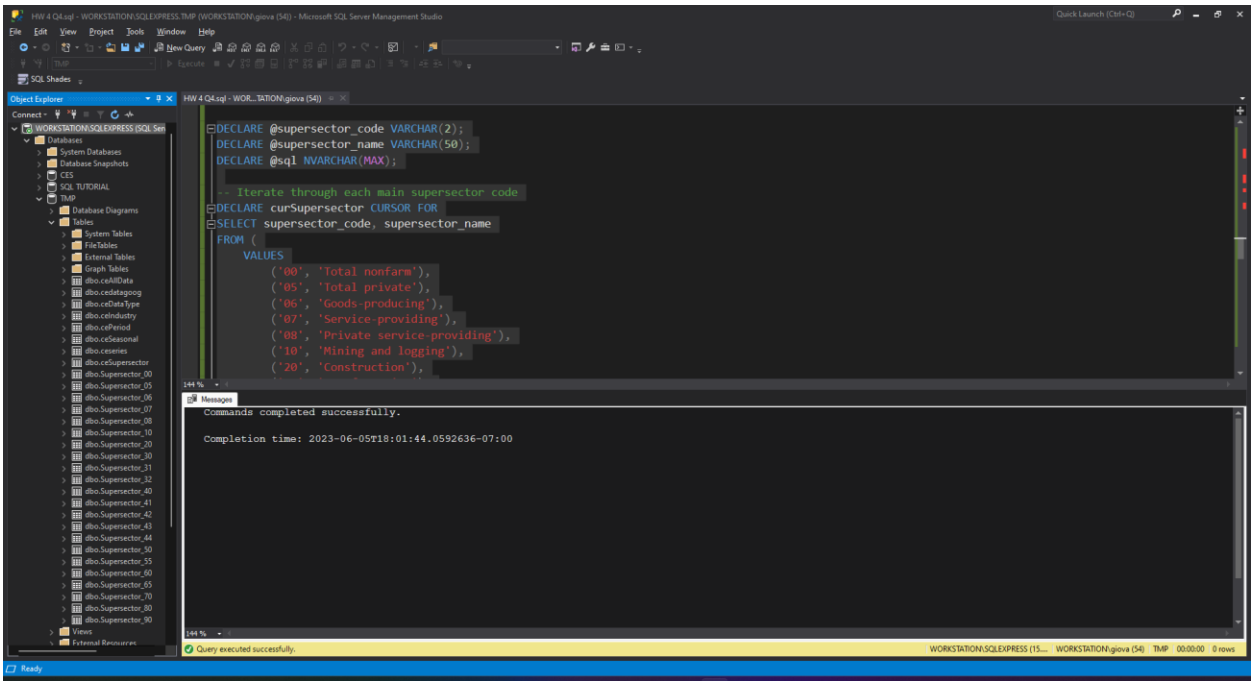

HW 4 Q3.sql - WORKSTATION\SQLEXPRESS.CES (WORKSTATION\giova (54)) - Microsoft SQL Server Management S...
Quick Launch (Ctrl+Q)
File Edit View Query Project Tools Window Help
New Query
CES
Execute
SQL Shades
Object Explorer
Connect
WORKSTATION\SQLEXPRESS (SQL Server
Databases
Security
Server Objects
Replication
PolyBase
Management
XEvent Profiler
HW 4 Q3.sql - WOR...TATION\giova (54))
-- Fetch first table
FETCH NEXT FROM cursorTable INTO @tableName;
-- Loop through all tables
WHILE @@FETCH_STATUS = 0
BEGIN
-- Construct dynamic SQL
SET @sql = 'SELECT TOP 10 * FROM ' + QUOTENAME(@tableNa
-- Execute dynamic SQL
EXEC sp_executesql @sql;
-- Fetch next table
FETCH NEXT FROM cursorTable INTO @tableName;
END;
-- Close and deallocate cursor
144 %
Results
Messages
6 10000000 - A Mining and l... 2 T 6
7 10113300 1133 A Logging 5 T 7
8 10210000 21 A Mining, quar... 3 T 8
9 10211000 211 A Oil and gas ... 4 T 9
10 10212000 212 A Mining (exce... 4 T 10
period mm month
1 M01 JAN January
2 M02 FEB February
3 M03 M... March
4 M04 A... April
5 M05 M... May
6 M06 JUN June
7 M07 JUL July
8 M08 A... August
seasonal_code seasonal_text
1 S Seasonally Adjusted
2 U Not Seasonally Adjusted
supersector_code supersector_name
Query executed successful...
WORKSTATION\SQLEXPRESS (15....
WORKSTATION\giova (54)
CES
00:00:00
82 rows
Item(s) Saved
Ln 83
Col 1
Ch 1
INS

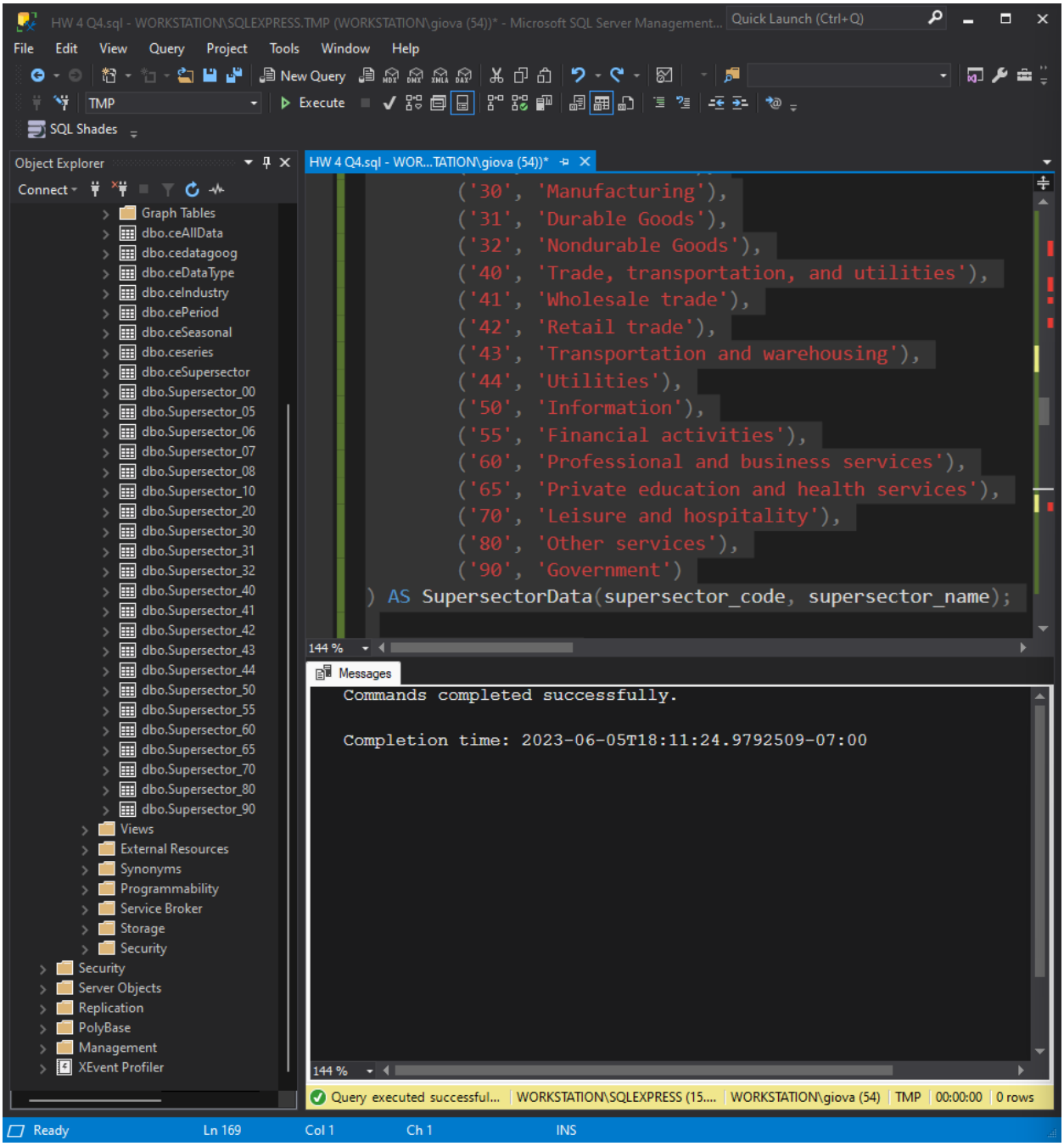

HW 4 Q3.sql - WORKSTATION\SQLEXPRESS.CES (WORKSTATION\giova (54))* - Microsoft SQL Server Management...
Quick Launch (Ctrl+Q)
File Edit View Query Project Tools Window Help
New Query
Execute
CES
SQL Shades
Object Explorer
Connect
WORKSTATION\SQLEXPRESS (SQL Server
Databases
Security
Server Objects
Replication
PolyBase
Management
XEvent Profiler
HW 4 Q3.sql - WOR...TATION\giova (54))*
-- Execute dynamic SQL
EXEC sp_executesql @sql;
-- Fetch next table
FETCH NEXT FROM cursorTable INTO @tableName;
END;
-- Close and deallocate cursor
CLOSE cursorTable;
DEALLOCATE cursorTable;
144 %
Results
Messages
series_id | series_title | data_type_code | data_type_text
1 | CES0000000001 | All employees, thousands, total nonfarm, seasonally a... | 1 | ALL EMPLOYEES, THOUS
2 | CES0000000010 | Women employees, thousands, total nonfarm, seaso... | 10 | WOMEN EMPLOYEES, TH
3 | CES0000000025 | All employees, quarterly averages, seasonally adjuste... | 25 | ALL EMPLOYEES, QUART
4 | CES0000000026 | All employees, 3-month average change, seasonally ... | 26 | ALL EMPLOYEES, 3-MONT
5 | CES0500000001 | All employees, thousands, total private, seasonally ad... | 1 | ALL EMPLOYEES, THOUS
6 | CES0500000002 | Average weekly hours of all employees, total private, ... | 2 | AVERAGE WEEKLY HOUR
7 | CES0500000003 | Average hourly earnings of all employees, total privat... | 3 | AVERAGE HOURLY EARN
8 | CES0500000006 | Production and nonsupervisory employees, thousand... | 6 | PRODUCTION AND NONS
series_id | year | period | value | footnote_codes
1 | CES0000000001 | 1939 | M01 | 29923
2 | CES0000000001 | 1939 | M02 | 30100
3 | CES0000000001 | 1939 | M03 | 30280
4 | CES0000000001 | 1939 | M04 | 30094
5 | CES0000000001 | 1939 | M05 | 30299
6 | CES0000000001 | 1939 | M06 | 30502
7 | CES0000000001 | 1939 | M07 | 30419
8 | CES0000000001 | 1939 | M08 | 30663
Query executed successful... | WORKSTATION\SQLEXPRESS (15.... | WORKSTATION\giova (54) | CES | 00:00:00 | 82 rows
Ready Ln 83 Col 1 Ch 1 INS

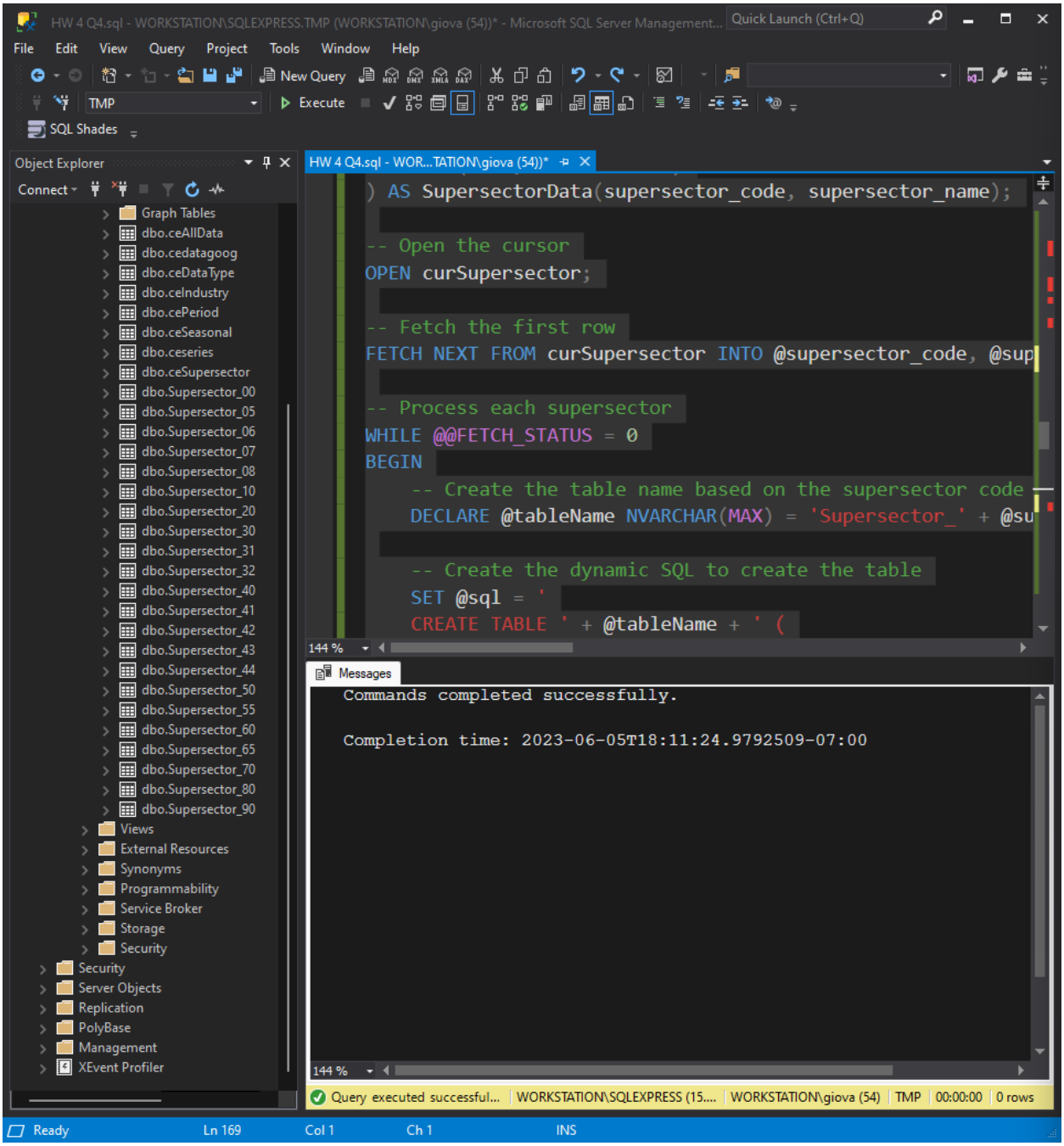
/* Part C. Using Dynamic SQL, create separate hard tables
in yourdatabase that create analysis heaps that have the
following Columns. Each table will be a series of
similar data that you will combine. You will keep
monthly records and drop annualized records. You will
keep seasonally adjusted records. There should be 20
tables in total, one for each main supersector type. Use
when the last six digits of Industry is equal to
'000000'
Parsed Series ID (hint, the SeriesID you
created to link together the different series.
Parsed Series_Title (hint, not the exact series
title. You are going to create a new one that
is generalized for the supersector)
Year
Month
Total Employment
Female Employment
Male Employment
Average Hourly Wages
Average Weekly Hours
Average Overtime hours

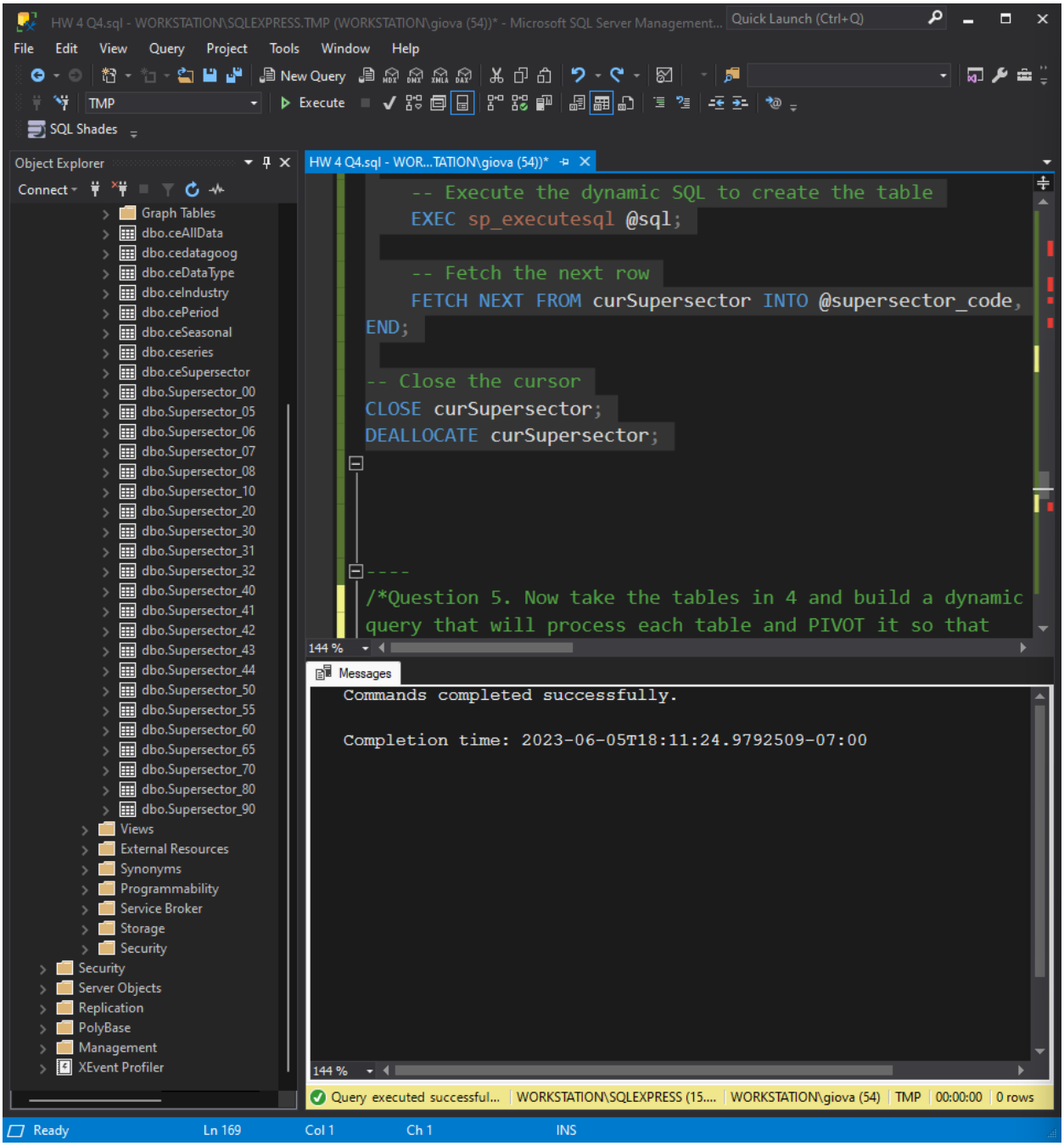
```
*/



DECLARE @supersector_code VARCHAR(2);
DECLARE @supersector_name VARCHAR(50);
DECLARE @sql NVARCHAR(MAX);

-- Iterate through each main supersector code
DECLARE curSupersector CURSOR FOR
SELECT supersector_code, supersector_name
FROM (
    VALUES
        ('00', 'Total nonfarm'),
        ('05', 'Total private'),
        ('06', 'Goods-producing'),
        ('07', 'Service-providing'),
        ('08', 'Private service-providing'),
        ('10', 'Mining and logging'),
        ('20', 'Construction'),
        ('30', 'Manufacturing'),
        ('31', 'Durable Goods'),
        ('32', 'Nondurable Goods'),
        ('40', 'Trade, transportation, and utilities'),
        ('41', 'Wholesale trade'),
        ('42', 'Retail trade'),
        ('43', 'Transportation and warehousing'),
        ('44', 'Utilities'),
        ('50', 'Information'),
        ('55', 'Financial activities'),
        ('60', 'Professional and business services'),
        ('65', 'Private education and health services'),
        ('70', 'Leisure and hospitality'),
        ('80', 'Other services'),
        ('90', 'Government')
) AS SupersectorData(supersector_code, supersector_name);

-- Open the cursor
OPEN curSupersector;

-- Fetch the first row
FETCH NEXT FROM curSupersector INTO @supersector_code, @supersector_name;

-- Process each supersector
WHILE @@FETCH_STATUS = 0
BEGIN
    -- Create the table name based on the supersector code
    DECLARE @tableName NVARCHAR(MAX) = 'Supersector_' + @supersector_code;

    -- Create the dynamic SQL to create the table
    SET @sql = '
    CREATE TABLE ' + @tableName + ' (
        ParsedSeriesID NVARCHAR(MAX),
```

```
        ParsedSeriesTitle NVARCHAR(MAX),
        Year INT,
        Month INT,
        TotalEmployment INT,
        FemaleEmployment INT,
        MaleEmployment INT,
        AverageHourlyWages DECIMAL(10, 2),
        AverageWeeklyHours DECIMAL(10, 2),
        AverageOvertimeHours DECIMAL(10, 2)
    );
    ';

    -- Execute the dynamic SQL to create the table
    EXEC sp_executesql @sql;

    -- Fetch the next row
    FETCH NEXT FROM curSupersector INTO @supersector_code, @supersector_name;
END;

-- Close the cursor
CLOSE curSupersector;
DEALLOCATE curSupersector;
```

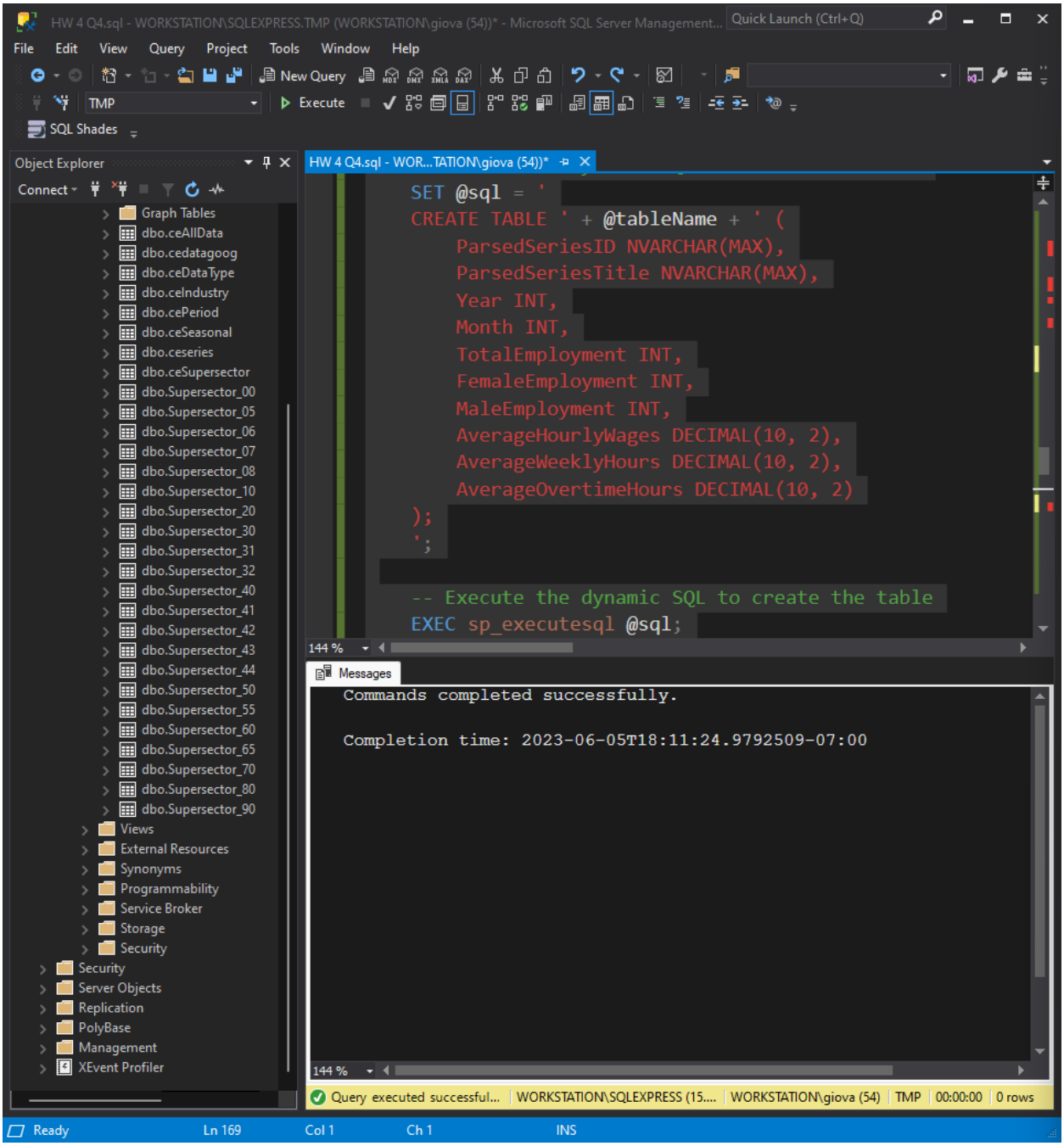

HW 4 Q4.sql - WORKSTATION\SQLEXPRESS.TMP (WORKSTATION\giova (54)) - Microsoft SQL Server Management Studio
File Edit View Project Tools Window Help
New Query
SQL Shades
Object Explorer
Connect
WORKSTATION\SQLEXPRESS (SQL Ser
Databases
System Databases
Database Snapshots
CES
SQL TUTORIAL
TMP
Database Diagrams
Tables
System Tables
FileTables
External Tables
Graph Tables
dbo.ceAllData
dbo.cedatagoog
dbo.ceDataType
dbo.ceIndustry
dbo.cePeriod
dbo.ceSeasonal
dbo.ceseries
dbo.ceSupersector
dbo.Supersector_00
dbo.Supersector_05
dbo.Supersector_06
dbo.Supersector_07
dbo.Supersector_08
dbo.Supersector_10
dbo.Supersector_20
dbo.Supersector_30
dbo.Supersector_31
dbo.Supersector_32
dbo.Supersector_40
dbo.Supersector_41
dbo.Supersector_42
dbo.Supersector_43
dbo.Supersector_44
dbo.Supersector_50
dbo.Supersector_55
dbo.Supersector_60
dbo.Supersector_65
dbo.Supersector_70
dbo.Supersector_80
dbo.Supersector_90
Views
External Resources
HW 4 Q4.sql - WOR...TATION\giova (54))
DECLARE @supersector_code VARCHAR(2);
DECLARE @supersector_name VARCHAR(50);
DECLARE @sql NVARCHAR(MAX);
-- Iterate through each main supersector code
DECLARE curSupersector CURSOR FOR
SELECT supersector_code, supersector_name
FROM (
VALUES
('00', 'Total nonfarm'),
('05', 'Total private'),
('06', 'Goods-producing'),
('07', 'Service-providing'),
('08', 'Private service-providing'),
('10', 'Mining and logging'),
('20', 'Construction'),
144 %
Messages
Commands completed successfully.
Completion time: 2023-06-05T18:01:44.0592636-07:00
144 %
Query executed successfully.
WORKSTATION\SQLEXPRESS (15.... WORKSTATION\giova (54) TMP 00:00:00 0 rows
Ready


HW 4 Q4.sql - WORKSTATION\SQLEXPRESS.TMP (WORKSTATION\giova (54))* - Microsoft SQL Server Management...
Quick Launch (Ctrl+Q)
File Edit View Query Project Tools Window Help
New Query
Execute
TMP
SQL Shades
Object Explorer
Connect
Graph Tables
dbo.ceAllData
dbo.cedatagoog
dbo.ceDataType
dbo.ceIndustry
dbo.cePeriod
dbo.ceSeasonal
dbo.ceseries
dbo.ceSupersector
dbo.Supersector_00
dbo.Supersector_05
dbo.Supersector_06
dbo.Supersector_07
dbo.Supersector_08
dbo.Supersector_10
dbo.Supersector_20
dbo.Supersector_30
dbo.Supersector_31
dbo.Supersector_32
dbo.Supersector_40
dbo.Supersector_41
dbo.Supersector_42
dbo.Supersector_43
dbo.Supersector_44
dbo.Supersector_50
dbo.Supersector_55
dbo.Supersector_60
dbo.Supersector_65
dbo.Supersector_70
dbo.Supersector_80
dbo.Supersector_90
Views
External Resources
Synonyms
Programmability
Service Broker
Storage
Security
Security
Server Objects
Replication
PolyBase
Management
XEvent Profiler
HW 4 Q4.sql - WOR...TATION\giova (54))*
('30', 'Manufacturing'),
('31', 'Durable Goods'),
('32', 'Nondurable Goods'),
('40', 'Trade, transportation, and utilities'),
('41', 'Wholesale trade'),
('42', 'Retail trade'),
('43', 'Transportation and warehousing'),
('44', 'Utilities'),
('50', 'Information'),
('55', 'Financial activities'),
('60', 'Professional and business services'),
('65', 'Private education and health services'),
('70', 'Leisure and hospitality'),
('80', 'Other services'),
('90', 'Government')
) AS SupersectorData(supersector_code, supersector_name);
144 %
Messages
Commands completed successfully.
Completion time: 2023-06-05T18:11:24.9792509-07:00
144 %
Query executed successful...
WORKSTATION\SQLEXPRESS (15....
WORKSTATION\giova (54)
TMP
00:00:00
0 rows
Ready
Ln 169
Col 1
Ch 1
INS


HW 4 Q4.sql - WORKSTATION\SQLEXPRESS.TMP (WORKSTATION\giova (54))* - Microsoft SQL Server Management...
Quick Launch (Ctrl+Q)
File Edit View Query Project Tools Window Help
New Query
Execute
TMP
SQL Shades
Object Explorer
Connect
Graph Tables
dbo.ceAllData
dbo.cedatagoog
dbo.ceDataType
dbo.ceIndustry
dbo.cePeriod
dbo.ceSeasonal
dbo.ceseries
dbo.ceSupersector
dbo.Supersector_00
dbo.Supersector_05
dbo.Supersector_06
dbo.Supersector_07
dbo.Supersector_08
dbo.Supersector_10
dbo.Supersector_20
dbo.Supersector_30
dbo.Supersector_31
dbo.Supersector_32
dbo.Supersector_40
dbo.Supersector_41
dbo.Supersector_42
dbo.Supersector_43
dbo.Supersector_44
dbo.Supersector_50
dbo.Supersector_55
dbo.Supersector_60
dbo.Supersector_65
dbo.Supersector_70
dbo.Supersector_80
dbo.Supersector_90
Views
External Resources
Synonyms
Programmability
Service Broker
Storage
Security
Security
Server Objects
Replication
PolyBase
Management
XEvent Profiler
HW 4 Q4.sql - WOR...TATION\giova (54))*
) AS SupersectorData(supersector_code, supersector_name);
-- Open the cursor
OPEN curSupersector;
-- Fetch the first row
FETCH NEXT FROM curSupersector INTO @supersector_code, @sup
-- Process each supersector
WHILE @@FETCH_STATUS = 0
BEGIN
-- Create the table name based on the supersector code
DECLARE @tableName NVARCHAR(MAX) = 'Supersector_' + @su
-- Create the dynamic SQL to create the table
SET @sql = '
CREATE TABLE ' + @tableName + ' (
144 %
Messages
Commands completed successfully.
Completion time: 2023-06-05T18:11:24.9792509-07:00
144 %
Query executed successful...
WORKSTATION\SQLEXPRESS (15....
WORKSTATION\giova (54)
TMP
00:00:00
0 rows
Ready
Ln 169
Col 1
Ch 1
INS


HW 4 Q4.sql - WORKSTATION\SQLEXPRESS.TMP (WORKSTATION\giova (54))* - Microsoft SQL Server Management...
Quick Launch (Ctrl+Q)
File Edit View Query Project Tools Window Help
New Query
Execute
TMP
SQL Shades
Object Explorer
Connect
Graph Tables
dbo.ceAllData
dbo.cedatagoog
dbo.ceDataType
dbo.ceIndustry
dbo.cePeriod
dbo.ceSeasonal
dbo.ceseries
dbo.ceSupersector
dbo.Supersector_00
dbo.Supersector_05
dbo.Supersector_06
dbo.Supersector_07
dbo.Supersector_08
dbo.Supersector_10
dbo.Supersector_20
dbo.Supersector_30
dbo.Supersector_31
dbo.Supersector_32
dbo.Supersector_40
dbo.Supersector_41
dbo.Supersector_42
dbo.Supersector_43
dbo.Supersector_44
dbo.Supersector_50
dbo.Supersector_55
dbo.Supersector_60
dbo.Supersector_65
dbo.Supersector_70
dbo.Supersector_80
dbo.Supersector_90
Views
External Resources
Synonyms
Programmability
Service Broker
Storage
Security
Security
Server Objects
Replication
PolyBase
Management
XEvent Profiler
HW 4 Q4.sql - WOR...TATION\giova (54))*
-- Execute the dynamic SQL to create the table
EXEC sp_executesql @sql;
-- Fetch the next row
FETCH NEXT FROM curSupersector INTO @supersector_code,
END;
-- Close the cursor
CLOSE curSupersector;
DEALLOCATE curSupersector;

/*Question 5. Now take the tables in 4 and build a dynamic
query that will process each table and PIVOT it so that
144 %
Messages
Commands completed successfully.
Completion time: 2023-06-05T18:11:24.9792509-07:00
144 %
Query executed successful...
WORKSTATION\SQLEXPRESS (15....
WORKSTATION\giova (54)
TMP
00:00:00
0 rows
Ready
Ln 169
Col 1
Ch 1
INS

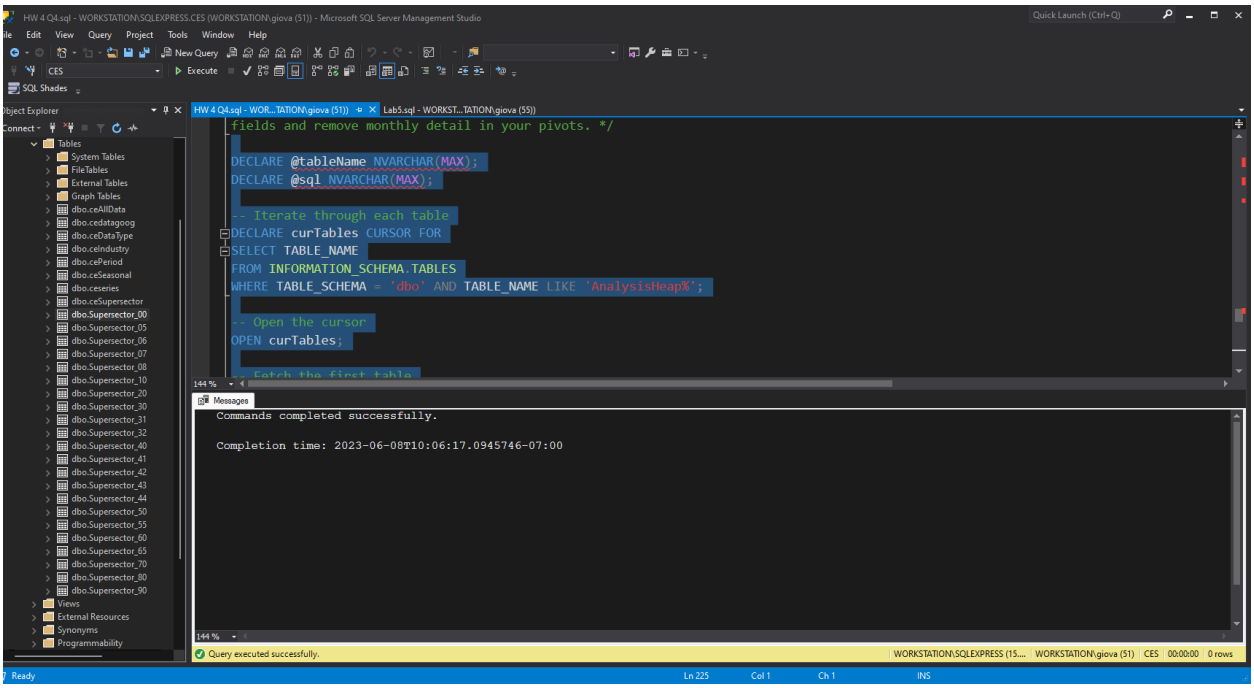


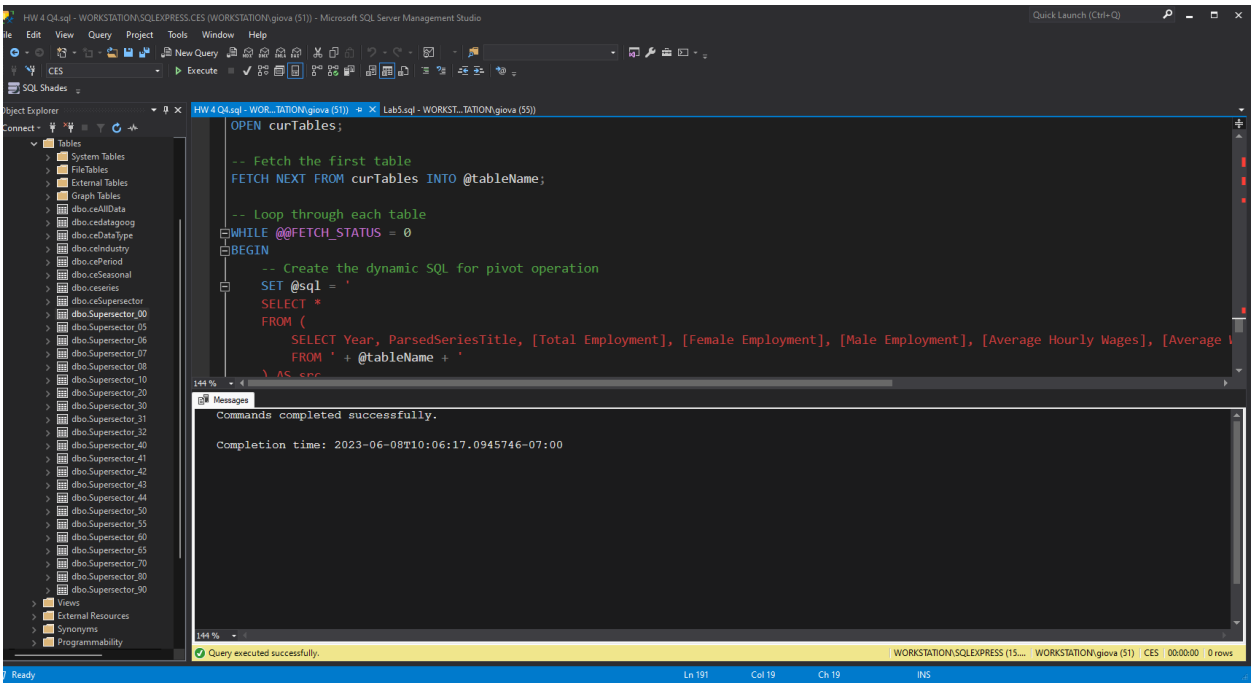
```
/*Question 5. Now take the tables in 4 and
build a dynamic SQL
```

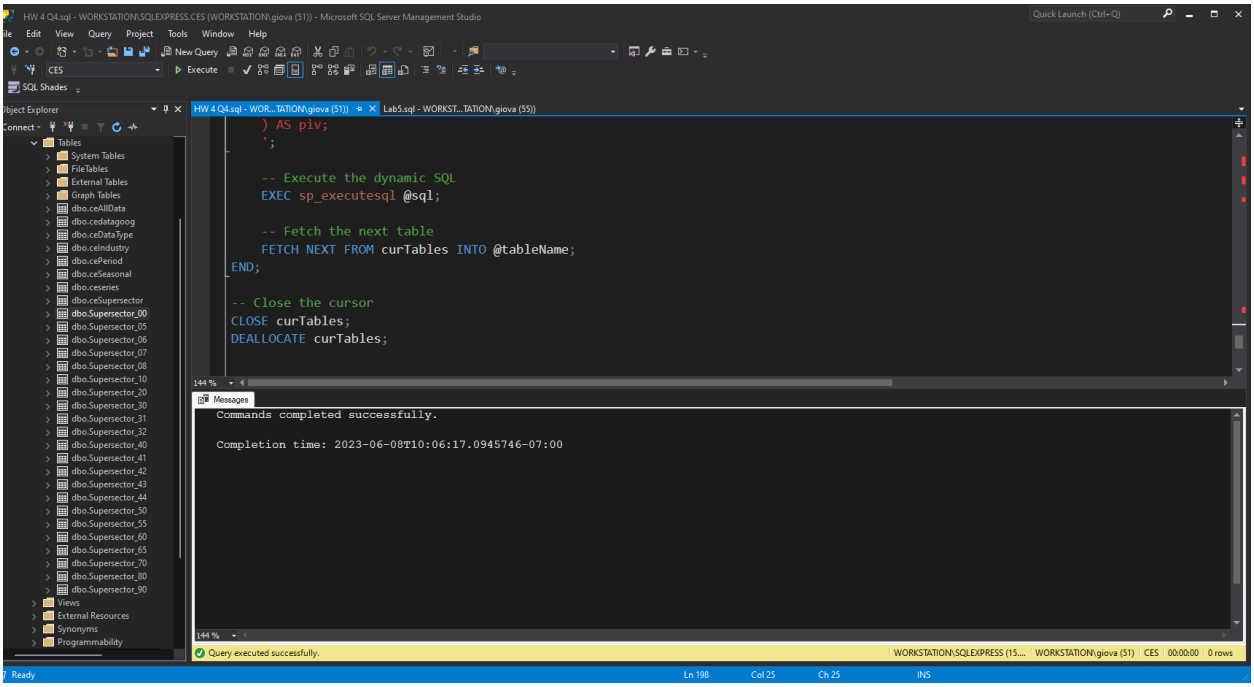
query that will process each table and PIVOT it so that
you have total employment, female employment, male
employment, average hourly wages, average weekly hours
average overtime hours on the left-hand side and year
across the top. You will want to average each of the
fields and remove monthly detail in your pivots. */

```
DECLARE @tableName NVARCHAR(MAX);
DECLARE @sql NVARCHAR(MAX);

-- Iterate through each table
DECLARE curTables CURSOR FOR
SELECT TABLE_NAME
FROM INFORMATION_SCHEMA.TABLES
WHERE TABLE_SCHEMA = 'dbo' AND TABLE_NAME LIKE 'AnalysisHeap%';

-- Open the cursor
OPEN curTables;

-- Fetch the first table
FETCH NEXT FROM curTables INTO @tableName;

-- Loop through each table
WHILE @@FETCH_STATUS = 0
BEGIN
    -- Create the dynamic SQL for pivot operation
    SET @sql = '
    SELECT *
    FROM (
        SELECT Year, ParsedSeriesTitle, [Total Employment], [Female Employment], [Male
Employment], [Average Hourly Wages], [Average Weekly Hours], [Average Overtime Hours]
        FROM ' + @tableName + '
    ) AS src
    PIVOT (
        AVG([Value])
        FOR Year IN ([2015], [2016], [2017], [2018], [2019], [2020])
    ) AS piv;
    ';
```

```
    -- Execute the dynamic SQL
    EXEC sp_executesql @sql;

    -- Fetch the next table
    FETCH NEXT FROM curTables INTO @tableName;
END;

-- Close the cursor
CLOSE curTables;
DEALLOCATE curTables;
```




HW 4 Q4.sql - WORKSTATION\SQLEXPRESS.CES (WORKSTATION\giova (51)) - Microsoft SQL Server Management Studio
Quick Launch (Ctrl+Q)
File Edit View Query Project Tools Window Help
New Query
Execute
CES
SQL Shades
Object Explorer
Connect
Tables
System Tables
FileTables
External Tables
Graph Tables
dbo.ceAllData
dbo.cedatagoog
dbo.ceDataType
dbo.ceIndustry
dbo.cePeriod
dbo.ceSeasonal
dbo.ceseries
dbo.ceSupersector
dbo.Supersector_00
dbo.Supersector_05
dbo.Supersector_06
dbo.Supersector_07
dbo.Supersector_08
dbo.Supersector_10
dbo.Supersector_20
dbo.Supersector_30
dbo.Supersector_31
dbo.Supersector_32
dbo.Supersector_40
dbo.Supersector_41
dbo.Supersector_42
dbo.Supersector_43
dbo.Supersector_44
dbo.Supersector_50
dbo.Supersector_55
dbo.Supersector_60
dbo.Supersector_65
dbo.Supersector_70
dbo.Supersector_80
dbo.Supersector_90
Views
External Resources
Synonyms
Programmability
HW 4 Q4.sql - WOR...TATION\giova (51))
Lab5.sql - WORKST...TATION\giova (55))
OPEN curTables;
-- Fetch the first table
FETCH NEXT FROM curTables INTO @tableName;
-- Loop through each table
WHILE @@FETCH_STATUS = 0
BEGIN
-- Create the dynamic SQL for pivot operation
SET @sql = '
SELECT *
FROM (
SELECT Year, ParsedSeriesTitle, [Total Employment], [Female Employment], [Male Employment], [Average Hourly Wages], [Average
FROM ' + @tableName + '
144 %
Messages
Commands completed successfully.
Completion time: 2023-06-08T10:06:17.0945746-07:00
144 %
Query executed successfully.
WORKSTATION\SQLEXPRESS (15....
WORKSTATION\giova (51)
CES
00:00:00
0 rows
Ready
Ln 191
Col 19
Ch 19
INS


HW 4 Q4.sql - WORKSTATION\SQLEXPRESS.CES (WORKSTATION\giova (51)) - Microsoft SQL Server Management Studio
Quick Launch (Ctrl+Q)
File Edit View Query Project Tools Window Help
New Query
Execute
CES
SQL Shades
Object Explorer
Connect
Tables
System Tables
FileTables
External Tables
Graph Tables
dbo.ceAllData
dbo.cedatagoog
dbo.ceDataType
dbo.ceIndustry
dbo.cePeriod
dbo.ceSeasonal
dbo.ceseries
dbo.ceSupersector
dbo.Supersector_00
dbo.Supersector_05
dbo.Supersector_06
dbo.Supersector_07
dbo.Supersector_08
dbo.Supersector_10
dbo.Supersector_20
dbo.Supersector_30
dbo.Supersector_31
dbo.Supersector_32
dbo.Supersector_40
dbo.Supersector_41
dbo.Supersector_42
dbo.Supersector_43
dbo.Supersector_44
dbo.Supersector_50
dbo.Supersector_55
dbo.Supersector_60
dbo.Supersector_65
dbo.Supersector_70
dbo.Supersector_80
dbo.Supersector_90
Views
External Resources
Synonyms
Programmability
HW 4 Q4.sql - WOR...TATION\giova (51))
Lab5.sql - WORKST...TATION\giova (55))
) AS piv;
';
-- Execute the dynamic SQL
EXEC sp_executesql @sql;
-- Fetch the next table
FETCH NEXT FROM curTables INTO @tableName;
END;
-- Close the cursor
CLOSE curTables;
DEALLOCATE curTables;
144 %
Messages
Commands completed successfully.
Completion time: 2023-06-08T10:06:17.0945746-07:00
144 %
Query executed successfully.
WORKSTATION\SQLEXPRESS (15....
WORKSTATION\giova (51)
CES
00:00:00
0 rows
Ready
Ln 198
Col 25
Ch 25
INS