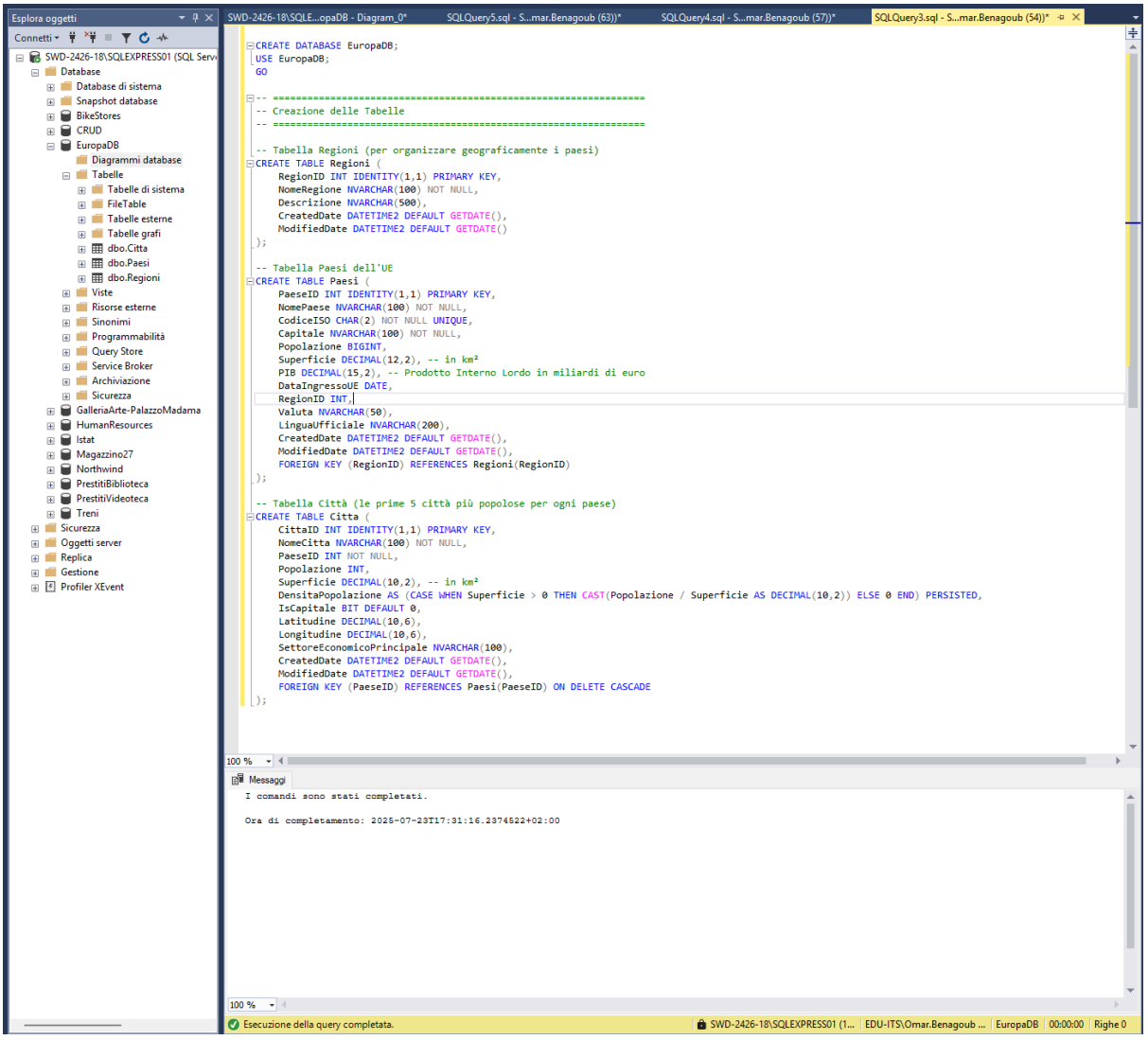
Benagoub
Omar
23-07-2025
Corso Software Developer
Base di dati - SQL

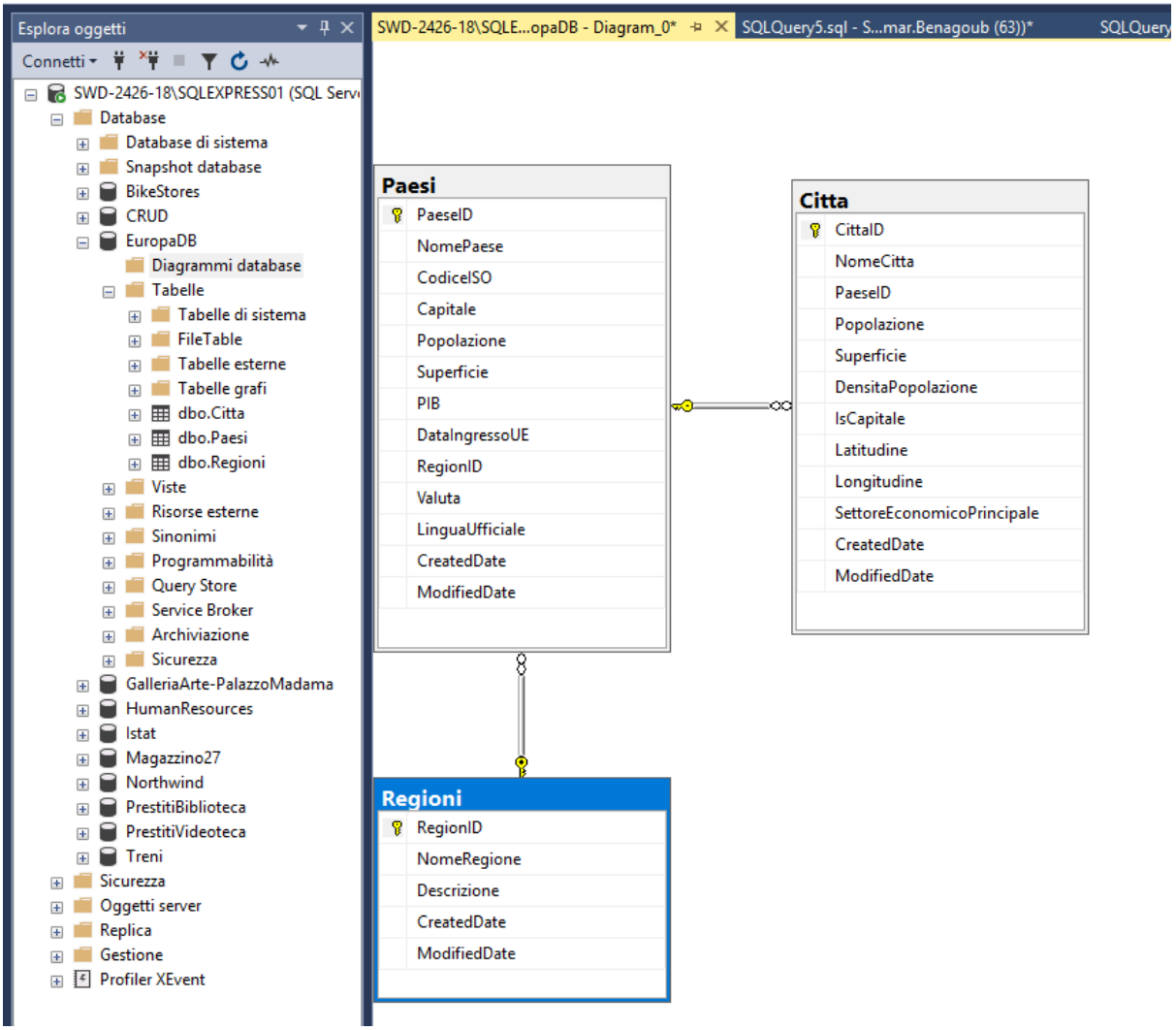
Utilizzo di Copilot

creazione database

```
CREATE DATABASE EuropaDB;
USE EuropaDB;
GO

-- ================================================================
-- Creazione delle Tabelle
-- ================================================================

-- Tabella Regioni (per organizzare geograficamente i paesi)
CREATE TABLE Regioni (
    RegionID INT IDENTITY(1,1) PRIMARY KEY,
    NomeRegione NVARCHAR(100) NOT NULL,
    Descrizione NVARCHAR(500),
    CreatedDate DATETIME2 DEFAULT GETDATE(),
    ModifiedDate DATETIME2 DEFAULT GETDATE()
);

-- Tabella Paesi dell'UE
CREATE TABLE Paesi (
    PaeseID INT IDENTITY(1,1) PRIMARY KEY,
    NomePaese NVARCHAR(100) NOT NULL,
    CodiceISO CHAR(2) NOT NULL UNIQUE,
    Capitale NVARCHAR(100) NOT NULL,
    Popolazione BIGINT,
    Superficie DECIMAL(12,2), -- in km²
    PIB DECIMAL(15,2), -- Prodotto Interno Lordo in miliardi di euro
    DataIngressoUE DATE,
    RegionID INT,
    Valuta NVARCHAR(50),
    LinguaUfficiale NVARCHAR(200),
    CreatedDate DATETIME2 DEFAULT GETDATE(),
    ModifiedDate DATETIME2 DEFAULT GETDATE(),
    FOREIGN KEY (RegionID) REFERENCES Regioni(RegionID)
);

-- Tabella Città (le prime 5 città più popolose per ogni paese)
CREATE TABLE Citta (
    CittaID INT IDENTITY(1,1) PRIMARY KEY,
```

```
    NomeCitta NVARCHAR(100) NOT NULL,
    PaeseID INT NOT NULL,
    Popolazione INT,
    Superficie DECIMAL(10,2), -- in km²
    DensitaPopolazione AS (CASE WHEN Superficie > 0 THEN CAST(Popolazione /
Superficie AS DECIMAL(10,2)) ELSE 0 END) PERSISTED,
    IsCapitale BIT DEFAULT 0,
    Latitudine DECIMAL(10,6),
    Longitudine DECIMAL(10,6),
    SettoreEconomicoPrincipale NVARCHAR(100),
    CreatedDate DATETIME2 DEFAULT GETDATE(),
    ModifiedDate DATETIME2 DEFAULT GETDATE(),
    FOREIGN KEY (PaeseID) REFERENCES Paesi(PaeseID) ON DELETE CASCADE
);
```





Popolamento dei dati

```
-- ============================================================
-- Inserimento delle Regioni Geografiche
-- ============================================================

INSERT INTO Regioni (NomeRegione, Descrizione) VALUES
('Europa Occidentale', 'Paesi dell''Europa occidentale inclusi Francia, Germania, Belgio, Paesi Bassi, Lussemburgo, Austria'),
('Europa Meridionale', 'Paesi del Mediterraneo inclusi Italia, Spagna, Portogallo, Grecia, Malta, Cipro'),
('Europa Settentrionale', 'Paesi nordici e baltici inclusi Svezia, Finlandia, Danimarca, Estonia, Lettonia, Lituania'),
('Europa Centrale', 'Paesi dell''Europa centrale inclusi Polonia, Repubblica Ceca, Slovacchia, Ungheria, Slovenia'),
('Europa Sud-orientale', 'Paesi dei Balcani inclusi Romania, Bulgaria, Croazia');

-- ============================================================
-- Inserimento dei Paesi dell'Unione Europea (27 paesi)
-- ============================================================
```

```
INSERT INTO Paesi (NomePaese, CodiceISO, Capitale, Popolazione, Superficie, PIB,
DataIngressoUE, RegionID, Valuta, LinguaUfficiale) VALUES
-- Europa Occidentale
('Germania', 'DE', 'Berlino', 83240525, 357022.00, 4259.93, '1957-03-25', 1, 'Euro',
'Tedesco'),
('Francia', 'FR', 'Parigi', 67391582, 643801.00, 2937.47, '1957-03-25', 1, 'Euro', 'Francese'),
('Paesi Bassi', 'NL', 'Amsterdam', 17441139, 41543.00, 909.07, '1957-03-25', 1, 'Euro',
'Olandese'),
('Belgio', 'BE', 'Bruxelles', 11555997, 30528.00, 521.84, '1957-03-25', 1, 'Euro', 'Olandese,
Francese, Tedesco'),
('Austria', 'AT', 'Vienna', 9006398, 83879.00, 432.07, '1995-01-01', 1, 'Euro', 'Tedesco'),
('Lussemburgo', 'LU', 'Lussemburgo', 625978, 2586.00, 71.16, '1957-03-25', 1, 'Euro',
'Lussemburghese, Francese, Tedesco'),
('Irlanda', 'IE', 'Dublino', 4982907, 70273.00, 425.89, '1973-01-01', 1, 'Euro', 'Irlandese,
Inglese'),

-- Europa Meridionale
('Italia', 'IT', 'Roma', 60461826, 301340.00, 2107.70, '1957-03-25', 2, 'Euro', 'Italiano'),
('Spagna', 'ES', 'Madrid', 47351567, 505370.00, 1397.87, '1986-01-01', 2, 'Euro', 'Spagnolo'),
('Portogallo', 'PT', 'Lisbona', 10295503, 92090.00, 238.31, '1986-01-01', 2, 'Euro',
'Portoghese'),
('Grecia', 'GR', 'Atene', 10423054, 131957.00, 189.41, '1981-01-01', 2, 'Euro', 'Greco'),
('Malta', 'MT', 'La Valletta', 441543, 316.00, 14.98, '2004-05-01', 2, 'Euro', 'Maltese, Inglese'),
('Cipro', 'CY', 'Nicosia', 1207359, 9251.00, 25.03, '2004-05-01', 2, 'Euro', 'Greco, Turco'),

-- Europa Settentrionale
('Svezia', 'SE', 'Stoccolma', 10379295, 450295.00, 541.22, '1995-01-01', 3, 'Corona
svedese', 'Svedese'),
('Finlandia', 'FI', 'Helsinki', 5540720, 338424.00, 269.75, '1995-01-01', 3, 'Euro', 'Finlandese,
Svedese'),
('Danimarca', 'DK', 'Copenaghen', 5831404, 43094.00, 356.08, '1973-01-01', 3, 'Corona
danese', 'Danese'),
('Estonia', 'EE', 'Tallinn', 1326535, 45228.00, 31.03, '2004-05-01', 3, 'Euro', 'Estone'),
('Lettonia', 'LV', 'Riga', 1886198, 64589.00, 35.04, '2004-05-01', 3, 'Euro', 'Lettone'),
('Lituania', 'LT', 'Vilnius', 2722289, 65300.00, 59.17, '2004-05-01', 3, 'Euro', 'Lituano'),

-- Europa Centrale
('Polonia', 'PL', 'Varsavia', 37846611, 312696.00, 679.44, '2004-05-01', 4, 'Zloty', 'Polacco'),
('Repubblica Ceca', 'CZ', 'Praga', 10708981, 78867.00, 281.78, '2004-05-01', 4, 'Corona
ceca', 'Ceco'),
('Slovacchia', 'SK', 'Bratislava', 5459642, 49035.00, 105.12, '2004-05-01', 4, 'Euro',
'Slovacco'),
('Ungheria', 'HU', 'Budapest', 9660351, 93028.00, 181.85, '2004-05-01', 4, 'Fiorino
ungherese', 'Ungherese'),
('Slovenia', 'SI', 'Lubiana', 2078938, 20273.00, 54.16, '2004-05-01', 4, 'Euro', 'Sloveno'),

-- Europa Sud-orientale
```

('Romania', 'RO', 'Bucarest', 19237691, 238391.00, 249.75, '2007-01-01', 5, 'Leu rumeno', 'Rumeno'),
('Bulgaria', 'BG', 'Sofia', 6948445, 110879.00, 84.05, '2007-01-01', 5, 'Lev bulgaro', 'Bulgaro'),
('Croazia', 'HR', 'Zagabria', 4105267, 56594.00, 60.42, '2013-07-01', 5, 'Euro', 'Croato');

-- ==
-- Inserimento delle Città (Prime 5 per ogni paese)
-- ==

-- GERMANIA - Prime 5 città
INSERT INTO Citta (NomeCitta, PaeseID, Popolazione, Superficie, IsCapitale, Latitudine, Longitudine, SettoreEconomicoPrincipale) VALUES
('Berlino', (SELECT PaeseID FROM Paesi WHERE CodiceISO = 'DE'), 3669491, 891.68, 1, 52.520008, 13.404954, 'Servizi, Tecnologia'),
('Amburgo', (SELECT PaeseID FROM Paesi WHERE CodiceISO = 'DE'), 1899160, 755.09, 0, 53.551086, 9.993682, 'Porto, Logistica'),
('Monaco di Baviera', (SELECT PaeseID FROM Paesi WHERE CodiceISO = 'DE'), 1471508, 310.43, 0, 48.135125, 11.581981, 'Tecnologia, Automotive'),
('Colonia', (SELECT PaeseID FROM Paesi WHERE CodiceISO = 'DE'), 1085664, 405.15, 0, 50.937531, 6.960279, 'Media, Cultura'),
('Francoforte sul Meno', (SELECT PaeseID FROM Paesi WHERE CodiceISO = 'DE'), 753056, 248.31, 0, 50.110922, 8.682127, 'Finanza, Banche');

-- FRANCIA - Prime 5 città
INSERT INTO Citta (NomeCitta, PaeseID, Popolazione, Superficie, IsCapitale, Latitudine, Longitudine, SettoreEconomicoPrincipale) VALUES
('Parigi', (SELECT PaeseID FROM Paesi WHERE CodiceISO = 'FR'), 2161000, 105.4, 1, 48.856614, 2.352222, 'Finanza, Turismo'),
('Marsiglia', (SELECT PaeseID FROM Paesi WHERE CodiceISO = 'FR'), 873076, 240.62, 0, 43.296482, 5.369780, 'Porto, Industria'),
('Lione', (SELECT PaeseID FROM Paesi WHERE CodiceISO = 'FR'), 522228, 47.87, 0, 45.764043, 4.835659, 'Farmaceutico, Tessile'),
('Tolosa', (SELECT PaeseID FROM Paesi WHERE CodiceISO = 'FR'), 486828, 118.3, 0, 43.604652, 1.444209, 'Aerospaziale'),
('Nizza', (SELECT PaeseID FROM Paesi WHERE CodiceISO = 'FR'), 342637, 71.92, 0, 43.710173, 7.261953, 'Turismo, Tecnologia');

-- ITALIA - Prime 5 città
INSERT INTO Citta (NomeCitta, PaeseID, Popolazione, Superficie, IsCapitale, Latitudine, Longitudine, SettoreEconomicoPrincipale) VALUES
('Roma', (SELECT PaeseID FROM Paesi WHERE CodiceISO = 'IT'), 2872800, 1285.0, 1, 41.902784, 12.496366, 'Turismo, Servizi pubblici'),
('Milano', (SELECT PaeseID FROM Paesi WHERE CodiceISO = 'IT'), 1374999, 181.76, 0, 45.464204, 9.189982, 'Finanza, Moda'),
('Napoli', (SELECT PaeseID FROM Paesi WHERE CodiceISO = 'IT'), 967069, 117.27, 0, 40.851775, 14.268124, 'Porto, Turismo'),
('Torino', (SELECT PaeseID FROM Paesi WHERE CodiceISO = 'IT'), 872367, 130.17, 0, 45.070312, 7.686856, 'Automotive'),

('Palermo', (SELECT PaeseID FROM Paesi WHERE CodiceISO = 'IT'), 673735, 158.9, 0, 38.115688, 13.361267, 'Turismo, Agricoltura');

-- SPAGNA - Prime 5 città
INSERT INTO Citta (NomeCitta, PaeseID, Popolazione, Superficie, IsCapitale, Latitudine, Longitudine, SettoreEconomicoPrincipale) VALUES
('Madrid', (SELECT PaeseID FROM Paesi WHERE CodiceISO = 'ES'), 3223334, 604.3, 1, 40.416775, -3.703790, 'Servizi, Finanza'),
('Barcellona', (SELECT PaeseID FROM Paesi WHERE CodiceISO = 'ES'), 1620343, 101.4, 0, 41.385064, 2.173403, 'Turismo, Tecnologia'),
('Valencia', (SELECT PaeseID FROM Paesi WHERE CodiceISO = 'ES'), 791413, 134.6, 0, 39.469907, -0.376288, 'Porto, Agricoltura'),
('Siviglia', (SELECT PaeseID FROM Paesi WHERE CodiceISO = 'ES'), 688711, 140.0, 0, 37.388081, -5.982077, 'Turismo, Aerospaziale'),
('Saragozza', (SELECT PaeseID FROM Paesi WHERE CodiceISO = 'ES'), 674997, 973.8, 0, 41.648823, -0.889085, 'Automotive, Logistica');

-- PAESI BASSI - Prime 5 città
INSERT INTO Citta (NomeCitta, PaeseID, Popolazione, Superficie, IsCapitale, Latitudine, Longitudine, SettoreEconomicoPrincipale) VALUES
('Amsterdam', (SELECT PaeseID FROM Paesi WHERE CodiceISO = 'NL'), 873338, 219.32, 1, 52.370216, 4.895168, 'Finanza, Turismo'),
('Rotterdam', (SELECT PaeseID FROM Paesi WHERE CodiceISO = 'NL'), 651446, 325.79, 0, 51.924420, 4.477733, 'Porto, Logistica'),
('L''Aia', (SELECT PaeseID FROM Paesi WHERE CodiceISO = 'NL'), 548320, 98.12, 0, 52.078663, 4.288788, 'Governo, Diritto internazionale'),
('Utrecht', (SELECT PaeseID FROM Paesi WHERE CodiceISO = 'NL'), 361924, 99.21, 0, 52.090737, 5.121420, 'Servizi, Educazione'),
('Eindhoven', (SELECT PaeseID FROM Paesi WHERE CodiceISO = 'NL'), 234235, 88.84, 0, 51.441642, 5.469722, 'Tecnologia, Design');

-- BELGIO - Prime 5 città
INSERT INTO Citta (NomeCitta, PaeseID, Popolazione, Superficie, IsCapitale, Latitudine, Longitudine, SettoreEconomicoPrincipale) VALUES
('Bruxelles', (SELECT PaeseID FROM Paesi WHERE CodiceISO = 'BE'), 1218255, 161.38, 1, 50.850340, 4.351710, 'Istituzioni UE, Servizi'),
('Anversa', (SELECT PaeseID FROM Paesi WHERE CodiceISO = 'BE'), 529247, 204.51, 0, 51.219448, 4.402464, 'Porto, Diamanti'),
('Gand', (SELECT PaeseID FROM Paesi WHERE CodiceISO = 'BE'), 262219, 156.18, 0, 51.054342, 3.717424, 'Industria, Università'),
('Charleroi', (SELECT PaeseID FROM Paesi WHERE CodiceISO = 'BE'), 201816, 102.08, 0, 50.411211, 4.444444, 'Industria, Aeroporto'),
('Liegi', (SELECT PaeseID FROM Paesi WHERE CodiceISO = 'BE'), 197355, 69.39, 0, 50.633333, 5.566667, 'Industria, Porto fluviale');

-- POLONIA - Prime 5 città
INSERT INTO Citta (NomeCitta, PaeseID, Popolazione, Superficie, IsCapitale, Latitudine, Longitudine, SettoreEconomicoPrincipale) VALUES

('Varsavia', (SELECT PaeseID FROM Paesi WHERE CodiceISO = 'PL'), 1790658, 517.24, 1, 52.229676, 21.012229, 'Finanza, Servizi'),
('Cracovia', (SELECT PaeseID FROM Paesi WHERE CodiceISO = 'PL'), 779115, 326.85, 0, 50.064651, 19.944981, 'Turismo, Tecnologia'),
('Danzica', (SELECT PaeseID FROM Paesi WHERE CodiceISO = 'PL'), 470907, 262.0, 0, 54.352025, 18.646638, 'Porto, Cantieristica'),
('Breslavia', (SELECT PaeseID FROM Paesi WHERE CodiceISO = 'PL'), 643782, 292.82, 0, 51.107885, 17.038538, 'Tecnologia, Servizi'),
('Lodz', (SELECT PaeseID FROM Paesi WHERE CodiceISO = 'PL'), 679941, 293.25, 0, 51.759445, 19.457216, 'Tessile, Film');

-- Continua con altri paesi principali...
-- REPUBBLICA CECA - Prime 5 città
INSERT INTO Citta (NomeCitta, PaeseID, Popolazione, Superficie, IsCapitale, Latitudine, Longitudine, SettoreEconomicoPrincipale) VALUES
('Praga', (SELECT PaeseID FROM Paesi WHERE CodiceISO = 'CZ'), 1357326, 496.21, 1, 50.075538, 14.437800, 'Turismo, Tecnologia'),
('Brno', (SELECT PaeseID FROM Paesi WHERE CodiceISO = 'CZ'), 382405, 230.22, 0, 49.195061, 16.606836, 'Tecnologia, Università'),
('Ostrava', (SELECT PaeseID FROM Paesi WHERE CodiceISO = 'CZ'), 287968, 214.23, 0, 49.834465, 18.282606, 'Industria pesante'),
('Plzen', (SELECT PaeseID FROM Paesi WHERE CodiceISO = 'CZ'), 175219, 137.65, 0, 49.747431, 13.377704, 'Birra, Industria'),
('Liberec', (SELECT PaeseID FROM Paesi WHERE CodiceISO = 'CZ'), 104261, 106.09, 0, 50.767220, 15.056360, 'Tessile, Vetro');

-- ROMANIA - Prime 5 città
INSERT INTO Citta (NomeCitta, PaeseID, Popolazione, Superficie, IsCapitale, Latitudine, Longitudine, SettoreEconomicoPrincipale) VALUES
('Bucarest', (SELECT PaeseID FROM Paesi WHERE CodiceISO = 'RO'), 1883425, 228.0, 1, 44.426767, 26.102538, 'Servizi, Finanza'),
('Cluj-Napoca', (SELECT PaeseID FROM Paesi WHERE CodiceISO = 'RO'), 324576, 179.52, 0, 46.770439, 23.591423, 'Tecnologia, Università'),
('Timisoara', (SELECT PaeseID FROM Paesi WHERE CodiceISO = 'RO'), 319279, 130.5, 0, 45.759723, 21.230019, 'Automotive, Tecnologia'),
('Iasi', (SELECT PaeseID FROM Paesi WHERE CodiceISO = 'RO'), 290422, 93.9, 0, 47.151726, 27.587914, 'Università, Farmaceutico'),
('Constanta', (SELECT PaeseID FROM Paesi WHERE CodiceISO = 'RO'), 283872, 124.89, 0, 44.181498, 28.635162, 'Porto, Turismo');

-- PORTOGALLO - Prime 5 città
INSERT INTO Citta (NomeCitta, PaeseID, Popolazione, Superficie, IsCapitale, Latitudine, Longitudine, SettoreEconomicoPrincipale) VALUES
('Lisbona', (SELECT PaeseID FROM Paesi WHERE CodiceISO = 'PT'), 544851, 100.05, 1, 38.722252, -9.139337, 'Turismo, Servizi'),
('Porto', (SELECT PaeseID FROM Paesi WHERE CodiceISO = 'PT'), 237591, 41.42, 0, 41.149685, -8.610016, 'Vino, Porto'),

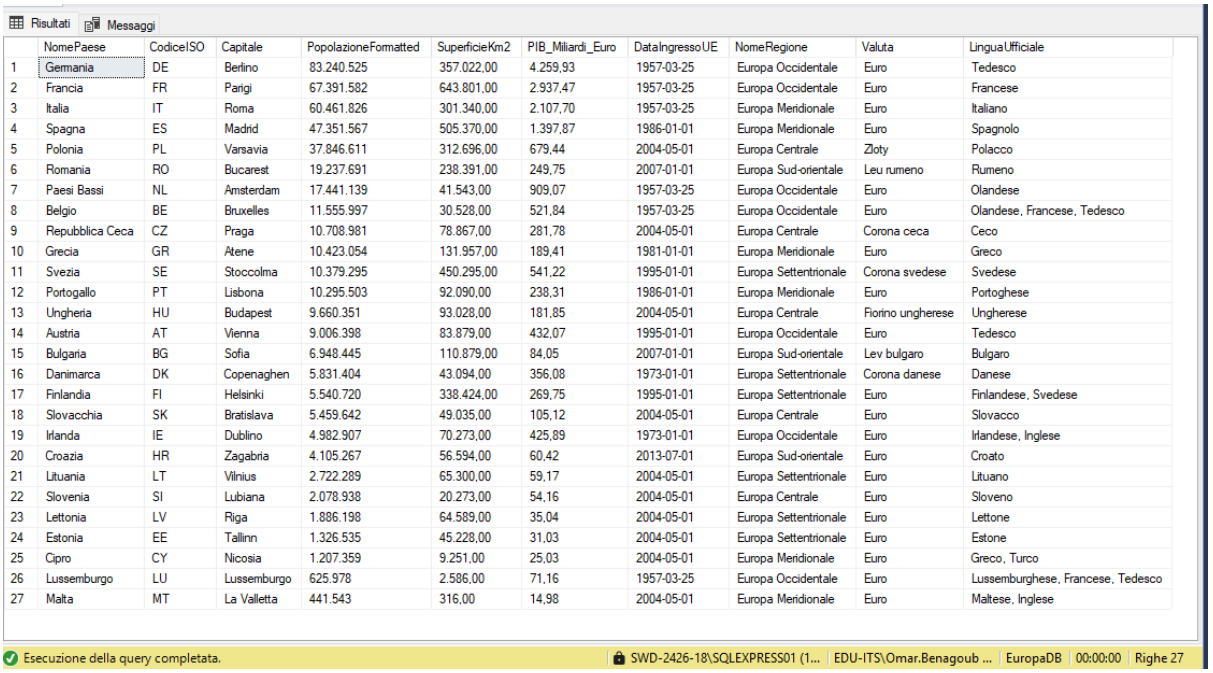
('Vila Nova de Gaia', (SELECT PaeseID FROM Paesi WHERE CodiceISO = 'PT'), 302295, 168.46, 0, 41.123981, -8.611108, 'Vino, Industria'),
('Amadora', (SELECT PaeseID FROM Paesi WHERE CodiceISO = 'PT'), 175872, 23.77, 0, 38.754190, -9.230243, 'Residenziale, Servizi'),
('Braga', (SELECT PaeseID FROM Paesi WHERE CodiceISO = 'PT'), 192494, 183.40, 0, 41.550323, -8.420546, 'Tessile, Tecnologia');

-- GRECIA - Prime 5 città
INSERT INTO Citta (NomeCitta, PaeseID, Popolazione, Superficie, IsCapitale, Latitudine, Longitudine, SettoreEconomicoPrincipale) VALUES
('Atene', (SELECT PaeseID FROM Paesi WHERE CodiceISO = 'GR'), 3153355, 412.0, 1, 37.975334, 23.734151, 'Turismo, Servizi'),
('Salonicco', (SELECT PaeseID FROM Paesi WHERE CodiceISO = 'GR'), 1110312, 111.703, 0, 40.640063, 22.944419, 'Porto, Università'),
('Patrasso', (SELECT PaeseID FROM Paesi WHERE CodiceISO = 'GR'), 213984, 125.42, 0, 38.244289, 21.734573, 'Porto, Università'),
('Larissa', (SELECT PaeseID FROM Paesi WHERE CodiceISO = 'GR'), 144651, 122.586, 0, 39.639565, 22.419807, 'Agricoltura, Trasporti'),
('Volos', (SELECT PaeseID FROM Paesi WHERE CodiceISO = 'GR'), 144449, 388.02, 0, 39.362077, 22.942665, 'Porto, Industria');

QUERIES:

-- Query 1: Elenco di tutti i paesi dell'UE con le loro informazioni principali

```
SELECT
    p.NomePaese,
    p.CodiceISO,
    p.Capitale,
    FORMAT(p.Popolazione, 'N0', 'it-IT') AS PopolazioneFormatted,
    FORMAT(p.Superficie, 'N2', 'it-IT') AS SuperficieKm2,
    FORMAT(p.PIB, 'N2', 'it-IT') AS PIB_Miliardi_Euro,
    p.DataIngressoUE,
    r.NomeRegione,
    p.Valuta,
    p.LinguaUfficiale
FROM Paesi p
INNER JOIN Regioni r ON p.RegionID = r.RegionID
ORDER BY p.Popolazione DESC;
```



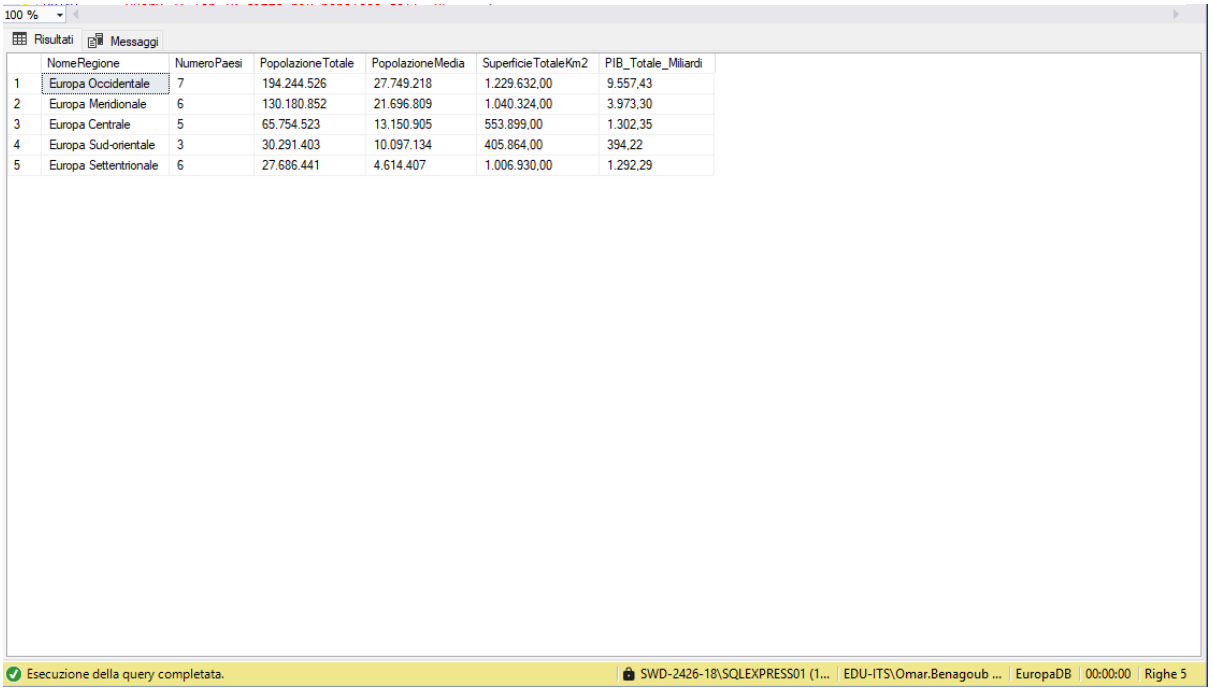
Risultati | Messaggi

	NomePaese	CodiceISO	Capitale	PopolazioneFormatted	SuperficieKm2	PIB_Miliardi_Euro	DataIngressoUE	NomeRegione	Valuta	LinguaUfficiale
1	Germania	DE	Berlino	83.240.525	357.022,00	4.259,93	1957-03-25	Europa Occidentale	Euro	Tedesco
2	Francia	FR	Parigi	67.391.582	643.801,00	2.937,47	1957-03-25	Europa Occidentale	Euro	Francese
3	Italia	IT	Roma	60.461.826	301.340,00	2.107,70	1957-03-25	Europa Meridionale	Euro	Italiano
4	Spagna	ES	Madrid	47.351.567	505.370,00	1.397,87	1986-01-01	Europa Meridionale	Euro	Spagnolo
5	Polonia	PL	Varsavia	37.846.611	312.696,00	679,44	2004-05-01	Europa Centrale	Zloty	Polacco
6	Romania	RO	Bucarest	19.237.691	238.391,00	249,75	2007-01-01	Europa Sud-orientale	Leu rumeno	Rumeno
7	Paesi Bassi	NL	Amsterdam	17.441.139	41.543,00	909,07	1957-03-25	Europa Occidentale	Euro	Olandese
8	Belgio	BE	Bruxelles	11.555.997	30.528,00	521,84	1957-03-25	Europa Occidentale	Euro	Olandese, Francese, Tedesco
9	Repubblica Ceca	CZ	Praga	10.708.981	78.867,00	281,78	2004-05-01	Europa Centrale	Corona ceca	Ceco
10	Grecia	GR	Atene	10.423.054	131.957,00	189,41	1981-01-01	Europa Meridionale	Euro	Greco
11	Svezia	SE	Stoccolma	10.379.295	450.295,00	541,22	1995-01-01	Europa Settentrionale	Corona svedese	Svedese
12	Portogallo	PT	Lisbona	10.295.503	92.090,00	238,31	1986-01-01	Europa Meridionale	Euro	Portoghese
13	Ungheria	HU	Budapest	9.660.351	93.028,00	181,85	2004-05-01	Europa Centrale	Fiorino ungherese	Ungherese
14	Austria	AT	Vienna	9.006.398	83.879,00	432,07	1995-01-01	Europa Occidentale	Euro	Tedesco
15	Bulgaria	BG	Sofia	6.948.445	110.879,00	84,05	2007-01-01	Europa Sud-orientale	Lev bulgaro	Bulgaro
16	Danimarca	DK	Copenaghen	5.831.404	43.094,00	356,08	1973-01-01	Europa Settentrionale	Corona danese	Danese
17	Finlandia	FI	Helsinki	5.540.720	338.424,00	269,75	1995-01-01	Europa Settentrionale	Euro	Finlandese, Svedese
18	Slovacchia	SK	Bratislava	5.459.642	49.035,00	105,12	2004-05-01	Europa Centrale	Euro	Slovacco
19	Irlanda	IE	Dublino	4.982.907	70.273,00	425,89	1973-01-01	Europa Occidentale	Euro	Irlandese, Inglese
20	Croazia	HR	Zagabria	4.105.267	56.594,00	60,42	2013-07-01	Europa Sud-orientale	Euro	Croato
21	Lituania	LT	Vilnius	2.722.289	65.300,00	59,17	2004-05-01	Europa Settentrionale	Euro	Lituano
22	Slovenia	SI	Lubiana	2.078.938	20.273,00	54,16	2004-05-01	Europa Centrale	Euro	Sloveno
23	Lettonia	LV	Riga	1.886.198	64.589,00	35,04	2004-05-01	Europa Settentrionale	Euro	Lettone
24	Estonia	EE	Tallinn	1.326.535	45.228,00	31,03	2004-05-01	Europa Settentrionale	Euro	Estone
25	Cipro	CY	Nicosia	1.207.359	9.251,00	25,03	2004-05-01	Europa Meridionale	Euro	Greco, Turco
26	Lussemburgo	LU	Lussemburgo	625.978	2.586,00	71,16	1957-03-25	Europa Occidentale	Euro	Lussemburghese, Francese, Tedesco
27	Malta	MT	La Valletta	441.543	316,00	14,98	2004-05-01	Europa Meridionale	Euro	Maltese, Inglese

Esecuzione della query completata. | SWD-2426-18\SQLEXPRESS01 (1... | EDU-ITS\Omar.Benagoub ... | EuropaDB | 00:00:00 | Righe 27

-- Query 2: Paesi per regione con statistiche aggregate

```
SELECT
    r.NomeRegione,
    COUNT(p.PaeseID) AS NumeroPaesi,
    FORMAT(SUM(p.Popolazione), 'N0', 'it-IT') AS PopolazioneTotale,
    FORMAT(AVG(CAST(p.Popolazione AS FLOAT)), 'N0', 'it-IT') AS PopolazioneMedia,
    FORMAT(SUM(p.Superficie), 'N2', 'it-IT') AS SuperficieTotaleKm2,
    FORMAT(SUM(p.PIB), 'N2', 'it-IT') AS PIB_Totale_Miliardi
FROM Regioni r
INNER JOIN Paesi p ON r.RegionID = p.RegionID
GROUP BY r.RegionID, r.NomeRegione
ORDER BY SUM(p.Popolazione) DESC;
```



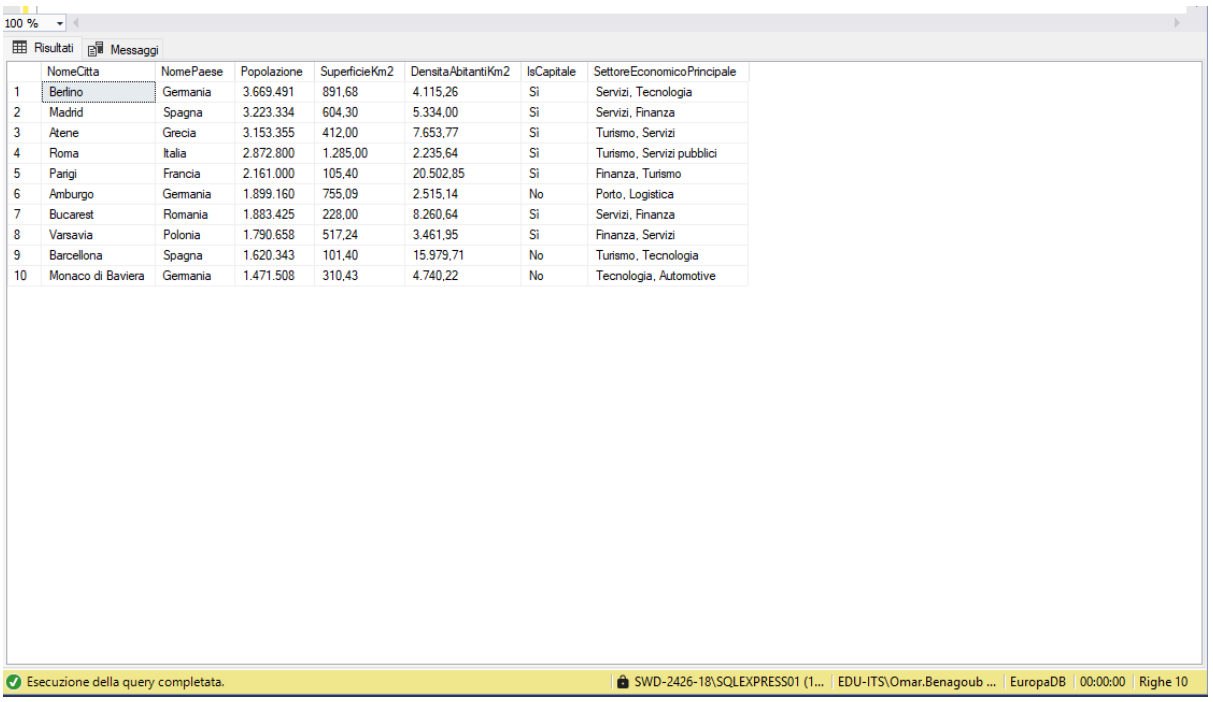
100 %

Risultati | Messaggi

	NomeRegione	NumeroPaesi	PopolazioneTotale	PopolazioneMedia	SuperficieTotaleKm2	PIB_Totale_Miliardi
1	Europa Occidentale	7	194.244.526	27.749.218	1.229.632,00	9.557,43
2	Europa Meridionale	6	130.180.852	21.696.809	1.040.324,00	3.973,30
3	Europa Centrale	5	65.754.523	13.150.905	553.899,00	1.302,35
4	Europa Sud-orientale	3	30.291.403	10.097.134	405.864,00	394,22
5	Europa Settentrionale	6	27.686.441	4.614.407	1.006.930,00	1.292,29

Esecuzione della query completata. | SWD-2426-18\SQLEXPRESS01 (1... | EDU-ITS\Omar.Benagoub ... | EuropaDB | 00:00:00 | Righe 5

```
-- Query 3: Top 10 città più popolose dell'UE
PRINT '=== Query 3: Top 10 città più popolose dell''UE ===';
SELECT TOP 10
    c.NomeCitta,
    p.NomePaese,
    FORMAT(c.Popolazione, 'N0', 'it-IT') AS Popolazione,
    FORMAT(c.Superficie, 'N2', 'it-IT') AS SuperficieKm2,
    FORMAT(c.DensitaPopolazione, 'N2', 'it-IT') AS DensitaAbitantiKm2,
    CASE WHEN c.IsCapitale = 1 THEN 'Sì' ELSE 'No' END AS IsCapitale,
    c.SettoreEconomicoPrincipale
FROM Citta c
INNER JOIN Paesi p ON c.PaeseID = p.PaeseID
ORDER BY c.Popolazione DESC;
```



100 %

Risultati | Messaggi

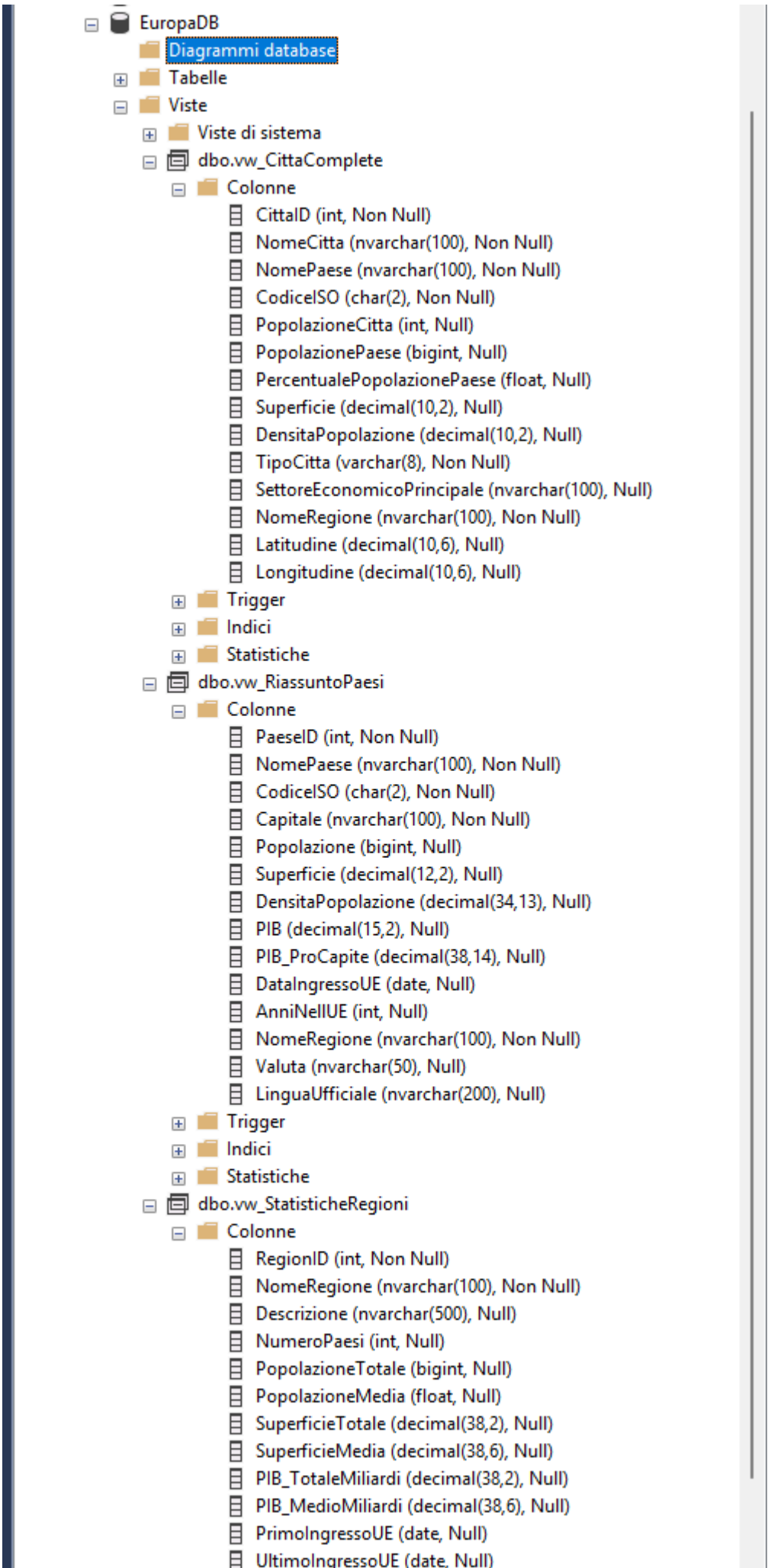
	NomeCitta	NomePaese	Popolazione	SuperficieKm2	DensitaAbitantiKm2	IsCapitale	SettoreEconomicoPrincipale
1	Berlino	Germania	3.669.491	891,68	4.115,26	Sì	Servizi, Tecnologia
2	Madrid	Spagna	3.223.334	604,30	5.334,00	Sì	Servizi, Finanza
3	Atene	Grecia	3.153.355	412,00	7.653,77	Sì	Turismo, Servizi
4	Roma	Italia	2.872.800	1.285,00	2.235,64	Sì	Turismo, Servizi pubblici
5	Parigi	Francia	2.161.000	105,40	20.502,85	Sì	Finanza, Turismo
6	Amburgo	Germania	1.899.160	755,09	2.515,14	No	Porto, Logistica
7	Bucarest	Romania	1.883.425	228,00	8.260,64	Sì	Servizi, Finanza
8	Varsavia	Polonia	1.790.658	517,24	3.461,95	Sì	Finanza, Servizi
9	Barcellona	Spagna	1.620.343	101,40	15.979,71	No	Turismo, Tecnologia
10	Monaco di Baviera	Germania	1.471.508	310,43	4.740,22	No	Tecnologia, Automotive

Esecuzione della query completata. | SWD-2426-18\SQLEXPRESS01 (1... | EDU-ITS\Omar.Benagoub ... | EuropaDB | 00:00:00 | Righe 10

VIEWS:

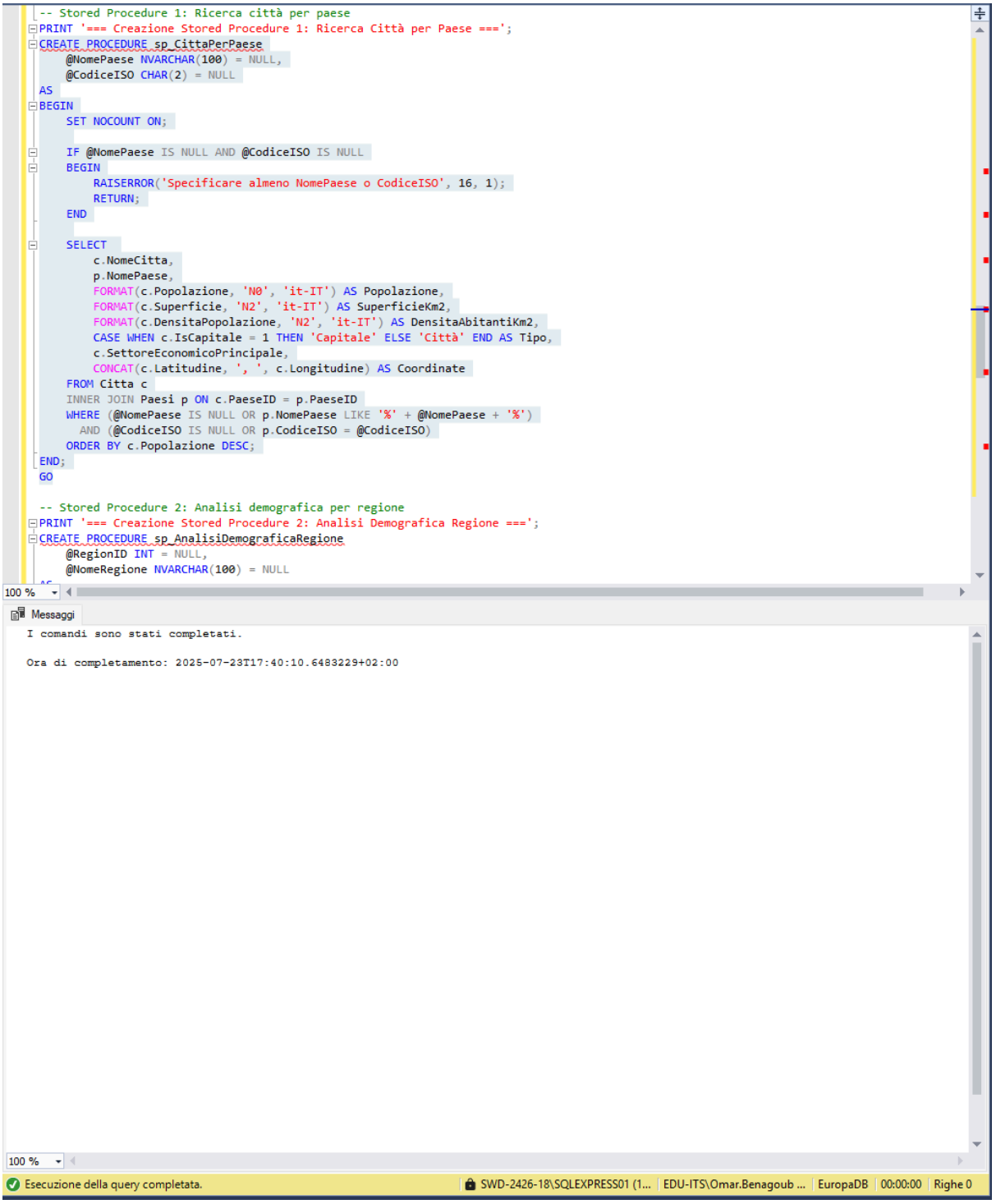
```
-- View 1: Vista riassuntiva paesi con informazioni geografiche ed economiche
PRINT '=== Creazione View 1: Riassunto Paesi ===';
CREATE VIEW vw_RiassuntoPaesi AS
SELECT
    p.PaeseID,
    p.NomePaese,
    p.CodiceISO,
    p.Capitale,
    p.Popolazione,
    p.Superficie,
    ROUND(p.Popolazione / p.Superficie, 2) AS DensitaPopolazione,
    p.PIB,
    ROUND(p.PIB * 1000000000 / p.Popolazione, 2) AS PIB_ProCapite,
    p.DataIngressoUE,
    DATEDIFF(YEAR, p.DataIngressoUE, GETDATE()) AS AnniNellUE,
    r.NomeRegione,
    p.Valuta,
    p.LinguaUfficiale
FROM Paesi p
INNER JOIN Regioni r ON p.RegionID = r.RegionID;
GO

-- View 2: Vista delle città con informazioni del paese
PRINT '=== Creazione View 2: Città Complete ===';
```

```
CREATE VIEW vw_CittaComplete AS
SELECT
    c.CittaID,
    c.NomeCitta,
    p.NomePaese,
    p.CodiceISO,
    c.Popolazione AS PopolazioneCitta,
    p.Popolazione AS PopolazionePaese,
    ROUND((CAST(c.Popolazione AS FLOAT) / CAST(p.Popolazione AS FLOAT)) * 100, 2)
AS PercentualePopolazionePaese,
    c.Superficie,
    c.DensitaPopolazione,
    CASE WHEN c.IsCapitale = 1 THEN 'Capitale' ELSE 'Città' END AS TipoCitta,
    c.SettoreEconomicoPrincipale,
    r.NomeRegione,
    c.Latitudine,
    c.Longitudine
FROM Citta c
INNER JOIN Paesi p ON c.PaeseID = p.PaeseID
INNER JOIN Regioni r ON p.RegionID = r.RegionID;
GO

-- View 3: Vista statistiche per regione
PRINT '=== Creazione View 3: Statistiche Regioni ===';
CREATE VIEW vw_StatisticheRegioni AS
SELECT
    r.RegionID,
    r.NomeRegione,
    r.Descrizione,
    COUNT(p.PaeseID) AS NumeroPaesi,
    SUM(p.Popolazione) AS PopolazioneTotale,
    AVG(CAST(p.Popolazione AS FLOAT)) AS PopolazioneMedia,
    SUM(p.Superficie) AS SuperficieTotale,
    AVG(p.Superficie) AS SuperficieMedia,
    SUM(p.PIB) AS PIB_TotaleMiliardi,
    AVG(p.PIB) AS PIB_MedioMiliardi,
    MIN(p.DataIngressoUE) AS PrimoIngressoUE,
    MAX(p.DataIngressoUE) AS UltimoIngressoUE
FROM Regioni r
INNER JOIN Paesi p ON r.RegionID = p.RegionID
GROUP BY r.RegionID, r.NomeRegione, r.Descrizione;
GO
```


EuropaDB
Diagrammi database
Tabelle
Viste
Viste di sistema
dbo.vw_CittaComplete
Colonne
CittaID (int, Non Null)
NomeCitta (nvarchar(100), Non Null)
NomePaese (nvarchar(100), Non Null)
CodiceISO (char(2), Non Null)
PopolazioneCitta (int, Null)
PopolazionePaese (bigint, Null)
PercentualePopolazionePaese (float, Null)
Superficie (decimal(10,2), Null)
DensitaPopolazione (decimal(10,2), Null)
TipoCitta (varchar(8), Non Null)
SettoreEconomicoPrincipale (nvarchar(100), Null)
NomeRegione (nvarchar(100), Non Null)
Latitudine (decimal(10,6), Null)
Longitudine (decimal(10,6), Null)
Trigger
Indici
Statistiche
dbo.vw_RiassuntoPaesi
Colonne
PaeseID (int, Non Null)
NomePaese (nvarchar(100), Non Null)
CodiceISO (char(2), Non Null)
Capitale (nvarchar(100), Non Null)
Popolazione (bigint, Null)
Superficie (decimal(12,2), Null)
DensitaPopolazione (decimal(34,13), Null)
PIB (decimal(15,2), Null)
PIB_ProCapite (decimal(38,14), Null)
DataIngressoUE (date, Null)
AnniNellUE (int, Null)
NomeRegione (nvarchar(100), Non Null)
Valuta (nvarchar(50), Null)
LinguaUfficiale (nvarchar(200), Null)
Trigger
Indici
Statistiche
dbo.vw_StatisticheRegioni
Colonne
RegionID (int, Non Null)
NomeRegione (nvarchar(100), Non Null)
Descrizione (nvarchar(500), Null)
NumeroPaesi (int, Null)
PopolazioneTotale (bigint, Null)
PopolazioneMedia (float, Null)
SuperficieTotale (decimal(38,2), Null)
SuperficieMedia (decimal(38,6), Null)
PIB_TotaleMiliardi (decimal(38,2), Null)
PIB_MedioMiliardi (decimal(38,6), Null)
PrimoIngressoUE (date, Null)
UltimoIngressoUE (date, Null)

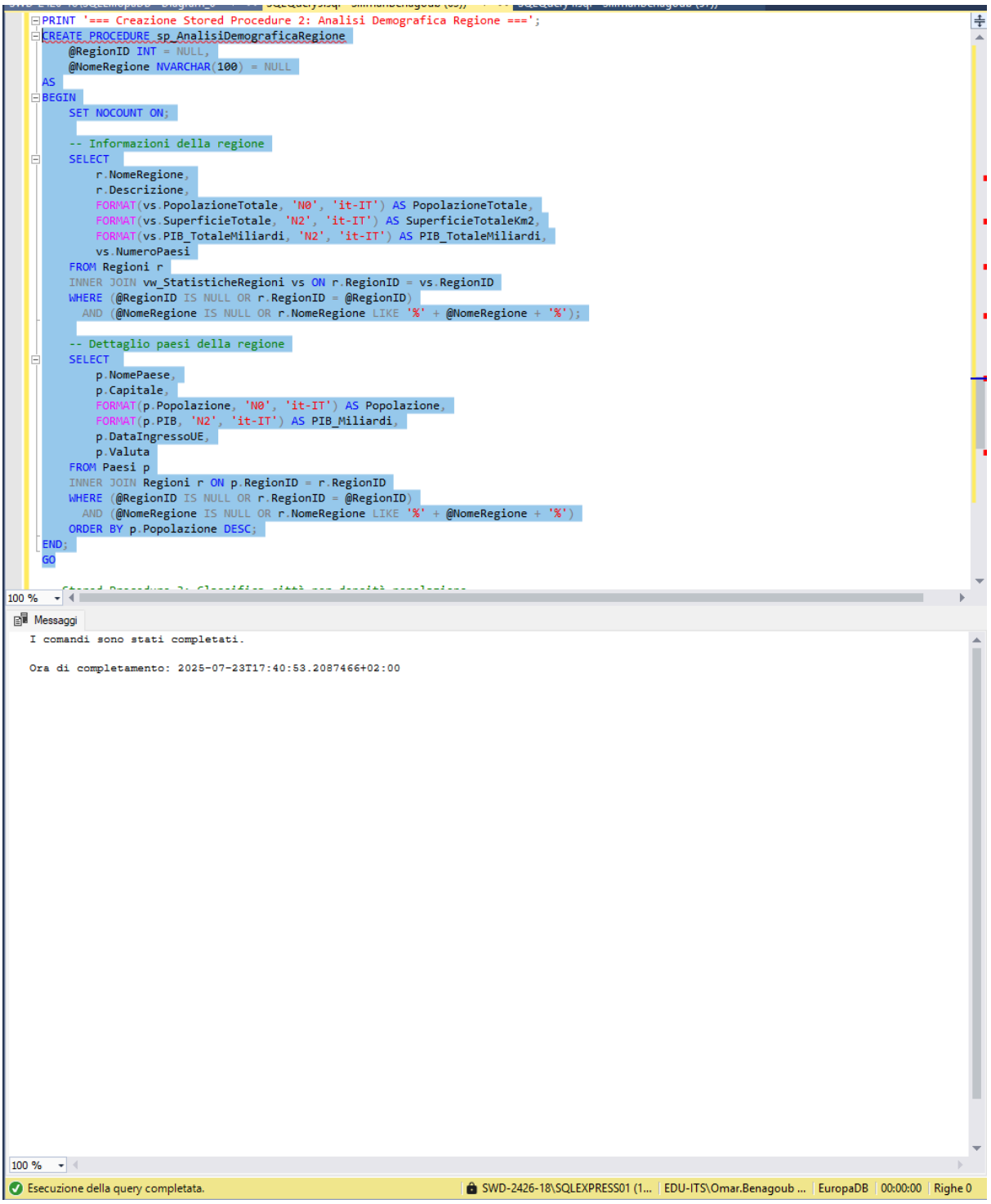
STORED PROCEDURES:

```
-- Stored Procedure 1: Ricerca città per paese
PRINT '=== Creazione Stored Procedure 1: Ricerca Città per Paese ===';
CREATE PROCEDURE sp_CittaPerPaese
    @NomePaese NVARCHAR(100) = NULL,
    @CodiceISO CHAR(2) = NULL
AS
BEGIN
    SET NOCOUNT ON;

    IF @NomePaese IS NULL AND @CodiceISO IS NULL
    BEGIN
        RAISERROR('Specificare almeno NomePaese o CodiceISO', 16, 1);
        RETURN;
    END

    SELECT
        c.NomeCitta,
        p.NomePaese,
        FORMAT(c.Popolazione, 'N0', 'it-IT') AS Popolazione,
        FORMAT(c.Superficie, 'N2', 'it-IT') AS SuperficieKm2,
        FORMAT(c.DensitaPopolazione, 'N2', 'it-IT') AS DensitaAbitantiKm2,
        CASE WHEN c.IsCapitale = 1 THEN 'Capitale' ELSE 'Città' END AS Tipo,
        c.SettoreEconomicoPrincipale,
        CONCAT(c.Latitudine, ', ', c.Longitudine) AS Coordinate
    FROM Citta c
    INNER JOIN Paesi p ON c.PaeseID = p.PaeseID
    WHERE (@NomePaese IS NULL OR p.NomePaese LIKE '%' + @NomePaese + '%')
      AND (@CodiceISO IS NULL OR p.CodiceISO = @CodiceISO)
    ORDER BY c.Popolazione DESC;
END;
GO
```



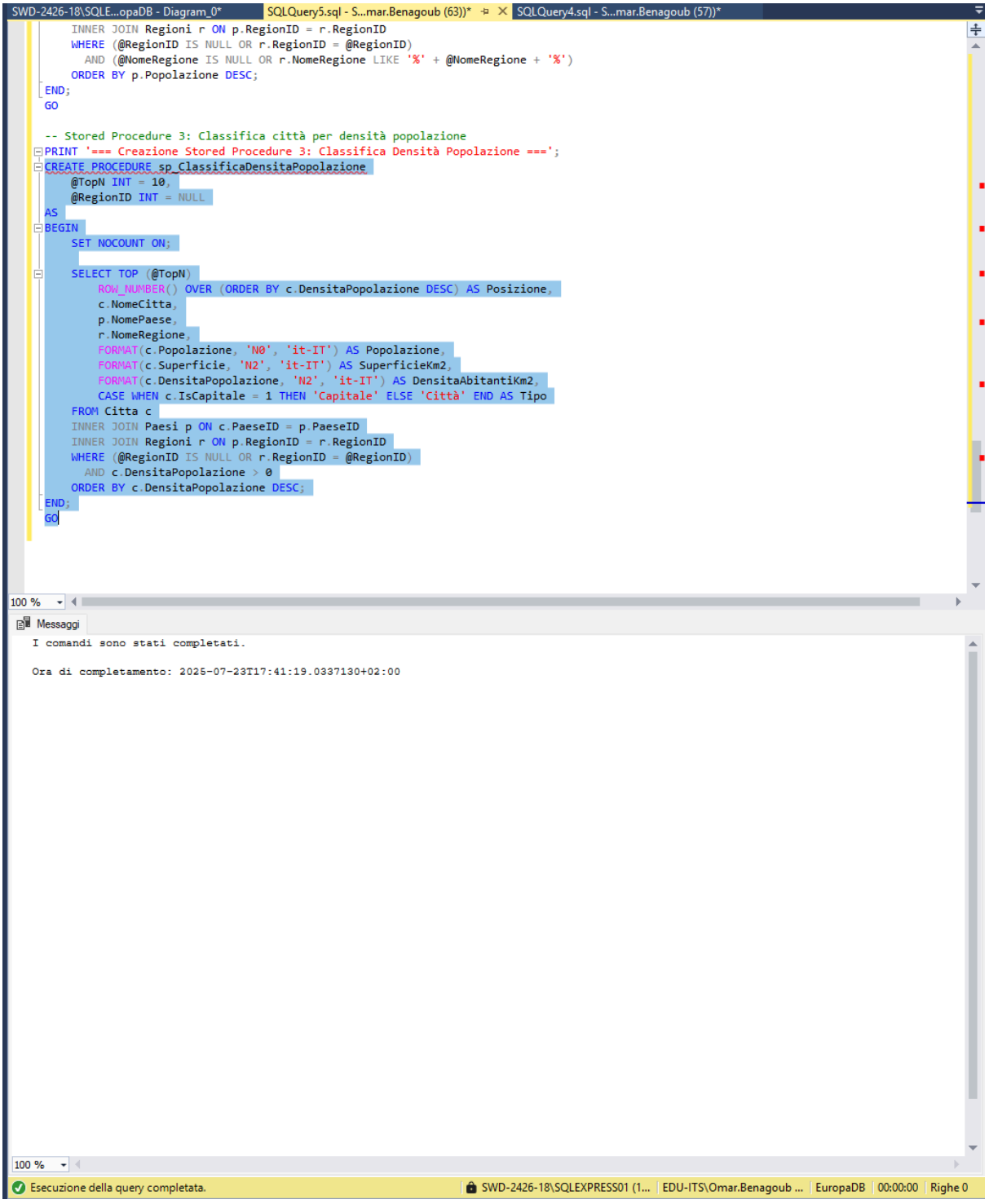

```
-- Stored Procedure 2: Analisi demografica per regione
PRINT '=== Creazione Stored Procedure 2: Analisi Demografica Regione ===';
CREATE PROCEDURE sp_AnalisiDemograficaRegione
    @RegionID INT = NULL,
    @NomeRegione NVARCHAR(100) = NULL
AS
BEGIN
    SET NOCOUNT ON;

    -- Informazioni della regione
```

```
    SELECT
        r.NomeRegione,
        r.Descrizione,
        FORMAT(vs.PopolazioneTotale, 'N0', 'it-IT') AS PopolazioneTotale,
        FORMAT(vs.SuperficieTotale, 'N2', 'it-IT') AS SuperficieTotaleKm2,
        FORMAT(vs.PIB_TotaleMiliardi, 'N2', 'it-IT') AS PIB_TotaleMiliardi,
        vs.NumeroPaesi
    FROM Regioni r
    INNER JOIN vw_StatisticheRegioni vs ON r.RegionID = vs.RegionID
    WHERE (@RegionID IS NULL OR r.RegionID = @RegionID)
      AND (@NomeRegione IS NULL OR r.NomeRegione LIKE '%' + @NomeRegione + '%');

    -- Dettaglio paesi della regione
    SELECT
        p.NomePaese,
        p.Capitale,
        FORMAT(p.Popolazione, 'N0', 'it-IT') AS Popolazione,
        FORMAT(p.PIB, 'N2', 'it-IT') AS PIB_Miliardi,
        p.DataIngressoUE,
        p.Valuta
    FROM Paesi p
    INNER JOIN Regioni r ON p.RegionID = r.RegionID
    WHERE (@RegionID IS NULL OR r.RegionID = @RegionID)
      AND (@NomeRegione IS NULL OR r.NomeRegione LIKE '%' + @NomeRegione + '%')
    ORDER BY p.Popolazione DESC;
END;
GO
```




```
-- Stored Procedure 3: Classifica città per densità popolazione
PRINT '=== Creazione Stored Procedure 3: Classifica Densità Popolazione ===';
CREATE PROCEDURE sp_ClassificaDensitaPopolazione
    @TopN INT = 10,
    @RegionID INT = NULL
AS
BEGIN
    SET NOCOUNT ON;

    SELECT TOP (@TopN)
```

```
        ROW_NUMBER() OVER (ORDER BY c.DensitaPopolazione DESC) AS Posizione,
        c.NomeCitta,
        p.NomePaese,
        r.NomeRegione,
        FORMAT(c.Popolazione, 'N0', 'it-IT') AS Popolazione,
        FORMAT(c.Superficie, 'N2', 'it-IT') AS SuperficieKm2,
        FORMAT(c.DensitaPopolazione, 'N2', 'it-IT') AS DensitaAbitantiKm2,
        CASE WHEN c.IsCapitale = 1 THEN 'Capitale' ELSE 'Città' END AS Tipo
    FROM Citta c
    INNER JOIN Paesi p ON c.PaeseID = p.PaeseID
    INNER JOIN Regioni r ON p.RegionID = r.RegionID
    WHERE (@RegionID IS NULL OR r.RegionID = @RegionID)
      AND c.DensitaPopolazione > 0
    ORDER BY c.DensitaPopolazione DESC;
END;
GO
```



backup

IF NOT EXISTS (SELECT name FROM sys.databases WHERE name = @DatabaseName)
BEGIN
    PRINT 'ERRORE: Database ' + @DatabaseName + ' non trovato!';

```
    PRINT 'Assicurarsi che il database sia stato creato correttamente.';
    RETURN;
END

PRINT 'Database ' + @DatabaseName + ' trovato. Procedendo con il backup...';
PRINT '';

-- ================================================================
-- BACKUP COMPLETO
-- ================================================================

DECLARE @FullBackupPath NVARCHAR(500);
SET @FullBackupPath = @BackupBasePath + @DatabaseName + '_Full_' + @TimeStamp
+ '.bak';

PRINT '=== BACKUP COMPLETO ===';
PRINT 'Percorso: ' + @FullBackupPath;
PRINT 'Inizio backup: ' + CONVERT(NVARCHAR(20), GETDATE(), 120);

BEGIN TRY
    BACKUP DATABASE EuropaDB
    TO DISK = @FullBackupPath
    WITH
        FORMAT,
        INIT,
        NAME = 'EuropaDB Full Backup ' + @TimeStamp,
        DESCRIPTION = 'Backup completo database Europa - paesi UE e città',
        COMPRESSION,
        CHECKSUM,
        STATS = 10;

    PRINT 'Backup completato con successo!';
    PRINT 'File: ' + @FullBackupPath;
    PRINT 'Completato: ' + CONVERT(NVARCHAR(20), GETDATE(), 120);

    -- Verifica backup
    RESTORE VERIFYONLY FROM DISK = @FullBackupPath;
    PRINT 'Verifica backup: OK';

END TRY
BEGIN CATCH
    PRINT 'ERRORE durante il backup:';
    PRINT 'Messaggio: ' + ERROR_MESSAGE();
    PRINT 'Numero: ' + CAST(ERROR_NUMBER() AS NVARCHAR(10));
    PRINT 'Severità: ' + CAST(ERROR_SEVERITY() AS NVARCHAR(10));
END CATCH

PRINT '';
```

```
-- ================================================================
-- INFORMAZIONI SUL BACKUP
-- ================================================================

PRINT '=== INFORMAZIONI BACKUP ===';

-- Dimensione database
SELECT
    name AS NomeFile,
    type_desc AS Tipo,
    size * 8 / 1024 AS DimensioneMB,
    physical_name AS PercorsoFisico
FROM sys.master_files
WHERE database_id = DB_ID(@DatabaseName);

PRINT '';

-- Statistiche backup recenti
PRINT '=== ULTIMI BACKUP ===';
SELECT TOP 5
    database_name AS Database,
    backup_start_date AS InizioBackup,
    backup_finish_date AS FineBackup,
    CASE type
        WHEN 'D' THEN 'Full'
        WHEN 'I' THEN 'Differential'
        WHEN 'L' THEN 'Log'
    END AS TipoBackup,
    backup_size / 1024 / 1024 AS DimensioneMB,
    compressed_backup_size / 1024 / 1024 AS DimensioneCompressa,
    physical_device_name AS PercorsoFile
FROM msdb.dbo.backupset bs
INNER JOIN msdb.dbo.backupmediafamily bmf ON bs.media_set_id = bmf.media_set_id
WHERE database_name = @DatabaseName
ORDER BY backup_finish_date DESC;

-- ================================================================
-- SCRIPT DI RESTORE (PER RIFERIMENTO)
-- ================================================================

PRINT '';
PRINT '=== SCRIPT DI RESTORE (PER RIFERIMENTO) ===';
PRINT '-- Per ripristinare il database utilizzare:';
PRINT '-- RESTORE DATABASE EuropaDB FROM DISK = ''' + @FullBackupPath + '''';
PRINT '-- WITH REPLACE, CHECKSUM;';
PRINT '';
```