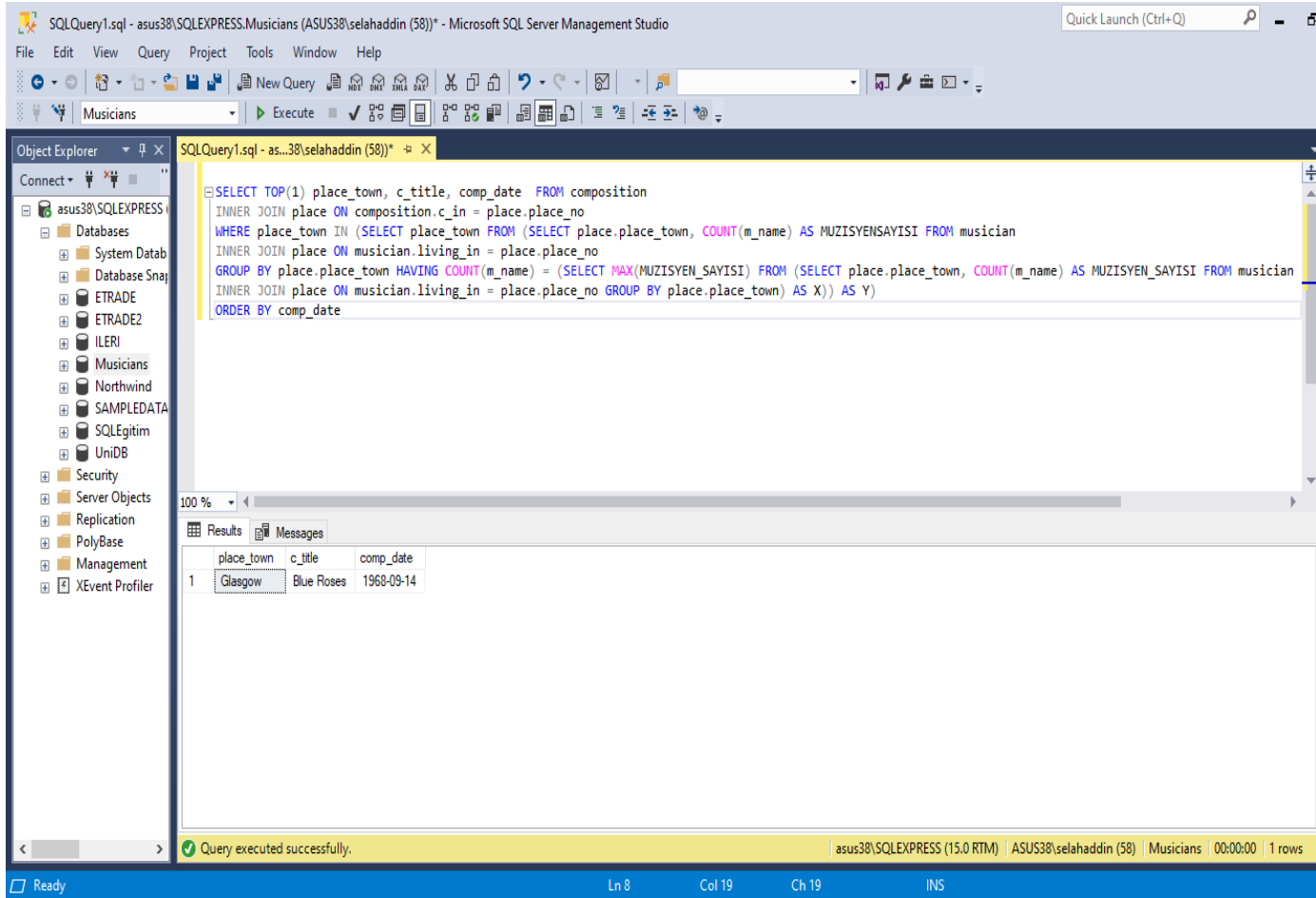


- En çok müzisyenin yaşadığı şehirde bestelenmiş eserler içinde en eski olanı bul (Find the oldest composition which was composed in the town that has the most musicians as a resident). İpucu: Engine MySQL seçili olduğunda SQL Server'daki TOP komutuna benzer LIMIT komutunu kullanabilirsiniz.



The screenshot shows the Microsoft SQL Server Management Studio interface. The query editor contains the following SQL code:

```
SELECT TOP(1) place_town, c_title, comp_date FROM composition
INNER JOIN place ON composition.c_in = place.place_no
WHERE place_town IN (SELECT place_town FROM (SELECT place.place_town, COUNT(m_name) AS MUZISYENSAYISI FROM musician
INNER JOIN place ON musician.living_in = place.place_no
GROUP BY place.place_town HAVING COUNT(m_name) = (SELECT MAX(MUZISYEN_SAYISI) FROM (SELECT place.place_town, COUNT(m_name) AS MUZISYEN_SAYISI FROM musician
INNER JOIN place ON musician.living_in = place.place_no GROUP BY place.place_town) AS X)) AS Y)
ORDER BY comp_date
```

The Results pane shows the following data:

	place_town	c_title	comp_date
1	Glasgow	Blue Roses	1968-09-14

The status bar at the bottom indicates "Query executed successfully." and "asus38\SQLEXPRESS (15.0 RTM) | ASUS38\selahaddin (58) | Musicians | 00:00:00 | 1 rows".

```
SELECT TOP(1) place_town, c_title, comp_date FROM composition
INNER JOIN place ON composition.c_in = place.place_no
WHERE place_town IN (SELECT place_town FROM (SELECT place.place_town, COUNT(m_name) AS
MUZISYENSAYISI FROM musician
INNER JOIN place ON musician.living_in = place.place_no
GROUP BY place.place_town HAVING COUNT(m_name) = (SELECT MAX(MUZISYEN_SAYISI) FROM
(SELECT place.place_town, COUNT(m_name) AS MUZISYEN_SAYISI FROM musician
INNER JOIN place ON musician.living_in = place. place_no GROUP BY place.place_town) AS
X)) AS Y )
ORDER BY comp_date
```

- ABD doğumlu olup ABD dışında kurulmuş bir grupta gitar çalarak performans sergilemiş müzisyenlerin adını ve grubunu listeleyin. (List the musicians and their bands, who have played guitar and were born in USA but have performed in a band whose home is outside USA).

The screenshot shows the Microsoft SQL Server Management Studio interface. The title bar indicates the file is 'SQLQuery1.sql - asus38\SQLEXPRESS.Musicians (ASUS38\selahaddin (76))* - Microsoft SQL Server Management Studio'. The menu bar includes File, Edit, View, Query, Project, Tools, Window, and Help. The toolbar contains various icons for file operations, query execution, and formatting. The Object Explorer on the left shows the database structure, including 'asus38\SQLEXPRESS (SQL Server)' with folders for Databases, System Databases, Database Snapshots, and a list of databases: ETRADE, ETRADE2, ILERI, Musicians, Northwind, SAMPLEDATA, SQLEgitim, and UniDB. The main query window displays the following SQL query:

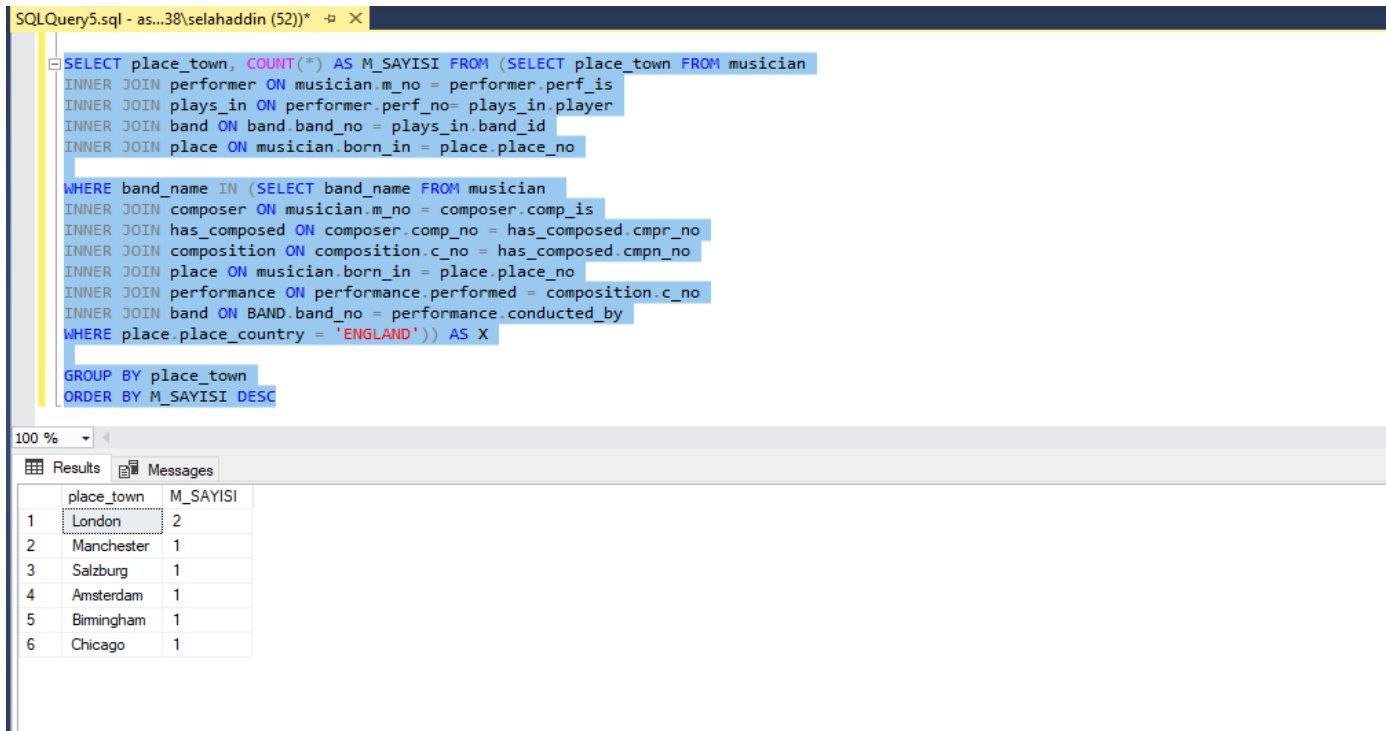
```
SELECT band.band_name, musician.m_name FROM band
INNER JOIN plays_in ON band.band_no = plays_in.band_id
INNER JOIN performer ON performer.perf_no = plays_in.player
INNER JOIN musician ON musician.m_no = performer.perf_is
WHERE band_name IN (SELECT band_name FROM band
INNER JOIN place ON band.band_home = place.place_no
WHERE place_country != 'USA') AND m_name IN (SELECT musician.m_name FROM musician
INNER JOIN place ON musician.born_in = place.place_no
WHERE place_country = 'USA' ) AND performer.instrument = 'guitar'
ORDER BY band_name
```

The Results pane at the bottom shows a single row of data:

	band_name	m_name
1	AASO	Elsie James

```
SELECT band.band_name, musician.m_name FROM band
INNER JOIN plays_in ON band.band_no = plays_in.band_id
INNER JOIN performer ON performer.perf_no = plays_in.player
INNER JOIN musician ON musician.m_no = performer.perf_is
WHERE band_name IN (SELECT band_name FROM band
INNER JOIN place ON band.band_home = place.place_no
WHERE place_country != 'USA') AND m_name IN (SELECT musician.m_name FROM musician
INNER JOIN place ON musician.born_in = place.place_no
WHERE place_country = 'USA' ) AND performer.instrument = 'guitar'
ORDER BY band_name
```

- İngiltere doğumlu bir müzisyen tarafından bestelenmiş herhangi bir eseri icra eden grupların üyelerinin yaşadıkları şehirleri bulun ve buralarda belirtilen kriterlere uyan müzisyenlerin sayılarını büyükten küçüğe sıralayın.



The screenshot shows a SQL query in a text editor window titled 'SQLQuery5.sql'. The query is a complex JOIN statement that filters musicians born in England and counts the number of bands they have performed with. The results are displayed in a table below the query.

```
SELECT place_town, COUNT(*) AS M_SAYISI FROM (SELECT place_town FROM musician
INNER JOIN performer ON musician.m_no = performer.perf_is
INNER JOIN plays_in ON performer.perf_no= plays_in.player
INNER JOIN band ON band.band_no = plays_in.band_id
INNER JOIN place ON musician.born_in = place.place_no
WHERE band_name IN (SELECT band_name FROM musician
INNER JOIN composer ON musician.m_no = composer.comp_is
INNER JOIN has_composed ON composer.comp_no = has_composed.cmpr_no
INNER JOIN composition ON composition.c_no = has_composed.cmpn_no
INNER JOIN place ON musician.born_in = place.place_no
INNER JOIN performance ON performance.performed = composition.c_no
INNER JOIN band ON BAND.band_no = performance.conducted_by
WHERE place.place_country = 'ENGLAND')) AS X
GROUP BY place_town
ORDER BY M_SAYISI DESC
```

	place_town	M_SAYISI
1	London	2
2	Manchester	1
3	Salzburg	1
4	Amsterdam	1
5	Birmingham	1
6	Chicago	1

```
SELECT place_town, COUNT(*) AS M_SAYISI FROM (SELECT place_town FROM musician
INNER JOIN performer ON musician.m_no = performer.perf_is
INNER JOIN plays_in ON performer.perf_no= plays_in.player
INNER JOIN band ON band.band_no = plays_in.band_id
INNER JOIN place ON musician.born_in = place.place_no
WHERE band_name IN (SELECT band_name FROM musician
INNER JOIN composer ON musician.m_no = composer.comp_is
INNER JOIN has_composed ON composer.comp_no = has_composed.cmpr_no
INNER JOIN composition ON composition.c_no = has_composed.cmpn_no
INNER JOIN place ON musician.born_in = place.place_no
INNER JOIN performance ON performance.performed = composition.c_no
INNER JOIN band ON BAND.band_no = performance.conducted_by
WHERE place.place_country = 'ENGLAND')) AS X
GROUP BY place_town
ORDER BY M_SAYISI DESC
```

- Doğduğu ülkede konser organize etmiş; fakat doğduğu ülkede hiç bir zaman bir grup elemanı olarak performans sergilememiş bir müzisyen var mıdır? Varsa yaşıyor mudur? Yaşıyorsa kaç yaşındadır?

The screenshot shows the Microsoft SQL Server Management Studio interface. The query editor contains a complex SQL query. The results pane at the bottom shows a single row of data.

```

SELECT m_name, died, DATEDIFF(YEAR, musician.born, GETDATE()) AS YAS FROM musician

WHERE m_name IN (SELECT S.m_name FROM (SELECT m_name, place_country FROM (SELECT X.m_name, Y.place_country FROM ((SELECT musician.m_name, place_country FROM musician
INNER JOIN place ON musician.born_in = place.place_no ) AS X
JOIN (SELECT m_name, place_country FROM place
INNER JOIN concert ON place.place_no = concert.concert_in
INNER JOIN musician ON musician.m_no = concert.concert_organiser) AS Y ON X.m_name = Y.m_name)
WHERE X.place_country = Y.place_country) AS M) AS S

JOIN (SELECT m_name, place_country FROM (SELECT DISTINCT A.m_name, B.place_country FROM ((SELECT m_name, place_country FROM musician
INNER JOIN place ON musician.born_in = place.place_no) AS A
RIGHT JOIN (SELECT m_name, place_country FROM (SELECT DISTINCT musician.m_name, band_name, place_country FROM band
INNER JOIN plays_in ON band.band_no = plays_in.band_id
INNER JOIN performer ON performer.perf_no = plays_in.player
INNER JOIN musician ON musician.m_no = performer.perf_is
INNER JOIN place ON place.place_no = band.band_no
INNER JOIN performance ON performance.gave = band.band_no) AS C) AS B ON A.m_name =
B.m_name )
WHERE A.place_country = B.place_country) AS N) AS D ON S.m_name = D.m_name
WHERE S.place_country = D.place_country)
  
```

m_name	died	YAS
1 Hany Forte	NULL	72

```
SELECT m_name, died, DATEDIFF(YEAR, musician.born, GETDATE()) AS YAS FROM musician
```

```

WHERE m_name IN (SELECT S.m_name FROM (SELECT m_name, place_country FROM (SELECT
X.m_name, Y.place_country FROM ((SELECT musician.m_name, place_country FROM musician
INNER JOIN place ON musician.born_in = place.place_no ) AS X
JOIN (SELECT m_name, place_country FROM place
INNER JOIN concert ON place.place_no = concert.concert_in
INNER JOIN musician ON musician.m_no = concert.concert_organiser) AS Y ON X.m_name =
Y.m_name)
WHERE X.place_country = Y.place_country) AS M) AS S
JOIN (SELECT m_name, place_country FROM (SELECT DISTINCT A.m_name, B.place_country FROM
((SELECT m_name, place_country FROM musician
INNER JOIN place ON musician.born_in = place.place_no) AS A
RIGHT JOIN (SELECT m_name, place_country FROM (SELECT DISTINCT musician.m_name,
band_name, place_country FROM band
INNER JOIN plays_in ON band.band_no = plays_in.band_id
INNER JOIN performer ON performer.perf_no = plays_in.player
INNER JOIN musician ON musician.m_no = performer.perf_is
INNER JOIN place ON place.place_no = band.band_no
INNER JOIN performance ON performance.gave = band.band_no) AS C) AS B ON A.m_name =
B.m_name )
WHERE A.place_country = B.place_country) AS N) AS D ON S.m_name = D.m_name
WHERE S.place_country = D.place_country)
  
```

