```sql
/*
1.	Display Artist names and their matching Album titles. Matches are required. (86
Rows)
Use the Artist and Album tables.
Display the Artist Name as "ArtistName", and Album Title as "AlbumTitle.
Only include those artists whose name starts with A through D and have an album record in
the Album table.
Order the results by ArtistName and AlbumTitle in ascending order.

*/
SELECT
	A.Name AS ArtistName
	,AL.Title AS AlbumTitle
FROM Artist AS A
JOIN Album AS AL
	ON AL.ArtistId = A.ArtistId
WHERE A.Name LIKE '[A-D]%'
ORDER BY A.Name, AL.Title

/*
2.	Display Artist names and their matching Album titles. No matches are okay. (103
Rows)
Use the Artist and Album tables.
Display the Artist Name as "ArtistName", and Album Title as "AlbumTitle.
Include all artists whose name starts with A through D even if they don't have a record
in the Album table.
Order the results by ArtistName and AlbumTitle in ascending order.

*/
SELECT
	A.Name AS ArtistName
	,AL.Title AS AlbumTitle
FROM Artist AS A
LEFT JOIN Album AS AL
	ON AL.ArtistId = A.ArtistId
WHERE A.Name LIKE '[A-D]%'
ORDER BY A.Name, AL.Title

/*
3.	Show the Artist and Track name for tracks that have the Genre name "Alternative".
(40 Rows)
Call Artist Name "ArtistName"
Call Track Name "TrackName"
Order by ArtistName then TrackName in ascending order.

*/
SELECT
	A.Name AS ArtistName
	,T.Name AS TrackName
FROM Artist A
JOIN Album AL
	ON AL.ArtistId = A.ArtistId
JOIN Track T
	ON T.AlbumId = AL.AlbumId
JOIN Genre G
	ON G.GenreId = T.GenreId
WHERE G.Name = 'Alternative'
ORDER BY A.Name, T.Name
```

```sql
/*
4.	Create a cartesian product using the first and last names of the Employee table.
(64 Rows)
Display FirstName and LastName.
Each first name should have a record match with every last name in the table.
Hint:  You'll need to use a self-join.

*/
SELECT
	E1.FirstName
	,E2.LastName
FROM Employee E1
CROSS JOIN Employee E2

/*
5.	Display the Artist, Album, Track and Genre names of the tracks on the "Grunge"
playlist. (15 Rows)
Use the Playlist table to identify "Grunge" tracks.
Call Playlist Name "PlaylistName"
Call Artist Name "ArtistName"
Call Album Title "AlbumName"
Call Track Name "TrackName"
Call Genre Name "GenreName"

*/
SELECT
	P.Name AS PlaylistName
	,A.Name AS ArtistName
	,AL.Title AS AlbumTitle
	,T.Name		AS TrackName
	,G.Name		AS GenreName
FROM Playlist P
JOIN PlaylistTrack PT
	ON PT.PlaylistId = P.PlaylistId
JOIN Track T
	ON T.TrackId = PT.TrackId
JOIN Genre G
	ON G.GenreId = T.GenreId
JOIN Album AL
	ON AL.AlbumId = T.AlbumId
JOIN Artist A
	ON A.ArtistId = AL.ArtistId
WHERE P.Name = 'Grunge'
```

```sql
/*
6.	Display the Album, Track and length in seconds for tracks on the “Let There Be Rock” album. (8 Rows)
Display the Album Title, Track Name and Milliseconds.
Display Milliseconds as seconds and name it Seconds.
Filter where the title = "Let There Be Rock".

*/
SELECT
	Title
	,Name
	,Milliseconds/1000 AS Seconds
FROM Album AL
JOIN Track T
	ON T.AlbumId = AL.AlbumId
WHERE AL.Title = 'Let There Be Rock'

/*
7.	Return all Customers and the Employees who are their support reps. (59 Rows)
Display Employee first and last name, Customer first and last name, and Customer country.
Concatenate the employee's first and last name with a space in between.  Call the column CustomerRep.
Concatenate the customer's first and last name with a space in between.  Call the column CustomerName.
Order by CustomerRep and the customer's Country.

*/
SELECT
	E.FirstName + ' ' + E.LastName AS CustomerRep
	,C.Country
	,CONCAT(C.Firstname,' ', C.LastName) AS CustomerName
FROM Employee E
JOIN Customer C
	ON C.SupportRepId = E.EmployeeId
ORDER BY CustomerRep, C.Country

/*
8.	Return all Track names, their Album titles and any associated Invoice IDs. (3759 Rows)
Display Album Title, Track Name, and InvoiceId.
Display the Track Name even if it does not have an associated InvoiceId.
(Note some Tracks may have more than one InvoiceId.)
Order by Track Name and InvoiceId in descending order.

*/
SELECT
	AL.Title
	,T.Name
	,IL.InvoiceId
FROM Album AL
JOIN Track T
	ON T.AlbumId = AL.AlbumId
LEFT JOIN InvoiceLine IL
	ON IL.TrackId = T.TrackId
ORDER BY Name, InvoiceId DESC
```

```sql
/*
9.	Return all employees and the name of the person to which they report. (8 Rows)
Display EmployeeId, LastName, FirstName, ReportsTo, ManagerName
Manager name is the concatenated first and last name of an employee's manager.
If employee doesn't have a manager then enter 'N/A' in the MangerName column.
Hint:  You'll need a self-join.

*/
SELECT
	E.EmployeeId
	,E.LastName
	,E.FirstName
	,E.ReportsTo
	,ISNULL(M.FirstName+' '+M.LastName,'N/A') AS ManagerName
	--,IIF(E.ReportsTo IS NULL,'N/A',CONCAT(M.FirstName,' ',M.LastName)) AS
ManagerName
FROM Employee E
LEFT JOIN Employee M
	ON M.EmployeeId = E.ReportsTo


/*
10.	Return all album tracks purchased by Julia Barnett. (38 Rows)
Display customer LastName, Album Title, Track Name, and Inventory InvoiceDate.
Display InvoiceDate in this format - dd/mm/yyyy - and rename the column PurchaseDate.
Order the records by InvoiceDate, Title and Name.

*/
SELECT
	C.LastName
	,AL.Title
	,T.Name
	,CONVERT(varchar,InvoiceDate, 103) PurchaseDate
FROM Customer C
JOIN Invoice I
	ON I.CustomerId = C.CustomerId
JOIN InvoiceLine IL
	ON IL.InvoiceId = I.InvoiceId
JOIN Track T
	ON T.TrackId = IL.TrackId
JOIN Album AL
	ON AL.AlbumId = T.AlbumId
WHERE C.FirstName = 'Julia'
	AND C.LastName = 'Barnett'
ORDER BY InvoiceDate, Title, Name
```