1. **Select all columns from the `Artist` table.**

QUERY: SELECT * FROM Artist;

2. **Select the `Name` column from the `Genre` table.**

QUERY: SELECT Name FROM Genre;

3. **Select the `Title` and `ArtistId` from the `Album` table.**

QUERY: SELECT Title, ArtistId FROM Album;

4. **Get all tracks with a `UnitPrice` greater than 0.99.**

QUERY: SELECT * FROM Tracks WHERE UnitPrice > 0.99

5. **Find all customers from the `USA from the customer`.**

QUERY: SELECT * FROM customer WHERE country = “USA”;

6. **List all employees with the title 'Sales Support Agent' from employee**

QUERY: SELECT * FROM employee WHERE title = ‘Sales Support Agent’;

7. **Get the `InvoiceId`, `CustomerId`, and `Total` for all invoices where the total is between 10 and 20 from the invoice table**

QUERY: SELECT InvoiceId, CustomerId, Total FROM Invoice WHERE Total BETWEEN 10 AND 20;

8. **Find all tracks with `GenreId` 1 or 2.**

QUERY: SELECT * FROM Track WHERE GenreId IN (1, 2);

9. **List all artists whose names start with 'A'.**

QUERY: SELECT * FROM Artist WHERE Name LIKE 'A%';

10. **Get all albums ordered alphabetically by `Title`.**

QUERY: SELECT * FROM Album ORDER BY Title ASC;

11. **Select `TrackId`, `Name`, and `Milliseconds` from `Track`, ordered by `Milliseconds` in descending order.**

QUERY: SELECT TrackId, Name, Milliseconds FROM Tracks ORDER BY Milliseconds ASC;

12. **Find the top 10 most expensive tracks.**

QUERY: SELECT TOP 10 * FROM Track ORDER BY UnitPrice DESC;

13. Get all invoices from the year 2009.

QUERY: SELECT * FROM Invoice WHERE YEAR(InvoiceDate) = 2009;

14. List all employees who were hired after '2003-01-01'

QUERY: SELECT * FROM Employee WHERE HireDate > '2003-01-01';

15. Find customers whose `FirstName` is 'Robert' or `LastName` is 'King'.

QUERY: SELECT * FROM Customer WHERE FirstName = 'Robert' AND LastName = 'King';

16. select `TrackId` and `Name` from `Track` where the `Composer` is 'AC/DC'

QUERY: SELECT TrackId, Name FROM Track WHERE Composer LIKE '%AC/DC%';

17. Get all tracks that are longer than 5 minutes

QUERY: SELECT * FROM Track WHERE Milliseconds > 300000;

18. Find all `MediaType` names.

QUERY: SELECT Name FROM MediaType;

19. Get the distinct `BillingCountry` from the `Invoice` table.

QUERY: SELECT DISTINCT BillingCountry FROM Invoice;

20. List the `Title` of all albums and the `Name` of their respective artists.

QUERY: SELECT Album.Title, Artist.Name FROM Album JOIN Artist ON Album.ArtistId = Artist.ArtistId;

21. Get the `Name` of each track and the `Title` of its album.

QUERY: SELECT Track.Name AS TrackName, Album.Title AS AlbumTitle FROM Track JOIN Album ON Track.AlbumId = Album.AlbumId;

22. Find the `Name` of each track and its `Genre` name.

QUERY: SELECT Track.Name AS TrackName, Genre.Name AS GenreName FROM Track JOIN Genre ON Track.GenreId = Genre.GenreId;

23. Get the `Name` of tracks and the `Name` of their `MediaType`

QUERY: SELECT Track.Name AS TrackName, MediaType.Name, FROM Track, JOIN MediaType ON Track.MediaTypeId = MediaType.MediaTypeId;

24. For each invoice, show the `InvoiceId`, `InvoiceDate`, `BillingCountry`, and the `FirstName` and `LastName` of the customer who made the purchase.

SELECT Invoice.InvoiceId, Invoice.InvoiceDate, Invoice.BillingCountry, Customer.FirstName, Customer.LastName FROM Invoice JOIN Customer ON Invoice.CustomerId = Customer.CustomerId;

25. Count the total number of tracks.

QUERY: SELECT COUNT(*) FROM Tracks;

26. Calculate the total sales (sum of `Total` from `Invoice`.

QUERY: SELECT SUM(Total) FROM Invoice;

27. How many customers are there in each `Country`?

SELECT Country, COUNT(*) AS NumberOfCustomers

FROM Customer

GROUP BY Country

ORDER BY NumberOfCustomers DESC;

28. What is the average `Total` of all invoices?

SELECT AVG(Total) AS AverageInvoiceTotal FROM Invoice;

29. Find the minimum and maximum `UnitPrice` of tracks.

SELECT

MIN(UnitPrice) AS MinimumPrice,

MAX(UnitPrice) AS MaximumPrice

FROM Track;

30. Count the number of tracks for each `Genre`.

SELECT Genre.Name AS GenreName, COUNT(*) AS NumberOfTracks

FROM Track

JOIN Genre ON Track.GenreId = Genre.GenreId

GROUP BY Genre.Name

ORDER BY NumberOfTracks DESC;

31. Count the number of Artist for each Genre

```
SELECT Genre.Name AS GenreName, COUNT(*) AS NumberOfTracks
FROM Track
JOIN Genre ON Track.GenreId = Genre.GenreId
GROUP BY Genre.Name
ORDER BY NumberOfTracks DESC;
```

32. Get the total `Milliseconds` for all tracks in the database.

```
QUERY: SELECT SUM(Milliseconds) AS TotalMilliseconds
FROM Track;
```

33. Find the `Name` of the artist and the number of albums they have.

```
SELECT Artist.Name AS ArtistName, COUNT(*) AS NumberOf Artist FROM Artist
JOIN ON Album Artist.AlbumId = Album.AlbumId
GROUP BY ArtistName
ORDER BY Numberof Artist DESC;
```

34. Find the `Name` of the artist and the number of albums they have.

```
SELECT Artist.Name AS ArtistName, COUNT(Album.AlbumId) AS NumberOfAlbums
FROM Artist
JOIN Album ON Artist.ArtistId = Album.ArtistId
GROUP BY Artist.Name
ORDER BY NumberOfAlbums DESC;
```

35. List the `FirstName` and `LastName` of employees and the number of customers they support.

```
SELECT
  e.FirstName,
  e.LastName,
  COUNT(c.CustomerId) AS NumberOfCustomers
```

```
FROM Employee e
LEFT JOIN Customer c ON e.EmployeeId = c.SupportRepId
GROUP BY e.FirstName, e.LastName
ORDER BY NumberOfCustomers DESC;
```

36. **For each `InvoiceId`, count the number of `InvoiceLine` items.**

```
SELECT
 InvoiceId,
 COUNT(*) AS NumberOfItems
FROM InvoiceLine
GROUP BY InvoiceId
ORDER BY NumberOfItems DESC;
```

37. Calculate the total quantity of tracks sold for each `TrackId`.

```
SELECT
 TrackId,
 SUM(Quantity) AS TotalQuantitySold
FROM InvoiceLine
GROUP BY TrackId
ORDER BY TotalQuantitySold DESC;
```

38. Find the `Name` of the playlist and the number of tracks in each playlist.

```
SELECT
 p.Name AS PlaylistName,
 COUNT(pt.TrackId) AS NumberOfTracks
FROM Playlist p
LEFT JOIN PlaylistTrack pt ON p.PlaylistId = pt.PlaylistId
GROUP BY p.Name
ORDER BY NumberOfTracks DESC;
```

39. Get the `Name` of the track, its `Album Title`, and the `Artist Name`.

```
SELECT
  Track.Name AS TrackName,
  Album.Title AS AlbumTitle,
  Artist.Name AS ArtistName
FROM Track
JOIN Album ON Track.AlbumId = Album.AlbumId
JOIN Artist  ON album.ArtistId = Artist.ArtistId
ORDER BY Artist.Name, Album.Title, Track.Name;
OR
SELECT
  t.Name AS TrackName,
  a.Title AS AlbumTitle,
  ar.Name AS ArtistName
FROM Track t
JOIN Album a ON t.AlbumId = a.AlbumId
JOIN Artist ar ON a.ArtistId = ar.ArtistId
ORDER BY ar.Name, a.Title, t.Name;
```

40. List the `FirstName`, `LastName` of customers, and the `FirstName`, `LastName` of their support representative.

```
QUERY: SELECT
  c.FirstName AS CustomerFirstName,
  c.LastName AS CustomerLastName,
  e.FirstName AS SupportRepFirstName,
  e.LastName AS SupportRepLastName
FROM Customer c
```

```
JOIN Employee e ON c.SupportRepId = e.EmployeeId
ORDER BY c.LastName, c.FirstName;
```

41. Calculate the total sales for each `BillingCountry`

```
SELECT
  BillingCountry,
  SUM(Total) AS TotalSales
FROM Invoice
GROUP BY BillingCountry
ORDER BY TotalSales DESC;
```

42. Find the `Genre` with the most tracks

```
SELECT TOP 1
  g.Name AS GenreName,
  COUNT(t.TrackId) AS NumberOfTracks
FROM Genre g
JOIN Track t ON g.GenreId = t.GenreId
GROUP BY g.Name
ORDER BY NumberOfTracks DESC;
```

43. Which `MediaType` has the most tracks?

```
SELECT TOP 1
  g.Name AS GenreName,
  COUNT(t.TrackId) AS NumberOfTracks
FROM Genre g
```

```
JOIN Track t ON g.GenreId = t.GenreId
GROUP BY g.Name
ORDER BY NumberOfTracks DESC;
```

44. **How many distinct `Composer` names are there?**

```
SELECT COUNT(DISTINCT Composer) AS DistinctComposerCount
FROM Track;
```

45. **Get the `Name` of each track and the `Name` of its `Playlist` (if it belongs to any playlist)**

```
SELECT
 t.Name AS TrackName,
 p.Name AS PlaylistName
FROM Track t
LEFT JOIN PlaylistTrack pt ON t.TrackId = pt.TrackId
LEFT JOIN Playlist p ON pt.PlaylistId = p.PlaylistId
ORDER BY t.Name, p.Name;
```

46. **List the `ArtistName` and the `Album Title` for all albums released by artists from 'AC/DC'**

```
SELECT
 ar.Name AS ArtistName,
 al.Title AS AlbumTitle
FROM Artist ar
JOIN Album al ON ar.ArtistId = al.ArtistId
WHERE ar.Name = 'AC/DC';
```

47. **Find the total sales for each `Customer`, along with their `FirstName` and `LastName`.**

```
SELECT
```

```
 c.FirstName,
 c.LastName,
 SUM(i.Total) AS TotalSales
FROM Customer c
JOIN Invoice i ON c.CustomerId = i.CustomerId
GROUP BY c.FirstName, c.LastName
ORDER BY TotalSales DESC;
```

48. **Get the `FirstName`, `LastName` of employees, and their `Title`, for employees who are not sales agents.**

```
SELECT
 FirstName,
 LastName,
 Title
FROM Employee
WHERE Title != 'Sales Support Agent' AND Title != 'Sales Manager';
```

49. **How many tracks are in the 'Classical' genre?**

```
SELECT COUNT(*) AS ClassicalTrackCount
FROM Track t
JOIN Genre g ON t.GenreId = g.GenreId
WHERE g.Name = 'Classical';
```

50. **Find the `Name` of artists who have at least 10 albums.**

```
SELECT
 ar.Name AS ArtistName,
 COUNT(al.AlbumId) AS AlbumCount
FROM Artist ar
JOIN Album al ON ar.ArtistId = al.ArtistId
```

```
GROUP BY ar.Name
HAVING COUNT(al.AlbumId) >= 10
ORDER BY AlbumCount DESC;
```

51. List the `Title` of albums that have more than 20 tracks.

```
SELECT
 al.Title AS AlbumTitle,
 COUNT(t.TrackId) AS TrackCount
FROM Album al
JOIN Track t ON al.AlbumId = t.AlbumId
GROUP BY al.Title
HAVING COUNT(t.TrackId) > 20
ORDER BY TrackCount DESC;
```

52. Get the `Name` of customers who have made more than 5 purchases (invoices).

```
SELECT
 c.FirstName,
 c.LastName,
 COUNT(i.InvoiceId) AS InvoiceCount
FROM Customer c
JOIN Invoice i ON c.CustomerId = i.CustomerId
GROUP BY c.FirstName, c.LastName
HAVING COUNT(i.InvoiceId) > 5
ORDER BY InvoiceCount DESC;
```

53. Find the `TrackId` and `Name` of tracks that have never been purchased.

```
SELECT
 t.TrackId,
 t.Name
```

```
FROM Track t
LEFT JOIN InvoiceLine il ON t.TrackId = il.TrackId
WHERE il.TrackId IS NULL;
```

54. **List the `EmployeeId`, `FirstName`, `LastName` of employees who are managers (they have employees reporting to them)**

```
SELECT DISTINCT
 m.EmployeeId,
 m.FirstName,
 m.LastName
FROM Employee e
JOIN Employee m ON e.ReportsTo = m.EmployeeId;
```

55. **Get the `Name` of tracks that are part of the 'Classical' genre and have a `UnitPrice` greater than 0.99.**

```
SELECT
 t.Name
FROM Track t
JOIN Genre g ON t.GenreId = g.GenreId
WHERE g.Name = 'Classical' AND t.UnitPrice > 0.99;
```

56. **Find the `ArtistName` who has the longest total `Milliseconds` across all their tracks.**

```
SELECT
 ar.Name AS ArtistName,
 SUM(t.Milliseconds) AS TotalMilliseconds
FROM Track t
JOIN Album al ON t.AlbumId = al.AlbumId
JOIN Artist ar ON al.ArtistId = ar.ArtistId
GROUP BY ar.Name
```

```
ORDER BY TotalMilliseconds DESC;
```

57. List the `Title` of the album that has the most tracks.

```
SELECT
  a.Title,
  COUNT(t.TrackId) AS TrackCount
FROM Album a
JOIN Track t ON a.AlbumId = t.AlbumId
GROUP BY a.AlbumId, a.Title
ORDER BY TrackCount DESC;
```

58. Get the `Name` of the customer who spent the most money.

```
SELECT
  c.FirstName,
  c.LastName,
  SUM(i.Total) AS TotalSpent
FROM Customer c
JOIN Invoice i ON c.CustomerId = i.CustomerId
GROUP BY c.CustomerId, c.FirstName, c.LastName
ORDER BY TotalSpent DESC;
```

59. Find the `Genre` that generated the highest total sales.

```
SELECT
  g.Name AS Genre,
  SUM(il.UnitPrice * il.Quantity) AS TotalSales
FROM InvoiceLine il
JOIN Track t ON il.TrackId = t.TrackId
JOIN Genre g ON t.GenreId = g.GenreId
```

```
GROUP BY g.GenreId, g.Name
ORDER BY TotalSales DESC;
```

60. List all `InvoiceIds` where the `Total` is greater than the average invoice total.

```
SELECT InvoiceId, Total
FROM Invoice
WHERE Total > (
  SELECT AVG(Total) FROM Invoice
);
```