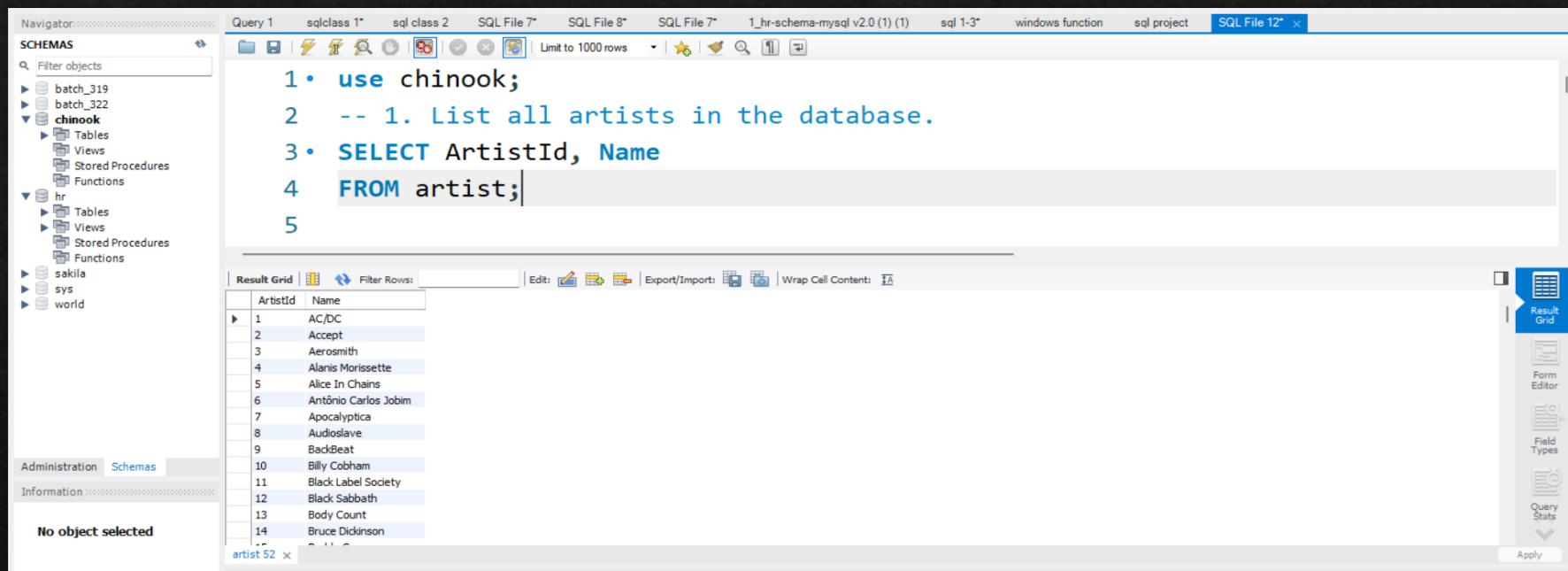




# Chinook Artist Database

Performing Artist

# List all artists in the database.



The screenshot shows the SSMS interface. The left pane displays the Navigator with the 'SCHEMAS' node expanded, showing databases like batch\_319, batch\_322, chinook, hr, sakila, sys, and world. The 'chinook' schema is selected. The right pane contains a query window with the following SQL code:

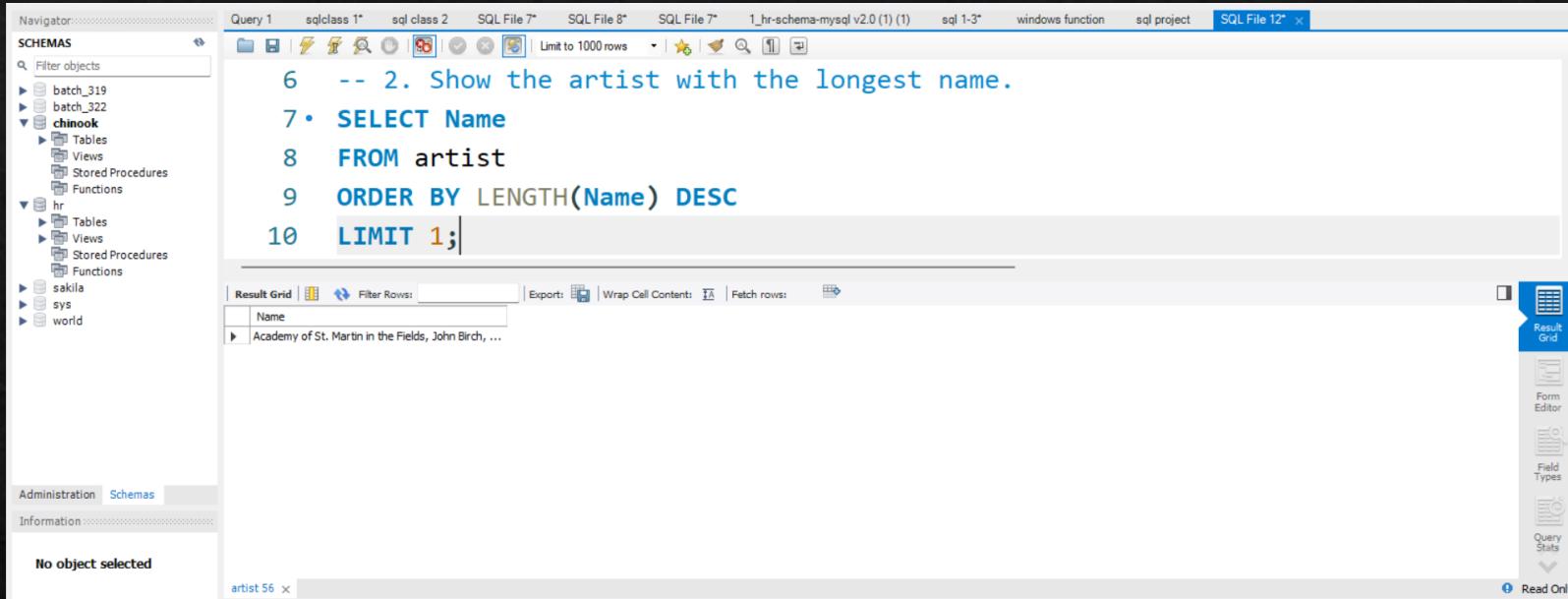
```
1 • use chinook;
2 -- 1. List all artists in the database.
3 • SELECT ArtistId, Name
4 FROM artist;
5
```

Below the code is a 'Result Grid' showing the results of the query:

ArtistId	Name
1	AC/DC
2	Accept
3	Aerosmith
4	Alanis Morissette
5	Alice In Chains
6	Antônio Carlos Jobim
7	Apocalyptica
8	Audioslave
9	BackBeat
10	Billy Cobham
11	Black Label Society
12	Black Sabbath
13	Body Count
14	Bruce Dickinson
15	C&H C

Simply selects all artist IDs and names from the artist table. It's the most basic lookup.

# Show the artist with the longest name.



The screenshot shows a MySQL query editor interface. On the left, there's a 'Navigator' pane titled 'SCHEMAS' with a tree view of databases: batch\_319, batch\_322, chinook (selected), hr, sakila, sys, and world. Below the navigator is an 'Information' pane stating 'No object selected'. The main area is a 'Query Editor' with tabs for 'Query 1' through 'SQL File 12'. The current tab is 'Query 1' which contains the following SQL code:

```
6 -- 2. Show the artist with the longest name.
7 • SELECT Name
8 FROM artist
9 ORDER BY LENGTH(Name) DESC
10 LIMIT 1;
```

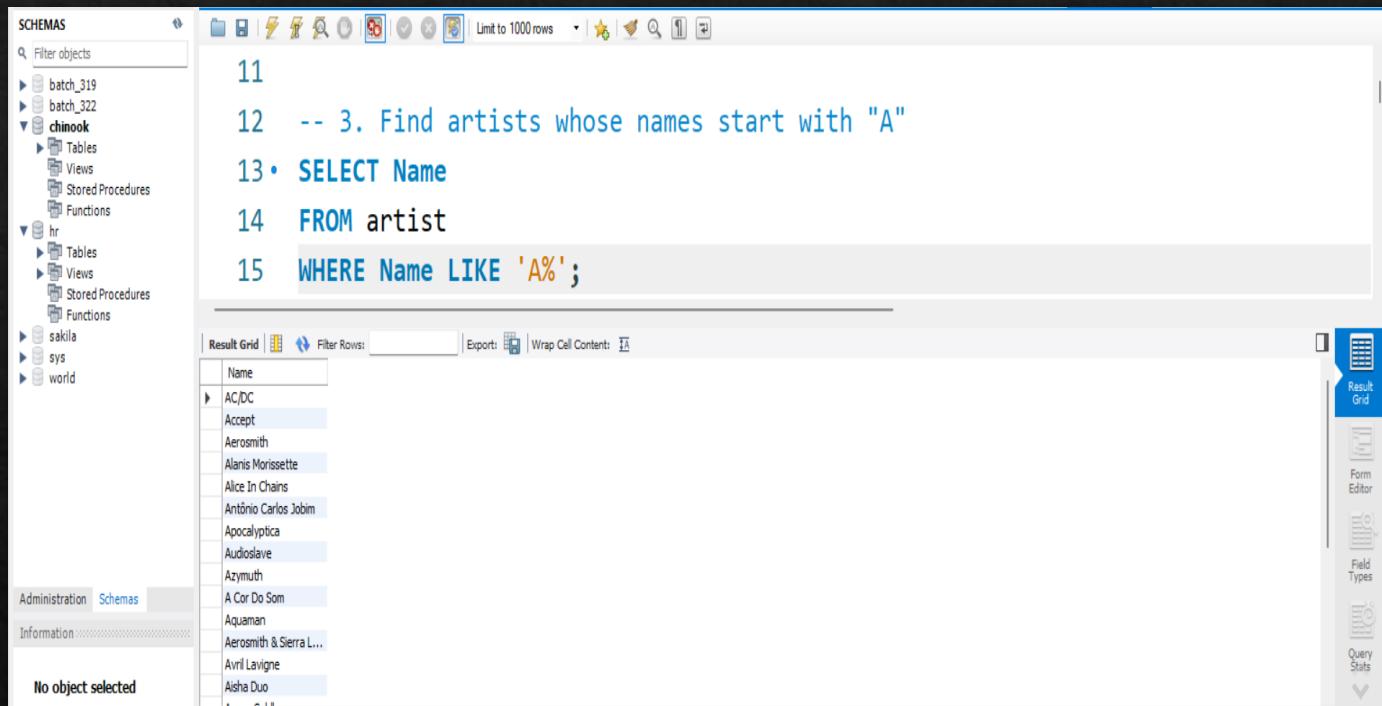
Below the code is a 'Result Grid' pane showing the output:

Name
Academy of St. Martin in the Fields, John Birch, ...

The result grid has a toolbar with 'Result Grid' selected, and a sidebar on the right labeled 'Read Only' with icons for 'Result Grid', 'Form Editor', 'Field Types', and 'Query Stats'.

Uses **LENGTH(Name)** to calculate the length of each artist's name, sorts in descending order, and picks the top one.

# Find artists whose names start with "A".



The screenshot shows the MySQL Workbench interface. On the left, the 'SCHEMAS' tree view is open, showing the 'chinook' schema selected. The main area displays a query editor with the following SQL code:

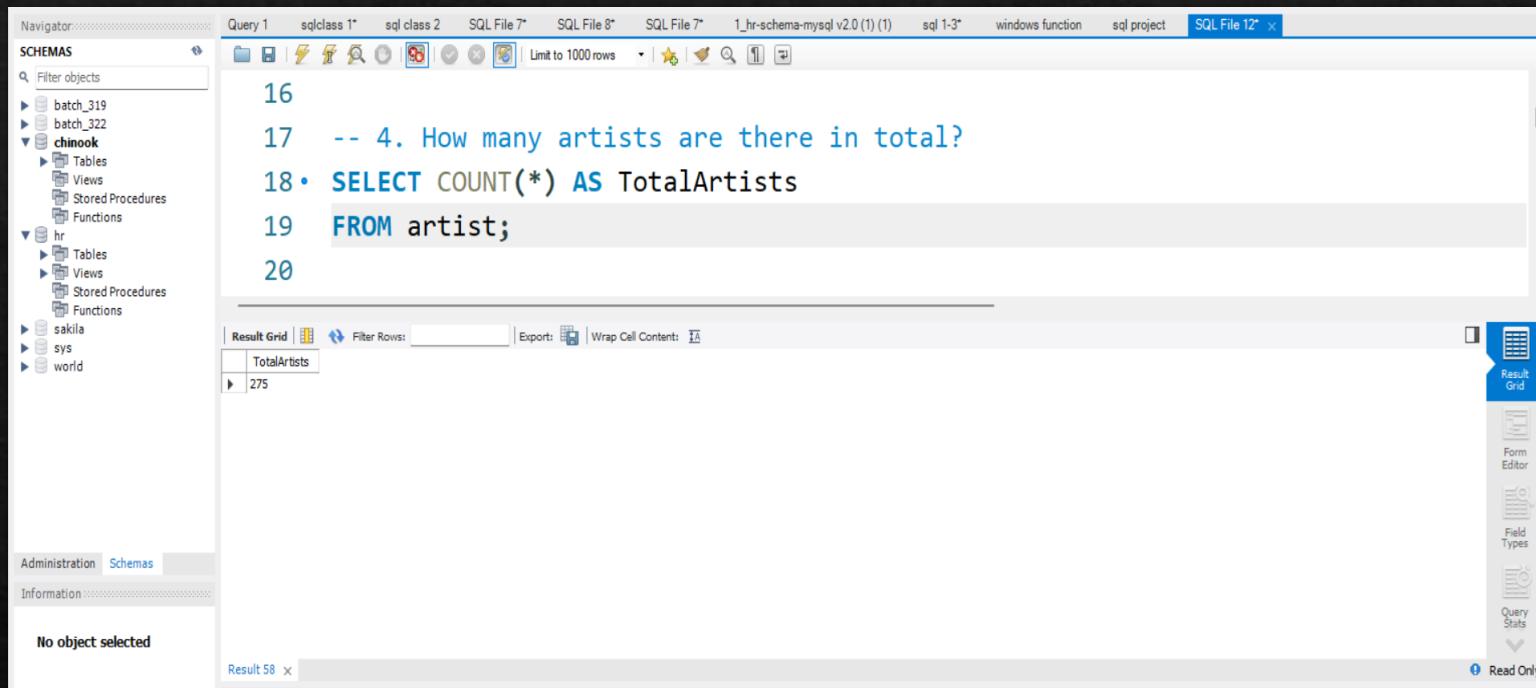
```
11
12 -- 3. Find artists whose names start with "A"
13 • SELECT Name
14 FROM artist
15 WHERE Name LIKE 'A%';
```

Below the code, the 'Result Grid' shows the following data:

Name
AC/DC
Accept
Aerosmith
Alanis Morissette
Alice In Chains
Antônio Carlos Jobim
Apocalyptica
Audioslave
Azymuth
A Cor Do Som
Aquaman
Aerosmith & Sierra L...
Avril Lavigne
Aisha Duo

Uses the `LIKE 'A%'` pattern to filter names starting with "A". Good for searching artists alphabetically.

# How many artists are there in total?



The screenshot shows a SQL query window in SSMS. The query is:

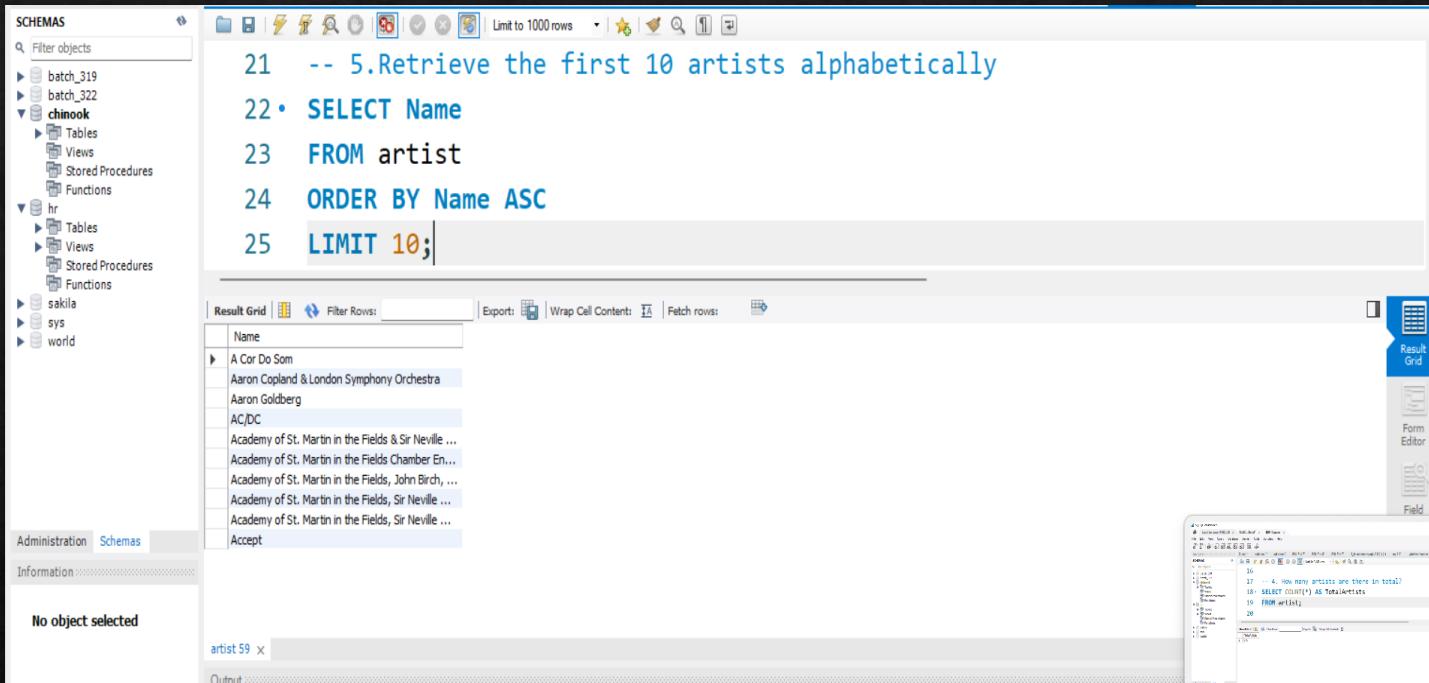
```
16  
17 -- 4. How many artists are there in total?  
18 • SELECT COUNT(*) AS TotalArtists  
19 FROM artist;  
20
```

The result grid shows a single row with the value 275 under the column 'TotalArtists'.

Below the query window, the status bar displays 'Result 58 x' and 'Read Only'.

Counts the total rows in the artist table using COUNT(\*) .

# Retrieve the first 10 artists alphabetically.



The screenshot shows the MySQL Workbench interface. On the left, the 'SCHEMAS' tree view is open, showing the 'chinook' schema expanded, revealing tables like 'Artist', 'Genre', and 'Album'. The main area contains a SQL query:

```
21 -- 5.Retrieve the first 10 artists alphabetically
22 • SELECT Name
23   FROM artist
24   ORDER BY Name ASC
25   LIMIT 10;
```

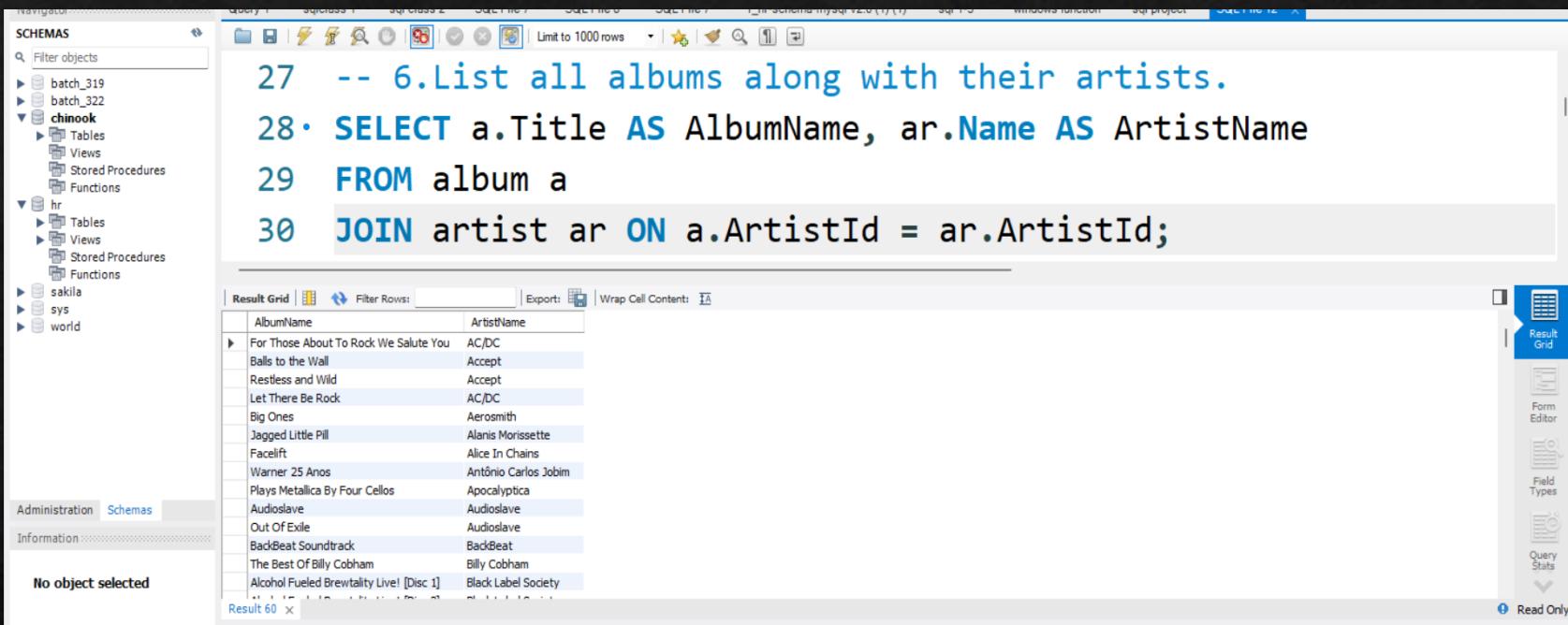
The results grid below the query shows the names of the first 10 artists in ascending order:

Name
A Cor Do Som
Aaron Copland & London Symphony Orchestra
Aaron Goldberg
AC/DC
Academy of St. Martin in the Fields & Sir Neville ...
Academy of St. Martin in the Fields Chamber En...
Academy of St. Martin in the Fields, John Birch, ...
Academy of St. Martin in the Fields, Sir Neville ...
Academy of St. Martin in the Fields, Sir Neville ...
Accept

On the right, there's a vertical toolbar with icons for Result Grid, Form Editor, and Field.

Orders artist names with ORDER BY Name ASC and limits results to 10 for quick previews.

# List all albums along with their artists.



The screenshot shows a MySQL Workbench interface. On the left, the Navigator pane displays the database schema with the 'chinook' schema expanded, showing tables like 'Artist', 'Album', and 'Track'. The main area contains a SQL editor window with the following query:

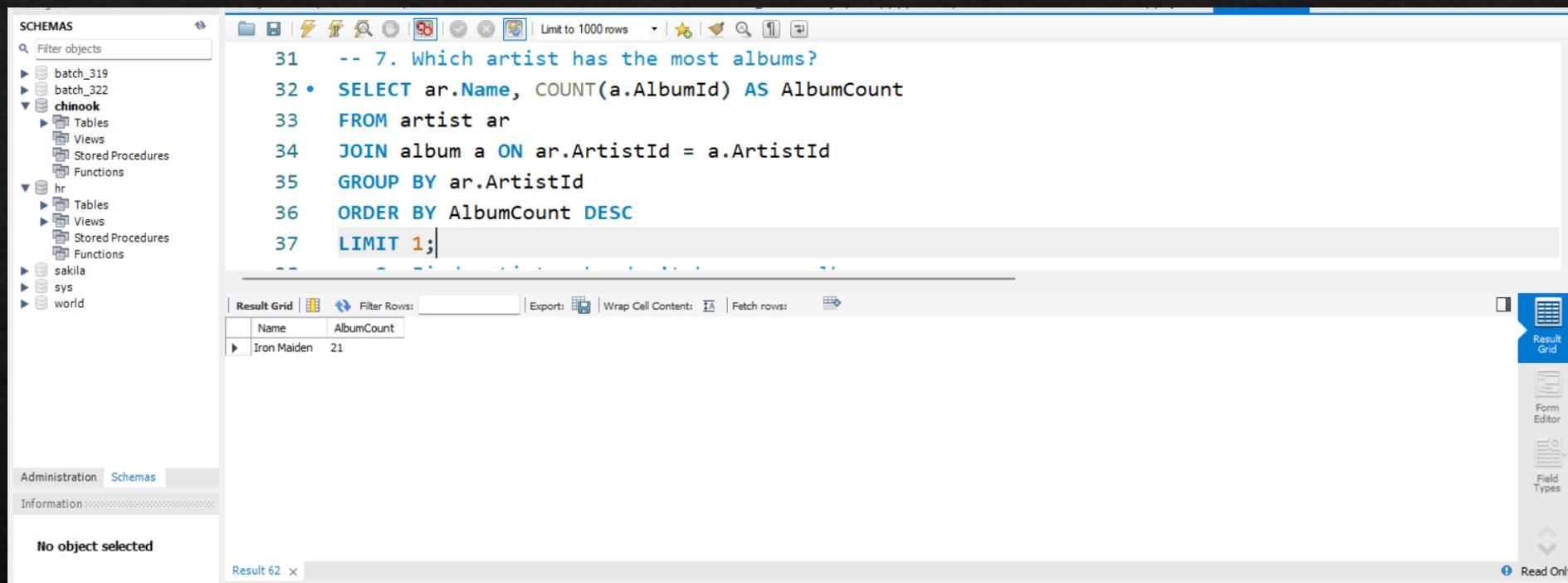
```
27 -- 6.List all albums along with their artists.  
28 • SELECT a.Title AS AlbumName, ar.Name AS ArtistName  
29 FROM album a  
30 JOIN artist ar ON a.ArtistId = ar.ArtistId;
```

The Result Grid shows the output of the query:

AlbumName	ArtistName
For Those About To Rock We Salute You	AC/DC
Balls to the Wall	Accept
Restless and Wild	Accept
Let There Be Rock	AC/DC
Big Ones	Aerosmith
Jagged Little Pill	Alanis Morissette
Facelift	Alice In Chains
Warner 25 Anos	Antônio Carlos Jobim
Plays Metallica By Four Cellos	Apocalyptica
Audioslave	Audioslave
Out Of Exile	Audioslave
BackBeat Soundtrack	BackBeat
The Best Of Billy Cobham	Billy Cobham
Alcohol Fueled Brutality Live! [Disc 1]	Black Label Society

Joins **album** with **artist** on **Artistid** to display albums and the artist who created them.

# Which artist has the most albums?



The screenshot shows a MySQL Workbench interface. On the left, the 'SCHEMAS' tree view is open, showing databases like batch\_319, batch\_322, chinook, hr, sakila, sys, and world. The 'chinook' database is selected. In the main pane, a SQL editor window contains the following query:

```
31 -- 7. Which artist has the most albums?
32 • SELECT ar.Name, COUNT(a.AlbumId) AS AlbumCount
33 FROM artist ar
34 JOIN album a ON ar.ArtistId = a.ArtistId
35 GROUP BY ar.ArtistId
36 ORDER BY AlbumCount DESC
37 LIMIT 1;
```

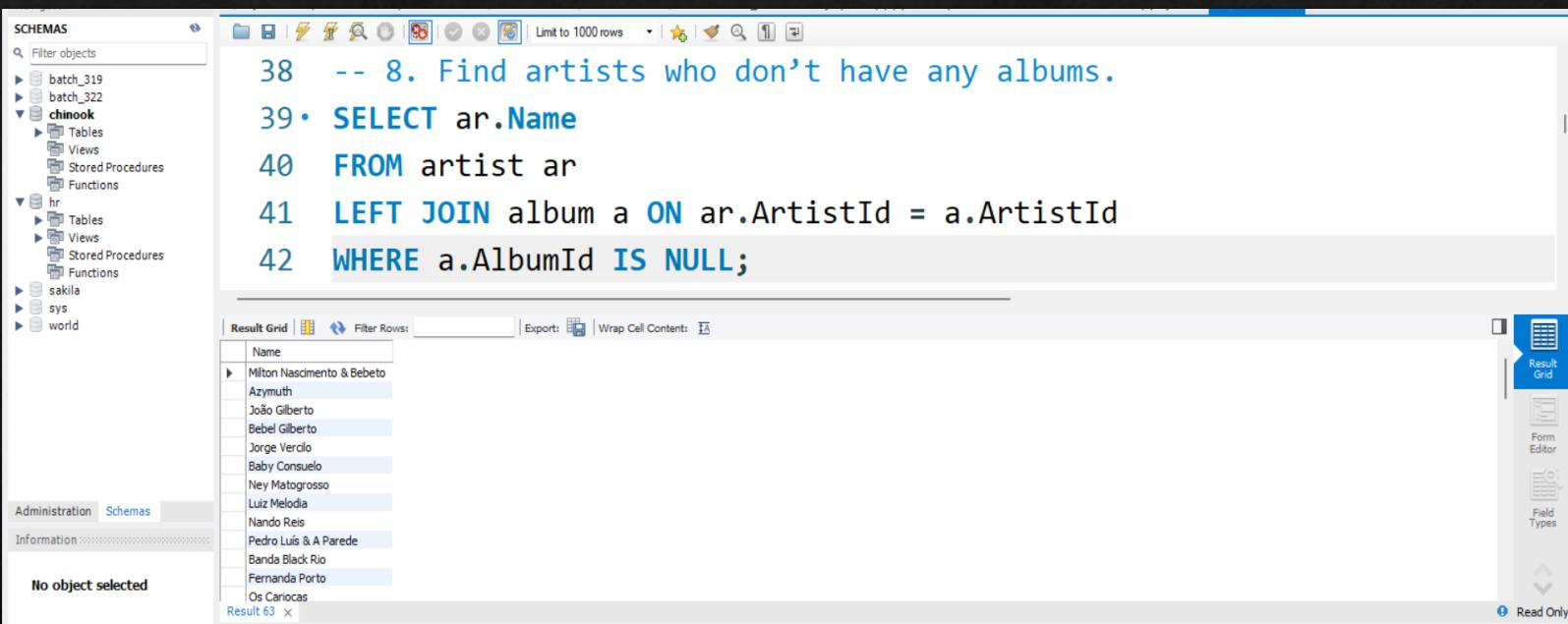
Below the SQL editor is a 'Result Grid' table with one row:

Name	AlbumCount
Iron Maiden	21

The interface includes various toolbars, a status bar at the bottom, and a sidebar on the right with icons for Result Grid, Form Editor, Field Types, and a Read Only button.

Groups albums by artist, counts them, and orders descending to find the artist with the maximum album count.

# Find artists who don't have any albums.



The screenshot shows a database management interface with a sidebar containing a 'SCHEMAS' tree. The 'chinook' schema is selected, revealing tables like 'Artist', 'Album', and 'Genre'. The main area displays a SQL query and its results. The query finds artists who have no albums:

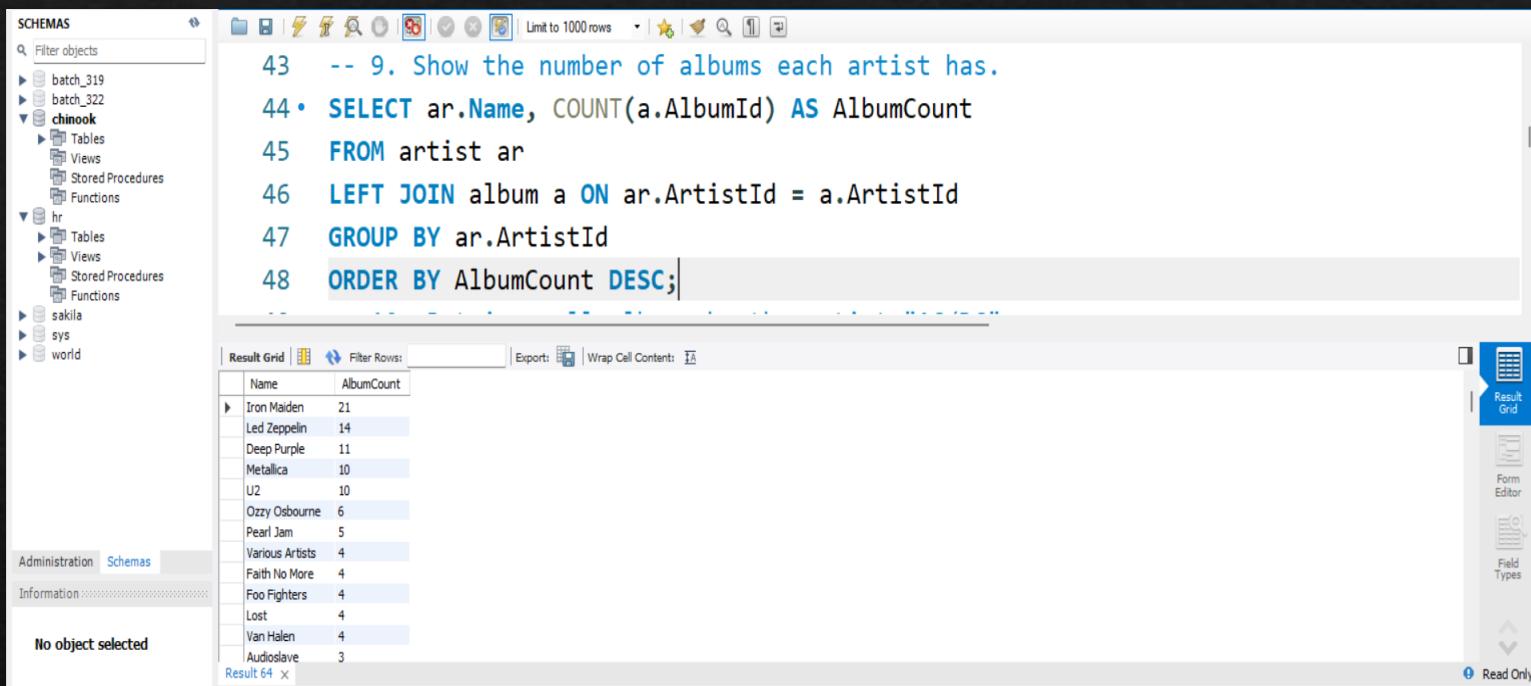
```
38 -- 8. Find artists who don't have any albums.  
39 • SELECT ar.Name  
40 FROM artist ar  
41 LEFT JOIN album a ON ar.ArtistId = a.ArtistId  
42 WHERE a.AlbumId IS NULL;
```

The results grid shows a single column 'Name' with the following data:

Name
Milton Nascimento & Bebeto
Azymuth
João Gilberto
Bebel Gilberto
Jorge Verílio
Baby Consuelo
Ney Matogrosso
Luiz Melodia
Nando Reis
Pedro Luís & A Parede
Banda Black Rio
Fernanda Porto
Os Cariocas

Uses a LEFT JOIN from **artist** to **album** and checks where **AlbumId IS NULL** → means no albums exist for that artist.

# Show the number of albums each artist has.



The screenshot shows a database management interface with a sidebar containing a 'SCHEMAS' tree. The 'chinook' schema is selected, revealing tables like 'Artist', 'Album', and 'Genre'. The main area displays a SQL query and its execution results.

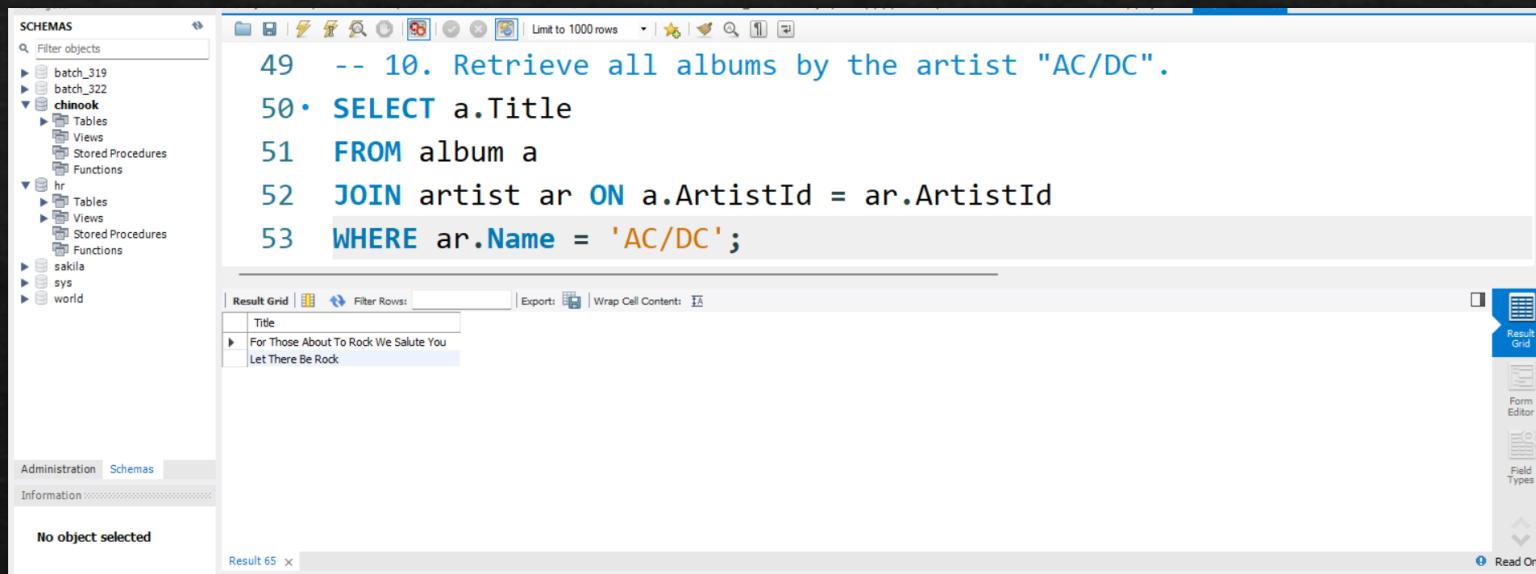
```
43 -- 9. Show the number of albums each artist has.
44 • SELECT ar.Name, COUNT(a.AlbumId) AS AlbumCount
45 FROM artist ar
46 LEFT JOIN album a ON ar.ArtistId = a.ArtistId
47 GROUP BY ar.ArtistId
48 ORDER BY AlbumCount DESC;
```

The results grid shows the following data:

Name	AlbumCount
Iron Maiden	21
Led Zeppelin	14
Deep Purple	11
Metallica	10
U2	10
Ozzy Osbourne	6
Pearl Jam	5
Various Artists	4
Faith No More	4
Foo Fighters	4
Lost	4
Van Halen	4
Audioslave	3

Groups by artist and counts albums. This gives insight into which artists are more prolific.

# Retrieve all albums by the artist "AC/DC".



The screenshot shows a MySQL Workbench interface. On the left, the 'SCHEMAS' tree view is open, showing databases like batch\_319, batch\_322, chinook, hr, sakila, sys, and world. The 'chinook' database is selected. In the main pane, a SQL editor window displays the following query:

```
49 -- 10. Retrieve all albums by the artist "AC/DC".
50 • SELECT a.Title
51 FROM album a
52 JOIN artist ar ON a.ArtistId = ar.ArtistId
53 WHERE ar.Name = 'AC/DC';
```

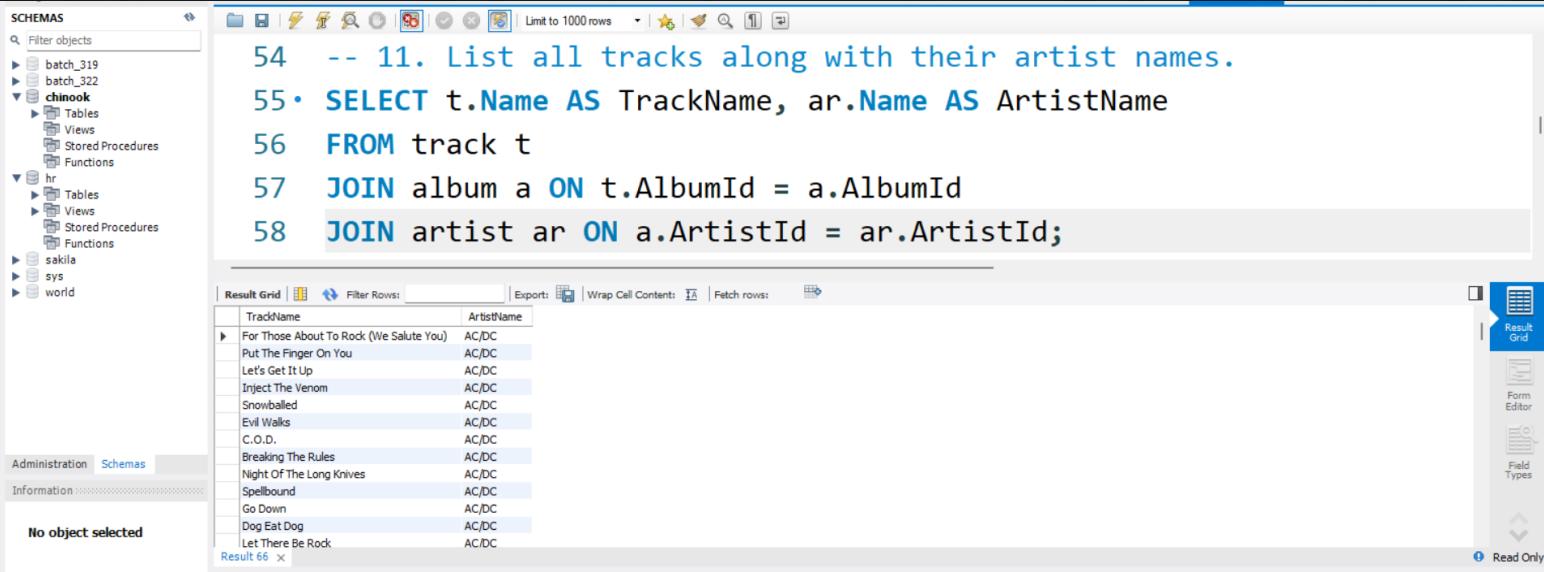
Below the query, the 'Result Grid' shows the results:

Title
For Those About To Rock We Salute You
Let There Be Rock

The interface includes various toolbars and panels on the right, such as 'Result Grid', 'Form Editor', and 'Field Types'. A status bar at the bottom indicates 'Result 65'.

Filters by artist name "AC/DC" after joining album with artist.

# List all tracks along with their artist names.



The screenshot shows a database interface with a sidebar containing 'SCHEMAS' (batch\_319, batch\_322, chinook, hr, sakila, sys, world) and a central area for writing and executing SQL queries. The query window displays the following code:

```
54 -- 11. List all tracks along with their artist names.
55 • SELECT t.Name AS TrackName, ar.Name AS ArtistName
56 FROM track t
57 JOIN album a ON t.AlbumId = a.AlbumId
58 JOIN artist ar ON a.ArtistId = ar.ArtistId;
```

The results grid shows the output of the query, listing 66 tracks and their artists:

TrackName	ArtistName
For Those About To Rock (We Salute You)	AC/DC
Put The Finger On You	AC/DC
Let's Get It Up	AC/DC
Inject The Venom	AC/DC
Snowballed	AC/DC
Evil Walks	AC/DC
C.O.D.	AC/DC
Breaking The Rules	AC/DC
Night Of The Long Knives	AC/DC
Spellbound	AC/DC
Go Down	AC/DC
Dog Eat Dog	AC/DC
Let There Be Rock	AC/DC

Joins track → album → artist to show which tracks belong to which artist

# Find the artist with the most tracks.

The screenshot shows a database management interface with a sidebar titled 'SCHEMAS' containing several database schemas: batch\_319, batch\_322, chinook (selected), hr, sakila, sys, and world. The main area displays a SQL query:

```
59 -- 12. Find the artist with the most tracks.
60 • SELECT ar.Name, COUNT(t.TrackId) AS TrackCount
61 FROM artist ar
62 JOIN album a ON ar.ArtistId = a.ArtistId
63 JOIN track t ON a.AlbumId = t.AlbumId
64 GROUP BY ar.ArtistId
65 ORDER BY TrackCount DESC
66 LIMIT 1;
```

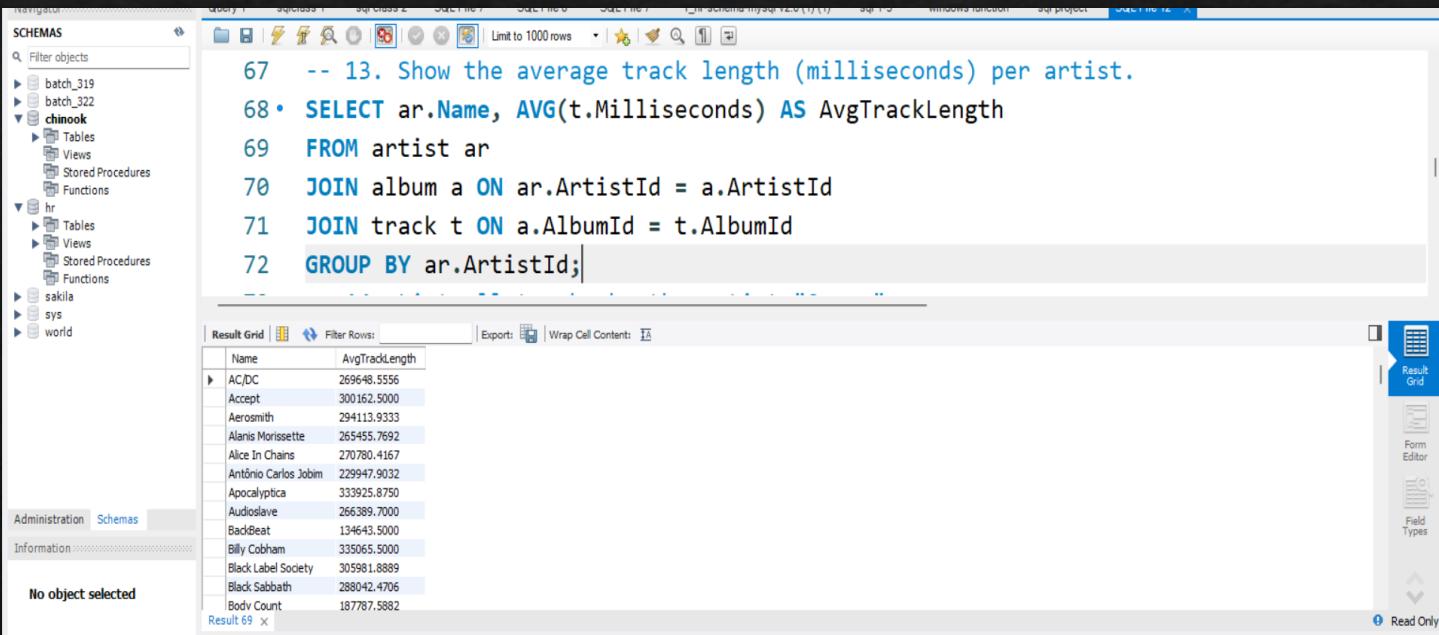
The results grid shows one row:

Name	TrackCount
Iron Maiden	213

Below the grid, the status bar indicates 'Result 77' and 'Read Only'.

Groups tracks by artist, counts them, sorts descending, and takes the top 1.

# Show the average track length (milliseconds) per artist.



The screenshot shows the MySQL Workbench interface. On the left, the 'Schemas' tree view is open, showing the 'chinook' schema selected. The main area contains a SQL editor window with the following code:

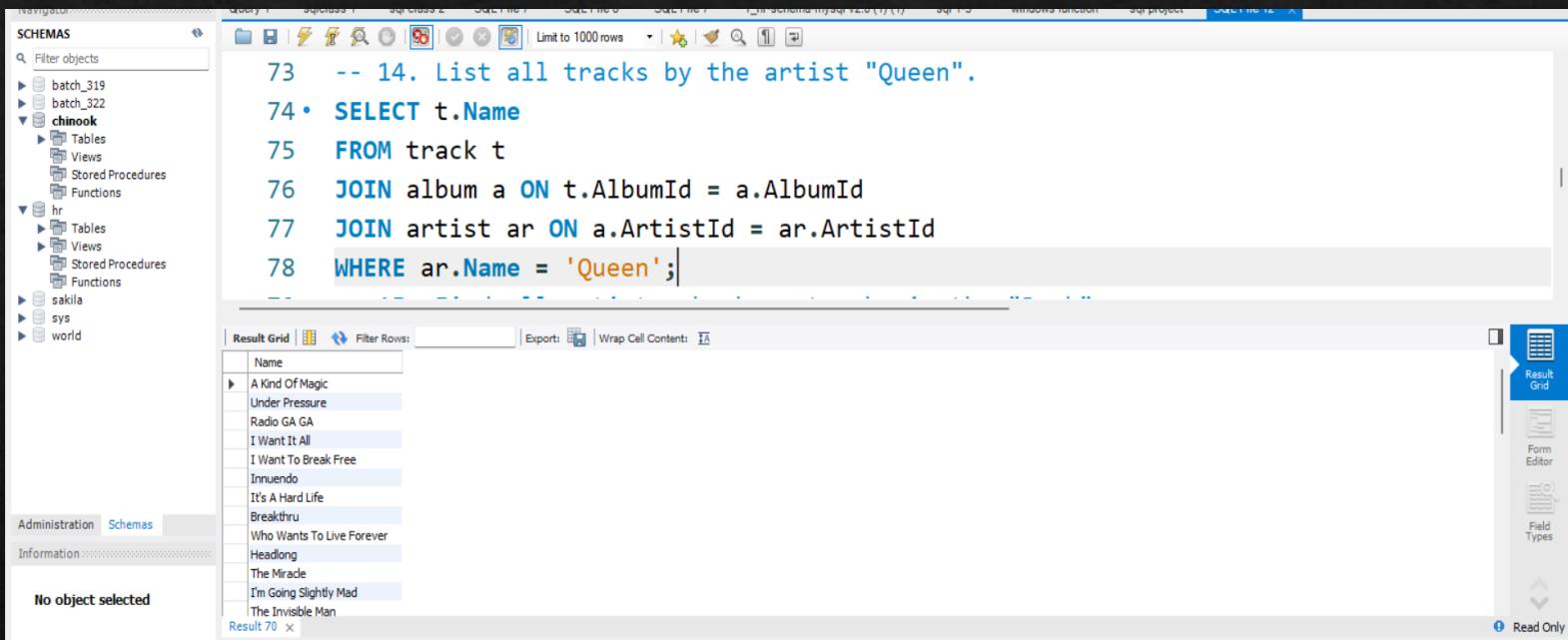
```
67 -- 13. Show the average track length (milliseconds) per artist.
68 • SELECT ar.Name, AVG(t.Milliseconds) AS AvgTrackLength
69 FROM artist ar
70 JOIN album a ON ar.ArtistId = a.ArtistId
71 JOIN track t ON a.AlbumId = t.AlbumId
72 GROUP BY ar.ArtistId;
```

Below the code, the 'Result Grid' shows the query results:

Name	AvgTrackLength
AC/DC	269648.5556
Accept	300162.5000
Aerosmith	294113.9333
Alanis Morissette	265455.7692
Alice In Chains	270780.4167
Antônio Carlos Jobim	229947.9032
Apocalyptica	333925.8750
Audioslave	266389.7000
BadBeat	134643.5000
Billy Cobham	335065.5000
Black Label Society	305981.8889
Black Sabbath	288042.4706
Body Count	187787.5882

Uses **AVG(Milliseconds)** grouped by artist to show how long an artist's songs typically are.

# List all tracks by the artist "Queen".



The screenshot shows a SQL query window in SSMS. The query is:

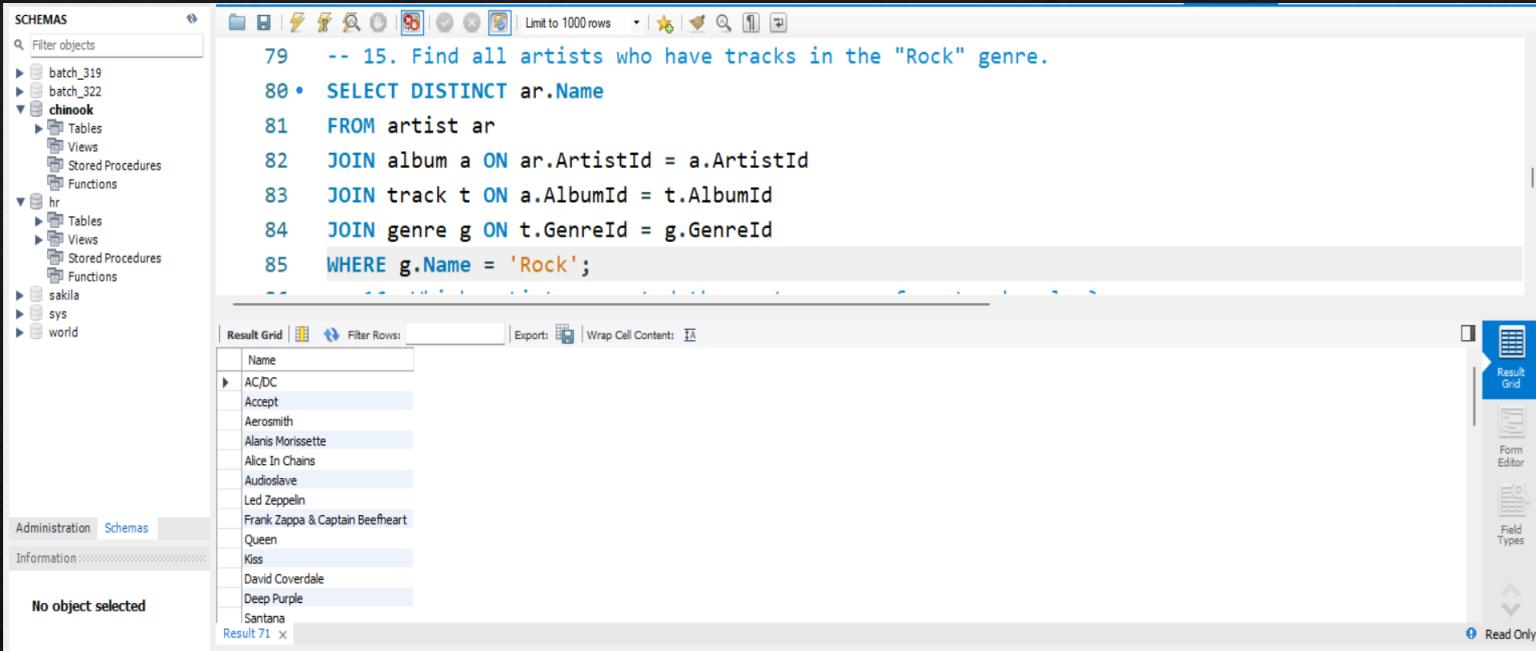
```
73  -- 14. List all tracks by the artist "Queen".
74  SELECT t.Name
75  FROM track t
76  JOIN album a ON t.AlbumId = a.AlbumId
77  JOIN artist ar ON a.ArtistId = ar.ArtistId
78  WHERE ar.Name = 'Queen';
```

The result grid displays the names of tracks by Queen:

Name
A Kind Of Magic
Under Pressure
Radio GA GA
I Want It All
I Want To Break Free
Innuendo
It's A Hard Life
Breakthru
Who Wants To Live Forever
Headlong
The Miracle
I'm Going Slightly Mad
The Invisible Man

Filters on Artist.Name = 'Queen' to list only their tracks.

# Find all artists who have tracks in the "Rock" genre.



The screenshot shows a MySQL Workbench interface. On the left, the 'SCHEMAS' tree view is open, showing databases like batch\_319, batch\_322, chinook, hr, sakila, sys, and world. The 'chinook' database is selected. In the center, a query editor window displays the following SQL code:

```
79 -- 15. Find all artists who have tracks in the "Rock" genre.
80 • SELECT DISTINCT ar.Name
81 FROM artist ar
82 JOIN album a ON ar.ArtistId = a.ArtistId
83 JOIN track t ON a.AlbumId = t.AlbumId
84 JOIN genre g ON t.GenreId = g.GenreId
85 WHERE g.Name = 'Rock';
```

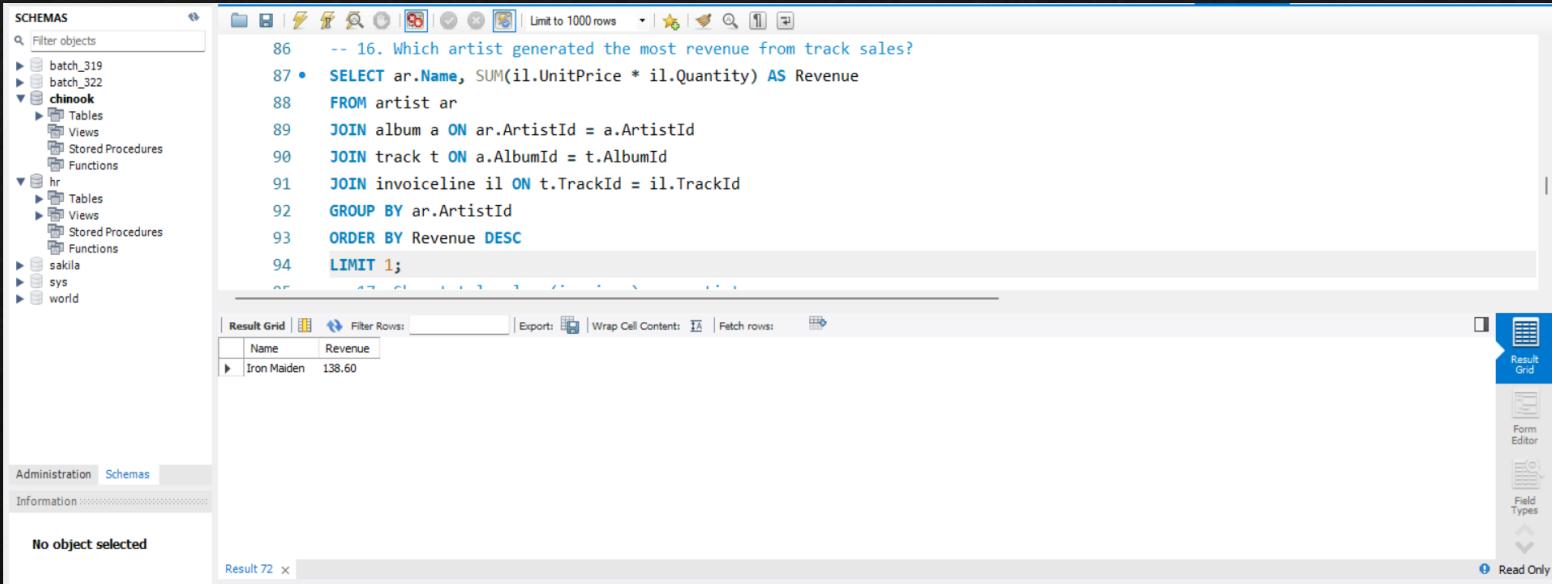
Below the code, the 'Result Grid' shows the names of artists who have tracks in the 'Rock' genre. The results are:

Name
Ac/DC
Accept
Aerosmith
Alanis Morissette
Alice In Chains
Audioslave
Led Zeppelin
Frank Zappa & Captain Beefheart
Queen
Kiss
David Coverdale
Deep Purple
Santana

At the bottom of the results grid, it says 'Result 71'.

Joins **track** → **genre** → **artist** and filters **G.name = 'Rock'**. Uses **DISTINCT** because one artist may have multiple Rock tracks.

# Which artist generated the most revenue from track sales?



The screenshot shows the MySQL Workbench interface. On the left, the 'SCHEMAS' tree view is open, showing databases like batch\_319, batch\_322, chinook, hr, sakila, sys, and world. The 'chinook' database is selected. In the center, a SQL editor window displays the following query:

```
86 -- 16. Which artist generated the most revenue from track sales?
87 • SELECT ar.Name, SUM(il.UnitPrice * il.Quantity) AS Revenue
88 FROM artist ar
89 JOIN album a ON ar.ArtistId = a.ArtistId
90 JOIN track t ON a.AlbumId = t.AlbumId
91 JOIN invoiceline il ON t.TrackId = il.TrackId
92 GROUP BY ar.ArtistId
93 ORDER BY Revenue DESC
94 LIMIT 1;
```

The result grid below the query shows one row of data:

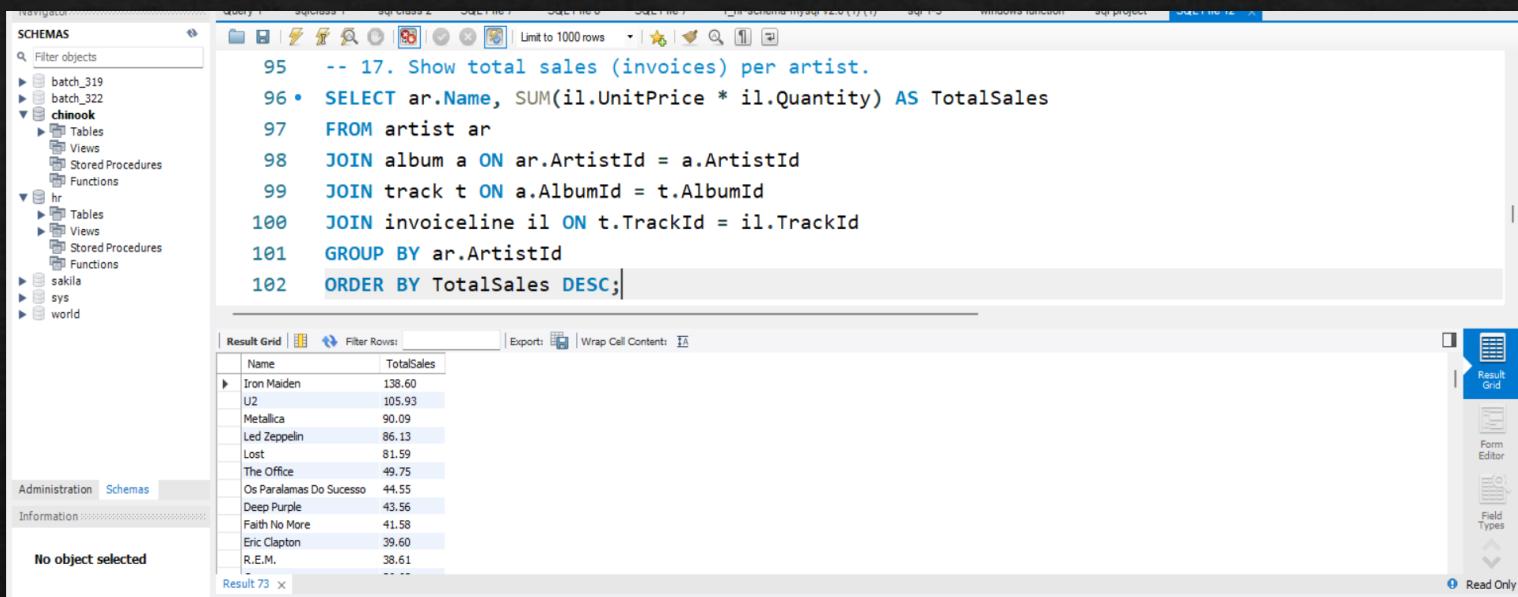
Name	Revenue
Iron Maiden	138.60

The bottom right corner of the result grid has a blue 'Result Grid' button.

Traverses relationships: artist → album → track → Invoiceline.

Calculates  $\text{SUM}(\text{Unitprice} * \text{Quantity})$  to get total revenue per artist, sorts, and picks the highest.

# Show total sales (invoices) per artist.



The screenshot shows the MySQL Workbench interface. On the left, the 'Schemas' tree view is open, showing the 'chinook' schema selected. The main area displays a SQL query:

```
95 -- 17. Show total sales (invoices) per artist.
96 • SELECT ar.Name, SUM(il.UnitPrice * il.Quantity) AS TotalSales
97 FROM artist ar
98 JOIN album a ON ar.ArtistId = a.ArtistId
99 JOIN track t ON a.AlbumId = t.AlbumId
100 JOIN invoiceline il ON t.TrackId = il.TrackId
101 GROUP BY ar.ArtistId
102 ORDER BY TotalSales DESC;
```

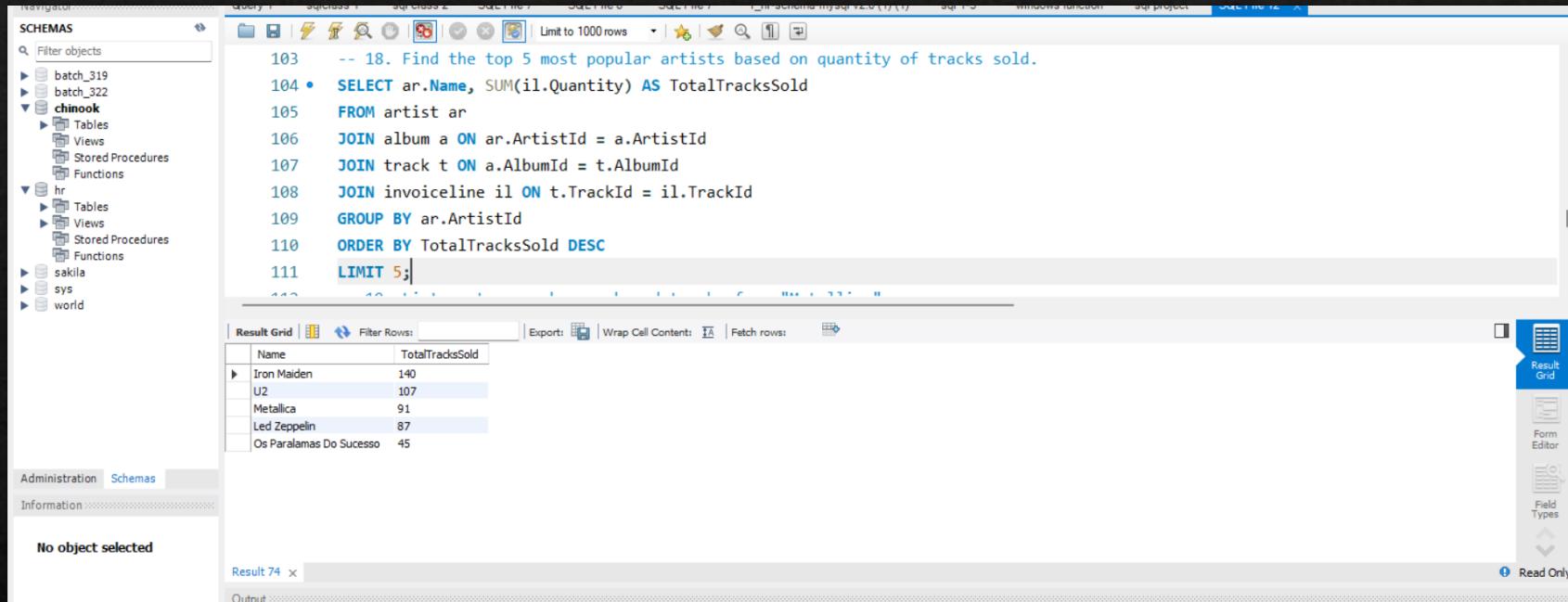
The results grid below the query shows the following data:

Name	TotalSales
Iron Maiden	138.60
U2	105.93
Metallica	90.09
Led Zeppelin	86.13
Lost	81.59
The Office	49.75
Os Paralamas Do Sucesso	44.55
Deep Purple	43.56
Faith No More	41.58
Eric Clapton	39.60
R.E.M.	38.61
...	...

The interface includes various toolbars and panels, and a sidebar on the right labeled 'Result Grid'.

Same join path as above but groups results for *all artists*, not just the top one. Gives the financial contribution of each artist

# Find the top 5 most popular artists based on quantity of tracks sold.



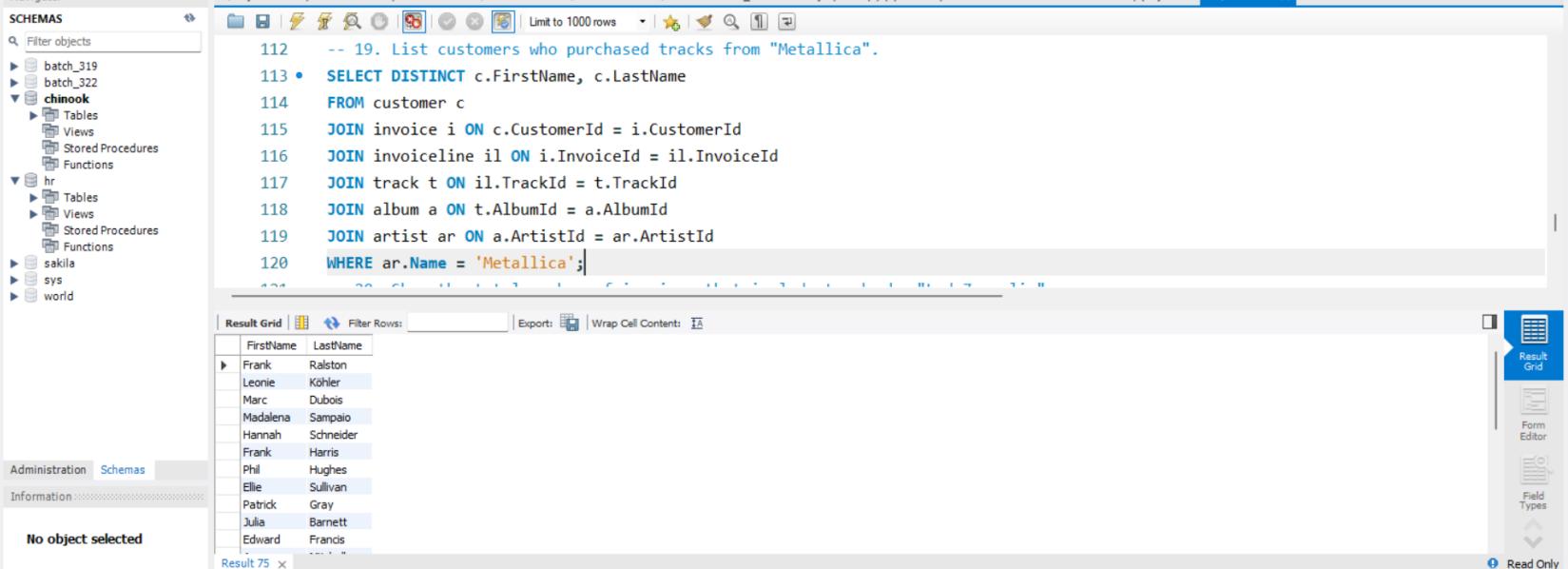
The screenshot shows a SQL query being run in SSMS against the Chinook database. The query retrieves the top 5 artists based on the total quantity of tracks sold. The results are displayed in a grid.

```
-- 18. Find the top 5 most popular artists based on quantity of tracks sold.  
SELECT ar.Name, SUM(il.Quantity) AS TotalTracksSold  
FROM artist ar  
JOIN album a ON ar.ArtistId = a.ArtistId  
JOIN track t ON a.AlbumId = t.AlbumId  
JOIN invoiceline il ON t.TrackId = il.TrackId  
GROUP BY ar.ArtistId  
ORDER BY TotalTracksSold DESC  
LIMIT 5;
```

Name	TotalTracksSold
Iron Maiden	140
U2	107
Metallica	91
Led Zeppelin	87
Os Paralamas Do Sucesso	45

Uses `SUM(il.Quantity)` (instead of revenue) to rank artists by number of units sold, then limits to top 5.

# List customers who purchased tracks from "Metallica".



The screenshot shows a database management interface with a sidebar containing a tree view of schemas: Schemas (batch\_319, batch\_322, chinook, hr, sakila, sys, world), Administration, Schemas, and Information. The main area displays a SQL query and its results.

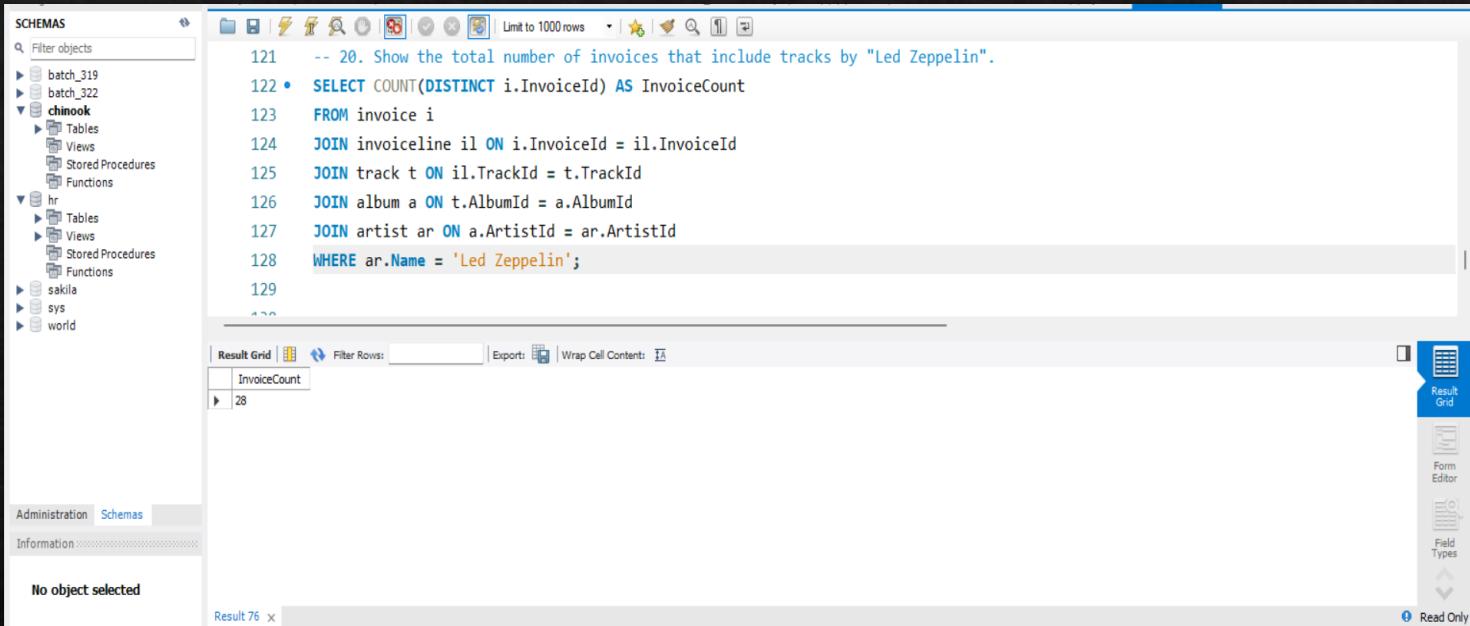
```
112 -- 19. List customers who purchased tracks from "Metallica".
113 • SELECT DISTINCT c.FirstName, c.LastName
114   FROM customer c
115   JOIN invoice i ON c.CustomerId = i.CustomerId
116   JOIN invoiceline il ON i.InvoiceId = il.InvoiceId
117   JOIN track t ON il.TrackId = t.TrackId
118   JOIN album a ON t.AlbumId = a.AlbumId
119   JOIN artist ar ON a.ArtistId = ar.ArtistId
120 WHERE ar.Name = 'Metallica';
```

The Result Grid shows the following data:

FirstName	LastName
Frank	Ralston
Leonie	Köhler
Marc	Dubois
Madalena	Sampaio
Hannah	Schneider
Frank	Harris
Phil	Hughes
Ellie	Sullivan
Patrick	Gray
Julia	Barnett
Edward	Francis

Joins across multiple tables:  
customer → invoice → invoiceline → track → album → artist.  
Filters by artist name = "Metallica" to get only those customers.

# Show the total number of invoices that include tracks by "Led Zeppelin".



The screenshot shows a database interface with the following details:

- Schemas:** A tree view showing databases like batch\_319, batch\_322, chinook, hr, sakila, sys, and world.
- Query Editor:** Displays a SQL query:

```
121 -- 20. Show the total number of invoices that include tracks by "Led Zeppelin".
122 • SELECT COUNT(DISTINCT i.InvoiceId) AS InvoiceCount
123   FROM invoice i
124   JOIN invoiceline il ON i.InvoiceId = il.InvoiceId
125   JOIN track t ON il.TrackId = t.TrackId
126   JOIN album a ON t.AlbumId = a.AlbumId
127   JOIN artist ar ON a.ArtistId = ar.ArtistId
128 WHERE ar.Name = 'Led Zeppelin';
129
```
- Result Grid:** Shows the result of the query: 

InvoiceCount
28
- Status Bar:** Shows "Result 76" and "Read Only".

Counts DISTINCT i.InvoiceId where purchased tracks belong to "Led Zeppelin". Ensures invoices are not double-counted if multiple Zeppelin tracks appear.



*By Vignesh*