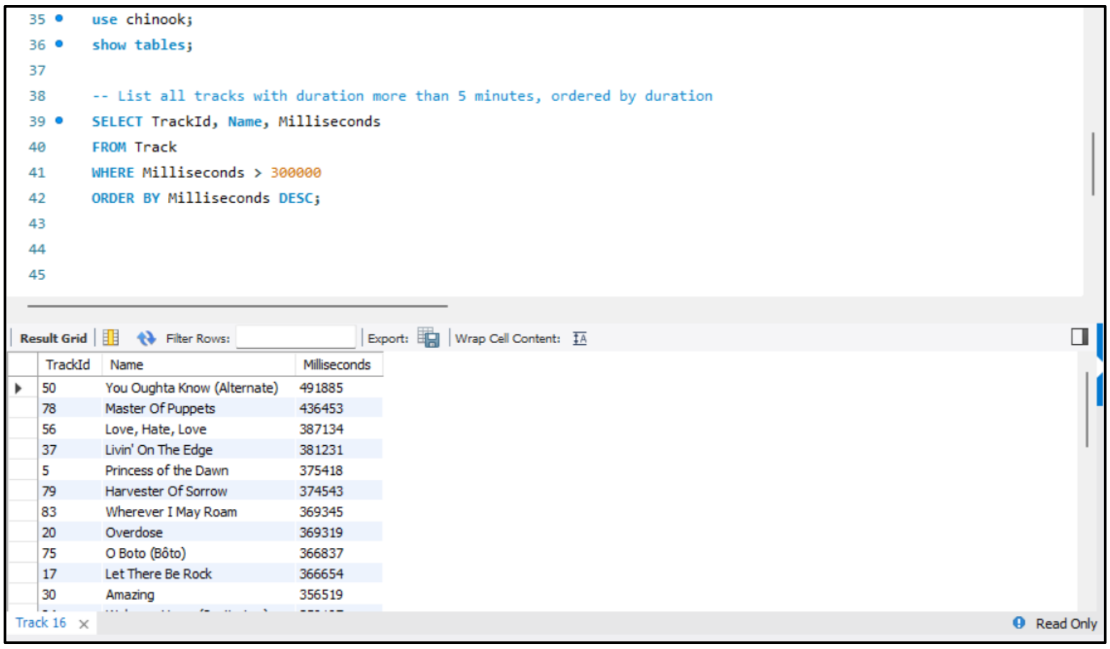
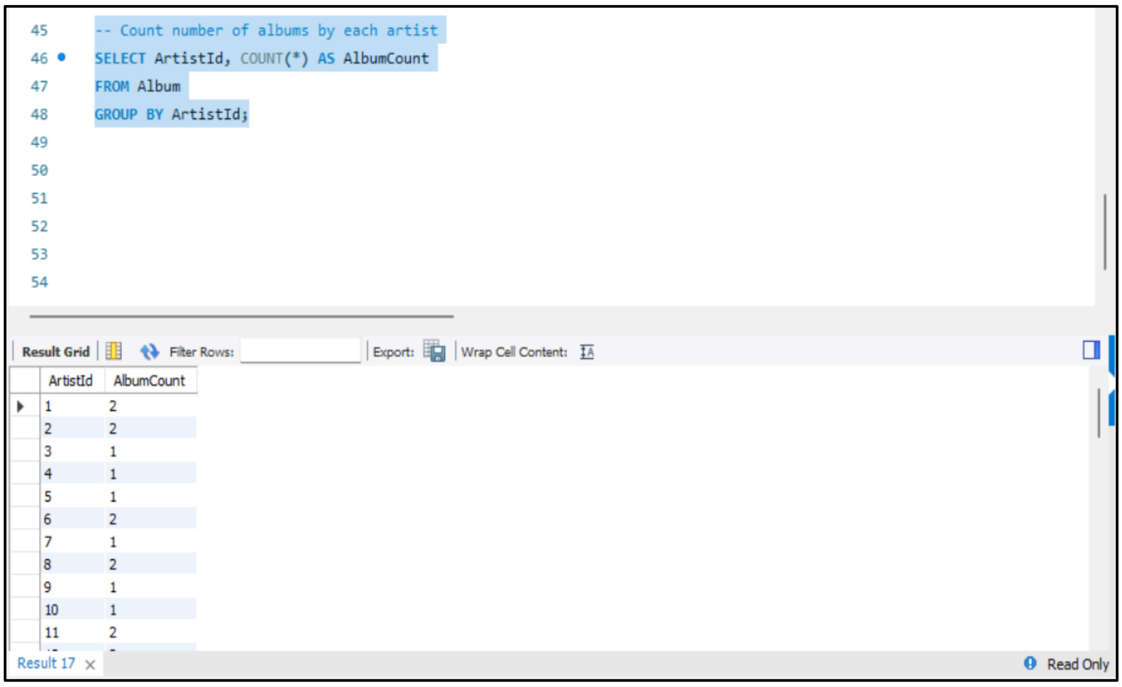
TASK 3: SQL FOR DATA ANALYSIS

A) Use of SELECT, WHERE ORDER BY, GROUP BY



```sql
use chinook;
show tables;

-- List all tracks with duration more than 5 minutes, ordered by duration
SELECT TrackId, Name, Milliseconds
FROM Track
WHERE Milliseconds > 300000
ORDER BY Milliseconds DESC;
```

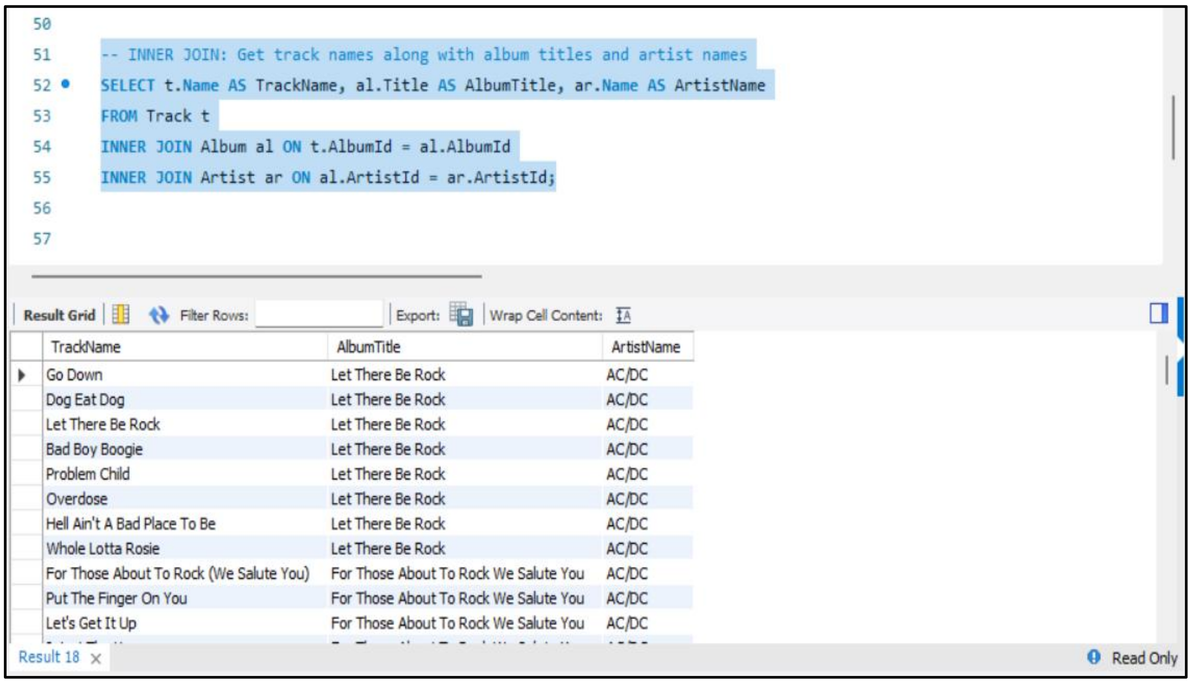
TrackId	Name	Milliseconds
50	You Oughta Know (Alternate)	491885
78	Master Of Puppets	436453
56	Love, Hate, Love	387134
37	Livin' On The Edge	381231
5	Princess of the Dawn	375418
79	Harvester Of Sorrow	374543
83	Wherever I May Roam	369345
20	Overdose	369319
75	O Boto (Bôto)	366837
17	Let There Be Rock	366654
30	Amazing	356519



```sql
-- Count number of albums by each artist
SELECT ArtistId, COUNT(*) AS AlbumCount
FROM Album
GROUP BY ArtistId;
```

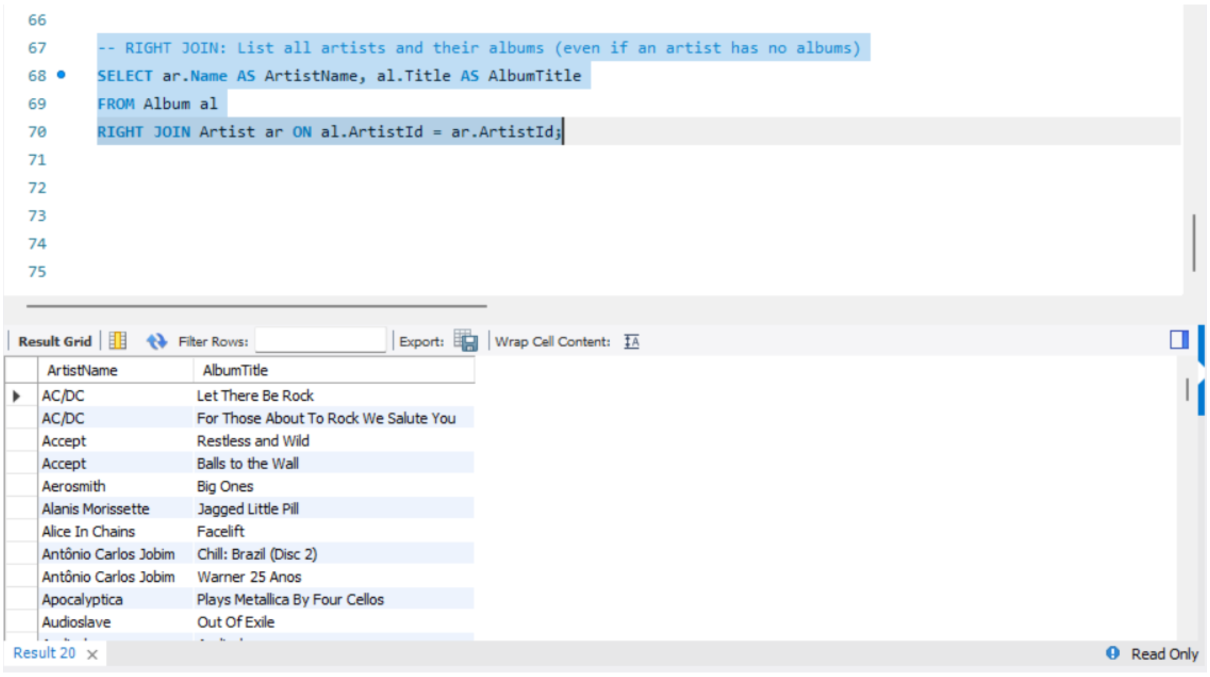
ArtistId	AlbumCount
1	2
2	2
3	1
4	1
5	1
6	2
7	1
8	2
9	1
10	1
11	2

B) Use of JOINS (INNER, LEFT, RIGHT)



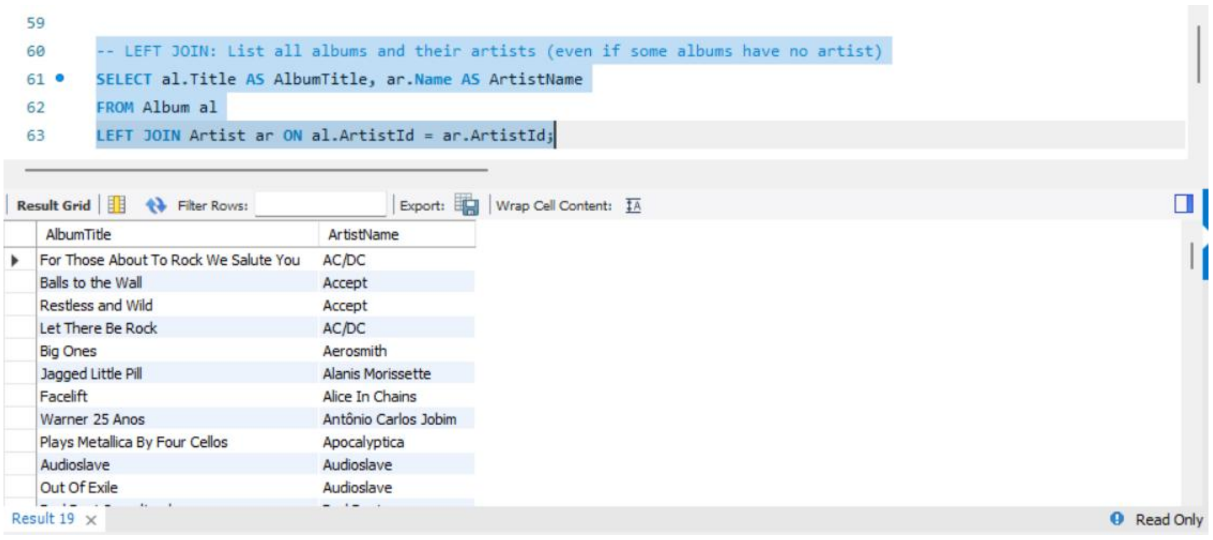
```sql
-- INNER JOIN: Get track names along with album titles and artist names
SELECT t.Name AS TrackName, al.Title AS AlbumTitle, ar.Name AS ArtistName
FROM Track t
INNER JOIN Album al ON t.AlbumId = al.AlbumId
INNER JOIN Artist ar ON al.ArtistId = ar.ArtistId;
```

TrackName	AlbumTitle	ArtistName
Go Down	Let There Be Rock	AC/DC
Dog Eat Dog	Let There Be Rock	AC/DC
Let There Be Rock	Let There Be Rock	AC/DC
Bad Boy Boogie	Let There Be Rock	AC/DC
Problem Child	Let There Be Rock	AC/DC
Overdose	Let There Be Rock	AC/DC
Hell Ain't A Bad Place To Be	Let There Be Rock	AC/DC
Whole Lotta Rosie	Let There Be Rock	AC/DC
For Those About To Rock (We Salute You)	For Those About To Rock We Salute You	AC/DC
Put The Finger On You	For Those About To Rock We Salute You	AC/DC
Let's Get It Up	For Those About To Rock We Salute You	AC/DC



```sql
-- RIGHT JOIN: List all artists and their albums (even if an artist has no albums)
SELECT ar.Name AS ArtistName, al.Title AS AlbumTitle
FROM Album al
RIGHT JOIN Artist ar ON al.ArtistId = ar.ArtistId;
```

ArtistName	AlbumTitle
AC/DC	Let There Be Rock
AC/DC	For Those About To Rock We Salute You
Accept	Restless and Wild
Accept	Balls to the Wall
Aerosmith	Big Ones
Alanis Morissette	Jagged Little Pill
Alice In Chains	Facelift
Antônio Carlos Jobim	Chill: Brazil (Disc 2)
Antônio Carlos Jobim	Warner 25 Anos
Apocalyptica	Plays Metallica By Four Cellos
Audioslave	Out Of Exile



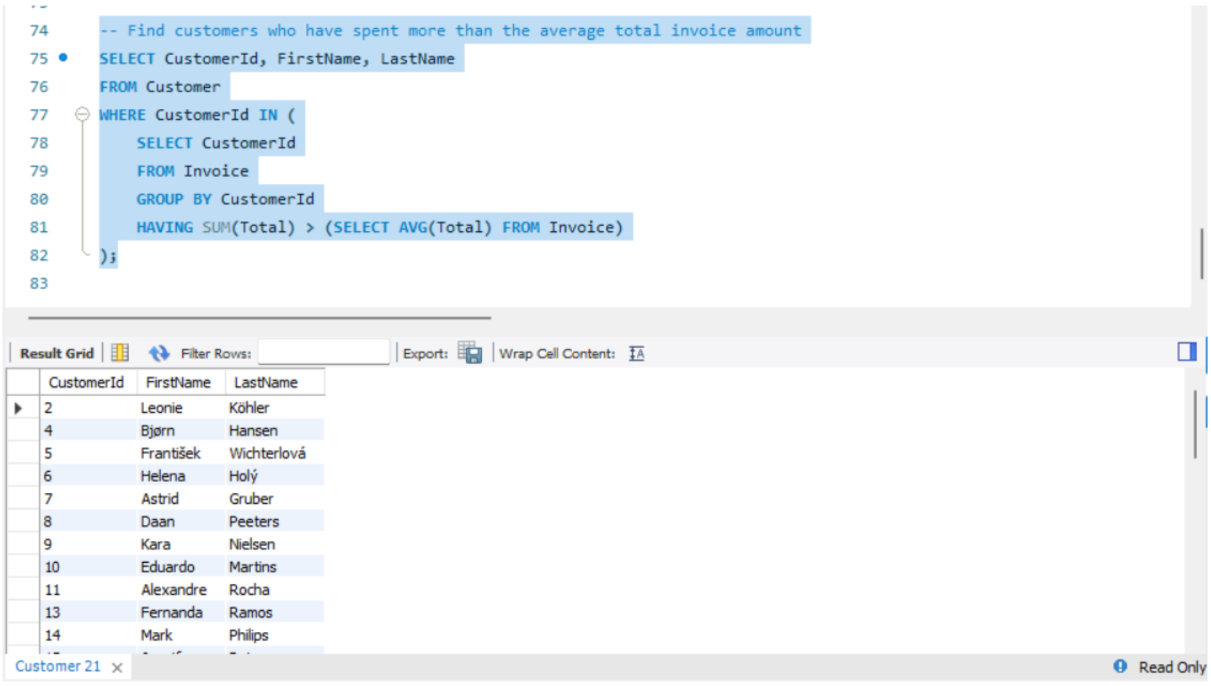
```sql
-- LEFT JOIN: List all albums and their artists (even if some albums have no artist)
SELECT al.Title AS AlbumTitle, ar.Name AS ArtistName
FROM Album al
LEFT JOIN Artist ar ON al.ArtistId = ar.ArtistId;
```

AlbumTitle	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

Result 19

Read Only

B) SUBQUERIES



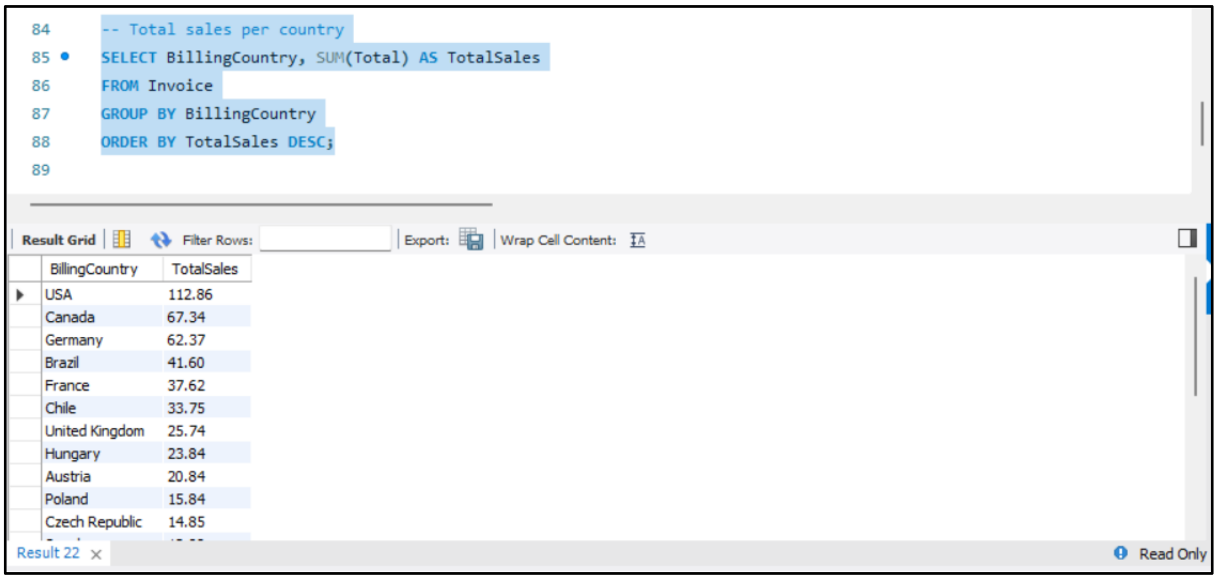
```sql
-- Find customers who have spent more than the average total invoice amount
SELECT CustomerId, FirstName, LastName
FROM Customer
WHERE CustomerId IN (
    SELECT CustomerId
    FROM Invoice
    GROUP BY CustomerId
    HAVING SUM(Total) > (SELECT AVG(Total) FROM Invoice)
);
```

CustomerId	FirstName	LastName
2	Leonie	Köhler
4	Bjørn	Hansen
5	František	Wichterlová
6	Helena	Holý
7	Astrid	Gruber
8	Daan	Peeters
9	Kara	Nielsen
10	Eduardo	Martins
11	Alexandre	Rocha
13	Fernanda	Ramos
14	Mark	Philips

Customer 21

Read Only

D) Use of AGGREGATE FUNCTIONS (SUM, AVG)

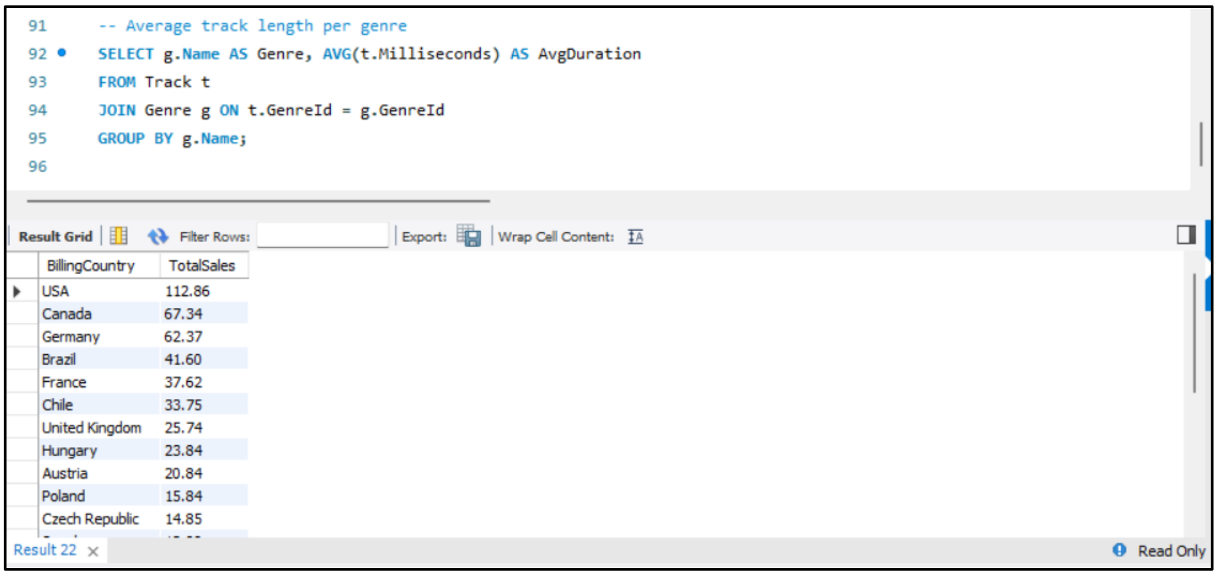


```
-- Total sales per country
SELECT BillingCountry, SUM(Total) AS TotalSales
FROM Invoice
GROUP BY BillingCountry
ORDER BY TotalSales DESC;
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

BillingCountry	TotalSales
USA	112.86
Canada	67.34
Germany	62.37
Brazil	41.60
France	37.62
Chile	33.75
United Kingdom	25.74
Hungary	23.84
Austria	20.84
Poland	15.84
Czech Republic	14.85

Result 22 × Read Only



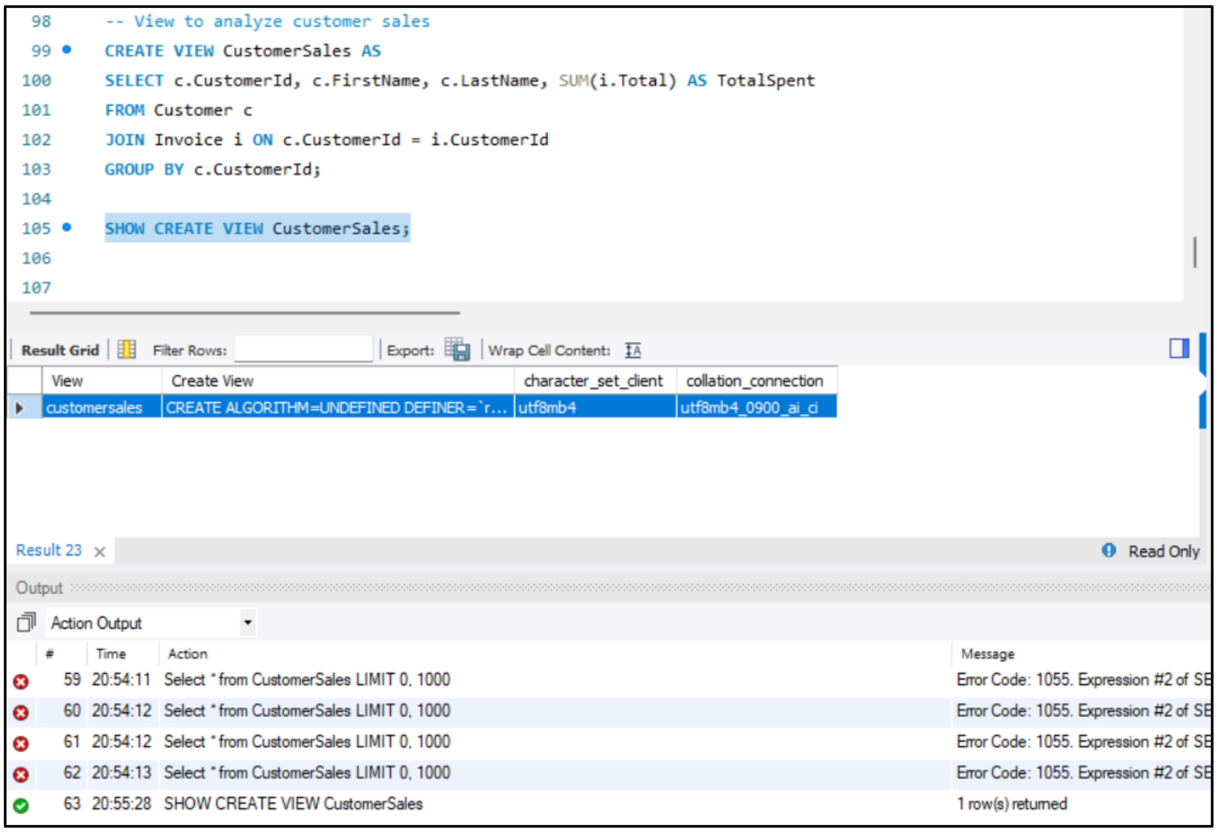
```
-- Average track length per genre
SELECT g.Name AS Genre, AVG(t.Milliseconds) AS AvgDuration
FROM Track t
JOIN Genre g ON t.GenreId = g.GenreId
GROUP BY g.Name;
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

BillingCountry	TotalSales
USA	112.86
Canada	67.34
Germany	62.37
Brazil	41.60
France	37.62
Chile	33.75
United Kingdom	25.74
Hungary	23.84
Austria	20.84
Poland	15.84
Czech Republic	14.85

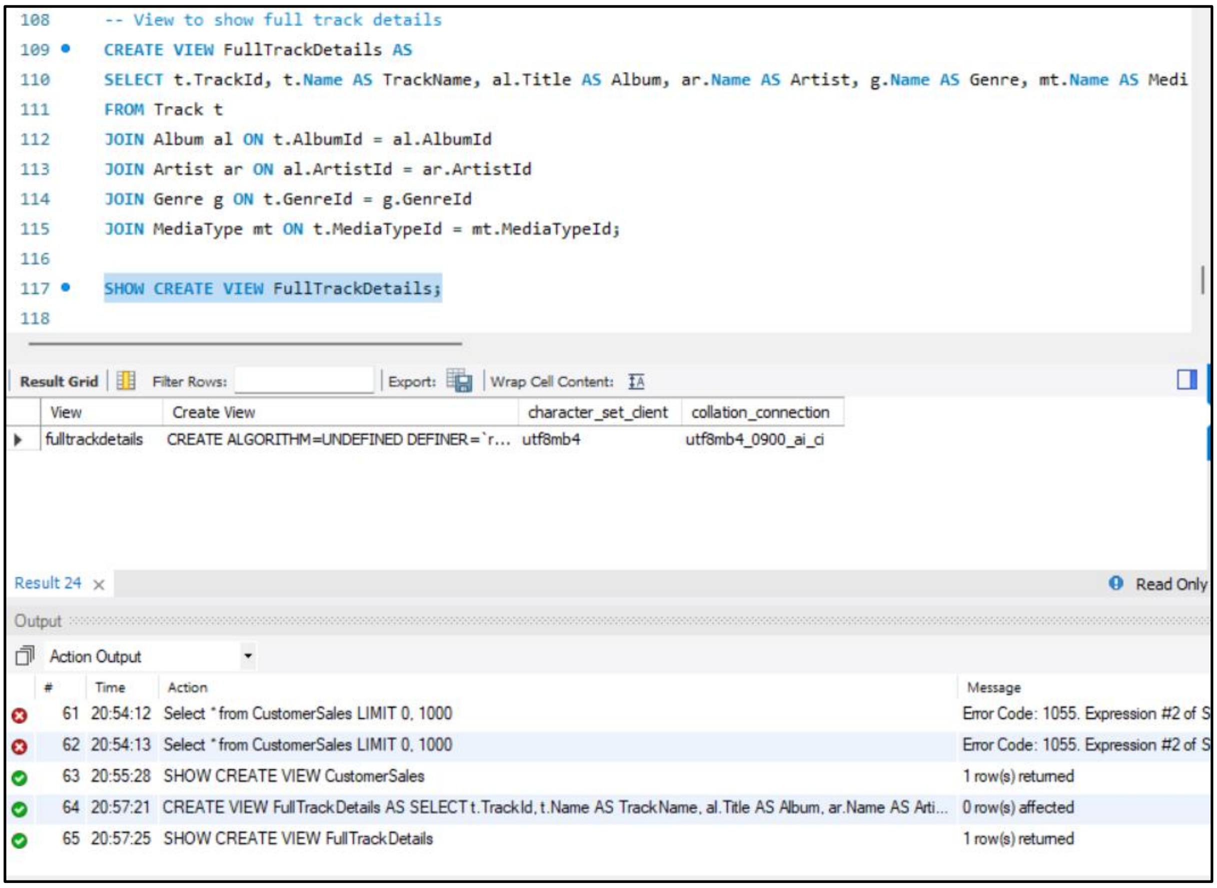
Result 22 × Read Only

E) CREATE VIEWS FOR ANALYSIS



```sql
-- View to analyze customer sales
CREATE VIEW CustomerSales AS
SELECT c.CustomerId, c.FirstName, c.LastName, SUM(i.Total) AS TotalSpent
FROM Customer c
JOIN Invoice i ON c.CustomerId = i.CustomerId
GROUP BY c.CustomerId;

SHOW CREATE VIEW CustomerSales;
```

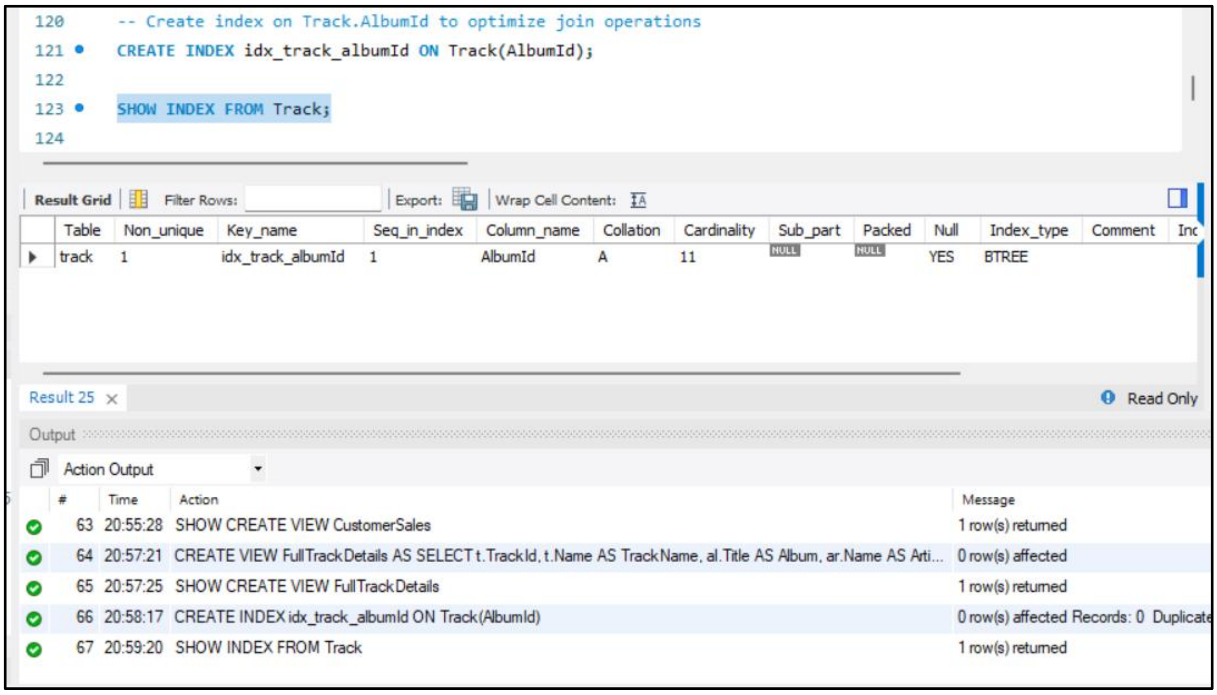
```
Result Grid | Filter Rows: | Export: | Wrap Cell Content:

  View            Create View                                  character_set_client   collation_connection
▶ customersales   CREATE ALGORITHM=UNDEFINED DEFINER=`r...     utf8mb4                utf8mb4_0900_ai_ci

Result 23 ×                                                                                  Read Only
Output
Action Output
   #   Time       Action                                          Message
   59  20:54:11   Select * from CustomerSales LIMIT 0, 1000       Error Code: 1055. Expression #2 of SE
   60  20:54:12   Select * from CustomerSales LIMIT 0, 1000       Error Code: 1055. Expression #2 of SE
   61  20:54:12   Select * from CustomerSales LIMIT 0, 1000       Error Code: 1055. Expression #2 of SE
   62  20:54:13   Select * from CustomerSales LIMIT 0, 1000       Error Code: 1055. Expression #2 of SE
   63  20:55:28   SHOW CREATE VIEW CustomerSales                  1 row(s) returned
```



```sql
-- View to show full track details
CREATE VIEW FullTrackDetails AS
SELECT t.TrackId, t.Name AS TrackName, al.Title AS Album, ar.Name AS Artist, g.Name AS Genre, mt.Name AS Medi
FROM Track t
JOIN Album al ON t.AlbumId = al.AlbumId
JOIN Artist ar ON al.ArtistId = ar.ArtistId
JOIN Genre g ON t.GenreId = g.GenreId
JOIN MediaType mt ON t.MediaTypeId = mt.MediaTypeId;

SHOW CREATE VIEW FullTrackDetails;
```

```
Result Grid | Filter Rows: | Export: | Wrap Cell Content:

  View               Create View                                  character_set_client   collation_connection
▶ fulltrackdetails   CREATE ALGORITHM=UNDEFINED DEFINER=`r...     utf8mb4                utf8mb4_0900_ai_ci

Result 24 ×                                                                                  Read Only
Output
Action Output
   #   Time       Action                                                                                  Message
   61  20:54:12   Select * from CustomerSales LIMIT 0, 1000                                               Error Code: 1055. Expression #2 of S
   62  20:54:13   Select * from CustomerSales LIMIT 0, 1000                                               Error Code: 1055. Expression #2 of S
   63  20:55:28   SHOW CREATE VIEW CustomerSales                                                          1 row(s) returned
   64  20:57:21   CREATE VIEW FullTrackDetails AS SELECT t.TrackId, t.Name AS TrackName, al.Title AS Album, ar.Name AS Arti...   0 row(s) affected
   65  20:57:25   SHOW CREATE VIEW FullTrackDetails                                                       1 row(s) returned
```

F) OPTIMIZED QUERIES WITH INDEXES



```
-- Create index on Track.AlbumId to optimize join operations
CREATE INDEX idx_track_albumId ON Track(AlbumId);

SHOW INDEX FROM Track;
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

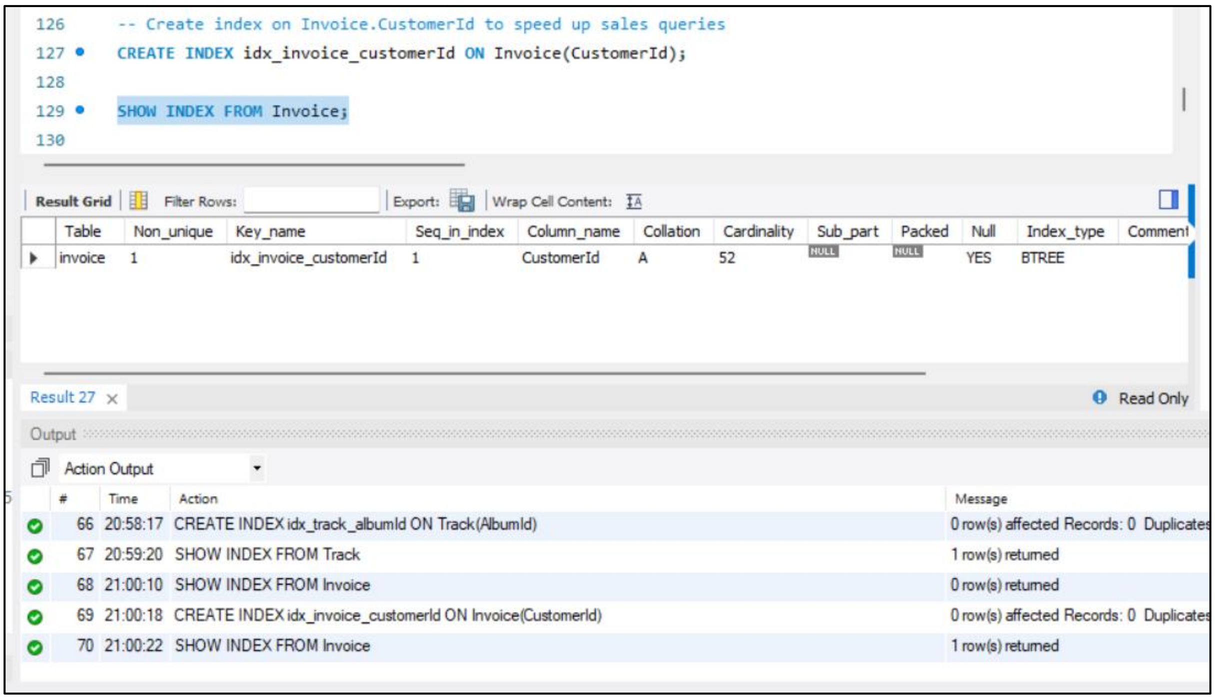
Table	Non_unique	Key_name	Seq_in_index	Column_name	Collation	Cardinality	Sub_part	Packed	Null	Index_type	Comment	Inc
track	1	idx_track_albumId	1	AlbumId	A	11	NULL	NULL	YES	BTREE		

Result 25 × Read Only

Output

Action Output

#	Time	Action	Message
63	20:55:28	SHOW CREATE VIEW CustomerSales	1 row(s) returned
64	20:57:21	CREATE VIEW FullTrackDetails AS SELECT t.TrackId, t.Name AS TrackName, al.Title AS Album, ar.Name AS Arti...	0 row(s) affected
65	20:57:25	SHOW CREATE VIEW FullTrackDetails	1 row(s) returned
66	20:58:17	CREATE INDEX idx_track_albumId ON Track(AlbumId)	0 row(s) affected Records: 0 Duplicate
67	20:59:20	SHOW INDEX FROM Track	1 row(s) returned



```
-- Create index on Invoice.CustomerId to speed up sales queries
CREATE INDEX idx_invoice_customerId ON Invoice(CustomerId);

SHOW INDEX FROM Invoice;
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

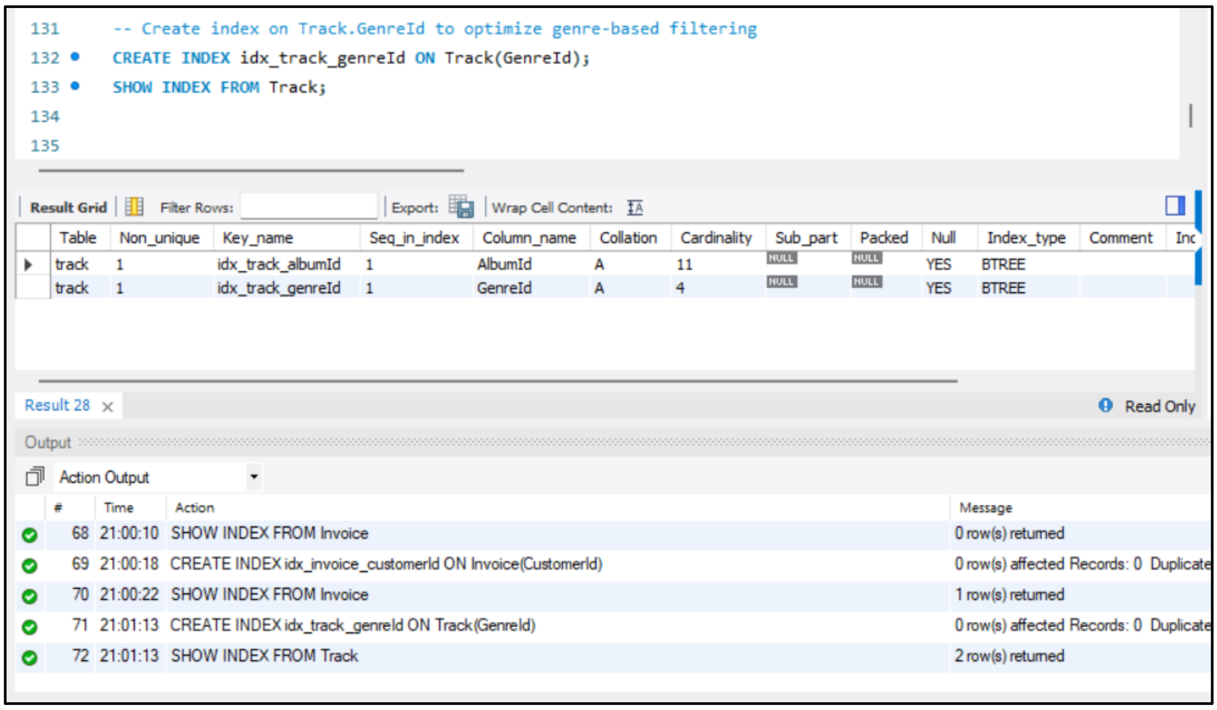
Table	Non_unique	Key_name	Seq_in_index	Column_name	Collation	Cardinality	Sub_part	Packed	Null	Index_type	Comment
invoice	1	idx_invoice_customerId	1	CustomerId	A	52	NULL	NULL	YES	BTREE	

Result 27 × Read Only

Output

Action Output

#	Time	Action	Message
66	20:58:17	CREATE INDEX idx_track_albumId ON Track(AlbumId)	0 row(s) affected Records: 0 Duplicates
67	20:59:20	SHOW INDEX FROM Track	1 row(s) returned
68	21:00:10	SHOW INDEX FROM Invoice	0 row(s) returned
69	21:00:18	CREATE INDEX idx_invoice_customerId ON Invoice(CustomerId)	0 row(s) affected Records: 0 Duplicates
70	21:00:22	SHOW INDEX FROM Invoice	1 row(s) returned



```
-- Create index on Track.GenreId to optimize genre-based filtering
CREATE INDEX idx_track_genreId ON Track(GenreId);
SHOW INDEX FROM Track;
```

Table	Non_unique	Key_name	Seq_in_index	Column_name	Collation	Cardinality	Sub_part	Packed	Null	Index_type	Comment	Inc
track	1	idx_track_albumId	1	AlbumId	A	11	NULL	NULL	YES	BTREE		
track	1	idx_track_genreId	1	GenreId	A	4	NULL	NULL	YES	BTREE		

Result 28 Read Only

Output

Action Output

#	Time	Action	Message
68	21:00:10	SHOW INDEX FROM Invoice	0 row(s) returned
69	21:00:18	CREATE INDEX idx_invoice_customerId ON Invoice(CustomerId)	0 row(s) affected Records: 0 Duplicate
70	21:00:22	SHOW INDEX FROM Invoice	1 row(s) returned
71	21:01:13	CREATE INDEX idx_track_genreId ON Track(GenreId)	0 row(s) affected Records: 0 Duplicate
72	21:01:13	SHOW INDEX FROM Track	2 row(s) returned