

```
2 • use musicstore;
3
4
5 #Select albums by a specific artist using of where,orderby and group by
6 • select *from album;
7 • SELECT AlbumId, Title, ArtistId
8 FROM album
9 WHERE Title LIKE '%Black%'
10 ORDER BY Title ASC;
11
```

Result Grid   Filter Rows: Edit:    Export/Import:   Wrap Cell Content: 

	AlbumId	Title	ArtistId
▶	321	Back to Black	252
	148	Black Album	50
	16	Black Sabbath	12
	17	Black Sabbath Vol. 4 (Remaster)	12
	208	[1997] Black Light Syndrome	136
•	NULL	NULL	NULL

album 30 x

Output

 Action Output 

#	Time	Action	Message
✓ 38	10:14:39	SELECT AlbumId, Title, ArtistId FROM album WHERE Title LIKE "%Black%" ORDER BY Title ASC LIMIT 0, 1000	5 row(s) returned
✓ 39	10:15:09	SELECT AlbumId, Title, ArtistId FROM album WHERE Title LIKE "%Black%" ORDER BY Title ASC LIMIT 0, 1000	5 row(s) returned

```

20      #Count albums per artist
21      SELECT artistid, COUNT(albumid) AS total_albums
22      FROM album
23      GROUP BY artistid
24      ORDER BY total_albums DESC;
25
26
27
28
29

```

Result Grid   Filter Rows: Export:  Wrap Cell Content: 

	artistid	total_albums
▶	90	21
	22	14
	58	11
	50	10
	150	10
	114	6
	118	5
	21	4
	82	4
	84	4
	149	4
	152	4
	8	3

Result 31 

Output

 Action Output 

	#	Time	Action	Message
✓	39	10:15:09	SELECT AlbumId, Title, ArtistId FROM album WHERE Title LIKE '%Black%' ORDER BY Title ASC LIMIT 0, 1000	5 row(s) returned
✓	40	10:16:53	SELECT artistid, COUNT(albumid) AS total_albums FROM album GROUP BY artistid ORDER BY total_albums ...	204 row(s) returned

```
--
32 # INNER JOIN: Albums with Artist names
33 • SELECT a.AlbumId, a.Title, ar.Name AS artist_name
34 FROM album a
35 INNER JOIN artist ar ON a.ArtistId = ar.ArtistId
36 ORDER BY artist_name;
37
38
39
```

Result Grid Filter Rows: Export: Wrap Cell Content:

	AlbumId	Title	artist_name
➤	296	A Copland Celebration, Vol. I	Aaron Copland & London Symphony Orchestra
	267	Worlds	Aaron Goldberg
	1	For Those About To Rock We Salute You	AC/DC
	4	Let There Be Rock	AC/DC
	280	The World of Classical Favourites	Academy of St. Martin in the Fields & Sir Neville ...
	281	Sir Neville Marriner: A Celebration	Academy of St. Martin in the Fields Chamber En...
	288	Fauré: Requiem, Ravel: Pavane & Others	Academy of St. Martin in the Fields, John Birch, ...
	327	Bach: Orchestral Suites Nos. 1 - 4	Academy of St. Martin in the Fields, Sir Neville ...
	2	Balls to the Wall	Accept
	3	Restless and Wild	Accept
	330	Górecki: Symphony No. 3	Adrian Leaper & Doreen de Feis
	5	Big Ones	Aerosmith
	262	Quiet Songs	Alpha Nine

Result 32 x

Output





Action Output

#	Time	Action	Message
40	10:16:53	SELECT artistid, COUNT(albumid) AS total_albums FROM album GROUP BY artistid ORDER BY total_albums ...	204 row(s) returned
41	10:18:00	SELECT a.AlbumId, a.Title, ar.Name AS artist_name FROM album a INNER JOIN artist ar ON a.ArtistId = ar.Arti...	347 row(s) returned

```

41
42 #LEFT JOIN: Customers with their invoices (including customers with no invoices)
43 • SELECT c.customerid, c.firstname, c.lastname, i.invoiceid, i.total
44 FROM customer c
45 LEFT JOIN invoice i ON c.customerid = i.customerid
46 ORDER BY c.lastname;
47
48

```

Result Grid   Filter Rows: Export:  Wrap Cell Content: 

	customerid	firstname	lastname	invoiceid	total
▶	12	Roberto	Almeida	34	0.99
	12	Roberto	Almeida	155	1.98
	12	Roberto	Almeida	166	13.86
	12	Roberto	Almeida	221	8.91
	12	Roberto	Almeida	350	1.98
	12	Roberto	Almeida	373	3.96
	12	Roberto	Almeida	395	5.94
	28	Julia	Barnett	71	1.98
	28	Julia	Barnett	82	13.86
	28	Julia	Barnett	137	8.91
	28	Julia	Barnett	266	1.98
	28	Julia	Barnett	289	3.96
	28	Julia	Barnett	311	11.94

Result 33 x

Output

 Action Output 

#	Time	Action	Message
✓ 41	10:18:00	SELECT a.AlbumId, a.Title, ar.Name AS artist_name FROM album a INNER JOIN artist ar ON a.ArtistId = ar.Arti...	347 row(s) returned
✓ 42	10:18:49	SELECT c.customerid, c.firstname, c.lastname, i.invoiceid, i.total FROM customer c LEFT JOIN invoice i ON c.c...	412 row(s) returned

```

54  #RIGHT JOIN: Employees and their managers
55  •  select *from employee;
56  •  SELECT e.FirstName AS employee_firstname, e.LastName AS employee_lastname,
57         m.firstname AS manager_firstname, m.lastname AS manager_lastname
58  FROM employee e
59  RIGHT JOIN employee m ON e.reportsto = m.employeeid;
60
61
62

```

Result Grid   Filter Rows: Export:  Wrap Cell Content: 

	employee_firstname	employee_lastname	manager_firstname	manager_lastname
▶	Nancy	Edwards	Andrew	Adams
	Michael	Mitchell	Andrew	Adams
	Jane	Peacock	Nancy	Edwards
	Margaret	Park	Nancy	Edwards
	Steve	Johnson	Nancy	Edwards
	NULL	NULL	Jane	Peacock
	NULL	NULL	Margaret	Park
	NULL	NULL	Steve	Johnson
	Robert	King	Michael	Mitchell
	Laura	Callahan	Michael	Mitchell
	NULL	NULL	Robert	King
	NULL	NULL	Laura	Callahan

Result 34 x

Output

 Action Output ▼

#	Time	Action	Message
✓ 42	10:18:49	SELECT c.customerid, c.firstname, c.lastname, i.invoiceid, i.total FROM customer c LEFT JOIN invoice i ON c.c...	412 row(s) returned
✓ 43	10:19:23	SELECT e.FirstName AS employee_firstname, e.LastName AS employee_lastname, m.firstname AS manage...	12 row(s) returned

```

66 #Customers who spent more than the average total amount
67 • SELECT CustomerId, FirstName, LastName, total_spent
68 FROM (SELECT c.CustomerId, c.FirstName, c.LastName, SUM(i.Total) AS total_spent
69       FROM customer c
70       JOIN invoice i ON c.CustomerId = i.CustomerId
71       GROUP BY c.CustomerId
72 ) AS customer_totals
73 WHERE total_spent > (
74     SELECT AVG(total_spent) FROM (SELECT SUM(total) AS total_spent FROM invoice GROUP BY customerId
75     ) AS avg_table
76 );

```

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

	CustomerId	FirstName	LastName	total_spent
▶	1	Luís	Gonçalves	39.62
	3	François	Tremblay	39.62
	4	Bjørn	Hansen	39.62
	5	František	Wichterlová	40.62
	6	Helena	Holý	49.62
	7	Astrid	Gruber	42.62
	17	Jack	Smith	39.62
	20	Dan	Miller	39.62
	22	Heather	Leacock	39.62
	24	Frank	Ralston	43.62
	25	Victor	Stevens	42.62
	26	Richard	Cunningham	47.62
	28	Julia	Barnett	43.62

Result 35 x

Output

Action Output

#	Time	Action	Message
✓ 43	10:19:23	SELECT e.FirstName AS employee_firstname, e.LastName AS employee_lastname, m.firstname AS manage...	12 row(s) returned
✓ 44	10:20:39	SELECT CustomerId, FirstName, LastName, total_spent FROM (SELECT c.CustomerId, c.FirstName, c.LastNam...	22 row(s) returned


```

78
79  # Average track length per genre for aggregate functions (SUM, AVG)
80 • SELECT g.Name AS genre_name, ROUND(AVG(t.milliseconds) / 60000, 2) AS avg_length_minutes
81 FROM track t
82 JOIN genre g ON t.GenreId = g.GenreId
83 GROUP BY g.name
84 ORDER BY avg_length_minutes DESC;
85
86
87
88 -- 8. Total revenue per country

```

Result Grid   Filter Rows: Export:  Wrap Cell Content: 

	genre_name	avg_length_minutes
▶	Sci Fi & Fantasy	48.53
	Science Fiction	43.76
	Drama	42.92
	TV Shows	35.75
	Comedy	26.42
	Metal	5.16
	Electronica/Dance	5.05
	Heavy Metal	4.96
	Classical	4.90
	Jazz	4.86
	Rock	4.73
	Blues	4.51
	Alternative	4.40

Result 36 ×

Output


 Action Output 

	#	Time	Action	Message
✓	44	10:20:39	SELECT CustomerId, FirstName, LastName, total_spent FROM (SELECT c.CustomerId, c.FirstName, c.LastNam...	22 row(s) returned
✓	45	10:21:58	SELECT g.Name AS genre_name, ROUND(AVG(t.milliseconds) / 60000, 2) AS avg_length_minutes FROM trac...	25 row(s) returned

```

94  #Top 5 selling tracks by revenue by creating views
95  • CREATE VIEW top_tracks AS
96  SELECT t.TrackId, t.Name AS track_name, SUM(il.UnitPrice * il.Quantity) AS total_revenue
97  FROM invoice il
98  JOIN track t ON il.TrackId = t.TrackId
99  GROUP BY t.TrackId, t.Name
100 ORDER BY total_revenue DESC
101 LIMIT 5;
102 • SELECT * FROM top_tracks;
103

```

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

	TrackId	track_name	total_revenue
▶	2832	The Woman King	3.98
	2868	Walkabout	3.98
	2850	The Fix	3.98
	3223	How to Stop an Exploding Man	3.98
	3214	Phyllis's Wedding	3.98

top_tracks 37 x

Output

 Action Output 

#	Time	Action	Message
✖ 46	10:23:13	CREATE VIEW top_tracks AS SELECT t.TrackId, t.Name AS track_name, SUM(il.UnitPrice * il.Quantity) AS tot...	Error Code: 1050. Table 'top_tracks' already exists
✔ 47	10:23:16	SELECT * FROM top_tracks LIMIT 0, 1000	5 row(s) returned