

Join with the Where clause

MySQL80 x

Query Database Server Tools Scripting Help

Query 1 x

Limit to 1000 rows

```
1 select firstname, lastname, invoice.invoiceId, invoiceDate, total,
2 trackID, unitprice, quantity
3 from invoice, customer, invoiceline
4 where billingcountry = 'USA'
5 and invoice.customerid = customer.customerid
6 and invoice.invoiceid = invoiceline.invoiceid
7 order by lastname, invoice.invoiceid, trackid
```

Result Grid

	firstname	lastname	invoiceId	invoiceDate	total	trackID	unitprice	quantity
	Julia	Barnett	71	2009-11-07 00:00:00	1.98	2322	0.99	1
	Julia	Barnett	71	2009-11-07 00:00:00	1.98	2324	0.99	1
	Julia	Barnett	82	2009-12-18 00:00:00	13.86	2651	0.99	1
	Julia	Barnett	82	2009-12-18 00:00:00	13.86	2660	0.99	1
	Julia	Barnett	82	2009-12-18 00:00:00	13.86	2669	0.99	1

Result Grid

Form

Join with the Join Statement

The screenshot shows the MySQL Workbench interface. The 'Query' tab is active, displaying a SQL query that performs a left join between the 'artist' and 'album' tables. The query is as follows:

```
1 select name, title
2 from artist left join album
3 on artist.artistid = album.ArtistId
4 order by name, title
```

The 'Result Grid' at the bottom shows the output of the query. The first row has a 'name' of 'A Cor Do Som' and a 'title' of 'NULL'. The subsequent rows show artists and their corresponding album titles.

name	title
A Cor Do Som	NULL
Aaron Copland & London Svmphony Orchestra	A Copland Celebration. Vol. I
Aaron Goldbera	Worlds
AC/DC	For Those About To Rock We Salute You
AC/DC	Let There Be Rock

Grouping and Summing

The screenshot displays the MySQL Workbench interface. The top menu bar includes File, Edit, View, Query, Database, Server, Tools, Scripting, and Help. Below the menu is a toolbar with various icons for file operations and database management. The left sidebar shows the 'SCHEMAS' tree with a filter box. Under the 'tunes' database, the 'Tables' folder is expanded, listing tables such as album, artist, custom, genre, invoice, invoice, mediat, playlist, playlist, and track. The main editor window shows a SQL query in 'Query 1' with a line number column on the left. The query is as follows:

```
1  
2  
3  
4 select customerid, sum(total) as 'TotalSale',  
5     max(total) as 'Largest Invoice Amount',  
6     min(total) as 'Smallest Invoice Amount',  
7     avg(total) as 'Average Invoice Amount'  
8 from invoice  
9 group by customerid  
10 order by customerid
```

The query is executed, and the results are limited to 1000 rows. The status bar at the bottom indicates 'No object selected'.

Nested queries

The screenshot shows a SQL IDE interface with a query editor and a result grid. The query editor displays a nested SQL query that filters customers based on their total invoice amount relative to the average invoice total.

```
1 select invoiceID, total
2 from invoice
3 where total >
4 (select avg(total)
5  from invoice)
6
7 select lastname, firstname
8 from customer
9 where customerID in
10 (select customerID from invoice
11  where total >
12   (select avg(total) from invoice))
```

The result grid shows the output of the query, displaying the last names and first names of the customers who meet the criteria.

lastname	firstname
Goncalves	Luís
Köhler	Leonie
Tremblay	Francois
Hansen	Biørn

The interface includes a sidebar with a tree view of database objects (Tables, Views, Stored Procedures) and a top toolbar with various SQL-related icons. The status bar at the bottom indicates the current object is 'customer 2' and the grid is in 'Read Only' mode.

