

Query

Query History

Scratch Pad

Q1 - Who is the senior most employee in the music store based on the job title?

SELECT * FROM employee

ORDER BY levels desc

limit 1

Data Output

Messages

Notifications

	employee_id [PK] character varying (50)	last_name character	first_name character	title character varying (50)	reports_to character varying (30)	levels character varying (10)	birthdate timestamp without time zon
1	9	Madan	Mohan	Senior General Manager	[null]	L7	1961-01-26 00:00:00

Query

Query History

Scratch Pad

limit 1

/*Q2 - Which countries have the most invoices? */

SELECT * FROM invoice

SELECT COUNT (*) as c, billing_country

FROM invoice

group by billing_country

order by c desc

Data Output

Messages

Notifications

	c bigint	billing_country character varying (30)
1	131	USA
2	76	Canada
3	61	Brazil
4	50	France
5	41	Germany
6	30	Czech Republic
7	29	Portugal
8	28	United Kingdom
9	21	India
10	13	Chile
11	13	Ireland
12	11	Spain
13	11	Finland

Total rows: 24 of 24

Query complete 00:00:00.100

Query

Query History

```

14
15 /*Q3 - Who is the best customer? Declare a person who has spent much money as the best customer. */
16 /*According to the schema table we have to link the customer with the invoice table. */
17
18 SELECT customer.customer_id, customer.first_name, customer.last_name, SUM(invoice.total) as total
19 FROM customer
20 JOIN invoice ON customer.customer_id = invoice.customer_id
21 GROUP BY customer.customer_id
22 ORDER BY total desc
23 LIMIT 3
24
25
26

```

Data Output

Messages

Notifications

	customer_id [PK] integer	first_name character	last_name character	total double precision
1	5	R	Madhav	144.54000000000002
2	6	Helena	Holy	128.7
3	46	Hugh	O'Reilly	114.83999999999997

Query

Query History

Scratch Pad

```

26 ✓ /* Q4: Write query to return the email, first name, last name, & Genre of all Rock Music listeners.
27 Return your list ordered alphabetically by email starting with A. */
28
29 ✓ /* From the schema table to obtain the rock genre we have to go to genre table. And to get the list of cust
30 Now we have to link customer table with the genre table. To do this we link Customer_id from customer t
31 Then we link invoice_id from invoice table with invoice_id from invoice_line table.
32 Now we link Track_id from Invoice_line table to track and finally genre_id from track to genre. */
33
34
35 SELECT DISTINCT email,first_name, last_name
36 FROM customer
37 JOIN invoice ON customer.customer_id = invoice.customer_id
38 JOIN invoice_line ON invoice.invoice_id = invoice_line.invoice_id
39 WHERE track_id IN(
40     SELECT track_id FROM track
41     JOIN genre ON track.genre_id = genre.genre_id
42     WHERE genre.name LIKE 'Rock'
43 )
44 ORDER BY email;

```

Data Output

Messages

Notifications

	email character varying (50)	first_name character	last_name character
1	aaronmitchell@yahoo.ca	Aaron	Mitchell
2	alero@uol.com.br	Alexandre	Rocha
3	astrid.gruber@apple.at	Astrid	Gruber
4	bjorn.hansen@yahoo.no	Bjorn	Hansen

Total rows: 59 of 59

Query complete 00:00:00.109

Ln 44, Col 16

Query Query History

```
46 ▾ /* Q5: Let's invite the artists who have written the most rock music in our dataset.
47 Write a query that returns the Artist name and total track count of the top 10 rock bands. */
48
49 ▾ SELECT artist.artist_id, artist.name, COUNT(artist.artist_id) AS number_of_songs
50 FROM track
51 JOIN album ON album.album_id = track.album_id
52 JOIN artist ON artist.artist_id = album.artist_id
53 JOIN genre ON genre.genre_id = track.genre_id
54 WHERE genre.name LIKE 'Rock'
55 GROUP BY artist.artist_id
56 ORDER BY number_of_songs DESC
57 LIMIT 10;
```

Data Output Messages Notifications



	artist_id [PK] character varying (50)	name character varying (120)	number_of_songs bigint
1	22	Led Zeppelin	114
2	150	U2	112
3	58	Deep Purple	92
4	90	Iron Maiden	81
5	118	Pearl Jam	54
6	152	Van Halen	52
7	51	Queen	45
8	142	The Rolling Stones	41
9	76	Creedence Clearwater Revival	40
10	52	Kiss	35

Query Query History

```
58
59 /* Q6: Return all the track names that have a song length longer than the average song length.
60 Return the Name and Milliseconds for each track. Order by the song length with the longest songs listed first.
61
62 SELECT name,milliseconds
63 FROM track
64 WHERE milliseconds > (
65     SELECT AVG(milliseconds) AS avg_track_length
66     FROM track)
67 ORDER BY milliseconds DESC;
68
```

[Data Output](#) [Messages](#) [Notifications](#)

	name character varying (150)	milliseconds integer
1	Occupation / Precipice	5286953
2	Through a Looking Glass	5088838
3	Greetings from Earth, Pt. 1	2960293
4	The Man With Nine Lives	2956998
5	Battlestar Galactica, Pt. 2	2956081
6	Battlestar Galactica, Pt. 1	2952702
7	Murder On the Rising Star	2935894
8	Battlestar Galactica, Pt. 3	2927802
9	Take the Celestra	2927677
10	Fire In Space	2926593
11	The Long Patrol	2925008
12	The Magnificent Warriors	2924716