

SQL PROJECT- MUSIC STORE DATA ANALYSIS

Questions

1. Who is the senior most employee based on job title?

Query

Query History

1

SELECT * FROM EMPLOYEE

2

order by levels desc

3

limit 1

4

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)

1

9

Madan

Mohan

Senior General Manager

[null]

L7

2. Which countries have the most Invoices?

Query		Query History	
1	SELECT COUNT(*)	as c,	billing_country
2	FROM	invoice	
3	GROUP BY	billing_country	
4	order by	c	desc
5			
6			
7			

Data Output		Messages		Notifications	
	c	billing_country			
	bigint	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			

3. What are top 3 values of total invoice?

QueryQuery History

1234

SELECT

total

FROM

invoice

order by

total

desc

limit

3

Data OutputMessagesNotifications

total

double precision

1

23.759999999999998

2

19.8

3

19.8

4. Which city has the best customers? We would like to throw a promotional Music Festival in the city we made the most money. Write a query that returns one city that has the highest sum of invoice totals. Return both the city name & sum of all invoice totals.

Query

Query History

1

SELECT SUM(total) as total_invoice , billing_city from invoice

2

GROUP BY billing_city

3

ORDER BY total_invoice desc

4

Data Output

Messages

Notifications

≡

📄

▼

📋

▼

🗑️

🗄️

⬇️

📈

	total_invoice double precision	billing_city character varying (30)
1	273.24000000000007	Prague
2	169.29	Mountain View
3	166.32	London
4	158.4	Berlin
5	151.47	Paris
6	129.69	São Paulo
7	114.83999999999997	Dublin
8	111.86999999999999	Delhi
9	108.89999999999998	São José dos Campos

5. Who is the best customer? The customer who has spent the most money will be declared the best customer. Write a query that returns the person who has spent the most money.

Query

Query History

1

SELECT customer.customer_id, customer.first_name, customer.last_name, SUM(invoice.total) as total

2

from customer

3

JOIN invoice ON customer.customer_id = invoice.customer_id

4

GROUP BY customer.customer_id

5

ORDER BY total desc

6

limit 1

7

Data Output

Messages

Notifications

≡

📄

▼

📋

▼

🗑️

🗄️

⬇️

📈

	customer_id [PK] integer	first_name character	last_name character	total double precision
1	5	R	Madhav	144.54000000000002

6. Write a query to return the email, first name, last name, & Genre of all Rock Music listeners. Return your list ordered alphabetically by email starting with A.

Query Query History

1

SELECT DISTINCT email, first_name, last_name

2

from customer

3

JOIN invoice ON customer.customer_id = invoice.customer_id

4

JOIN invoice_line ON invoice.invoice_id = invoice_line.invoice_id

5

WHERE track_id IN(

6

SELECT track_id FROM track

7

JOIN genre ON track.genre_id = genre.genre_id

8

WHERE genre.name LIKE 'Rock'

9

)

10

11

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	Bjørn	Hansen
5	camille.bernard@yahoo.fr	Camille	Bernard
6	daan_peeters@apple.be	Daan	Peeters
7	diego.gutierrez@yahoo.ar	Diego	Gutiérrez
8	dmiller@comcast.com	Dan	Miller
9	dominiquelefebvre@gmail.c...	Dominique	Lefebvre
10	edfrancis@yachoo.ca	Edward	Francis
11	eduardo@woodstock.com.br	Eduardo	Martins
12	ellie.sullivan@shaw.ca	Ellie	Sullivan
13	emma_jones@hotmail.com	Emma	Jones
14	enrique_munoz@yahoo.es	Enrique	Muñoz
15	fernadaramos4@uol.com.br	Fernanda	Ramos
16	fharris@google.com	Frank	Harris

7. Let's invite the artists who have written the most rock music in our dataset. Write a query that returns the Artist name and total track count of the top 10 rock bands.

Query Query History

1

SELECT artist.artist_id, artist.name,COUNT(artist.artist_id) AS number_of_songs

2

FROM track

3

JOIN album ON album.album_id = track.album_id

4

JOIN artist ON artist.artist_id = album.artist_id

5

JOIN genre ON genre.genre_id = track.genre_id

6

WHERE genre.name LIKE 'Rock'

7

GROUP BY artist.artist_id

8

ORDER BY number_of_songs DESC

9

LIMIT 10;

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

- Return all the track names that have a song length longer than the average song length. Return the Name and Milliseconds for each track. Order by the song length with the longest songs listed first.

Query Query History	
1	SELECT name,milliseconds
2	FROM track
3	WHERE milliseconds > (
4	SELECT AVG(milliseconds) AS avg_track_length
5	FROM track)
6	ORDER BY milliseconds DESC;
7	
8	SELECT name,milliseconds
9	FROM track
10	WHERE milliseconds > 393599
11	ORDER BY milliseconds DESC;
12	

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
13	The Living Legend, Pt. 1	2924507
14	The Gun On Ice Planet Zero, Pt. 2	2924341
15	The Hand of God	2924007

9. Find how much amount spent by each customer on artists? Write a query to return customer name, artist name and total spent.

Query Query History	
1	--CTE
2	WITH best_selling_artist AS (
3	SELECT artist.artist_id AS artist_id, artist.name AS artist_name,
4	SUM(invoice_line.unit_price*invoice_line.quantity) AS total_sales
5	FROM invoice_line
6	JOIN track ON track.track_id = invoice_line.track_id
7	JOIN album ON album.album_id = track.album_id
8	JOIN artist ON artist.artist_id = album.artist_id
9	GROUP BY 1
10	ORDER BY 3 DESC
11	LIMIT 1
12	
13)
14	SELECT c.customer_id, c.first_name, c.last_name, bsa.artist_name,
15	SUM(il.unit_price*il.quantity) AS amount_spent
16	FROM invoice i
17	JOIN customer c ON c.customer_id = i.customer_id
18	JOIN invoice_line il ON il.invoice_id = i.invoice_id
19	JOIN track t ON t.track_id = il.track_id
20	JOIN album alb ON alb.album_id = t.album_id
21	JOIN best_selling_artist bsa ON bsa.artist_id = alb.artist_id
22	GROUP BY 1,2,3,4
23	ORDER BY 5 DESC;

Data Output							Messages	Notifications
	customer_id integer	first_name character	last_name character	artist_name character varying (120)	amount_spent double precision			
1	46	Hugh	O'Reilly	Queen	27.719999999999985			
2	38	Niklas	Schröder	Queen	18.81	...		
3	3	François	Tremblay	Queen	17.82	...		
4	34	João	Fernandes	Queen	16.830000000000002	...		
5	53	Phil	Hughes	Queen	11.88	...		
6	41	Marc	Dubois	Queen	11.88			
7	47	Lucas	Mancini	Queen	10.89	...		
8	33	Ellie	Sullivan	Queen	10.89			
9	20	Dan	Miller	Queen	3.96			
10	5	R	Madhav	Queen	3.96	...		
11	23	John	Gordon	Queen	2.9699999999999998	...		
12	54	Steve	Murray	Queen	2.9699999999999998			
13	31	Martha	Silk	Queen	2.9699999999999998	...		
14	16	Frank	Harris	Queen	1.98			
15	17	Jack	Smith	Queen	1.98			