Query | Query History

```sql
--Q1: Who is the senior most employee based on job title?
SELECT * FROM employee
ORDER BY levels desc
LIMIT 1;
```

Data Output | Messages | Notifications

Showing rows: 1 to 1 | Page No: 1 of 1

	employee_id [PK] character varying (50)	last_name character (50)	first_name character (50)	title character varying (50)	reports_to character varying (30)	levels character varying (10)	birthdate timestamp w
1	9	Madan ...	Mohan ...	Senior General Manager	[null]	L7	1961-01-26 0

Query | Query History

```
6  --Q2: Which countries have the most Invoices?
7  SELECT COUNT(*) AS c, billing_country
8  FROM invoice
9  GROUP BY  billing_country
10 ORDER BY c desc;
11
```

Data Output | Messages | Notifications

Showing rows: 1 to 24 | Page No: 1 of 1

	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

Query | Query History

```
12  --Q3: What are top 3 values of total invoice?
13  SELECT total FROM invoice
14  ORDER BY total desc
15  LIMIT 3;
16
17
```

Data Output | Messages | Notifications

Showing rows: 1 to 3 | Page No: 1 of 1

	total double precision
1	23.759999999999998
2	19.8
3	19.8

Query | Query History

```sql
/* Q4: 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 */
SELECT SUM(total) AS invoice_total, billing_city
FROM invoice
GROUP BY billing_city
ORDER BY invoice_total desc
LIMIT 1;
```

Data Output | Messages | Notifications

Showing rows: 1 to 1 | Page No: 1 of 1

	invoice_total double precision	billing_city character varying (30)
1	273.24000000000007	Prague

Query | Query History

```
28
29  /* Q5: Who is the best customer?
30      The customer who has spent the most money will be declared the best customer.
31      Write a query that returns the person who has spent the most money.*/
32  SELECT customer.customer_id, customer.first_name, customer.last_name, SUM(invoice.total) AS total
33  FROM customer
34  JOIN invoice ON customer.customer_id = invoice.customer_id
35  GROUP BY customer.customer_id
36  ORDER BY total desc
37  LIMIT 1;
```

Data Output | Messages | Notifications

Showing rows: 1 to 1 | Page No: 1 of 1

	customer_id [PK] integer	first_name character (50)	last_name character (50)	total double precision
1	5	R ...	Madhav ...	144.54000000000002

Query | Query History

```sql
/* Q6: Write 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. */
select DISTINCT email, first_name, last_name
FROM customer
JOIN invoice ON customer.customer_id = invoice.customer_id
JOIN invoice_line ON invoice.invoice_id = invoice_line.invoice_id
WHERE track_id IN (SELECT track_id FROM track
JOIN genre ON track.genre_id = genre.genre_id
WHERE genre.name LIKE 'Rock')
ORDER BY email;
```

Data Output | Messages | Notifications

Showing rows: 1 to 59 | Page No: 1 of 1

	email character varying (50)	first_name character (50)	last_name character (50)
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

Query | Query History

```sql
/* Q7: 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. */
SELECT artist.artist_id, artist.name,COUNT(artist.artist_id) AS number_of_songs
FROM track
JOIN album ON album.album_id = track.album_id
JOIN artist ON artist.artist_id = album.artist_id
JOIN genre ON genre.genre_id = track.genre_id
WHERE genre.name LIKE 'Rock'
GROUP BY artist.artist_id
ORDER BY number_of_songs desc
LIMIT 10;
```

Data Output | Messages | Notifications

Showing rows: 1 to 10 | Page No: 1 of 1

	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

```sql
/* Q8: 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. */
SELECT name, milliseconds
FROM track
WHERE milliseconds >(SELECT AVG(milliseconds) AS Average_track_Length
FROM track)
ORDER BY milliseconds desc;
```

Data Output | Messages | Notifications

Showing rows: 1 to 494 | Page No: 1 of 1

	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

Query | Query History

```sql
/* Q9: Find how much amount spent by each customer on artists?
    Write a query to return customer name, artist name and total spent */
WITH best_selling_artist AS (
SELECT artist.artist_id AS artist_id, artist.name AS artist_name,
SUM(invoice_line.unit_price*invoice_line.quantity) AS total_sales
FROM invoice_line
JOIN track ON track.track_id = invoice_line.track_id
JOIN album ON album.album_id = track.album_id
JOIN artist ON artist.artist_id = album.artist_id
GROUP BY 1
ORDER BY 3 desc
LIMIT 1
    )
    SELECT c.customer_id, c.first_name, c.last_name, bsa.artist_name,
    SUM(il.unit_price*il.quantity) AS amount_spent
    FROM invoice i
    JOIN customer c ON c.customer_id = i.customer_id
    JOIN invoice_line il ON il.invoice_id = i.invoice_id
    JOIN track t ON t.track_id = il.track_id
    JOIN album alb ON alb.album_id = t.album_id
    JOIN best_selling_artist bsa ON bsa.artist_id = alb.artist_id
    GROUP BY 1,2,3,4
    ORDER BY 5 desc;
```

Data Output | Messages | Notifications

Showing rows: 1 to 43 | Page No: 1 of 1

	customer_id integer	first_name character (50)	last_name character (50)	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

Query | Query History

```
95
96  /* Q10: We want to find out the most popular music Genre for each country.
97      We determine the most popular genre as the genre
98      with the highest amount of purchases.
99      Write a query that returns each country along with the top Genre.
100     For countries where the maximum number of purchases is shared return all Genres. */
101
102 WITH popular_genre AS
103 (
104 SELECT COUNT(invoice_line.quantity) AS purchases, customer.country, genre.name, genre.genre_id,
105 row_number() OVER(PARTITION BY customer.country
106 ORDER BY COUNT(invoice_line.quantity) desc) AS RowNo
107 FROM invoice_line
108 JOIN invoice ON invoice.invoice_id = invoice_line.invoice_id
109 JOIN customer ON customer.customer_id = invoice.customer_id
110 JOIN track ON track.track_id = invoice_line.track_id
111 JOIN genre ON genre.genre_id = track.genre_id
112 GROUP BY 2,3,4
113 ORDER BY 2 ASC, 1 desc
114 )
115 SELECT  * FROM popular_genre WHERE RowNo <= 1;
```

Data Output | Messages | Notifications

Showing rows: 1 to 24 | Page No: 1 of 1

	purchases bigint	country character varying (50)	name character varying (120)	genre_id character varying (50)	rowno bigint
1	17	Argentina	Alternative & Punk	4	1
2	34	Australia	Rock	1	1
3	40	Austria	Rock	1	1
4	26	Belgium	Rock	1	1
5	205	Brazil	Rock	1	1
6	333	Canada	Rock	1	1
7	61	Chile	Rock	1	1
8	143	Czech Republic	Rock	1	1
9	24	Denmark	Rock	1	1
10	46	Finland	Rock	1	1
11	211	France	Rock	1	1
12	194	Germany	Rock	1	1

Query | Query History

Execute script F5

```
/* Q11: Write a query that determi                tomer that has spent the most on music for each country.
    Write a query that returns the country along with the top customer and how much they spent.
    For countries where the top amount spent is shared, provide all customers who spent this amount. */
with Customter_with_country as (
select customer.customer_id,first_name,last_name,billing_country,sum(total) as total_spending,
row_number() over(partition by billing_country
order by sum(total) desc) as RowNo
from invoice
join customer on customer.customer_id = invoice.customer_id
group by 1,2,3,4
order by 4 asc,5 desc)
select * from Customter_with_country where RowNo <= 1;
```

Data Output | Messages | Notifications

Showing rows: 1 to 24 | Page No: 1 of 1

	customer_id integer	first_name character (50)	last_name character (50)	billing_country character varying (30)	total_spending double precision	rowno bigint
1	56	Diego ...	Gutiérrez ...	Argentina	39.6	1
2	55	Mark ...	Taylor ...	Australia	81.18	1
3	7	Astrid ...	Gruber ...	Austria	69.3	1
4	8	Daan ...	Peeters ...	Belgium	60.38999999999999	1
5	1	Luís ...	Goncalves ...	Brazil	108.89999999999998	1