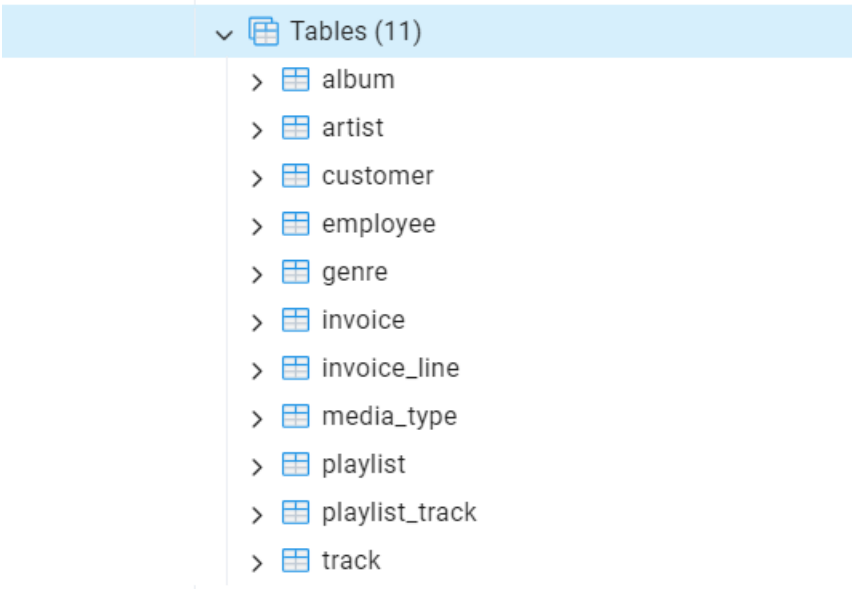
Music Store Analysis PostgreSQL

LinkedIn

By Om Gurav

Total Tables



/* Question Set 1 - Easy */

/* Q1: Who is the senior most employee based on job title? */

```
SELECT title, last_name, first_name
FROM employee
ORDER BY levels DESC
LIMIT 1
```

	title character varying (50)	last_name character	first_name character
1	Senior General Manager	Madan	Mohan

/* Q2: Which countries have the most Invoices? */

```
SELECT COUNT(*) AS c, billing_country
FROM invoice
GROUP BY billing_country
ORDER BY c DESC
```

	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
14	10	Australia
15	10	Netherlands
16	10	Sweden
17	10	Poland
18	10	Hungary
19	10	Denmark
20	9	Austria
21	9	Norway

/* Q3: What are top 3 values of total invoice? */

```
SELECT total
FROM invoice
ORDER BY total DESC
```

	total double precision
1	23.759999999999998
2	19.8
3	19.8
4	19.8
5	19.8
6	18.81
7	17.82
8	17.82
9	17.82
10	17.82
11	17.82
12	17.82
13	17.82
14	16.83
15	16.83
16	16.83
17	16.83
18	16.83
19	16.83
20	16.83

/* 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 billing_city,SUM(total) AS InvoiceTotal
FROM invoice
GROUP BY billing_city
ORDER BY InvoiceTotal DESC
LIMIT 1;
```

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

/* Q5: 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.*/

```
SELECT customer.customer_id, first_name, last_name, SUM(total) AS total_spending
FROM customer
JOIN invoice ON customer.customer_id = invoice.customer_id
GROUP BY customer.customer_id
ORDER BY total_spending DESC
LIMIT 1;
```

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

/* Question Set 2 - Moderate */

/* Q1: 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. */

/*Method 1 */

```
SELECT DISTINCT email,first_name, last_name
FROM customer
JOIN invoice ON customer.customer_id = invoice.customer_id
JOIN invoiceline ON invoice.invoice_id = invoiceline.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;
```

	first_name character	last_name character	email character varying (50)
1	Aaron	Mitchell	aaronmitchell@yahoo.ca
2	Alexandre	Rocha	alero@uol.com.br
3	Astrid	Gruber	astrid.gruber@apple.at
4	Bjørn	Hansen	bjorn.hansen@yahoo.no
5	Camille	Bernard	camille.bernard@yahoo.fr
6	Daan	Peeters	daan_peeters@apple.be
7	Diego	Gutiérrez	diego.gutierrez@yahoo.ar
8	Dan	Miller	dmiller@comcast.com
9	Dominique	Lefebvre	dominiquelefebvre@gmail.c...
10	Edward	Francis	edfrancis@yachoo.ca
11	Eduardo	Martins	eduardo@woodstock.com.br
12	Ellie	Sullivan	ellie.sullivan@shaw.ca
13	Emma	Jones	emma_jones@hotmail.com
14	Enrique	Muñoz	enrique_munoz@yahoo.es
15	Fernanda	Ramos	fernadaramos4@uol.com.br
16	Frank	Harris	fharris@google.com
17	Frank	Ralston	fralston@gmail.com
18	François	Tremblay	ftremblay@gmail.com
19	Fynn	Zimmermann	fzimmermann@yahoo.de
20	Hannah	Schneider	hannah.schneider@yahoo.de
21	Helena	Holý	hholy@gmail.com

/* Method 2 */

```
SELECT DISTINCT email AS Email,first_name AS FirstName, last_name AS LastName, genre.name AS Name
FROM customer
JOIN invoice ON invoice.customer_id = customer.customer_id
JOIN invoiceline ON invoiceline.invoice_id = invoice.invoice_id
JOIN track ON track.track_id = invoiceline.track_id
JOIN genre ON genre.genre_id = track.genre_id
WHERE genre.name LIKE 'Rock'
ORDER BY email;
```

	email character varying (50)	firstname character	lastname character	name character varying (120)
1	aaronmitchell@yahoo.ca	Aaron	Mitchell	Rock
2	alero@uol.com.br	Alexandre	Rocha	Rock
3	astrid.gruber@apple.at	Astrid	Gruber	Rock
4	bjorn.hansen@yahoo.no	Bjørn	Hansen	Rock
5	camille.bernard@yahoo.fr	Camille	Bernard	Rock
6	daan_peeters@apple.be	Daan	Peeters	Rock
7	diego.gutierrez@yahoo.ar	Diego	Gutiérrez	Rock
8	dmiller@comcast.com	Dan	Miller	Rock
9	dominiquelefebvre@gmail.c...	Dominique	Lefebvre	Rock
10	edfrancis@yachoo.ca	Edward	Francis	Rock
11	eduardo@woodstock.com.br	Eduardo	Martins	Rock
12	ellie.sullivan@shaw.ca	Ellie	Sullivan	Rock
13	emma_jones@hotmail.com	Emma	Jones	Rock
14	enrique_munoz@yahoo.es	Enrique	Muñoz	Rock
15	fernadaramos4@uol.com.br	Fernanda	Ramos	Rock
16	fharris@google.com	Frank	Harris	Rock
17	fralston@gmail.com	Frank	Ralston	Rock
18	ftremblay@gmail.com	François	Tremblay	Rock
19	fzimmermann@yahoo.de	Fynn	Zimmermann	Rock
20	hannah.schneider@yahoo.de	Hannah	Schneider	Rock
21	hholy@gmail.com	Helena	Holý	Rock

/* Q2: 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;
```

	artist_id [PK] character varying (50)	name character varying (120)	num_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

/* Q3: 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 > 393599.212103910933
order by milliseconds desc
```

	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
16	Experiment In Terra	2923548
17	War of the Gods, Pt. 2	2923381
18	The Living Legend, Pt. 2	2923298
19	War of the Gods, Pt. 1	2922630
20	Lost Planet of the Gods, Pt. 1	2922547
21	Baltar's Escape	2922088

/* Question Set 3 - Advance */

/* Q1: Find how much amount spent by each customer on artists? Write a query to return customer name, artist name and total spent */

/* Steps to Solve: First, find which artist has earned the most according to the InvoiceLines. Now use this artist to find which customer spent the most on this artist. For this query, you will need to use the Invoice, InvoiceLine, Track, Customer, Album, and Artist tables. Note, this one is tricky because the Total spent in the Invoice table might not be on a single product, so you need to use the InvoiceLine table to find out how many of each product was purchased, and then multiply this by the pricefor each artist. */

```
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;
```

	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	6	Helena	Holý	Red Hot Chili Peppers	19.799999999999997
3	46	Hugh	O'Reilly	Nirvana	18.81
4	38	Niklas	Schröder	Queen	18.81
5	3	François	Tremblay	Queen	17.82
6	28	Julia	Barnett	Jimi Hendrix	16.830000000000002
7	58	Manoj	Pareek	Jimi Hendrix	16.830000000000002
8	34	João	Fernandes	Jimi Hendrix	16.830000000000002
9	34	João	Fernandes	Queen	16.830000000000002
10	37	Fynn	Zimmermann	Jimi Hendrix	16.830000000000002
11	6	Helena	Holý	Jimi Hendrix	16.830000000000002
12	27	Patrick	Gray	Nirvana	16.830000000000002
13	3	François	Tremblay	Jimi Hendrix	16.830000000000002
14	50	Enrique	Muñoz	Jimi Hendrix	16.830000000000002
15	12	Roberto	Almeida	Jimi Hendrix	16.830000000000002
16	52	Emma	Jones	Red Hot Chili Peppers	15.840000000000002
17	25	Victor	Stevens	Pearl Jam	15.840000000000002
18	13	Fernanda	Ramos	Red Hot Chili Peppers	15.840000000000002
19	57	Luis	Rojas	Red Hot Chili Peppers	14.850000000000001
20	40	Dominique	Lefebvre	Pearl Jam	14.850000000000001
21	11	Alexandre	Rocha	Pearl Jam	13.860000000000001

/* Q2: We want to find out the most popular music Genre for each country. We determine the most popular genre as the genre with the highest amount of purchases. Write a query that returns each country along with the top Genre. For countries where the maximum number of purchases is shared return all Genres. */
/* Steps to Solve: There are two parts in question- first most popular music genre and second need data at country level. */

/* Method 1: Using CTE */

```
WITH popular_genre AS
(
   SELECT COUNT(invoice_line.quantity) AS purchases, customer.country, genre.name,
genre.genre_id,
          ROW_NUMBER() OVER(PARTITION BY customer.country ORDER BY
COUNT(invoice_line.quantity) DESC) AS RowNo
   FROM invoice_line
          JOIN invoice ON invoice.invoice_id = invoice_line.invoice_id
          JOIN customer ON customer.customer_id = invoice.customer_id
          JOIN track ON track.track_id = invoice_line.track_id
          JOIN genre ON genre.genre_id = track.genre_id
          GROUP BY 2,3,4
          ORDER BY 2 ASC, 1 DESC
)
SELECT * FROM popular_genre WHERE RowNo <= 1
```

Data Output | Messages | Notifications

	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
13	44	Hungary	Rock	1	1
14	102	India	Rock	1	1
15	72	Ireland	Rock	1	1
16	35	Italy	Rock	1	1
17	33	Netherlands	Rock	1	1
18	40	Norway	Rock	1	1
19	40	Poland	Rock	1	1
20	108	Portugal	Rock	1	1
21	46	Spain	Rock	1	1
22	60	Sweden	Rock	1	1

Total rows: 24 of 24 | Query complete 00:00:00.086

/* Method 2: : Using Recursive */

```
WITH RECURSIVE
	sales_per_country AS(
		SELECT COUNT(*) AS purchases_per_genre, customer.country, genre.name,
genre.genre_id
		FROM invoice_line
		JOIN invoice ON invoice.invoice_id = invoice_line.invoice_id
		JOIN customer ON customer.customer_id = invoice.customer_id
		JOIN track ON track.track_id = invoice_line.track_id
		JOIN genre ON genre.genre_id = track.genre_id
		GROUP BY 2,3,4
		ORDER BY 2
	),
	max_genre_per_country AS (SELECT MAX(purchases_per_genre) AS max_genre_number,
country
		FROM sales_per_country
		GROUP BY 2
		ORDER BY 2)

SELECT sales_per_country.*
FROM sales_per_country
JOIN max_genre_per_country ON sales_per_country.country = max_genre_per_country.country
WHERE sales_per_country.purchases_per_genre = max_genre_per_country.max_genre_number;
```

	purchases_per_genre bigint	country character varying (50)	name character varying (120)	genre_id character varying (50)
1	17	Argentina	Alternative & Punk	4
2	34	Australia	Rock	1
3	40	Austria	Rock	1
4	26	Belgium	Rock	1
5	205	Brazil	Rock	1
6	333	Canada	Rock	1
7	61	Chile	Rock	1
8	143	Czech Republic	Rock	1
9	24	Denmark	Rock	1
10	46	Finland	Rock	1
11	211	France	Rock	1
12	194	Germany	Rock	1
13	44	Hungary	Rock	1
14	102	India	Rock	1
15	72	Ireland	Rock	1
16	35	Italy	Rock	1
17	33	Netherlands	Rock	1
18	40	Norway	Rock	1
19	40	Poland	Rock	1
20	108	Portugal	Rock	1
21	46	Spain	Rock	1
22	60	Sweden	Rock	1

Total rows: 24 of 24 Query complete 00:00:00.057

Q3: Write a query that determines the customer 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. */

/* Steps to Solve: Similar to the above question. There are two parts in question-
first find the most spent on music for each country and second filter the data for respective customers. */

/* Method 1: using CTE */

```
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
```

	customer_id integer	first_name character	last_name character	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	Gonçalves	Brazil	108.89999999999998	1
6	3	François	Tremblay	Canada	99.99	1
7	57	Luis	Rojas	Chile	97.02000000000001	1
8	5	R	Madhav	Czech Republic	144.54000000000002	1
9	9	Kara	Nielsen	Denmark	37.61999999999999	1
10	44	Terhi	Hämäläinen	Finland	79.2	1
11	42	Wyatt	Girard	France	99.99	1
12	37	Fynn	Zimmermann	Germany	94.05000000000001	1
13	45	Ladislav	Kovács	Hungary	78.21	1
14	58	Manoj	Pareek	India	111.86999999999999	1
15	46	Hugh	O'Reilly	Ireland	114.83999999999997	1
16	47	Lucas	Mancini	Italy	50.49	1
17	48	Johannes	Van der Berg	Netherlands	65.34	1
18	4	Bjørn	Hansen	Norway	72.27000000000001	1
19	49	Stanisław	Wójcik	Poland	76.22999999999999	1
20	34	João	Fernandes	Portugal	102.96000000000001	1
21	50	Enrique	Muñoz	Spain	98.01	1
22	51	Joakim	Johansson	Sweden	75.24	1

Total rows: 24 of 24 Query complete 00:00:00.060

/* Method 2: Using Recursive */

```
WITH RECURSIVE
	customter_with_country AS (
		SELECT customer.customer_id,first_name,last_name,billing_country,SUM(total) AS
total_spending
		FROM invoice
		JOIN customer ON customer.customer_id = invoice.customer_id
		GROUP BY 1,2,3,4
		ORDER BY 2,3 DESC),

	country_max_spending AS(
		SELECT billing_country,MAX(total_spending) AS max_spending
		FROM customter_with_country
		GROUP BY billing_country)

SELECT cc.billing_country, cc.total_spending, cc.first_name, cc.last_name, cc.customer_id
FROM customter_with_country cc
JOIN country_max_spending ms
ON cc.billing_country = ms.billing_country
WHERE cc.total_spending = ms.max_spending
ORDER BY 1;
```

	billing_country character varying (30)	total_spending double precision	first_name character	last_name character	customer_id integer
1	Argentina	39.6	Diego	Gutiérrez	56
2	Australia	81.18	Mark	Taylor	55
3	Austria	69.3	Astrid	Gruber	7
4	Belgium	60.38999999999999	Daan	Peeters	8
5	Brazil	108.89999999999998	Luís	Gonçalves	1
6	Canada	99.99	François	Tremblay	3
7	Chile	97.02000000000001	Luis	Rojas	57
8	Czech Republic	144.54000000000002	R	Madhav	5
9	Denmark	37.61999999999999	Kara	Nielsen	9
10	Finland	79.2	Terhi	Hämäläinen	44
11	France	99.99	Wyatt	Girard	42
12	Germany	94.05000000000001	Fynn	Zimmermann	37
13	Hungary	78.21	Ladislav	Kovács	45
14	India	111.86999999999999	Manoj	Pareek	58
15	Ireland	114.83999999999997	Hugh	O'Reilly	46
16	Italy	50.49	Lucas	Mancini	47
17	Netherlands	65.34	Johannes	Van der Berg	48
18	Norway	72.27000000000001	Bjørn	Hansen	4
19	Poland	76.22999999999999	Stanisław	Wójcik	49
20	Portugal	102.96000000000001	João	Fernandes	34
21	Spain	98.01	Enrique	Muñoz	50
22	Sweden	75.24	Joakim	Johansson	51

Total rows: 24 of 24 | Query complete 00:00:00.042