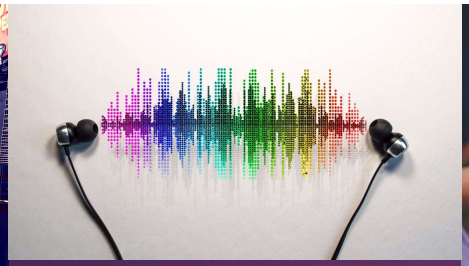
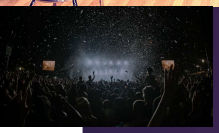
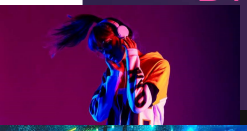







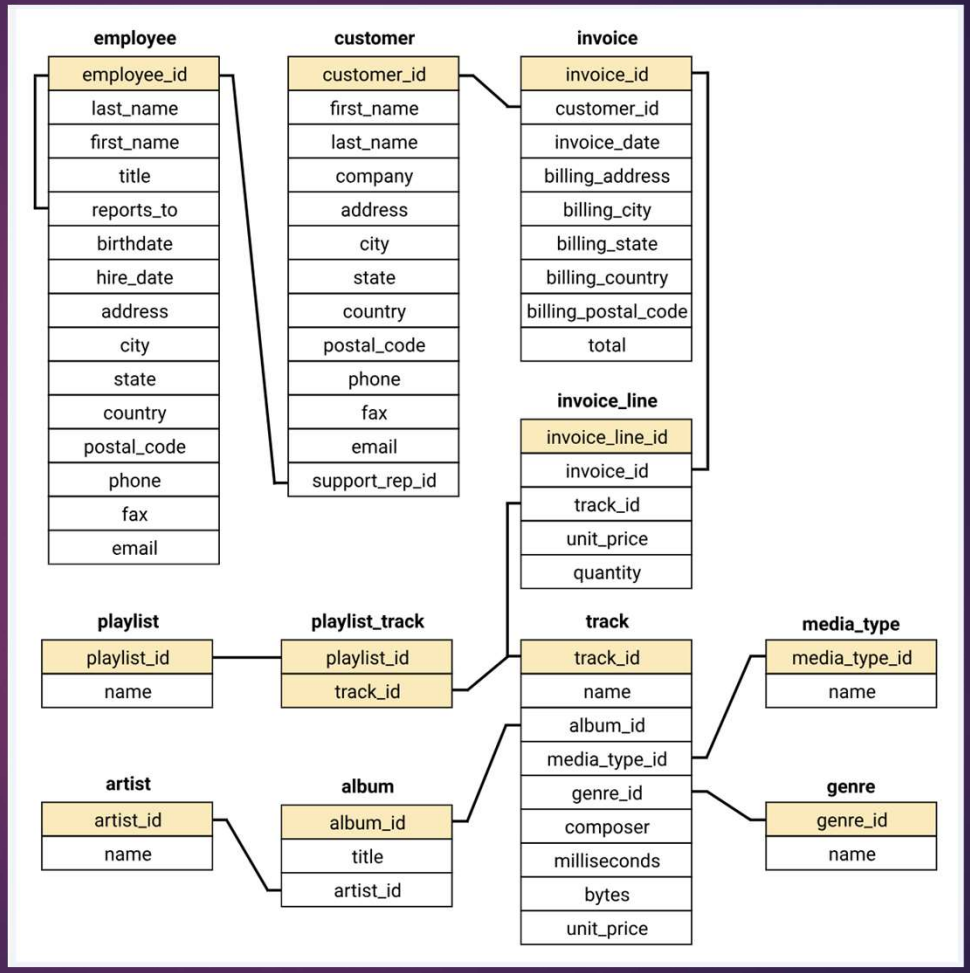
MUSIC STORE DATA ANALYSIS SQL PROJECT

-BY RITUSREE SAMANTA





SCHEMA


employee
employee_id
last_name
first_name
title
reports_to
birthdate
hire_date
address
city
state
country
postal_code
phone
fax
email
customer
customer_id
first_name
last_name
company
address
city
state
country
postal_code
phone
fax
email
support_rep_id
invoice
invoice_id
customer_id
invoice_date
billing_address
billing_city
billing_state
billing_country
billing_postal_code
total
invoice_line
invoice_line_id
invoice_id
track_id
unit_price
quantity
playlist
playlist_id
name
playlist_track
playlist_id
track_id
track
track_id
name
album_id
media_type_id
genre_id
composer
milliseconds
bytes
unit_price
media_type
media_type_id
name
genre
genre_id
name
artist
artist_id
name
album
album_id
title
artist_id

Who is the senior most employee based on job title?

```
SELECT * FROM employee
ORDER BY levels DESC
LIMIT 1;
```

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 zone
9	Madan	Mohan	Senior General Manager	[null]	L7	1961-01-26 00:00:00

hire_date timestamp without time zone	address character varying (120)	city character varying (50)	state character varying (50)	country character varying (30)	postal_code character varying (30)	phone character varying (30)	fax character va
2016-01-14 00:00:00	1008 Vrinda Ave MT	Edmonton	AB	Canada	T5K 2N1	+1 (780) 428-9482	+1 (780) 42

email character varying (30)
madan.mohan@chinookcorp.com

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
22	9	Italy
23	7	Belgium
24	5	Argentina

What are top 3 values of total invoice?

```
SELECT total FROM invoice
ORDER BY total DESC
LIMIT 3;
```

	total double precision
1	23.759999999999998
2	19.8
3	19.8

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;
```

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

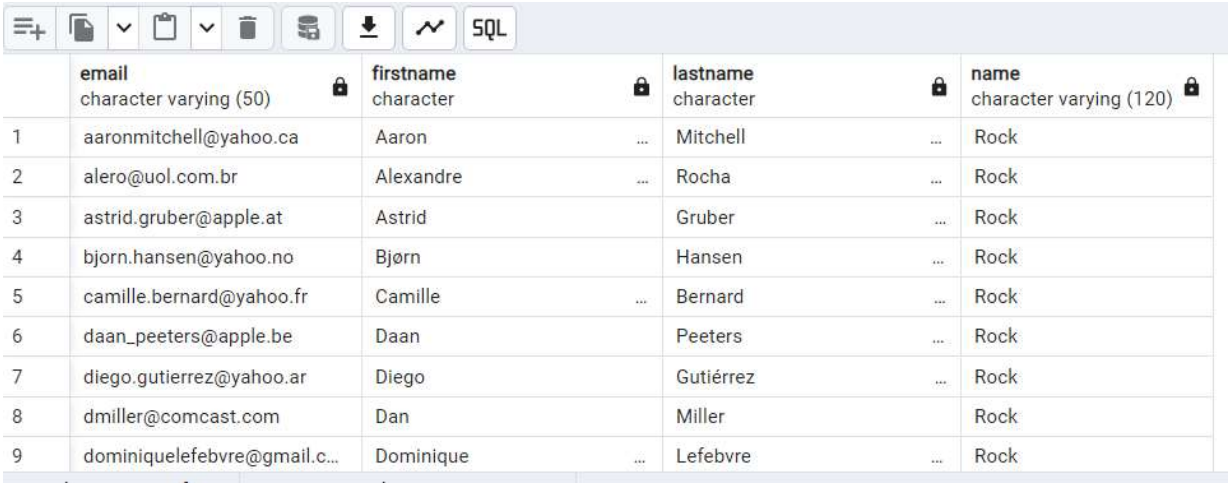
Who is the best customer? The customer who has spent the most money will be declared the best customer.
Write a query that returns the person who has spent the most money.

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

first_name character	last_name character	total double precision
R ...	Madhav	144.54000000000002

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.

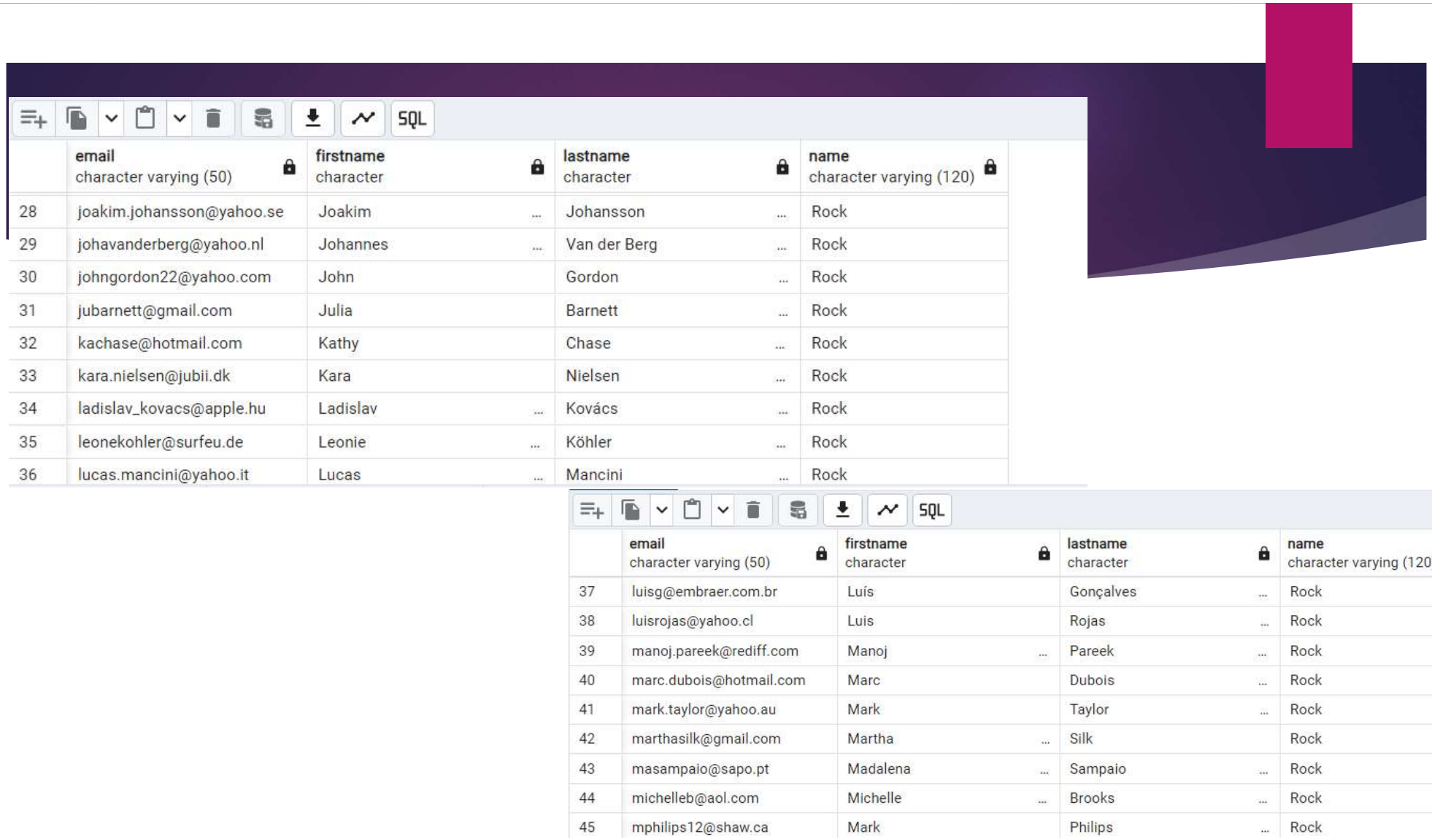
```
SELECT DISTINCT email AS Email,first_name AS FirstName, last_name AS LastName, g.name AS Name
FROM customer as c
JOIN invoice as i ON i.customer_id = c.customer_id
JOIN invoice_line AS il ON il.invoice_id = i.invoice_id
JOIN track AS t ON t.track_id = il.track_id
JOIN genre AS g ON g.genre_id = t.genre_id
WHERE g.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

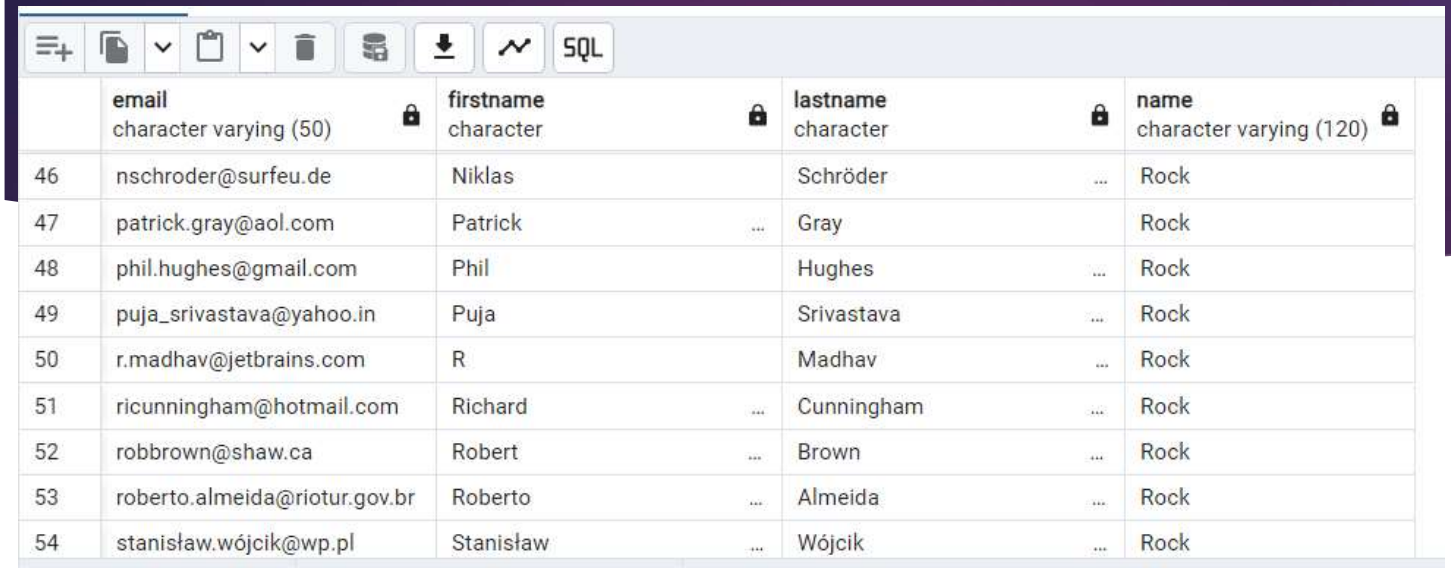
	email character varying (50)	firstname character	lastname character	name character varying (120)
10	edfrancis@yachoo.ca	Edward	Francis	Rock
11	eduardo@woodstock.com.br	Eduardo	Martins	Rock
12	ellie.sullivan@shaw.ca	Ellie	Sullivan	Rock
13	emma_jones@hotmail.com	Emma	Jones	Rock
14	enrique_munoz@yahoo.es	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	Francois	Tremblay	Rock

	email character varying (50)	firstname character	lastname character	name character varying (120)
19	fzimmermann@yahoo.de	Fynn	Zimmermann	Rock
20	hannah.schneider@yahoo.de	Hannah	Schneider	Rock
21	hholy@gmail.com	Helena	Holý	Rock
22	hleacock@gmail.com	Heather	Leacock	Rock
23	hughoreilly@apple.ie	Hugh	O'Reilly	Rock
24	isabelle_mercier@apple.fr	Isabelle	Mercier	Rock
25	jacksmith@microsoft.com	Jack	Smith	Rock
26	jenniferp@rogers.ca	Jennifer	Peterson	Rock
27	jfernandes@yahoo.pt	João	Fernandes	Rock



	email character varying (50)	firstname character	lastname character	name character varying (120)
28	joakim.johansson@yahoo.se	Joakim	Johansson	Rock
29	johavanderberg@yahoo.nl	Johannes	Van der Berg	Rock
30	johngordon22@yahoo.com	John	Gordon	Rock
31	jubarnett@gmail.com	Julia	Barnett	Rock
32	kachase@hotmail.com	Kathy	Chase	Rock
33	kara.nielsen@jubii.dk	Kara	Nielsen	Rock
34	ladislav_kovacs@apple.hu	Ladislav	Kovács	Rock
35	leonekohler@surfeu.de	Leonie	Köhler	Rock
36	lucas.mancini@yahoo.it	Lucas	Mancini	Rock

	email character varying (50)	firstname character	lastname character	name character varying (120)
37	luisg@embraer.com.br	Luís	Gonçalves	Rock
38	luisrojas@yahoo.cl	Luis	Rojas	Rock
39	manoj.pareek@rediff.com	Manoj	Pareek	Rock
40	marc.dubois@hotmail.com	Marc	Dubois	Rock
41	mark.taylor@yahoo.au	Mark	Taylor	Rock
42	marthasilk@gmail.com	Martha	Silk	Rock
43	masampaio@sapo.pt	Madalena	Sampaio	Rock
44	michelleb@aol.com	Michelle	Brooks	Rock
45	mphilips12@shaw.ca	Mark	Philips	Rock

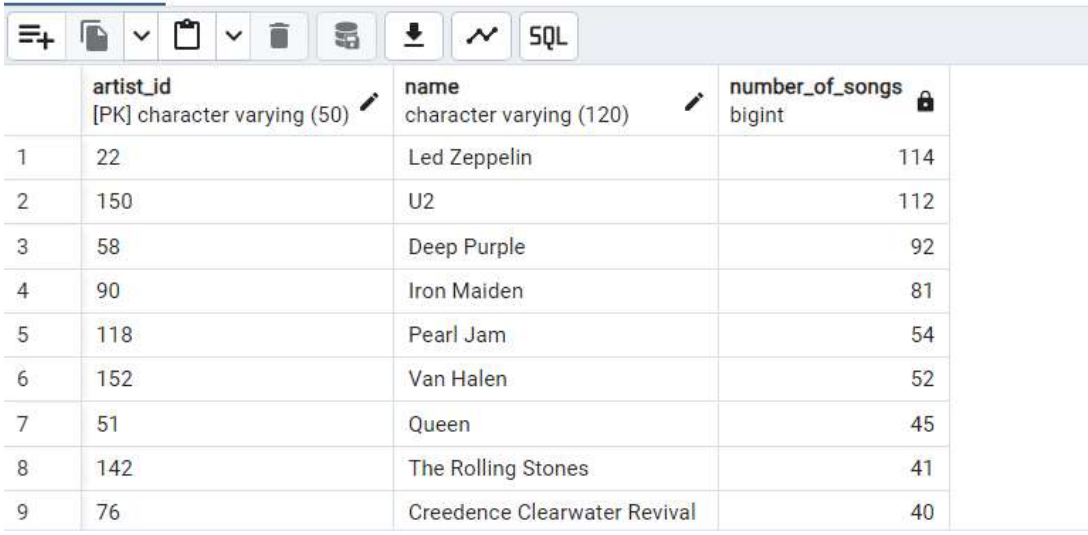


	email character varying (50)	firstname character	lastname character	name character varying (120)
46	nschroder@surfeu.de	Niklas	Schröder	Rock
47	patrick.gray@aol.com	Patrick	Gray	Rock
48	phil.hughes@gmail.com	Phil	Hughes	Rock
49	puja_srivastava@yahoo.in	Puja	Srivastava	Rock
50	r.madhav@jetbrains.com	R	Madhav	Rock
51	ricunningham@hotmail.com	Richard	Cunningham	Rock
52	robbrown@shaw.ca	Robert	Brown	Rock
53	roberto.almeida@riotur.gov.br	Roberto	Almeida	Rock
54	stanisław.wójcik@wp.pl	Stanisław	Wójcik	Rock
55	steve.murray@yahoo.uk	Steve	Murray	Rock
56	terhi.hamalainen@apple.fi	Terhi	Hämäläinen	Rock
57	tgoyer@apple.com	Tim	Goyer	Rock
58	vstevens@yahoo.com	Victor	Stevens	Rock
59	wyatt.girard@yahoo.fr	Wyatt	Girard	Rock

Total rows: 59 of 59 Query complete 00:00:00.061

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 ar.artist_id, ar.name,COUNT(ar.artist_id) AS number_of_songs
FROM track AS t
JOIN album AS a ON a.album_id = t.album_id
JOIN artist AS ar ON ar.artist_id = a.artist_id
JOIN genre AS g ON g.genre_id = t.genre_id
WHERE g.name LIKE 'Rock'
GROUP BY ar.artist_id
ORDER BY number_of_songs DESC
LIMIT 10;
```



	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

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 averae_track_length
    FROM track )
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

Total rows: 494 of 494 Query complete 00:00:00.084

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

	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
16	24	Frank	Ralston	Queen	1.98
17	30	Edward	Francis	Queen	1.98
18	35	Madalena	Sampaio	Queen	1.98
19	36	Hannah	Schneider	Queen	1.98
20	11	Alexandre	Rocha	Queen	1.98
21	8	Daan	Peeters	Queen	1.98
22	42	Wyatt	Girard	Queen	1.98

Total rows: 43 of 43 Query complete 00:00:00.069

	customer_id integer	first_name character	last_name character	artist_name character varying (120)	amount_spent double precision
22	42	Wyatt	Girard	Queen	1.98
23	44	Terhi	Hämäläinen	Queen	1.98
24	1	Luís	Gonçalves	Queen	1.98
25	48	Johannes	Van der Berg	Queen	1.98
26	49	Stanisław	Wójcik	Queen	1.98
27	52	Emma	Jones	Queen	1.98
28	57	Luis	Rojas	Queen	1.98
29	15	Jennifer	Peterson	Queen	1.98
30	28	Julia	Barnett	Queen	1.98
31	27	Patrick	Gray	Queen	0.99
32	58	Manoj	Pareek	Queen	0.99
33	45	Ladislav	Kovács	Queen	0.99
34	26	Richard	Cunningham	Queen	0.99
35	59	Puja	Srivastava	Queen	0.99
36	13	Fernanda	Ramos	Queen	0.99
37	6	Helena	Holý	Queen	0.99
38	22	Heather	Leacock	Queen	0.99
39	19	Tim	Goyer	Queen	0.99
40	39	Camille	Bernard	Queen	0.99
41	55	Mark	Taylor	Queen	0.99
42	50	Enrique	Muñoz	Queen	0.99
43	43	Isabelle	Mercier	Queen	0.99

Total rows: 43 of 43 Query complete 00:00:00.069

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.

```
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
```

	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
23	166	United Kingdom	Rock	1	1
24	561	USA	Rock	1	1

Total rows: 24 of 24 Query complete 00:00:00.070

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.

```
WITH Cust_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 Cust_with_country WHERE RowNo = 1
```

23	53	Phil	Hughes	...	United Kingdom	98.01	1
24	17	Jack	Smith	...	USA	98.01	1

Total rows: 24 of 24 | Query complete 00:00:00.135

Data Output | Messages | Notifications

	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

THANK YOU!!