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

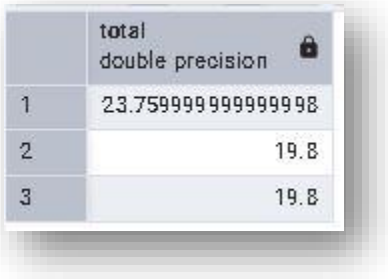
-- 2) Which countries have the most Invoices?

```
SELECT COUNT(*) total_counts, billing_country
FROM invoice
GROUP BY billing_country
ORDER BY total_counts desc;
```

	total_counts 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

-- 3) 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

-- 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

```
SELECT SUM(total) AS invoice_total, billing_city
FROM invoice
GROUP BY billing_city
ORDER BY invoice_total DESC;
```

	invoice_total 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
10	106.91999999999999	Brasília

-- 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

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

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

-- 6) 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 c
JOIN invoice i ON c.customer_id = i.customer_id
JOIN invoice_line il ON i.invoice_id = il.invoice_id
WHERE track_id IN(
    SELECT track_id FROM track t
    JOIN genre g ON t.genre_id = g.genre_id
    WHERE g.name LIKE 'Rock'
)
ORDER BY email;
```

	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
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

-- 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

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

	artist_id [PK] character varying (50)	name character varying (120)	number_of_song 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 Reviv...	40
10	52	Kiss	35

-- 8) 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 avg_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
10	Fire In Space	2926593

```
-- 9) 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 ar.artist_id AS artist_id, ar.name AS artist_name,
        SUM(il.unit_price*il.quantity) AS total_sale
        FROM invoice_line il
        JOIN track t ON t.track_id = il.track_id
        JOIN album a ON a.album_id = t.album_id
        JOIN artist ar ON a.artist_id = a.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 a ON a.album_id = t.album_id
JOIN best_selling_artist bsa ON bsa.artist_id = a.artist_id
GROUP BY 1,2,3,4
ORDER BY 5 DESC;
```

	customer_id integer	first_name character (50)	last_name character (50)	artist_nam character	amount_spent double precision
1	54	Steve ...	Murray ...	AC/DC	17.82
2	21	Kathy ...	Chase ...	AC/DC	10.89
3	53	Phil ...	Hughes ...	AC/DC	10.89
4	49	Stanisław ...	Wójcik ...	AC/DC	9.9
5	1	Luís ...	Gonçalves ...	AC/DC	7.920000000000001
6	24	Frank ...	Ralston ...	AC/DC	7.920000000000001
7	31	Martha ...	Silk ...	AC/DC	3.96
8	6	Helena ...	Holý ...	AC/DC	2.9699999999999998
9	35	Madalena ...	Sampaio ...	AC/DC	2.9699999999999998
10	38	Niklas ...	Schröder ...	AC/DC	2.9699999999999998

-- 10) 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(il.quantity) AS purchase, c.country, g.name, g.genre_id,
    ROW_NUMBER() OVER(PARTITION BY c.country ORDER BY COUNT(il.quantity) DESC) AS row_no
    FROM invoice_line il
    JOIN invoice i ON i.invoice_id = il.invoice_id
    JOIN customer c ON c.customer_id = i.customer_id
    JOIN track t ON t.track_id = il.track_id
    JOIN genre g ON g.genre_id = t.genre_id
    GROUP BY 2,3,4
    ORDER BY 2 ASC, 1 DESC
)
```

SELECT * FROM popular_genre WHERE row_no <= 1;

	purchase bigint	country character varying (50)	name character varying (120)	genre_id character varying (50)	row_no 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) 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 RECURSIVE
    customter_with_country AS (
    SELECT c.customer_id,first_name,last_name,billing_country,SUM(total) AS total_spending
    FROM invoice i
    JOIN customer c ON c.customer_id = i.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 cwc.billing_country, cwc.total_spending, cwc.first_name, cwc.last_name, cwc.customer_id
FROM customter_with_country cwc
JOIN country_max_spending cms ON cwc.billing_country = cms.billing_country
WHERE cwc.total_spending = cms.max_spending
ORDER BY 1;
```

	billing_country character varying (30)	total_spending double precision	first_name character (50)	last_name character (50)	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	37.619999999999999	Kara ...	Nielsen ...	9
10	Finland	79.2	Terhi ...	Hämäläinen ...	44

```
-- Method 2

WITH customter_with_country AS (
        SELECT c.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 row_no
        FROM invoice i
        JOIN customer c ON c.customer_id = i.customer_id
        GROUP BY 1,2,3,4
        ORDER BY 4 ASC, 5 DESC)
```

SELECT * FROM customter_with_country WHERE row_no <= 1;

	customer_id integer	first_name character (50)	last_name character (50)	billing_country character varying (30)	total_spending double precision	row_no 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