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

       SELECT \* FROM employee\*/

       ORDER BY levels DESC

       LIMIT 1;

/\* Q2. Which countries have the most invoices? \*/

SELECT billing\_country, COUNT(invoice\_id) as invoice\_count

        FROM invoice

        GROUP BY billing\_country

        ORDER BY invoice\_count DESC;

/\* Q3. What are the top 3 values of total invoice? \*/

SELECT round(total, 2)

       FROM invoice

       ORDER by total DESC

       LIMIT 3;

/\* Q4. Which city has the best customers? We would like to throw a 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 and the sum of total invoices. \*/

SELECT billing\_city as City , ROUND(SUM(total), 2) as Invoice\_total

       FROM invoice

       GROUP BY billing\_city

       ORDER BY Invoice\_total DESC

       LIMIT 1;

/\* Q5. Who is the best customer? The customer who has spent the most money will be declared as the best customer. Write a query that returns the customer that has spent the most money.\*/

       SELECT c.customer\_id, c.first\_name, c.last\_name,

       ROUND(SUM(i.total), 2) as invoice\_total

       FROM customer c

       JOIN invoice i

       ON c.customer\_id = i.customer\_id

       GROUP BY c.customer\_id

       ORDER BY invoice\_total DESC

       LIMIT 1;

/\* Q6. Write a query to return the email, first name, last name and genre of all rock music listeners. Return your list ordered alphabetically by email starting with A. \*/

 SELECT distinct first\_name, last\_name, email\_id

      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\_id ASC;

/\* 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 a.artist\_id, a.name, COUNT(a.artist\_id) as track\_count

      FROM artist a

      JOIN album2 al ON a.artist\_id = al.artist\_id

      JOIN track t ON al.album\_id = t.album\_id

      JOIN genre g ON t.genre\_id = g.genre\_id

      WHERE g.name = 'Rock'

      GROUP BY a.artist\_id

      ORDER BY track\_count DESC

      LIMIT 10;

/\* Q8. Return all the track names that have a song length longer than the average song length. Return the name and the milliseconds for each track. Order by the song length with the longest songs listed. \*/

 SELECT name, milliseconds as song\_length

     FROM track

     WHERE milliseconds > (SELECT ROUND(AVG(milliseconds))

FROM track)

     order by song\_length DESC;

/\* Q9. Find how much amount is spent by each customer on artists? Write a query to return customer name, artist name and total amount spent.\*/

       WITH best\_selling\_artist as

(SELECT a.artist\_id, a.name as artist\_name,

SUM(il.unit\_price\*il.quantity) as total\_amount\_spent

FROM invoice\_line il

JOIN track t ON t.track\_id = il.track\_id

JOIN album2 al ON al.album\_id = t.album\_id

JOIN artist a ON a.artist\_id = al.artist\_id

GROUP BY a.artist\_id

ORDER BY total\_amount\_spent DESC

LIMIT 1)

SELECT  c.customer\_id, CONCAT(c.first\_name,' ', c.last\_name) as customer\_name,

        bsa.artist\_name, SUM(il.unit\_price \* il.quantity) as total\_amount\_spent

       FROM customer c

  JOIN invoice i ON c.customer\_id = i.customer\_id

  JOIN invoice\_line il ON i.invoice\_id = il.invoice\_id

  JOIN track t ON il.track\_id = t.track\_id

  JOIN album2 al ON t.album\_id = al.album\_id

  JOIN best\_selling\_artist bsa ON al.artist\_id = bsa.artist\_id

  GROUP BY 1, 2, 3

  ORDER BY total\_amount\_spent DESC;

/\* Q10. 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 c.country as country, g.name,

g.genre\_id, COUNT(il.quantity) as purchases,

       ROW\_NUMBER() OVER(PARTITION BY c.country ORDER BY COUNT(il.quantity) DESC) as RowNo

      FROM customer c

      JOIN invoice i ON c.customer\_id = i.customer\_id

      JOIN invoice\_line il ON i.invoice\_id = il.invoice\_id

      JOIN track t ON t.track\_id = il.track\_id

      JOIN genre g ON g.genre\_id = t.genre\_id

      GROUP BY 1, 2, 3

      ORDER BY c.country asc, purchases DESC)

SELECT \* FROM popular\_genre WHERE RowNo <= 1;

/\* Q11. Write a query that determines the customer that has spent the most 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 the customers who spent this amount. \*/

        WITH RECRUSIVE customer\_with\_country as

            (SELECT

c.customer\_id, CONCAT(c.first\_name, ' ', c.last\_name) as full\_name,

i.billing\_country, ROUND(SUM(i.total), 2) as total\_spending

FROM customer c

JOIN invoice i ON c.customer\_id = i.customer\_id

    GROUP BY 1, 2, 3

ORDER BY 1, 4 DESC),

max\_country\_spending as

            (SELECT billing\_country, MAX(total\_spending) as max\_spending

FROM customer\_with\_country

GROUP BY billing\_country)

SELECT cc.customer\_id, cc.billing\_country, cc.full\_name, cc.total\_spending

FROM customer\_with\_country cc

JOIN max\_country\_spending m

ON cc.billing\_country = m.billing\_country

WHERE cc.total\_spending = m.max\_spending

ORDER BY 2;