/\* 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

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

SELECT COUNT(\*) AS c, billing\_country

FROM invoice

GROUP BY billing\_country

ORDER BY c DESC

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

SELECT total

FROM invoice

ORDER BY total DESC

/\* 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;

/\* 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;

/\* 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. \*/

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;

'Rock'

ORDER BY email;

/\* 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;

/\* 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,miliseconds

FROM track

WHERE miliseconds > (

SELECT AVG(miliseconds) AS avg\_track\_length

FROM track )

ORDER BY miliseconds DESC;