1. Who is the senior most employee based on job title

select max(levels) from employee

/\*select \* from employee

where levels = 'l7'\*/

select \* from employee

order by levels desc

Limit 1

1. Which countries have the most invoices

select \* from invoice

select count(\*) cnt, billing\_country

from invoice

group by billing\_country

order by cnt desc

limit 1

1. What are top 3 values of total invoices

Select total

From invoice

Order by total desc

Limit 3

1. 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.

select \* from invoice

select billing\_city, sum(total) as invoice\_total

from invoice

group by billing\_city

order by invoice\_total desc

1. 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 \* from customer

select \* from invoice

select c.first\_name,c.last\_name, sum(i.total) as total

from customer c, invoice i

where c.customer\_id = i.customer\_id

group by c.customer\_id

order by total desc

limit 1

======================================================================

1. Write a 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 first\_name, last\_name, email

from customer c, invoice i, invoice\_line i1, genre g, track t

where c.customer\_id = i.customer\_id

and i.invoice\_id = i1.invoice\_id

and i1.track\_id = t.track\_id

and t.genre\_id = g.genre\_id

and g.name = 'Rock'

order by email

1. 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 distinct ar.name as artist\_name, count(ar.artist\_id) as num\_of\_songs

from artist ar

join album a on a.artist\_id = ar.artist\_id

join track t on a.album\_id = t.album\_id

join genre g on g.genre\_id = t.genre\_id

where g.name like 'Rock'

group by ar.artist\_id

order by num\_of\_songs desc

limit 10;

1. 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;

ADVANCED Pending