/\* The senior most employee based on job title. \*/

select top 1 first\_name, last\_name, title, levels from employee

--where ROW\_NUMBER=1

order by levels desc

/\* Countries have the most Invoices. \*/

select top 1 billing\_country, count(\*) Total from invoice

group by billing\_country

order by Total desc

/\* Top 3 values of total invoice? \*/

select top 3 total from invoice

--group by billing\_country

order by Total desc

/\* Find the 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 top 1 billing\_city, sum(total) Invoice\_Total from invoice

group by billing\_city

order by Invoice\_Total desc

/\* Find 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 top 1 cus.first\_name, cus.last\_name,

sum(inv.total) over (partition by inv.customer\_id ) as Total\_Invoice

from customer cus join invoice as inv

on cus.customer\_id = inv.customer\_id

order by Total\_Invoice desc

/\* 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 cus.email, cus.first\_name, cus.last\_name,gen.name, count(gen.name) Total\_Genre

from customer cus join invoice inv

on cus.customer\_id = inv.customer\_id

join invoice\_line inl

on inv.invoice\_id = inl.invoice\_id

join track trk

on inl.track\_id = trk.track\_id

join genre gen

on trk.genre\_id = gen.genre\_id

where gen.name = 'Rock'

group by cus.email, cus.first\_name, cus.last\_name, gen.name

order by cus.email

/\* 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, artist.name

ORDER BY number\_of\_songs DESC

/\* 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) from track)

order by milliseconds desc