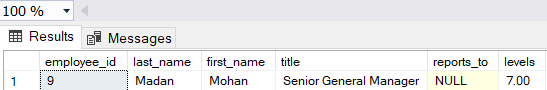
**MUSIC STORE DATA ANALYSIS**

1. **What is the senior most employee based on job title**

SELECT TOP 1 \* FROM employee ORDER BY levels DESC



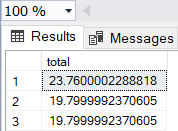
1. **Which countries have the most invoices**

SELECT COUNT(\*) AS Number\_of\_invoices, billing\_country FROM invoice GROUP BY billing\_country ORDER BY Number\_of\_invoices DESC



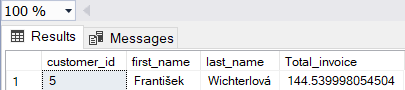
1. **What are top 3 values of total invoice**

SELECT TOP 3 total FROM invoice ORDER BY total DESC



1. **Which City has the best customer? We would like to throw 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 total**

SELECT TOP 1 billing\_city, SUM(total) AS Total\_invoice FROM invoice GROUP BY billing\_city ORDER BY Total\_invoice DESC



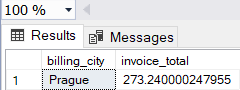
1. **Who is the best Customer? The Customer who has spend the most money will be declared the best customer, Write a query that return the person who has spent the most money**

SELECT TOP 1 customer.customer\_id, customer.first\_name, customer.last\_name, SUM(invoice.total) AS Total\_invoice FROM customer

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

GROUP BY customer.customer\_id, customer.first\_name, customer.last\_name

ORDER BY Total\_invoice DESC



1. **Write query to return the email, first name, last name & Genre of all Rock music listeners, Return your list ordered alphabetically by email sorting with A**

SELECT DISTINCT email, first\_name, last\_name FROM customer

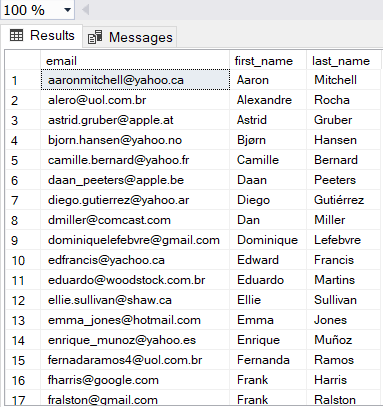
JOIN invoice on customer.customer\_id = invoice.customer\_id

JOIN invoice\_line ON invoice.invoice\_id = invoice\_line.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;



1. **Let’s invite the artist 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 TOP 10 artist.artist\_id, artist.name, COUNT(artist.artist\_id) AS total\_trackes

FROM track

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

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

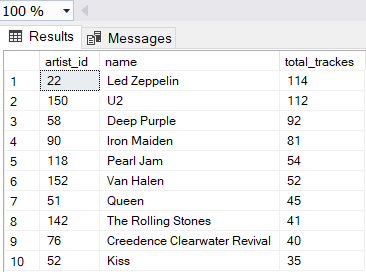
JOIN artist ON artist.artist\_id = album2.artist\_id

JOIN genre ON genre.genre\_id = track.genre\_id

WHERE genre.name LIKE 'Rock'

GROUP BY artist.artist\_id, artist.name

ORDER BY total\_trackes DESC



1. **Return all the tracks names that have a song length longer than the average song length. Return the name and milliseconds for each track. Orders 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



1. **Find how much amount spent by each customer on artist? Write a query to return customer name, artist name and total spent**

SELECT customer.first\_name, customer.last\_name, artist.name, SUM(invoice.total) AS total\_spent FROM invoice

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

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

JOIN track ON track.track\_id = invoice\_line.track\_id

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

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

JOIN artist ON artist.artist\_id = album.artist\_id

GROUP BY customer.first\_name, customer.last\_name, artist.name ORDER BY total\_spent DESC

