Question Set 1 - Easy

Q1: Who is the senior most employee based on job title?

```
SELECT *
FROM music_database.employee
ORDER BY levels desc
LIMIT 1
```

Q2: Which countries have the most Invoices?

```
SELECT count(*) as number,billing_country
FROM music_database.invoice
GROUP BY billing_country
ORDER BY number desc
```

Q3: What are the top 3 values of the total invoice?

```
SELECT total
FROM music_database.invoice
ORDER BY total desc
LIMIT 3
```

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 SUM(total)as invoice_total, billing_city
FROMfrom music_database.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 the best customer.

Write a query that returns the person who has spent the most money.*/

```
SELECT
music_database.customer. customer_id, first_name, last_name, SUM(total) AS total_spending
FROM music_database.customer
JOIN music_database.invoice ON music_database.customer.customer_id = music_database.invoice.customer_id
GROUP BY music_database.customer.customer_id,first_name,last_name
ORDER BY total_spending DESC
LIMIT 1;
```

/* Question Set 2 - Moderate */

Q1: 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 email,first_name, last_name
FROM music_database.customer
JOIN music_database.invoice ON music_database.customer.customer_id = music_database.invoice.customer_id
JOIN music_database.invoice_line ON music_database.invoice.invoice_id = music_database.invoice_line.invoice_id
WHERE track_id IN(
SELECT track_id FROM music_database.track
JOIN music_database.genre ON music_database.track.genre_id = music_database.genre.genre_id
WHERE music_database.genre.name LIKE '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 music_database.artist.artist_id,music_database.artist.name,COUNT(music_database.artist.artist_id)
as number_of_songs
FROM music_database.track
INNER JOIN music_database.album ON music_database.album.album_id = music_database.track.album_id
INNER JOIN music_database.artistON music_database.artist.artist_id = music_database.album.artist_id
INNER JOIN music_database.genre ON music_database.genre.genre_id = music_database.track.genre_id
WHERE music_database.genre.name LIKE "Rock"
GROUP BY music_database.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, milliseconds
FROM music_database.track
WHERE milliseconds > (
SELECT AVG(milliseconds) AS avg_track_length
FROM music_database.track )
ORDER BY milliseconds DESC;
```