


MUSIC STORE DATA EXPLORATION USING SQL

-- 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
```

```
1  -- Q1: Who is the senior most employee based on job title?
2
3  ✓ SELECT title, last_name, first_name
4    FROM employee
5    ORDER BY levels DESC
6    LIMIT 1
```

Data Output Messages Notifications				
				
	title character varying (50)	last_name character (50)		first_name character (50)
1	Senior General Manager	Madan	...	Mohan

-- Q2: Which countries have the most Invoices?

```
SELECT COUNT(*) AS c, billing_country
FROM invoice
GROUP BY billing_country
ORDER BY c DESC
```

-- Q2: Which countries have the most Invoices?

```
SELECT COUNT(*) AS c, billing_country
FROM invoice
GROUP BY billing_country
ORDER BY c DESC
```


Query Query History

```
24 ▾ /* Q4: Which city has the best customers?
25 We would like to throw a promotional Music Festival in the city we made the most money.
26 Write a query that returns one city that has the highest sum of invoice totals.
27 Return both the city name & sum of all invoice totals */
28
29 SELECT billing_city,SUM(total) AS InvoiceTotal
30 FROM invoice
31 GROUP BY billing_city
32 ORDER BY InvoiceTotal DESC
33 LIMIT 1;
```

Data Output Messages Notifications

	billing_city character varying (30)	invoicetotal double precision
1	Prague	273.240000000000007

/* 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;
```

```
/* 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;
```

Data Output Messages Notifications					
	customer_id [PK] integer	first_name character (50)	last_name character (50)	total_spending double precision	
1	5	R	...	Madhav	144.54000000000002

/* Q6: 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 AS Email, first_name AS FirstName, last_name AS LastName, genre.name AS
Name
FROM customer
JOIN invoice ON invoice.customer_id = customer.customer_id
JOIN invoice_line ON invoice_line.invoice_id = invoice.invoice_id
JOIN track ON track.track_id = invoice_line.track_id
JOIN genre ON genre.genre_id = track.genre_id
WHERE genre.name LIKE 'Rock'
ORDER BY email;
```

```
/* Q6: 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 AS Email, first_name AS FirstName, last_name AS LastName, genre.name
AS Name
FROM customer
JOIN invoice ON invoice.customer_id = customer.customer_id
JOIN invoice_line ON invoice_line.invoice_id = invoice.invoice_id
JOIN track ON track.track_id = invoice_line.track_id
JOIN genre ON genre.genre_id = track.genre_id
WHERE genre.name LIKE 'Rock'
ORDER BY email;
```

	email character varying (50)	firstname character (50)	lastname character (50)	name character varying (120)
1	aaronmitchell@yahoo.ca	Aaron	Mitchell	Rock
2	alero@uol.com.br	Alexandre	Rocha	Rock
3	astrid_gruber@apple.at	Astrid	Gruber	Rock
4	bjorn.hansen@yahoo.no	Bjorn	Hansen	Rock
5	camille.bernard@yahoo.fr	Camille	Bernard	Rock
6	daan_peeters@apple.be	Daan	Peeters	Rock
7	diego.gutierrez@yahoo.ar	Diego	Gutiérrez	Rock
8	dmiller@comcast.com	Dan	Miller	Rock
9	dominiquelefebvre@gmail.c...	Dominique	Lefebvre	Rock
10	edfrancis@yachoo.ca	Edward	Francis	Rock
11	eduardo@woodstock.com.br	Eduardo	Martins	Rock
12	ellie.sullivan@shaw.ca	Ellie	Sullivan	Rock
13	emma_jones@hotmail.com	Emma	Jones	Rock
14	enrique_munoz@yahoo.es	Enrique	Muñoz	Rock
15	fernadaramos4@uol.com.br	Fernanda	Ramos	Rock
Total rows: 59 of 59 Query complete 00:00:00.333 Ln 60, Col 16				

**/* Q7: 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;
```

/ Q7: 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;
```

Data Output Messages Notifications			
	artist_id [PK] character varying (50)	name character varying (120)	number_of_songs bigint
1	22	Led Zeppelin	114
2	150	U2	112
3	58	Deep Purple	92
4	90	Iron Maiden	81
5	118	Pearl Jam	54
6	152	Van Halen	52
7	51	Queen	45
8	142	The Rolling Stones	41
9	76	Creedence Clearwater Revival	40
10	52	Kiss	35

/* Q8: 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;
```

/ Q8: 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;
```



	name character varying (150)	milliseconds integer
1	Occupation / Precipice	5286953
2	Through a Looking Glass	5088838
3	Greetings from Earth, Pt. 1	2960293
4	The Man With Nine Lives	2956998
5	Battlestar Galactica, Pt. 2	2956081
6	Battlestar Galactica, Pt. 1	2952702
7	Murder On the Rising Star	2935894
8	Battlestar Galactica, Pt. 3	2927802
9	Take the Celestra	2927677
10	Fire In Space	2926593
11	The Long Patrol	2925008
12	The Magnificent Warriors	2924716
13	The Living Legend, Pt. 1	2924507
14	The Gun On Ice Planet Zero, Pt. 2	2924341
15	The Hand of God	2924007
16	Experiment In Terra	2923548
Total rows: 494 of 494 Query complete 00:00:00.796 Ln 86, Col 1		