MUSIC SALES DATABASE ANALYSIS
-SHASHWAT PANDEY

employee

employee_id
last_name
first_name
title
reports_to
birthdate
hire_date
address
city
state
country
postal_code
phone
fax
email

customer

customer_id
first_name
last_name
company
address
city
state
country
postal_code
phone
fax
email
support_rep_id

invoice

invoice_id
customer_id
invoice_date
billing_address
billing_city
billing_state
billing_country
billing_postal_code
total

invoice_line

invoice_line_id
invoice_id
track_id
unit_price
quantity

playlist

playlist_id
name

playlist_track

playlist_id
track_id

track

track_id
name
album_id
media_type_id
genre_id
composer
milliseconds
bytes
unit_price

media_type

media_type_id
name

artist

artist_id
name

album

album_id
title
artist_id

genre

genre_id
name

- ∨ ⊞ Tables (11)
  - › ⊞ album
  - › ⊞ artist
  - › ⊞ customer
  - › ⊞ employee
  - › ⊞ genre
  - › ⊞ invoice
  - › ⊞ invoice_line
  - › ⊞ media_type
  - › ⊞ playlist
  - › ⊞ playlist_track
  - › ⊞ track

Query   Query History

```sql
 1    -- Write a query that returns the Artist name and total track count
 2    -- of the top 10 rock bands
 3
 4 v  SELECT artist.artist_id, artist.name,COUNT(artist.artist_id) AS number_of_songs
 5    FROM track
 6    JOIN album ON album.album_id = track.album_id
 7    JOIN artist ON artist.artist_id = album.artist_id
 8    JOIN genre ON genre.genre_id = track.genre_id
 9    WHERE genre.name LIKE 'Rock'
10    GROUP BY artist.artist_id
11    ORDER BY number_of_songs DESC
12    LIMIT 10;
```

Scratch Pad ✕

Data Output   Messages   Notifications

Showing rows: 1 to 10   Page No: 1   of 1

	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

Total rows: 10   Query complete 00:00:00.150   CRLF   Ln 13, Col 1

```sql
-- Q2: 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;
```

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

Total rows: 1   Query complete 00:00:00.509

```
 1    -- Return all the track names that have a song length longer than the average song
 2    -- length. Return the Name and Milliseconds for each track.
 3    -- Order by the song length with the longest songs listed first.
 4
 5  ∨ SELECT name,milliseconds
 6    FROM track
 7    WHERE milliseconds > (
 8        SELECT AVG(milliseconds) AS avg_track_length
 9        FROM track )
10    ORDER BY milliseconds DESC;
11
12
```

Data Output   Messages   Notifications   ↗

≡₊ ▣ ∨ ▢ ∨ ▤ ▦ ⬇ ∿ SQL   Showing rows: 1 to 494 ✎   Page No: 1   of 1   ◁ ◀◀ ▶▶ ▷|

	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

Total rows: 494   Query complete 00:00:00.106   CRLF   Ln 10, Col 14

```sql
1    -- : Find how much amount spent by each customer on artists?
2    -- Write a query to return customer name, artist name and total spent
3
4  ⌄ WITH best_selling_artist AS (
5        SELECT artist.artist_id AS artist_id, artist.name AS artist_name, SUM(invoice_line.unit_price*invoice_line.quantity) AS total_sales
6        FROM invoice_line
7        JOIN track ON track.track_id = invoice_line.track_id
8        JOIN album ON album.album_id = track.album_id
9        JOIN artist ON artist.artist_id = album.artist_id
10       GROUP BY 1
11       ORDER BY 3 DESC
12       LIMIT 1
13   )
14   SELECT c.customer_id, c.first_name, c.last_name, bsa.artist_name, SUM(il.unit_price*il.quantity) AS amount_spent
15   FROM invoice i
16   JOIN customer c ON c.customer_id = i.customer_id
17   JOIN invoice_line il ON il.invoice_id = i.invoice_id
18   JOIN track t ON t.track_id = il.track_id
19   JOIN album alb ON alb.album_id = t.album_id
20   JOIN best_selling_artist bsa ON bsa.artist_id = alb.artist_id
21   GROUP BY 1,2,3,4
22   ORDER BY 5 DESC;
23
```

Scratch Pad ✕

Data Output   Messages   Notifications

Showing rows: 1 to 494   Page No: 1   of 1

	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   Query complete 00:00:00.106   CRLF   Ln 23, Col 1

Scratch Pad ×

```
1    -- . Write a query that returns each country along with the top Genre.
2    -- For countries where the maximum number of purchases is shared return all Genres.
3
4  ∨ WITH popular_genre AS
5    (
6        SELECT COUNT(invoice_line.quantity) AS purchases, customer.country, genre.name, genre.genre_id,
7        ROW_NUMBER() OVER(PARTITION BY customer.country ORDER BY COUNT(invoice_line.quantity) DESC) AS RowNo
8        FROM invoice_line
9        JOIN invoice ON invoice.invoice_id = invoice_line.invoice_id
10       JOIN customer ON customer.customer_id = invoice.customer_id
11       JOIN track ON track.track_id = invoice_line.track_id
12       JOIN genre ON genre.genre_id = track.genre_id
13       GROUP BY 2,3,4
14       ORDER BY 2 ASC, 1 DESC
15   )
16   SELECT * FROM popular_genre WHERE RowNo <= 1
17
```

Data Output   Messages   Notifications

Showing rows: 1 to 24   Page No: 1   of 1

	purchases bigint	country character varying (50)	name character varying (120)	genre_id character varying (50)	rowno bigint
1	17	Argentina	Alternative & Punk	4	1
2	34	Australia	Rock	1	1
3	40	Austria	Rock	1	1
4	26	Belgium	Rock	1	1
5	205	Brazil	Rock	1	1
6	333	Canada	Rock	1	1
7	61	Chile	Rock	1	1
8	143	Czech Republic	Rock	1	1
9	24	Denmark	Rock	1	1
10	46	Finland	Rock	1	1
11	211	France	Rock	1	1
12	194	Germany	Rock	1	1
13	44	Hungary	Rock	1	1
14	102	India	Rock	1	1
15	72	Ireland	Rock	1	1
16	35	Italy	Rock	1	1

Total rows: 24   Query complete 00:00:00.085   CRLF   Ln 17, Col 1

```sql
1    -- Write a query that determines the customer that has spent the most on music for each country.
2    -- Write a query that returns the country along with the top customer and how much they spent.
3    -- For countries where the top amount spent is shared, provide all customers who spent this amount
4
5  v WITH RECURSIVE
6      customter_with_country AS (
7          SELECT customer.customer_id,first_name,last_name,billing_country,SUM(total) AS total_spending
8          FROM invoice
9          JOIN customer ON customer.customer_id = invoice.customer_id
10         GROUP BY 1,2,3,4
11         ORDER BY 2,3 DESC),
12
13     country_max_spending AS(
14         SELECT billing_country,MAX(total_spending) AS max_spending
15         FROM customter_with_country
16         GROUP BY billing_country)
17
18  SELECT cc.billing_country, cc.total_spending, cc.first_name, cc.last_name, cc.customer_id
19  FROM customter_with_country cc
20  JOIN country_max_spending ms
21  ON cc.billing_country = ms.billing_country
22  WHERE cc.total_spending = ms.max_spending
23  ORDER BY 1;
```

Data Output   Messages   Notifications

Showing rows: 1 to 24   Page No: 1   of 1

	billing_country character varying (30)	total_spending double precision	first_name character (50)	last_name character (50)	customer_id integer
1	Argentina	39.6	Diego	Gutiérrez	56
2	Australia	81.18	Mark	Taylor	55
3	Austria	69.3	Astrid	Gruber	7
4	Belgium	60.38999999999999	Daan	Peeters	8
5	Brazil	108.89999999999998	Luís	Gonçalves	1
6	Canada	99.99	François	Tremblay	3
7	Chile	97.02000000000001	Luis	Rojas	57
8	Czech Republic	144.54000000000002	R	Madhav	5
9	Denmark	37.61999999999999	Kara	Nielsen	9
10	Finland	79.2	Terhi	Hämäläinen	44
11	France	99.99	Wyatt	Girard	42
12	Germany	94.05000000000001	Fynn	Zimmermann	37

Total rows: 24   Query complete 00:00:00.120   CRLF   Ln 24, Col 1