THE MUSIC STORE

SQL PROJECTS

```
/* Set 1 - Easy */

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

Select employee_id , first_name , last_name , title , levels
From employee
Order By levels DESC
Limit 1
```

OUTPUT

	employee_id [PK] character varying (50)	first_name character		last_name character		title character varying (50)	levels character varying (10)
1	9	Mohan	...	Madan	...	Senior General Manager	L7

```
/* Q2: Which countries have the most Invoices? */

Select billing_country , Count(billing_country) As no_of_invoice
From invoice
Group By billing_country
Order By Count(billing_country) DESC
Limit 1;
```

OUTPUT

	billing_country character varying (30)	no_of_invoice bigint
1	USA	131

```
/* Q3: What are top 3 values of total invoice? */

Select *
From invoice
Order By total DESC
Limit 3;
```

OUTPUT

	invoice_id [PK] integer	customer_id integer	invoice_date timestamp without time zone	billing_address character varying (120)	billing_city character varying (30)	billing_state character varying (30)	billing_country character varying (30)	billing_postal character varying
1	183	42	2018-02-09 00:00:00	9, Place Louis Barthou	Bordeaux	None	France	33 000
2	92	32	2017-07-02 00:00:00	696 Osborne Street	Winnipeg	MB	Canada	R3L 2B9
3	31	3	2017-02-21 00:00:00	1498 rue Bélanger	Montréal	QC	Canada	H2G 1A7

```
/* 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 billing_city , Sum(total) As invoice_total
From invoice
Group by billing_city
Order by Sum(total) DESC
Limit 1;
```

OUTPUT

	billing_city character varying (30) 🔒	invoice_total double precision 🔒
1	Prague	273.24000000000007

```sql
/* 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 c.customer_id , c.first_name , c.last_name , c.country , Sum(i.total) As total_invoice
From customer As c
Join invoice As i
On c.customer_id = i.customer_id
Group by c.customer_id
Order by Sum(i.total) DESC
Limit 1;
```

OUTPUT

customer_id [PK] integer	first_name character	last_name character	country character varying (50)	total_invoice double precision
5	R ...	Madhav ...	Czech Republic	144.540000000000002

```
/* Set 2 - Moderate */

/* Q1: 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 c.first_name , c.last_name , c.email
From customer As c
Join invoice As i1
On c.customer_id = i1.customer_id
Join invoice_line As i2
On i1.invoice_id = i2.invoice_id
Join track As t
On i2.track_id = t.track_id
Join genre As g
On t.genre_id = g.genre_id
Where g.name = 'Rock'
Group By c.first_name , c.last_name , c.email
Order By c.email ASC
```

	first_name character		last_name character		email character varying (50)	
1	Aaron	...	Mitchell	...	aaronmitchell@yahoo.ca	
2	Alexandre	...	Rocha	...	alero@uol.com.br	
3	Astrid	...	Gruber	...	astrid.gruber@apple.at	
4	Bjørn	...	Hansen	...	bjorn.hansen@yahoo.no	
5	Camille	...	Bernard	...	camille.bernard@yahoo.fr	
6	Daan	...	Peeters	...	daan_peeters@apple.be	
7	Diego	...	Gutiérrez	...	diego.gutierrez@yahoo.ar	
8	Dan	...	Miller	...	dmiller@comcast.com	

Total rows: 59 of 59   Query complete 00:00:00.159

```sql
/* 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 al.artist_id As artist_id, al.name As artist_name , Count(g.name) As Total_track_count
From artist As al
Join album As a2
On al.artist_id = a2.artist_id
Join track As t
On a2.album_id = t.album_id
Join genre As g
On t.genre_id = g.genre_id
Where g.name = 'Rock'
Group By al.artist_id , al.name
Order By Count(g.name) DESC
Limit 10;
```

OUTPUT

	artist_id [PK] character varying (50)	artist_name character varying (120)	total_track_count 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 Reviv...	40
10	52	Kiss	35

```
/* 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 t.name As track_name , t.milliseconds As track_milliseconds
From track As t
Where t.milliseconds > (Select Avg(milliseconds) As Average_Length From track)
Order By t.milliseconds DESC;
```

OUTPUT

	track_name character varying (150)	track_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	

Total rows: 494 of 494   Query complete 00:00:00.405

PostgreSQL 16/Music S

```sql
/* Question Set 3 - Advance */

/* Q1: Find how much amount spent by each customer on artists? Write a query to return
customer name, artist name and total spent */

SELECT CONCAT(c.first_name, c.last_name) AS customer_name, a2.name AS artist_name,
SUM(i2.unit_price * i2.quantity) AS total_spent
FROM customer AS c
JOIN invoice AS i1 ON c.customer_id = i1.customer_id
JOIN invoice_line AS i2 ON i1.invoice_id = i2.invoice_id
JOIN track AS t ON i2.track_id = t.track_id
JOIN album AS a1 ON t.album_id = a1.album_id
JOIN artist AS a2 ON a1.artist_id = a2.artist_id
WHERE a2.name =
(
        SELECT artist.name
        FROM artist
        JOIN album ON artist.artist_id = album.artist_id
        JOIN track ON album.album_id = track.album_id
        JOIN invoice_line ON track.track_id = invoice_line.track_id
        GROUP BY artist.artist_id, artist.name
        ORDER BY SUM(invoice_line.unit_price * invoice_line.quantity) DESC
        LIMIT 1
)
GROUP BY customer_name, artist_name
ORDER BY total_spent DESC;
```

OUTPUT

	customer_name text		artist_name character varying (120)	total_spent double precision
1	Hugh	O'Reilly	...	Queen	27.719999999999985
2	Niklas	Schröder	...	Queen	18.81
3	François	Tremblay	...	Queen	17.82
4	João	Fernandes	...	Queen	16.830000000000002
5	Phil	Hughes	...	Queen	11.88
6	Marc	Dubois	...	Queen	11.88

```
/* Q2: We want to find out the most popular music Genre for each country. We determine the most popular genre as the genre
with the highest amount of purchases. Write a query that returns each country along with the top Genre. For countries where
the maximum number of purchases is shared return all Genres. */

Select subquery_2.* From
(

    Select subquery_1.* ,
    Rank() Over (PARTITION BY subquery_1.country ORDER BY subquery_1.purchase_per_genre DESC) As rank
    From
        (

            Select Count(*) As purchase_per_genre , c.country , g.name , g.genre_id
            From customer As c
            Join invoice As i1 On c.customer_id = i1.customer_id
            Join invoice_line As i2 On i1.invoice_id = i2.invoice_id
            Join track As t On i2.track_id = t.track_id
            Join genre As g On t.genre_id = g.genre_id
            Group By c.country , g.name , g.genre_id
            Order By c.country ASC , Count(*) DESC
        )   As subquery_1
)   As subquery_2

Where subquery_2.rank = 1
```
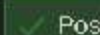
OUTPUT

	purchase_per_genre bigint	country character varying (50)	name character varying (120)	genre_id character varying (50)	rank 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

Total rows: 24 of 24   Query complete 00:00:00.447

```
/* Q3: Write a query that determines the customer that has spent the most on music for each country.
Write a query that returns the country along with the top customer and how much they spent.
For countries where the top amount spent is shared, provide all customers who spent this amount. */

Select subquery_2.*
From
(
    Select subquery_1.billing_country , subquery_1.total_spending , subquery_1.first_name ,
    subquery_1.last_name , subquery_1.customer_id,
    Rank() Over (Partition By subquery_1.billing_country Order By subquery_1.total_spending DESC)
    From
    (
        Select c.customer_id , c.first_name , c.last_name , i.billing_country , Sum(i.total) As total_spending
        From customer As c
        Join invoice As i
        On c.customer_id = i.customer_id
        Group By c.customer_id , c.first_name , c.last_name , i.billing_country
    )   As subquery_1
Group By subquery_1.billing_country , subquery_1.total_spending ,subquery_1.first_name ,
subquery_1.last_name , subquery_1.customer_id
)   As subquery_2
Where  subquery_2.rank = 1
Order By subquery_2.billing_country ASC , subquery_2.total_spending DESC
```

Activate Windows

OUTPUT

	billing_country character varying (30)	total_spending double precision	first_name character	last_name character	customer_id integer	rank bigint
1	Argentina	39.6	Diego	...	Gutiérrez	...	56	1
2	Australia	81.18	Mark	...	Taylor	...	55	1
3	Austria	69.3	Astrid	...	Gruber	...	7	1
4	Belgium	60.38999999999999	Daan	...	Peeters	...	8	1
5	Brazil	108.89999999999998	Luís	...	Gonçalves	...	1	1