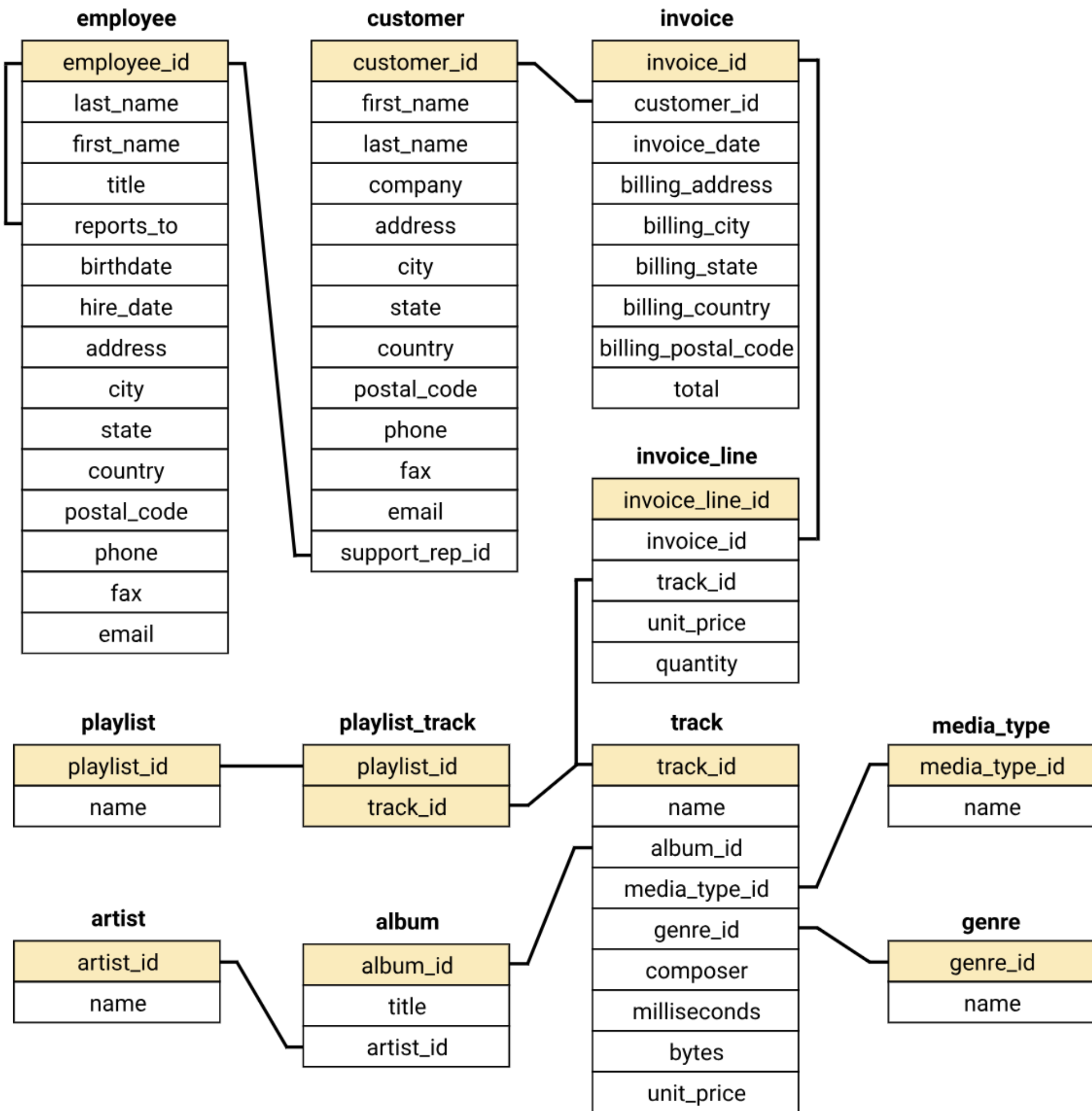


MUSIC STORE DATA ANALYSIS USING SQL

BY SHREYASH BHAGAT



SCHEMA DIAGRAM

Data Output Messages Notifications

pgAdmin 4

FileObjectToolsHelp

Object Explorer

public

Aggregates

Collations

Domains

FTS Configurations

FTS Dictionaries

FTS Parsers

FTS Templates

Foreign Tables

Functions

Materialized Views

Operators

Procedures

Sequences

Tables (11)

album

artist

customer

employee

genre

invoice

invoice_line

media_type

playlist

playlist_track

track

Trigger Functions

Types

Views

Subscriptions

postgres

Casts

Catalogs

Event Triggers

Extensions

Foreign Data Wrappers

Languages

Publications

Schemas

Subscriptions

DashboardPropertiesSQLStatisticsDependenciesDependentsProcessesMusic_Databa...SQL MUSIC DATABASE PROJECT.sql

Music_Database/postgres@PostgreSQL 16

No limit

QueryQuery History

16Q3: What are top 3 values of total invoice?

17

18SELECT TOTAL FROM INVOICE

19ORDER BY TOTAL DESC

20LIMIT 3

21

22Q4: Which city has the best customers? We would like to throw a promotional Music

23Festival in the city we made the most money. Write a query that returns one city that

24has the highest sum of invoice totals. Return both the city name & sum of all invoice

25totals

26

27SELECT SUM(TOTAL) AS INVOICE_TOTAL, BILLING_CITY

28FROM INVOICE

29GROUP BY BILLING_CITY

30ORDER BY INVOICE_TOTAL DESC

31

32Q5:Who is the best customer? The customer who has spent the most money will be

33declared the best customer. Write a query that returns the person who has spent the

34most money

35

Data OutputMessagesNotifications

	total	
	double precision	
1	23.759999999999998	
2	19.8	
3	19.8	

Total rows: 3 of 3Query complete 00:00:00.044Ln 20, Col 8

23°C
Mostly cloudy

Search

ENG
IN

04:25
10-01-2024

pgAdmin 4

FileObjectToolsHelp

Object Explorer

public

Aggregates

Collations

Domains

FTS Configurations

FTS Dictionaries

FTS Parsers

FTS Templates

Foreign Tables

Functions

Materialized Views

Operators

Procedures

Sequences

Tables (11)

album

artist

customer

employee

genre

invoice

invoice_line

media_type

playlist

playlist_track

track

Trigger Functions

Types

Views

Subscriptions

postgres

Casts

Catalogs

Event Triggers

Extensions

Foreign Data Wrappers

Languages

Publications

Schemas

Subscriptions

DashboardPropertiesSQLStatisticsDependenciesDependentsProcessesMusic_Databa...SQL MUSIC DATABASE PROJECT.sql

Music_Database/postgres@PostgreSQL 16

No limit

QueryQuery History

31

32 Q5:Who is the best customer? The customer who has spent the most money will be

33 declared the best customer. Write a query that returns the person who has spent the

34 most money

35

36 SELECT CUSTOMER.CUSTOMER_ID, CUSTOMER.FIRST_NAME, CUSTOMER.LAST_NAME, SUM(INVOICE.TOTAL) AS TOTAL

37 FROM CUSTOMER

38 JOIN INVOICE ON CUSTOMER.CUSTOMER_ID = INVOICE.CUSTOMER_ID

39 GROUP BY CUSTOMER.CUSTOMER_ID

40 ORDER BY TOTAL DESC

41 LIMIT 1

42

43

44 MODERATE QUESTIONS

45

46 Q1: Write query to return the email, first name, last name, & Genre of all Rock Music

47 listeners. Return your list ordered alphabetically by email starting with A

48

49 SELECT DISTINCT EMAIL, FIRST_NAME, LAST_NAME

50 FROM CUSTOMER

Data OutputMessagesNotifications

customer_idfirst_namelast_nametotal

[PK] integercharactercharacterdouble precision

15RMadhav144.54000000000002

Total rows: 1 of 1

Query complete 00:00:00.041

Ln 41, Col 8

23°C

Mostly cloudy

Search

ENG IN

04:26

10-01-2024

pgAdmin 4

FileObjectToolsHelp

Object Explorer

public

Aggregates

Collations

Domains

FTS Configurations

FTS Dictionaries

FTS Parsers

FTS Templates

Foreign Tables

Functions

Materialized Views

Operators

Procedures

Sequences

Tables (11)

album

artist

customer

employee

genre

invoice

invoice_line

media_type

playlist

playlist_track

track

Trigger Functions

Types

Views

Subscriptions

postgres

Casts

Catalogs

Event Triggers

Extensions

Foreign Data Wrappers

Languages

Publications

Schemas

Subscriptions

DashboardPropertiesSQLStatisticsDependenciesDependentsProcessesMusic_Databa...SQL MUSIC DATABASE PROJECT.sql

Music_Database/postgres@PostgreSQL 16

No limit

QueryQuery History

41LIMIT 1

42

43

44MODERATE QUESTIONS

45

46Q1: Write query to return the email, first name, last name, & Genre of all Rock Music

47listeners. Return your list ordered alphabetically by email starting with A

48

49SELECT DISTINCT EMAIL, FIRST_NAME, LAST_NAME

50FROM CUSTOMER

51JOIN INVOICE ON CUSTOMER.CUSTOMER_ID = INVOICE.CUSTOMER_ID

52JOIN INVOICE_LINE ON INVOICE.INVOICE_ID = INVOICE_LINE.INVOICE_ID

53WHERE TRACK_ID IN(

54SELECT TRACK_ID FROM TRACK

55JOIN GENRE ON TRACK.GENRE_ID = GENRE.GENRE_ID

56WHERE GENRE.NAME LIKE 'Rock'

57)

58ORDER BY EMAIL;

59

60Q2: Lets invite the artists who have written the most rock music in our dataset. Write a

Data OutputMessagesNotifications

email

first_name

last_name

character varying (50)

character

character

1

aaronmitchell@yahoo.ca

Aaron

Mitchell

2

alero@uol.com.br

Alexandre

Rocha

3

astrid.gruber@apple.at

Astrid

Gruber

4

bjorn.hansen@yahoo.no

Bjørn

Hansen

5

camille.bernard@yahoo.fr

Camille

Bernard

6

daan_peeters@apple.be

Daan

Peeters

7

diego.gutierrez@yahoo.ar

Diego

Gutiérrez

8

dmiller@comcast.com

Dan

Miller

9

dominiquelefebvre@gmail.c...

Dominique

Lefebvre

10

edfrancis@yahoo.ca

Edward

Francis

11

eduardo@woodstock.com.br

Eduardo

Martins

12

ellie.sullivan@shaw.ca

Ellie

Sullivan

13

emma.jones@hotmail.com

Emma

Jones

Total rows: 59 of 59

Query complete 00:00:00.043

Ln 49, Col 1

23°C

Mostly cloudy

Search

ENG IN

04:26

10-01-2024

pgAdmin 4

FileObjectToolsHelp

Object Explorer

public

Aggregates

Collations

Domains

FTS Configurations

FTS Dictionaries

FTS Parsers

FTS Templates

Foreign Tables

Functions

Materialized Views

Operators

Procedures

Sequences

Tables (11)

album

artist

customer

employee

genre

invoice

invoice_line

media_type

playlist

playlist_track

track

Trigger Functions

Types

Views

Subscriptions

postgres

Casts

Catalogs

Event Triggers

Extensions

Foreign Data Wrappers

Languages

Publications

Schemas

Subscriptions

DashboardPropertiesSQLStatisticsDependenciesDependentsProcessesMusic_Databa...SQL MUSIC DATABASE PROJECT.sql

Music_Database/postgres@PostgreSQL 16

No limit

Query

Query History

69

GROUP BY ARTIST.ARTIST_ID

70

ORDER BY NUMBER_OF_SONGS DESC

71

LIMIT 10;

72

73

Q3: Return all the track names that have a song length longer than the average song length.

74

Return the Name and Milliseconds for each track. Order by the song length with the

75

longest songs listed first

76

77

SELECT NAME, MILLISECONDS

78

FROM TRACK

79

WHERE MILLISECONDS > (

80

SELECT AVG(MILLISECONDS) AS AVG_TRACK_LENGTH

81

FROM TRACK)

82

ORDER BY MILLISECONDS DESC;

83

84

85

ADVANCED QUESTIONS

86

87

Q1: Find how much amount spent by each customer on artists? Write a query to return

88

customer name, artist name and total spent

89

Data Output

Messages

Notifications

	name	milliseconds
	character varying (150)	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

Total rows: 494 of 494

Query complete 00:00:00.076

Ln 82, Col 29

23°C

Mostly cloudy

Search

ENG IN

04:27

10-01-2024

pgAdmin 4

FileObjectToolsHelp

Object Explorer

public

Aggregates

Collations

Domains

FTS Configurations

FTS Dictionaries

FTS Parsers

FTS Templates

Foreign Tables

Functions

Materialized Views

Operators

Procedures

Sequences

Tables (11)

album

artist

customer

employee

genre

invoice

invoice_line

media_type

playlist

playlist_track

track

Trigger Functions

Types

Views

Subscriptions

postgres

Casts

Catalogs

Event Triggers

Extensions

Foreign Data Wrappers

Languages

Publications

Schemas

Subscriptions

DashboardPropertiesSQLStatisticsDependenciesDependentsProcessesMusic_Databa...SQL MUSIC DATABASE PROJECT.sql

Music_Database/postgres@PostgreSQL 16

No limit

QueryQuery History

85

ADVANCED QUESTIONS

86

87

Q1: Find how much amount spent by each customer on artists? Write a query to return

88

customer name, artist name and total spent

89

90

WITH best_selling_artist AS (

91

SELECT artist.artist_id AS artist_id, artist.name AS artist_name,

92

SUM(invoice_line.unit_price*invoice_line.quantity) AS total_sales

93

FROM invoice_line

94

JOIN track ON track.track_id = invoice_line.track_id

95

JOIN album ON album.album_id = track.album_id

96

JOIN artist ON artist.artist_id = album.artist_id

97

GROUP BY 1

98

ORDER BY 3 DESC

99

LIMIT 1

100

)

101

SELECT c.customer_id, c.first_name, c.last_name, bsa.artist_name, SUM(il.unit_price*il.quantity) AS amount_spent

102

FROM invoice i

103

JOIN customer c ON c.customer_id = i.customer_id

104

JOIN invoice_line il ON il.invoice_id = i.invoice_id

105

JOIN track t ON t.track_id = il.track_id

106

JOIN album alb ON alb.album_id = t.album_id

107

JOIN best_selling_artist bsa ON bsa.artist_id = alb.artist_id

108

GROUP BY 1,2,3,4

109

ORDER BY 5 DESC;

Data OutputMessagesNotifications

	customer_id integer	first_name character	last_name character	artist_name character varying (120)	amount_spent double precision
1	46	Hugh	O'Reilly	Queen	27.719999999999985
2	38	Niklas	Schröder	Queen	18.81
3	3	François	Tremblay	Queen	17.82
4	34	João	Fernandes	Queen	16.830000000000002
5	53	Phil	Hughes	Queen	11.88
6	41	Marc	Dubois	Queen	11.88
7	47	Lucas	Mancini	Queen	10.89
8	33	Ellie	Sullivan	Queen	10.89
9	20	Dan	Miller	Queen	3.96

Total rows: 43 of 43Query complete 00:00:00.037

Ln 89, Col 1

23°C
Mostly cloudy

Search

ENG
IN

04:27
10-01-2024


```

110
111 Q2: We want to find out the most popular music Genre for each country. We determine the
112 most popular genre as the genre with the highest amount of purchases. Write a query
113 that returns each country along with the top Genre. For countries where the maximum
114 number of purchases is shared return all Genres
115
116 WITH popular_genre AS
117 (
118     SELECT COUNT(invoice_line.quantity) AS purchases, customer.country, genre.name, genre.genre_id,
119     ROW_NUMBER() OVER(PARTITION BY customer.country ORDER BY COUNT(invoice_line.quantity) DESC) AS RowNo
120     FROM invoice_line
121     JOIN invoice ON invoice.invoice_id = invoice_line.invoice_id
122     JOIN customer ON customer.customer_id = invoice.customer_id
123     JOIN track ON track.track_id = invoice_line.track_id
124     JOIN genre ON genre.genre_id = track.genre_id
125     GROUP BY 2,3,4
126     ORDER BY 2 ASC, 1 DESC
127 )
128 SELECT * FROM popular_genre WHERE RowNo <= 1
129 --
130 Q3: Write a query that determines the customer that has spent the most on music for each
131 country. Write a query that returns the country along with the top customer and how
132 much they spent. For countries where the top amount spent is shared, provide all
133 customers who spent this amount
134

```

	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	

Total rows: 24 of 24 Query complete 00:00:00.047

Ln 115, Col 1



THANK YOU

