

Pergunta 1: Quais países possuem mais faturas?

Use a tabela **Invoice** (Fatura) para determinar quais países possuem mais faturas. Forneça as tabelas de **BillingCountry** (país de cobrança) e **Invoices** (faturas) ordenadas pelo número de faturas para cada país. O país com mais faturas deve aparecer primeiro.

Solução

```
SELECT BillingCountry, COUNT(BillingCountry)
FROM Invoice
GROUP BY 1
ORDER BY 2 DESC;
```

BillingCountry	COUNT(BillingCountry)
USA	91
Canada	56
Brazil	35
France	35
Germany	28
United Kingdom	21
Czech Republic	14
Portugal	14
India	13
Argentina	7
Australia	7
Austria	7
Belgium	7
Chile	7
Denmark	7
Finland	7
Hungary	7
Ireland	7
Italy	7
Netherlands	7
Norway	7
Poland	7
Spain	7
Sweden	7

Pergunta 2: Qual cidade tem os melhores clientes?

Gostaríamos de lançar um festival de música promocional na cidade que nos gerou mais dinheiro. Escreva uma consulta que retorna a cidade que possui a maior soma dos totais de fatura. Retorne tanto o nome da cidade quanto a soma de todos os totais de fatura.

Solução

```
SELECT BillingCity, SUM(Total)
FROM Invoice
GROUP BY 1
ORDER BY 2 DESC
LIMIT 1;
```

A cidade com mais dólares em fatura foi Praga com uma quantia de 90.24.

BillingCity	SUM(Total)
Prague	90.24

Pergunta 3: Quem é o melhor cliente?

O cliente que gastou mais dinheiro será declarado o melhor cliente. Crie uma consulta que retorna a pessoa que mais gastou dinheiro. Eu encontrei essa informação ao linkar três tabelas: **Invoice** (fatura), **InvoiceLine** (linha de faturamento), e **Customer** (cliente). Você provavelmente consegue achar a solução com menos tabelas!

Solução

```
SELECT CustomerId, SUM(Total)
FROM Invoice i
GROUP BY 1
ORDER BY 2 DESC
LIMIT 1;
```

O cliente que mais gastou de acordo com as faturas foi o Cliente 6 com 49.62 em compras.

CustomerId	SUM(Total)
6	49.62

Pergunta 4: Use sua consulta para retornar o e-mail, nome, sobrenome e gênero de todos os ouvintes de Rock. Retorne sua lista ordenada alfabeticamente por endereço de e-mail, começando por “A”. Você consegue encontrar um jeito de lidar com e-mails duplicados para que ninguém receba vários e-mails?

Solução

```
SELECT DISTINCT c.Email, c.FirstName, c.LastName, g.Name
FROM Genre g
JOIN Track t
ON g.GenreId = t.GenreId
JOIN InvoiceLine il
ON t.TrackId = il.TrackId
JOIN Invoice i
ON i.InvoiceId = il.InvoiceId
JOIN Customer c
ON i.CustomerId = c.CustomerId
WHERE g.Name = 'Rock'
ORDER BY 1;
```

Todos os clientes possuem uma conexão ao Rock (isso pode ser verificado olhando para o comprimento original da tabela de clientes). A tabela final tem 59 linhas e 4 colunas.

	Email	FirstName	LastName	Name
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	Bjørn	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	dominiquedefebvre@gmail.com	Dominique	Lefebvre	Rock
10	edfrancis@yahoo.ca	Edward	Francis	Rock
11	eduardo@woodstock.com.br	Eduardo	Martins	Rock
...				
59	wyatt.girard@yahoo.fr	Wyatt	Girard	Rock

Pergunta 5: Quem está escrevendo as músicas de rock?

Agora que sabemos que nossos clientes amam rock, podemos decidir quais músicos convidar para tocar no show.

Vamos convidar os artistas que mais escreveram as músicas de rock em nosso banco de dados. Escreva uma consulta que retorna o nome do **Artist** (artista) e a contagem total de músicas das dez melhores bandas de rock.

Você precisará usar as tabelas **Genre** (gênero), **Track** (música) , **Album** (álbum), and **Artist** (artista).

Solução

```
SELECT ar.ArtistId, ar.Name, COUNT(g.Name) Songs
FROM (
    SELECT GenreId, Name
    FROM Genre
    WHERE Name = 'Rock') AS g
JOIN Track t
ON g.GenreId = t.GenreId
JOIN Album al
ON al.AlbumId = t.AlbumId
JOIN Artist ar
ON ar.ArtistId = al.ArtistId
GROUP BY 2
ORDER BY 3 DESC
LIMIT 10;
```

As dez melhores bandas são mostradas abaixo junto do número de canções que cada uma possui em registro.

ArtistId	Name	Songs
22	Led Zeppelin	114
150	U2	112
58	Deep Purple	92
90	Iron Maiden	81
118	Pearl Jam	54
152	Van Halen	52
51	Queen	45
142	The Rolling Stones	41
76	Creedence Clearwater Revival	40
52	Kiss	35

Pergunta 6: Primeiro, descubra qual artista ganhou mais de acordo com **InvoiceLines** (linhas de faturamento).

Agora encontre qual cliente gastou mais com o artista que você encontrou acima.

Para essa consulta, você precisará usar as tabelas **Invoice** (fatura), **InvoiceLine** (linha de faturamento), **Track** (música), **Customer** (cliente), **Album** (álbum) e **Artist** (artista).

Observe que essa consulta é complicada porque a quantia **Total** gasta na tabela **Invoice** (fatura) pode não ser em um só produto, então você precisa usar a tabela **InvoiceLine** (linha de faturamento) para descobrir quanto de cada produto foi comprado e, então, multiplicar isso pelo preço de cada artista.

Solução

```
WITH t1 AS (  
    SELECT il.TrackId TrackId, SUM(il.UnitPrice * il.Quantity) Price  
    FROM InvoiceLine il  
    GROUP BY 1),  
  
    t2 AS (  
    SELECT ar.ArtistId, ar.Name, t.TrackId TrackId  
    FROM Track t  
    JOIN Album al  
    ON al.AlbumId = t.AlbumId  
    JOIN Artist ar  
    ON ar.ArtistId = al.ArtistId)  
  
SELECT t2.ArtistId, t2.Name , SUM(t1.Price) AmountSpent  
FROM t1  
JOIN t2  
ON t1.TrackId = t2.TrackId  
GROUP BY 1  
ORDER BY 3 DESC  
LIMIT 1;
```

Os artistas com maior faturamento são mostrados na tabela abaixo. Em primeiro lugar, temos Iron Maiden.

ArtistId	Name	AmountSpent
90	Iron Maiden	138.6

Continuação da solução com o maior comprador

Então, os maiores compradores são mostrados na tabela abaixo. O cliente com a maior quantia total em fatura é o cliente 55, Mark Taylor.

```
WITH t1 AS (  
    SELECT il.TrackId TrackId, SUM(il.UnitPrice * il.Quantity) Price  
    FROM InvoiceLine il  
    GROUP BY 1),  
  
    t2 AS (  
    SELECT ar.ArtistId, ar.Name, t.TrackId TrackId  
    FROM Track t  
    JOIN Album al  
    ON al.AlbumId = t.AlbumId  
    JOIN Artist ar  
    ON ar.ArtistId = al.ArtistId),  
  
    t3 AS (  
    SELECT t2.ArtistId, t2.Name , SUM(t1.Price) AmountSpent  
    FROM t1  
    JOIN t2  
    ON t1.TrackId = t2.TrackId  
    GROUP BY 1  
    ORDER BY 3 DESC  
    LIMIT 1)  
  
SELECT t2.Name,  
    SUM(il.UnitPrice * il.Quantity) AmountSpent,  
    c.CustomerId,  
    c.FirstName,  
    c.LastName  
FROM Customer c  
JOIN Invoice i  
ON c.CustomerId = i.CustomerId  
JOIN InvoiceLine il  
ON i.InvoiceId = il.InvoiceId  
JOIN t2  
ON t2.TrackId = il.TrackId  
JOIN t3  
ON t3.ArtistId = t2.ArtistId  
GROUP BY 3  
ORDER BY 2 DESC;
```


Name	AmountSpent	CustomerId	FirstName	LastName
Iron Maiden	17.82	55	Mark	Taylor

Pergunta 7: Queremos descobrir o gênero musical mais popular em cada país. Determinamos o gênero mais popular como o gênero com o maior número de compras. Escreva uma consulta que retorna cada país juntamente a seu gênero mais vendido. Para países onde o número máximo de compras é compartilhado retorne todos os gêneros.

Para essa consulta você precisará usar as tabelas **Invoice** (fatura), **InvoiceLine** (linha de faturamento), **Track** (música), **Customer** (cliente) e **Genre** (gênero).

Solução

```
WITH t1 AS (  
    SELECT c.Country, t.GenreId, g.Name, COUNT(*) Purchases  
    FROM Customer c  
    JOIN Invoice i  
    ON c.CustomerId = i.CustomerId  
    JOIN InvoiceLine il  
    ON i.InvoiceId = il.InvoiceId  
    JOIN Track t  
    ON t.TrackId = il.TrackId  
    JOIN Genre g  
    ON g.GenreId = t.GenreId  
    GROUP BY 1, 2),  
  
    t2 AS (  
    SELECT t1.Country, t1.Purchases, MAX(t1.Purchases)  
    FROM t1  
    GROUP BY 1)  
  
SELECT t2.Purchases, t2.Country, t1.Name, t1.GenreId  
FROM t1  
JOIN t2  
ON t1.Country = t2.Country AND t1.Purchases = t2.Purchases
```

Embora existam apenas 24 países, a consulta retorna 25 linhas, pois a Argentina possui 2 gêneros que compartilham o máximo.

Purchases	Country	Name	GenreId
9	Argentina	Rock	1
9	Argentina	Alternative & Punk	4
22	Australia	Rock	1
15	Austria	Rock	1
21	Belgium	Rock	1
81	Brazil	Rock	1
107	Canada	Rock	1
9	Chile	Rock	1
25	Czech Republic	Rock	1
21	Denmark	Rock	1
18	Finland	Rock	1
65	France	Rock	1
62	Germany	Rock	1
11	Hungary	Rock	1
25	India	Rock	1
12	Ireland	Rock	1
18	Italy	Rock	1
18	Netherlands	Rock	1
17	Norway	Rock	1
22	Poland	Rock	1
31	Portugal	Rock	1
22	Spain	Rock	1
12	Sweden	Latin	7
157	USA	Rock	1
37	United Kingdom	Rock	1

Pergunta 8: Retorne todos os nomes de músicas que possuem um comprimento de canção maior que o comprimento médio de canção. Embora você possa fazer isso com duas consultas. Imagine que você queira que sua consulta atualize com base em onde os dados são colocados no banco de dados. Portanto, você não quer fazer um hard code da média na sua consulta. Você só precisa da tabela **Track** (música) para completar essa consulta.

Retorne o **Name** (nome) e os **Milliseconds** (milissegundos) para cada música. Ordene pelo comprimento da canção com as músicas mais longas sendo listadas primeiro.

Solução

```
SELECT Name, Milliseconds
FROM Track t
WHERE Milliseconds > (SELECT AVG(Milliseconds) FROM Track)
ORDER BY 2 DESC
```

Abaixo, a tabela com 494 das 3503 músicas em sua tabela.

	Name	Milliseconds
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
...		
494	Wicked Ways	393691

Pergunta 9: Escreva uma consulta que determina qual cliente gastou mais em músicas por país. Escreva uma consulta que retorna o país junto ao principal cliente e quanto ele gastou. Para países que compartilham a quantia total gasta, forneça todos os clientes que gastaram essa quantia.

Você só precisará usar as tabelas **Customer** (cliente) e **Invoice** (fatura).

Solução

```
WITH t1 AS (  
    SELECT BillingCountry Country, CustomerId, SUM(Total) TotalSpent  
    FROM Invoice  
    GROUP BY 1, 2  
    ORDER BY 1),  
  
    t2 AS (  
    SELECT Country, MAX(t1.TotalSpent) TotalSpent  
    FROM t1  
    GROUP BY 1)  
  
SELECT t1.Country, t1.TotalSpent, c.FirstName, c.LastName, c.CustomerId  
FROM Customer c  
JOIN t1  
ON c.CustomerId = t1.CustomerId  
JOIN t2  
ON t1.Country = t2.Country AND t1.TotalSpent = t2.TotalSpent  
ORDER BY 1;
```

Embora existam apenas 24 países, a consulta deve retornar 25 linhas. Observe que o Reino Unido tem 2 clientes que compartilham o máximo.

	Country	TotalSpent	FirstName	LastName	CustomerId
1	Argentina	37.62	Diego	Gutiérrez	56
2	Australia	37.62	Mark	Taylor	55
3	Austria	42.62	Astrid	Gruber	7
4	Belgium	37.62	Daan	Peeters	8
5	Brazil	39.62	Luís	Gonçalves	1
6	Canada	39.62	François	Tremblay	3
7	Chile	46.62	Luis	Rojas	57
8	Czech Republic	49.62	Helena	Holý	6
9	Denmark	37.62	Kara	Nielsen	9
10	Finland	41.62	Terhi	Hämäläinen	44
11	France	40.62	Isabelle	Mercier	43
12	Germany	43.62	Fynn	Zimmermann	37
13	Hungary	45.62	Ladislav	Kovács	45
14	India	38.62	Manoj	Pareek	58
15	Ireland	45.62	Hugh	O'Reilly	46
16	Italy	37.62	Lucas	Mancini	47
17	Netherlands	40.62	Johannes	Van der Berg	48
18	Norway	39.62	Bjørn	Hansen	4
19	Poland	37.62	Stanisław	Wójcik	49
20	Portugal	39.62	João	Fernandes	34
21	Spain	37.62	Enrique	Muñoz	50
22	Sweden	38.62	Joakim	Johansson	51
23	USA	47.62	Richard	Cunningham	26
24	United Kingdom	37.62	Emma	Jones	52
25	United Kingdom	37.62	Phil	Hughes	53