# **Chinook Database project**

## **Исследуйте данные музыкального магазина с помощью SQL!**

### **Обзор проекта**

В этом проекте вы будете выполнять запросы для базы данных Chinook. Chinook содержит информацию о музыкальном магазине.

Ваши задачи:

* использовать основные функционалы SQL
* отвечать на вопросы с помощью правильных запросов
* найти интересные инсайты в данных используя JOIN-ы и подзапросы
* переносить данные с одной платформы в другую
* получить удовольствие от процесса

### **Ваша роль**

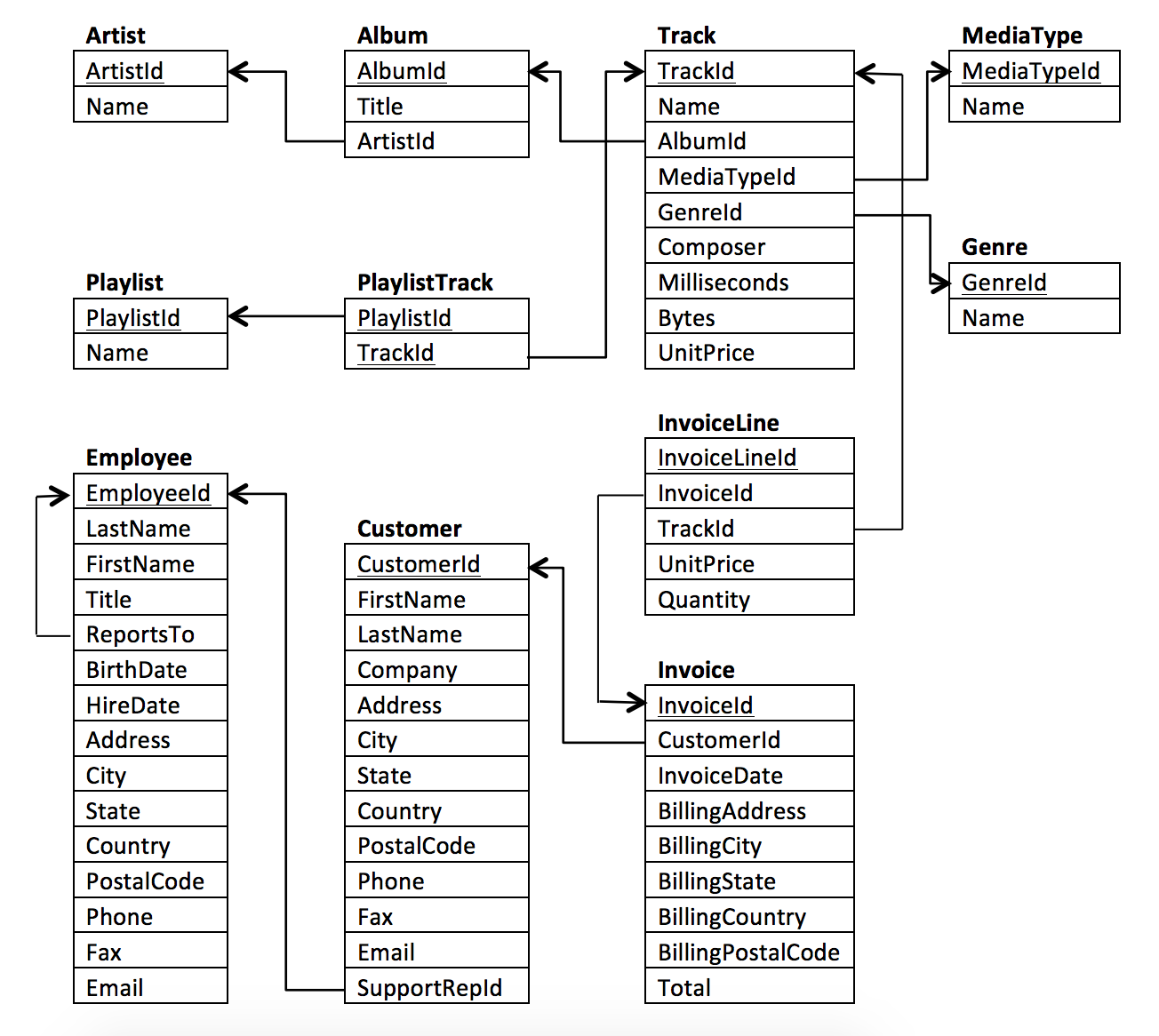
В рамках этого проекта вы будете помогать команде Chinook разобраться в товарах в их магазине, их покупателях и сотрудниках, а также в информации об их счетах.

### **База данных**

Чтобы помочь вам с дальнейшими запросами, ниже представлена схема базы данных Chinook. Вы можете увидеть столбцы, которые связывают таблицы вместе с помощью стрелок. (Hint: База данных Chinook уже есть в BigQuery, ничего создавать не нужно).

Для работы с базой данных, перенесите все данные с Google Sheets в BigQuery.

Изучите таблицы и данные в таблицах перед тем как начать работать над проектом.



## **Инструкции по сдаче проекта**

* После выполнения задания нажмите на кнопку "+ Добавить или создать" в Google Classroom и сохранить файл Google sheets + Google docs
* После загрузки файлов нажмите на кнопку "Сдать".
* Ожидайте обратную связь от менторов.

**Часть 1**

#### **Вопрос 1: В каких странах больше всего счетов (Invoices)?**

Используйте таблицу **Invoice**, чтобы определить страны с наибольшим количеством счетов. Предоставьте таблицу со столбцами **BillingCountry** и **Invoices**, упорядоченную по количеству счетов для каждой страны. Страна с наибольшим количеством счетов должна отображаться первой.

| 1 | USA | 91 |  |
| --- | --- | --- | --- |
| 2 | Canada | 56 |  |
| 3 | Brazil | 35 |  |
| 4 | France | 35 |  |
| 5 | Germany | 28 |  |
| 6 | United Kingdom | 21 |  |
| 7 | Portugal | 14 |  |
| 8 | Czech Republic | 14 |  |
| 9 | India | 13 |  |
| 10 | Chile | 7 |  |
| 11 | Italy | 7 |  |
| 12 | Spain | 7 |  |
| 13 | Norway | 7 |  |

Ваш запрос:

select

i.billingcountry,

count(i.InvoiceId) as invoices

from da-nfactorial.chinook.invoice i

group by i.billingcountry

order by invoices desc

#### **Вопрос 2: В каком городе самые лучшие клиенты?**

Мы хотели бы устроить рекламный музыкальный фестиваль в городе, в котором мы заработали больше всего денег. Напишите запрос, который возвращает город с наибольшей суммой итоговых счетов (invoice totals). Предоставьте название города и общую сумму итоговых счетов.

| 1 | Prague | 90.24 |
| --- | --- | --- |

Ваш запрос:

select

i.billingcity,

sum(i.total) as InvoiceTotals

from da-nfactorial.chinook.invoice i

group by i.billingcity

order by InvoiceTotals desc

limit 1

#### **Вопрос 3: Кто является лучшим клиентом?**

Клиент, потративший больше всего денег, будет объявлен лучшим клиентом. Напишите запрос, который возвращает человека, который потратил больше всего денег. Мы нашли решение, связав следующие три таблицы: **Invoice, InvoiceLine** и **Customer**, чтобы получить эту информацию, но вы, вероятно, можете сделать это с меньшим количеством таблиц.

| Puja | Srivastava | puja\_srivastava@yahoo.in | 36.64 |
| --- | --- | --- | --- |

Ваш запрос:

select

c.firstname,

c.lastname,

c.email,

sum(i.total) as totalspend

from da-nfactorial.chinook.customer c

join da-nfactorial.chinook.invoice i

on c.customerid = i.customerid

group by c.firstname,c.lastname, c.email

order by totalspend

limit 1

#### 

#### 

#### **Вопрос 4: Кто слушает рок-музыку?**

Напишите запрос, чтобы вернуть адрес электронной почты, имя, фамилию и жанр всех слушателей рок-музыки. Верните свой список в алфавитном порядке по адресу электронной почты.

| **firstname** | **lastname** | **email** | **name** |  |  |
| --- | --- | --- | --- | --- | --- |
| 1 | Aaron | Mitchell | aaronmitchell@yahoo.ca | Rock |  |
| 2 | Alexandre | Rocha | alero@uol.com.br | Rock |  |
| 3 | Astrid | Gruber | astrid.gruber@apple.at | Rock |  |
| 4 | Bjørn | Hansen | bjorn.hansen@yahoo.no | Rock |  |

Ваш запрос:

select distinct

c.firstname,

c.lastname,

c.email,

g.name

from da-nfactorial.chinook.genre g

join da-nfactorial.chinook.track tr

on g.genreid = tr.genreid

join da-nfactorial.chinook.invoiceline i

on tr.trackid = i.trackid

join da-nfactorial.chinook.invoice inv

on i.invoiceid = inv.invoiceid

join da-nfactorial.chinook.customer c

on inv.customerid = c.customerid

where g.genreid = 1

order by c.email asc

#### **Вопрос 5: Кто пишет рок-музыку?**

Теперь, когда мы знаем, что наши клиенты любят рок-музыку, мы можем решить, каких музыкантов пригласить сыграть на концерте.

Давайте пригласим артистов, которые написали больше всего рок-музыки в нашем наборе данных. Напишите запрос, который возвращает имя исполнителя и общее количество треков топ 10 рок-групп по количеству треков.

|  |  |  |  |
| --- | --- | --- | --- |
| 1 | Led Zeppelin | 114 |  |
| 2 | U2 | 112 |  |
| 3 | Deep Purple | 92 |  |
| 4 | Iron Maiden | 81 |  |
| 5 | Pearl Jam | 54 |  |
| 6 | Van Halen | 52 |  |
| 7 | Queen | 45 |  |
| 8 | The Rolling Stones | 41 |  |
| 9 | Creedence Clearwater Revival | 40 |  |
| 10 | Kiss | 35 |  |

Ваш запрос:

select

art.name,

count(tr.trackid) as qtytracks

from da-nfactorial.chinook.artist art

join da-nfactorial.chinook.album alb

on art.artistid = alb.artistid

join da-nfactorial.chinook.track tr

on alb.albumid = tr.albumid

join da-nfactorial.chinook.genre g

on tr.genreid = g.genreid

where g.name = 'Rock'

group by art.name

order by qtytracks desc

limit 10

#### **Вопрос 6: Кто зарабатывает и кто тратит больше всего?**

Во-первых, найдите, какой артист заработал больше всего по данным из **InvoiceLines**. Теперь используйте этого исполнителя, чтобы определить, какой клиент потратил больше всего на этого исполнителя.

|  |  |  |
| --- | --- | --- |
| **name** | **totalearning** |  |
| 1 | Iron Maiden | 138.5999999999998 |

|  |  |  |  |
| --- | --- | --- | --- |
| **firstname** | **lastname** | **totalspent** |  |
| 1 | Mark | Taylor | 17.82 |

Ваш запрос:

select

art.name,

sum (i.unitprice\*i.quantity) as totalearning

from da-nfactorial.chinook.artist art

join da-nfactorial.chinook.album alb

on art.artistid = alb.artistid

join da-nfactorial.chinook.track tr

on alb.albumid = tr.albumid

join da-nfactorial.chinook.invoiceline i

on tr.trackid = i.trackid

group by art.name

order by totalearning desc

limit 1

select

c.firstname,

c.lastname,

sum(i.quantity \* i.unitprice) as totalspent

from da-nfactorial.chinook.artist art

join da-nfactorial.chinook.album alb

on art.artistid = alb.artistid

join da-nfactorial.chinook.track tr

on alb.albumid = tr.albumid

join da-nfactorial.chinook.invoiceline i

on tr.trackid = i.trackid

join da-nfactorial.chinook.invoice inv

on i.invoiceid = inv.invoiceid

join da-nfactorial.chinook.customer c

on inv.customerid = c.customerid

where art.artistid = 90

group by c.firstname,

c.lastname

order by totalspent desc

limit 1

### **Вопрос 7: Какой жанр самый популярный?**

Мы хотим найти самый популярный музыкальный жанр для каждой страны. Мы определяем самый популярный жанр как жанр с наибольшим количеством покупок. Напишите запрос, который возвращает каждую страну вместе с топ 1 жанром. Для стран, где несколько жанров с максимальное количество покупок, верните все жанры соответствующие критериям.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **billingcountry** | **name** | **total** |  |  |
| 1 | India | Rock | 24.749999999999989 |  |
| 2 | Belgium | Rock | 20.789999999999996 |  |
| 3 | Czech Republic | Rock | 24.749999999999989 |  |
| 4 | Ireland | TV Shows | 13.93 |  |

Ваш запрос:

with genretotals as (

select

inv.billingcountry,

g.name,

sum(i.unitprice \* i.quantity) as total

from da-nfactorial.chinook.genre g

join da-nfactorial.chinook.track tr

on g.genreid = tr.genreid

join da-nfactorial.chinook.invoiceline i

on tr.trackid = i.trackid

join da-nfactorial.chinook.invoice inv

on i.invoiceid = inv.invoiceid

group by inv.billingcountry, g.name

order by inv.billingcountry

)

SELECT

billingcountry,

name,

total

from

(

select

billingcountry,

name,

total,

rank() over (partition by billingcountry order by total desc) as rank

from genretotals

) as rankedgenres

where rank = 1;

#### **Вопрос 8: Какая песня самая долгая?**

Предоставьте названия всех песен, длина которых превышает среднюю длину песни в наборе данных.

Предоставьте столбцы **Name** и **Milliseconds** для каждой песни. Упорядочьте по длине песни, самые длинные песни укажите первыми.

| 1 | Occupation / Precipice | 5286953 |  |
| --- | --- | --- | --- |
| 2 | Through a Looking Glass | 5088838 |  |
| 3 | Greetings from Earth, Pt. 1 | 2960293 |  |
| 4 | The Man With Nine Lives | 2956998 |  |

Ваш запрос:

select

t.name,

t.milliseconds

from da-nfactorial.chinook.track t

where t.milliseconds > (select avg(t2.milliseconds) from da-nfactorial.chinook.track t2)

group by t.name, t.milliseconds

order by t.milliseconds desc

#### **Вопрос 9:**

Напишите запрос, определяющий **клиента, который потратил больше всего на музыку для каждой страны.** Напишите запрос, который возвращает страну вместе с топ 1 покупателем и суммой, которую он потратил. Для стран, в которых есть несколько покупателей потратившие максимальное количество денег, укажите всех клиентов, потративших эту сумму.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **firstname** | **lastname** | **country** | **total** |  |  |
| 1 | Enrique | Muñoz | Spain | 37.61999999999999 |  |
| 2 | Terhi | Hämäläinen | Finland | 41.62 |  |
| 3 | Johannes | Van der Berg | Netherlands | 40.62 |  |
| 4 | Stanisław | Wójcik | Poland | 37.61999999999999 |  |

Ваш запрос:

with totalspend as(

select

c.firstname,

c.lastname,

i.billingcountry as country,

sum(il.unitprice \* il.quantity) as total

from da-nfactorial.chinook.customer c

join da-nfactorial.chinook.invoice i

on c.customerid = i.customerid

join da-nfactorial.chinook.invoiceline il

on i.invoiceid = il.invoiceid

group by

c.firstname,

c.lastname,

i.billingcountry

)

select

firstname,

lastname,

country,

total

from (

select

firstname,

lastname,

country,

total,

rank() over (partition by country order by total desc) as rank

from totalspend

) as ranked

where rank = 1;

## **Часть 2**

* Определите 4 вопроса о Chinook, на который вы хотите ответить на основе анализа данных.
* Затем напишете SQL-запросы, чтобы получить данные, необходимые для успешного ответа на ваши вопросы.
* Визуализируйте полученные данные (используя гистограммы или другие графики), отвечающие на ваш вопрос.
* Объясните ответ в 1-2 предложениях.
* Вопросы, которые вы задаете, зависят от вас, но все четыре запроса должны содержать JOIN и AGGREGATION.

**Вопрос 1:**

**Объяснение:**

Этот запрос был написан для того, чтобы вычислить среднюю продолжительность треков в секундах для каждого жанра. Результат помогает понять, какие жанры имеют более длинные или короткие треки в среднем.

Ваш запрос и объяснение с графиком:

select

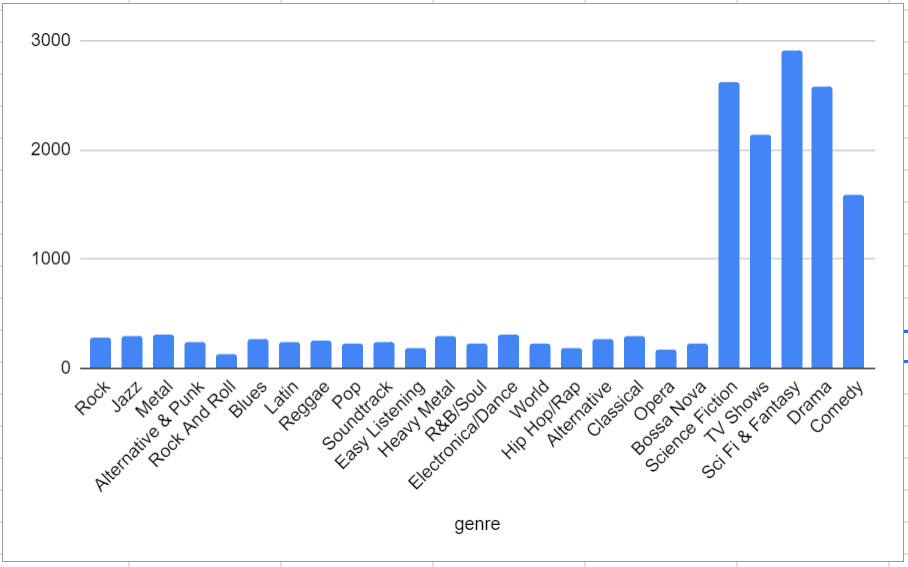
g.name as genre,

round(avg(t.milliseconds/1000),2) as seconds

from da-nfactorial.chinook.track t

join da-nfactorial.chinook.genre g

on t.genreid = g.genreid

group by genre;  


**Вопрос 2:**

**Объяснение:**

Этот запрос был написан для того, чтобы вычислить среднюю цену за трек для каждого жанра. Результат помогает понять, какие жанры имеют более дорогие или дешевые треки в среднем. Запрос возвращает название жанра и среднюю цену за трек для каждого жанра, отсортированную по убыванию цены

Ваш запрос и объяснение с графиком:

select

g.name as genre,

avg(i.unitprice) as unitprice

from da-nfactorial.chinook.track t

join da-nfactorial.chinook.genre g

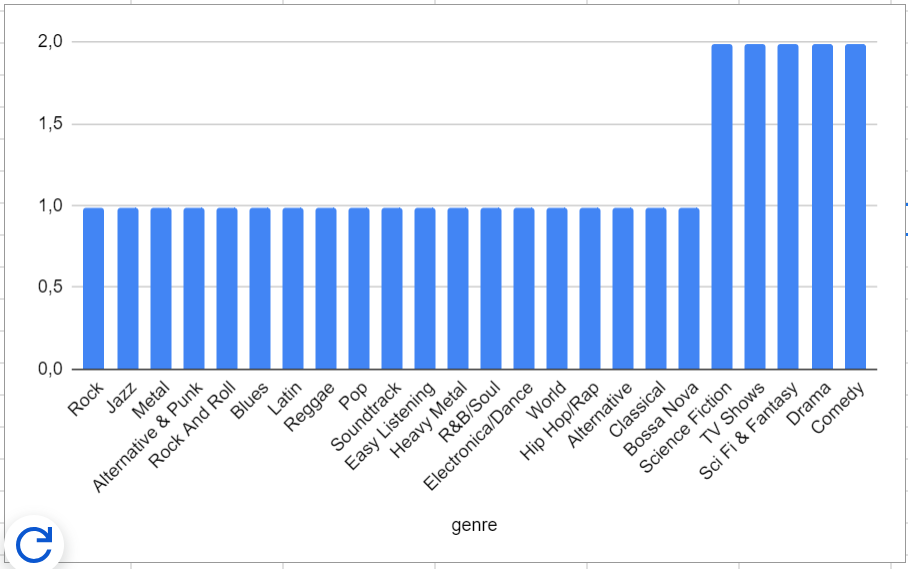
on t.genreid = g.genreid

join da-nfactorial.chinook.invoiceline i

on t.trackid = i.trackid

group by genre

order by unitprice desc;



**Вопрос 3:**

**Объяснение:**

Этот запрос был написан для того, чтобы вычислить общий объём продаж (в денежном выражении) для каждого сотрудника(sales support agent), который поддерживает клиентов. Он показывает, сколько денег заработал каждый сотрудник, отвечающий за поддержку, на основе покупок их клиентов. Результат возвращает фамилию, имя сотрудника и сумму продаж, отсортированную по убыванию.

Ваш запрос и объяснение с графиком:

select

e.lastname,

e.firstname,

sum(inv.unitprice \* inv.quantity) as result

from da-nfactorial.chinook.employee e

join da-nfactorial.chinook.customer c

on e.employeeid = c.supportrepid

join da-nfactorial.chinook.invoice i

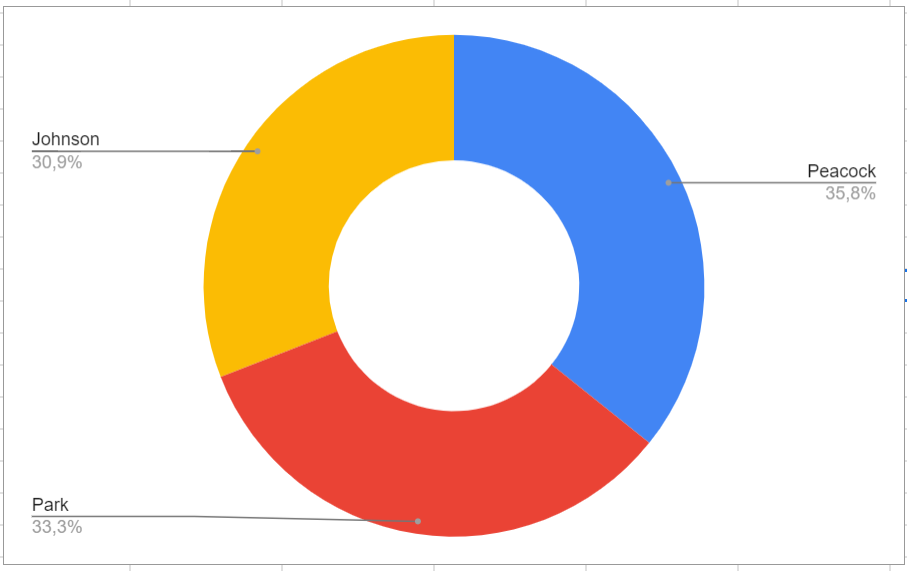
on c.customerid = i.customerid

join da-nfactorial.chinook.invoiceline inv

on i.invoiceid = inv.invoiceid

group by e.lastname, e.firstname

order by result desc



**Вопрос 4:**

**Объяснение:**

Этот запрос был написан для того, чтобы найти артистов, которые продали наибольшее количество своих треков в каждой стране. Он сначала вычисляет общее количество продаж по каждому артисту и каждой стране, а затем с помощью ранжирования определяет, какие артисты имеют наибольшие продажи в каждой стране. Результат возвращает имена артистов, страны и количество проданных треков, отсортированные по убыванию количества продаж.

Ваш запрос и объяснение с графиком:

with totalquantity as (

select

art.name as artist\_name,

i.billingcountry as country,

sum (il.quantity) as quantity

from da-nfactorial.chinook.artist art

join da-nfactorial.chinook.album alb

on art.artistid = alb.artistid

join da-nfactorial.chinook.track tr

on alb.albumid = tr.albumid

join da-nfactorial.chinook.invoiceline il

on tr.trackid = il.trackid

join da-nfactorial.chinook.invoice i

on il.invoiceid = i.invoiceid

group by artist\_name, country

)

select

artist\_name,

country,

quantity

from (

select

artist\_name,

country,

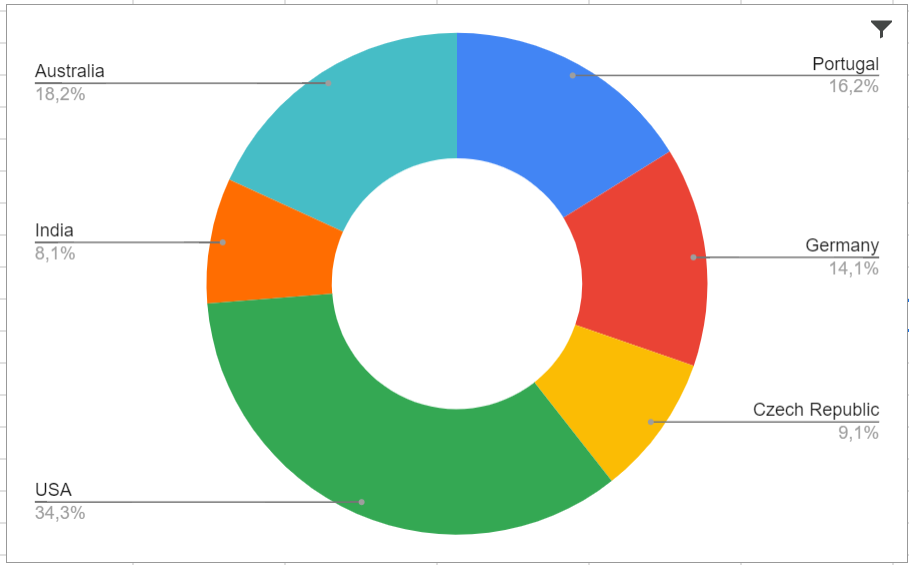
quantity,

rank() over (partition by country order by quantity desc) as rank

from totalquantity

) as ranked

where rank = 1

order by quantity desc  
  
**В круговой диаграмме соотношение стран по sold songs самого знаменитого артиста в database**  
  


Вы закончили свой первый SQL анализ! Отличная работа!