# ИИ с нуля - SQL - домашнее задание:

* Следуя логике задач, вам нужно написать запросы, которые будут возвращать данные в требуемом виде.
* Под каждой задачей указан искомый результат в виде скриншота с таблицей. Как этот результат получить - решать вам.
* После каждой задачи впишите ваше решение, чтобы куратор мог его проверить.
* Всё необходимое для решения домашнего задания можно найти в материалах лекции.
* Среда, в которой можно проверить работу запросов - <https://www.sqlitetutorial.net/tryit/>

### 1. Агрегация значений

Вывести дату самой ранней известной покупки из таблицы с покупками.

* Назвать получившийся столбец *first\_date*.

**Образец:**



Запрос:

SELECT MIN(invoicedate) AS first\_date

FROM invoices;

### 2. Сортировка и уникальные значения

Вывести все страны, в которых когда-либо осуществлялись покупки.

* Названия стран не должны повторяться.
* Отсортировать страны по алфавиту.

**Образец:**



Запрос:

SELECT DISTINCT BillingCountry

FROM invoices

ORDER BY BillingCountry ASC;

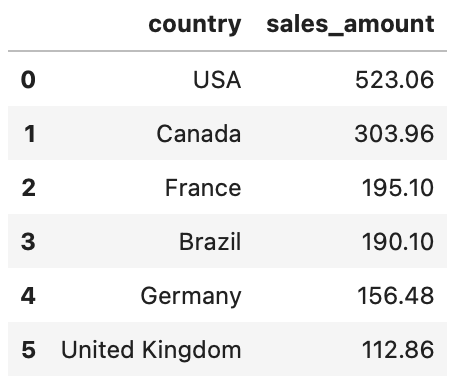
### 

### 3. Агрегация и срезы данных

Вывести суммарные продажи каждой страны.

* Столбец со страной назвать country, а с продажами - *sales\_amount*.
* Оставить только те страны, где продажи более 100$.
* Упорядочить по убыванию *sales\_amount*.

**Образец:**



Запрос:

SELECT DISTINCT BillingCountry AS country,

SUM(Total) AS sales\_amount

FROM invoices

GROUP BY country

HAVING SUM(Total) > 100

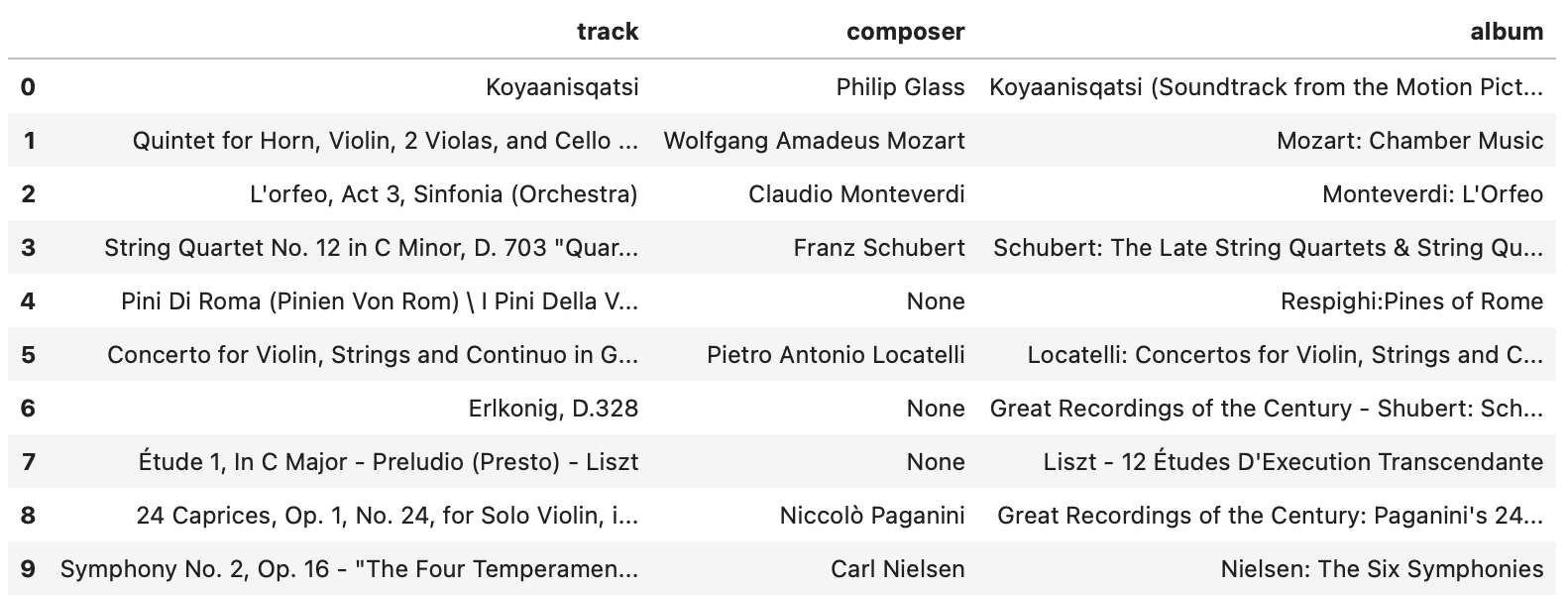
ORDER BY sales\_amount DESC;

### 4. Объединение таблиц

Вывести названия, композиторов и альбомы последних 10 треков. (JOIN)

* Порядок треков определяется их *id*.
* Дать получившимся столбцам имена *track*, *composer* и *album*.
* Пропуски оставить как есть.

**Образец:**



Запрос:

SELECT --TrackId,

tracks.Name AS track,

tracks.Composer AS composer,

albums.Title AS album

FROM tracks

LEFT JOIN albums

ON albums.AlbumId = tracks.AlbumId

ORDER BY TrackId DESC

LIMIT 10;

### 5. Объединение, группировка, агрегация\* (За дополнительные баллы)

Вывести названия топ-10 композиций по суммарным продажам и сами продажи (назвать столбец *sales\_amount*) в порядке убывания. (GROUP BY, Агрегация, JOIN)

* Если продажи треков равны по сумме, отсортировать названия в алфавитном порядке.
* Столбцы назвать *track* и *sales\_amount*.

**Образец:**



Запрос:

SELECT tracks.Name AS track,

SUM(invoice\_items.UnitPrice) AS sales\_amount

FROM invoice\_items

LEFT JOIN tracks ON tracks.TrackId = invoice\_items.TrackId

GROUP BY tracks.TrackId,

track

ORDER BY sales\_amount DESC,

track

LIMIT 10;

***PS: Я не согласен с требуемым решением: в данном случае композиция The Troopers продалась 5 раз по цене 0.99, хоть и в разных вариациях. Композиция-то одна, композитор один. Исполнение разное, продолжительность разная, но трек - один. Хоть и TrackID разные.***

***Мы же не считаем сколько заработала КАЖДАЯ ОТДЕЛЬНАЯ ВЕРСИЯ фильма “Назад в будущее”(1985г) (оригинальная, режиссерская, удлиненная, реставрированная, для “Дня-Назад-В-Будущее ”), мы считаем ОБЩИЕ продажи фильма - $381,906,762.***

***Тогда запрос будет выглядеть вот так (отличие в группировке по имени трека, а не по ID трека):***

***SELECT tracks.Name AS track,***

***SUM(invoice\_items.UnitPrice) AS sales\_amount***

***FROM invoice\_items***

***LEFT JOIN tracks ON tracks.TrackId = invoice\_items.TrackId***

***GROUP BY track***

***ORDER BY sales\_amount DESC,***

***track***

***LIMIT 10;***

6. Объединение, срезы данных\* (За дополнительные баллы)

Вывести имена, фамилии и должности подчинённых сотрудника Nancy Edwards. (JOIN, WHERE)

* Подчинённые должны быть отсортированы по дате приёма на работу.
* \*Попробуйте сделать, *не используя CROSS JOIN*: эта процедура требует много ресурсов в реальных условиях.

**Образец:**



Запрос:

*SELECT e1.FirstName,*

*e1.LastName,*

*e1.Title*

*FROM employees AS e1*

*LEFT JOIN employees AS e2 ON e2.EmployeeId = e1.ReportsTo*

*WHERE e2.LastName == "Edwards"*

*AND e2.FirstName == "Nancy"*

*ORDER BY e1.HireDate ASC;*