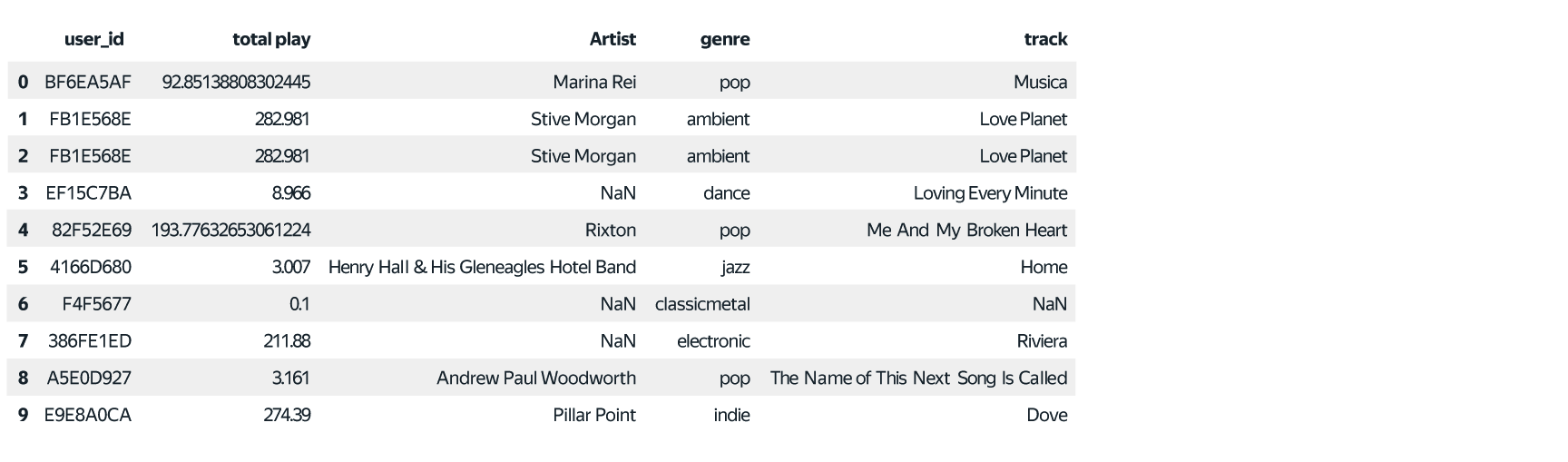
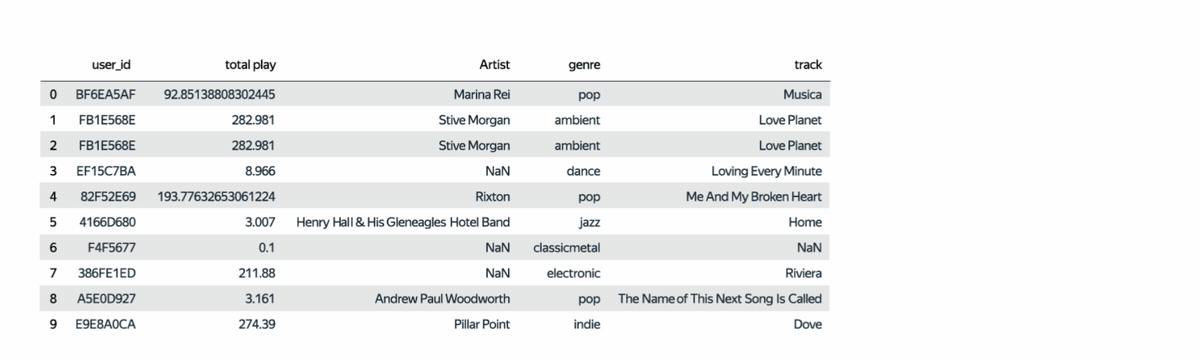
## 5. Объект DataFrame

Таблица, которую мы получили, хранится в структуре данных DataFrame. Давайте подробно разберём, из чего состоит этот объект и какие операции с ним можно выполнять.



DataFrame — это двумерная структура данных Pandas, где у каждого элемента есть две координаты: по строке и по столбцу.



Вы видите две оси, которые формируют объект DataFrame. Первая ось называется индексы, вторая ось — столбцы. По умолчанию индексация в DataFrame начинается с нуля.

Каждая строка — это одно наблюдение, запись об объекте исследования. А столбцы — признаки объектов. В нашем случае одна строка — это одно действие одного пользователя. Прослушивание такой-то композиции в исполнении такой-то группы в течение такого-то времени.

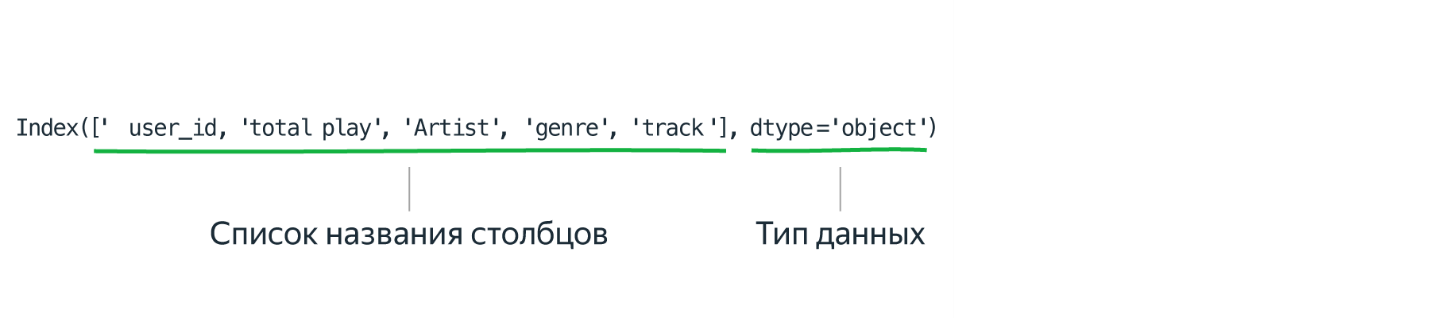
Для лучшего понимания данных полезно получить доступ к их описанию. Это либо документация со сведениями о содержании каждого столбца, либо – не самый лучший вариант – рассказ человека, который предоставил вам эту информацию. Сейчас документация выглядит так:

* *userid* — содержит информацию об уникальном идентификаторе пользователя;
* *total play* — сколько секунд пользователь слушал трек;
* *Artist* — имя исполнителя;
* *genre* — жанр (рок, поп, электронная музыка, классическая и др.);
* *track* — название трека.

Такое описание поможет нам ставить себе корректные задачи.

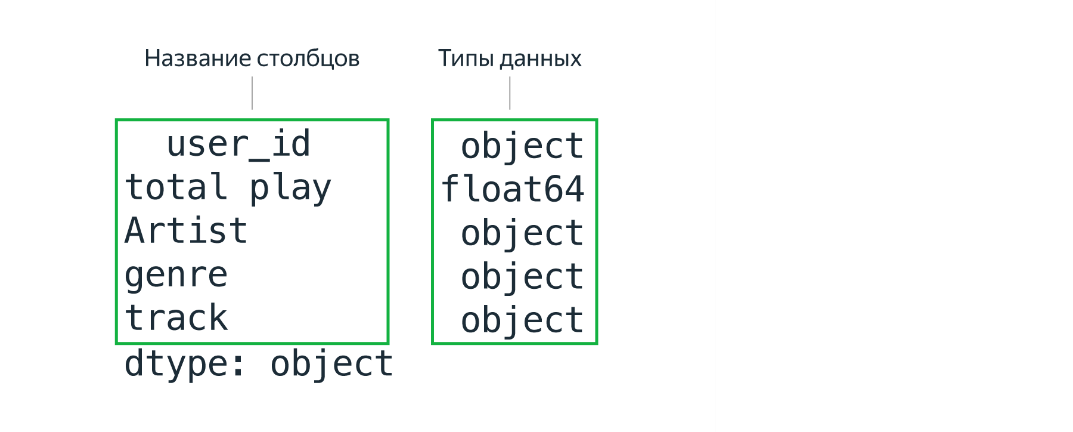
У DataFrame есть неотъемлемые свойства, значения которых можно запросить. Они называются **атрибуты**. Например, атрибут columns содержит информацию о названиях столбцов в наборе данных.

print(df.columns)



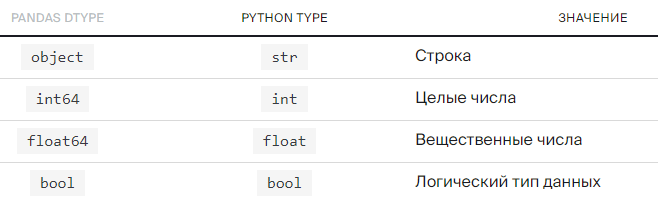
В данном случае атрибут columns вернул список названий столбцов и сообщил, что каждое из них имеет тип данных object.

Вообще типы данных могут быть разные. Для просмотра типа данных каждого столбца лучше всего использовать атрибут dtypes.



Типы данных, о которых сообщают нам атрибуты — это типы данных библиотеки Pandas. Каждому из них соответствует определённый тип данных языка Python.

Так, для *int* таким «двойником» в Pandas будет *int64*. Тип данных *object* используется, когда данные не подходят ни под одну категорию или соответствуют в Python типу «строка». Вот таблица соответствия типов данных Pandas и Python:



О размерах таблицы с данными сообщает её атрибут **shape** (англ. shape, «форма»). В результате получается **кортеж** (неизменяемый список) из двух чисел: первое – количество строк, второе – количество столбцов.

print(df.shape)



В таблице 67963 строк (наблюдений) и 5 столбцов.

**Кортеж** – одномерная неизменяемая последовательность. Это структура данных, похожая на список, её тоже можно сохранять в переменной. Например, кортеж *artist* содержит имена исполнителей:

artist = ('Queen', 'Led Zeppelin', 'Scorpions')

Но если мы попытаемся изменить элемент кортежа, то Python вернёт ошибку:

artist[0] = 'Spice Girls'

---------------------------------------------------------------------------

TypeError Traceback (most recent call last)

<ipython-input-25-4409a5f0dbe8> in <module>()

----> 1 artist[0] = 'Spice Girls'

TypeError: 'tuple' object does not support item assignment

Можно получить информацию как обо всём кортеже, так и об отдельных его элементах:

print(artist)

('Queen', 'Led Zeppelin', 'Scorpions')

print(artist[0])

'Queen'

Кортеж нужен для хранения и чтения данных, которые лучше не изменять. Он похож на текстовый документ, защищённый от редактирования.

Всю информацию, которую предоставляют разные атрибуты DataFrame, можно получить вызовом одного-единственного метода info(). Изучив результаты, которые этот метод возвращает, аналитик выбирает тактику дальнейшей работы с таблицей.

df.info()

<class 'pandas.core.frame.DataFrame'>

RangeIndex: 67963 entries, 0 to 67962

Data columns (total 5 columns):

user\_id 67963 non-null object

total play 67963 non-null float64

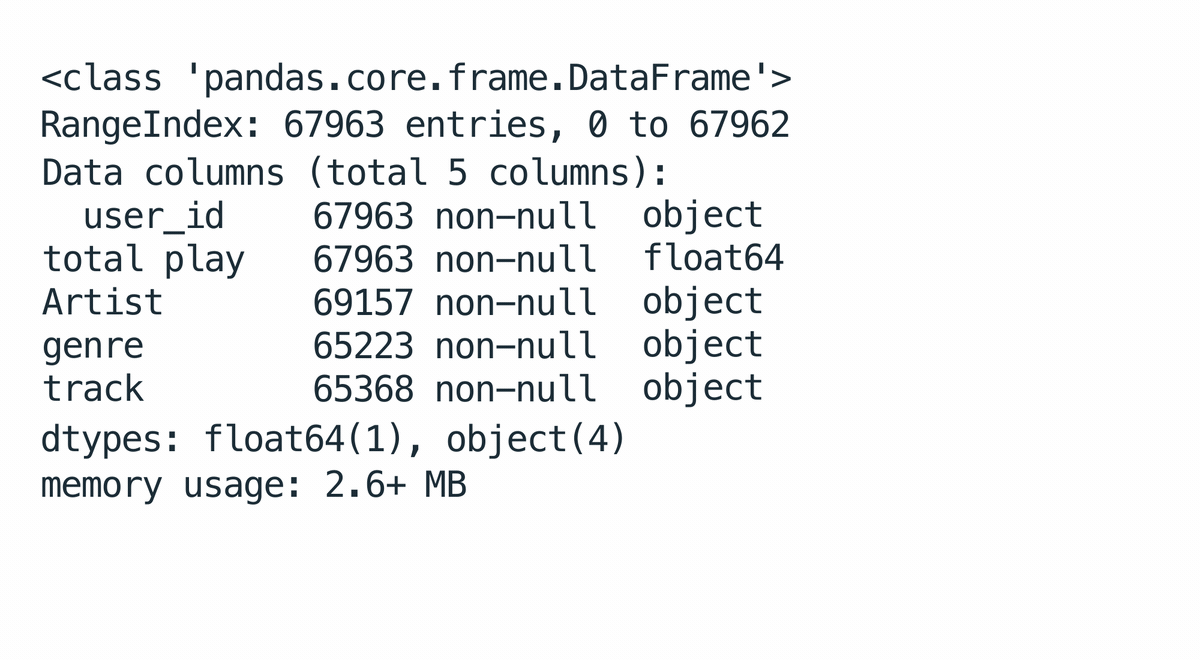
Artist 60157 non-null object

genre 65223 non-null object

track 65368 non-null object

dtypes: float64(1), object(4)

memory usage: 2.6+ MB

Также с

-------- Также смотрим файл 05-05-02-DF\_info\_method.gif ---------------------

Например, здесь в разных столбцах разное количество элементов с определёнными значениями (non-null). Следовательно, в таблице есть пропущенные значения (null). Прежде чем анализировать такие данные, их нужно обработать. Это одна из самых интересных задач аналитика, и мы поговорим о ней подробнее в следующей теме.