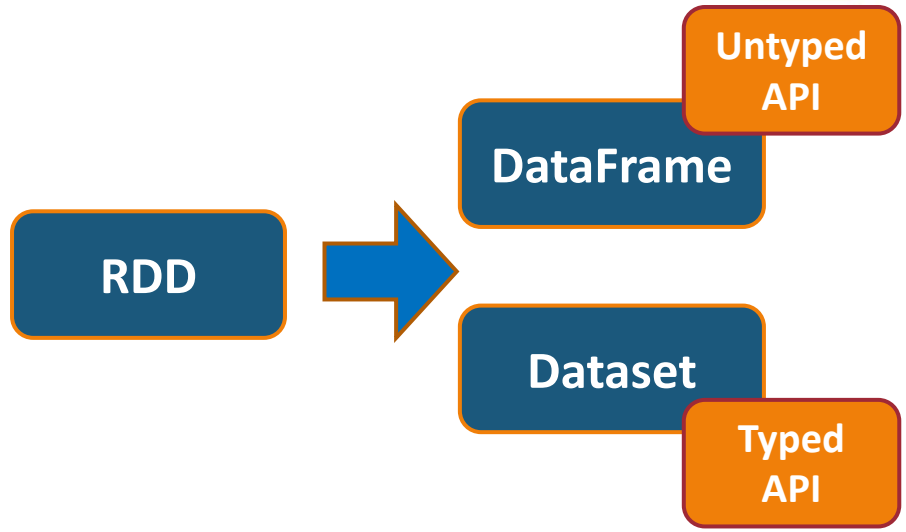
Dataframe и Structured API в Spark

МОДУЛЬ 4

ШКОЛА БОЛЬШИХ ДАННЫХ

Из чего состоят dataframe



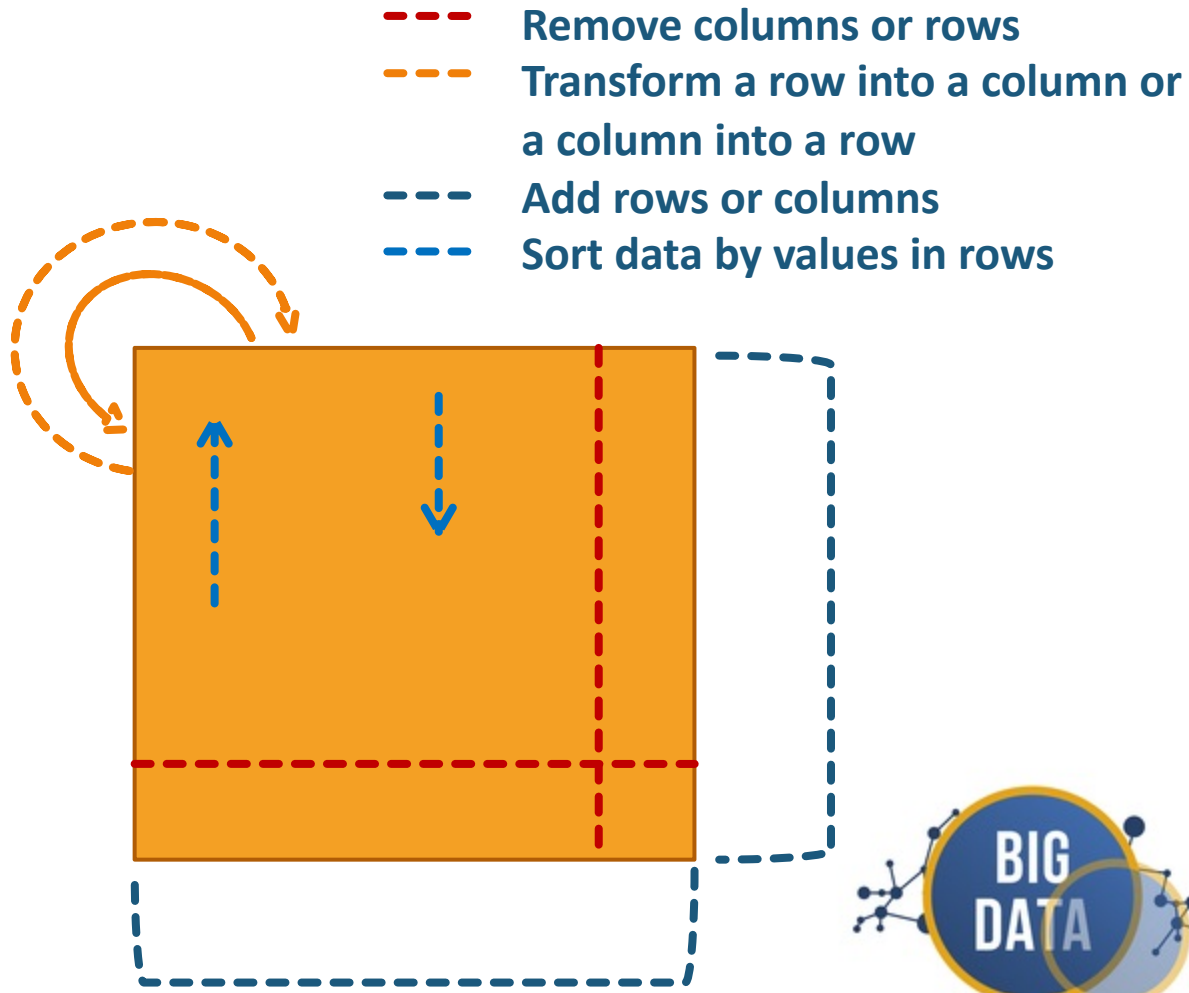
- **датафрейм** состоит из **строк** (объект **“Row”**)
- **строка** состоит из **колонок** (объект **“Column”**)
- **“Row”** не имеет методов, **“Column”** имеет множество методов
- значение в колонке всегда имеет **тип** и определяется схемой (**“Schema”**)
- spark работает с собственным **набором типов данных** (см. документацию)

Выражения и колонки

- **колонка** - это значение, вычисленное для каждой строки с помощью **выражения**
- **выражение** (**“expression”**) - это функция, которая преобразует множество колонок в значение для каждой строки dataframe
- в простейшем случае выражение - само значение колонки датафрейма (**“col("a")”**)
- пример выражения (**“col("a") + col("b") – 5”**)
- функция expr() - создает выражение из строки (**“expr("a + b - 5")”**)

Операции с dataframe

- добавляем или удаляем **колонки**
- добавляем или удаляем **строки**
- **трансформируем колонку** в строку или наоборот
- **меняем порядок** строк
- dataframe изменить нельзя, можно лишь создать новый



Информация о dataframe

- **“schema”** - атрибут dataframe, содержащий **StructType** со схемой
- **“printSchema()”** - печать схемы **в виде дерева**
- **“columns”** - атрибут dataframe, содержащий **список колонок**
- **“describe()”** - вычисляет count, mean, standard deviation, min и max для всех цифровых и стоковых колонок dataframe

Создание dataframe

- методы спарк сессии
  - ✓ **“range()”** - создает датафрейм с колонкой `id` и количеством строк из параметра
  - ✓ **“createDataFrame()”** - создает датафрейм из списка списков

Удаление и добавление колонок

- **“select()”** - выбор (набора) колонок dataframe
- **“selectExpr()”** - аналогично, но с использованием выражений
- **“drop()”** - удаление колонок dataframe
- **“withColumn()”** - добавление колонки
- **“withColumnRenamed()”** - переименование колонки
- **“toDF()”** - переименование всех колонок
- функция **“lit()”** - литерал (используется для добавления колонок с константными значениями)

Удаление и добавление строк

- **“filter()”** - отбор строк dataframe
- **“where()”** - синоним

Добавление строк

- объединение датафреймов (**“join()”**)
- конкатенация датафреймов (**“union()”**)

Объединение dataframe

“join()/union()” - добавление строк и/или колонок

- **всегда два участника**
- **“join()”** - "джойнит" один датафрейм с другим
- **“union()”** - конкатенирует один датафрейм с другим

Join() и их виды

- **“Inner join”:** остаются строки, которые есть в левом и правом датафреймах
- **“Outer join”:** остаются строки, ключи которых есть в левом или правом датафрейме
- **“Left outer join”:** остаются строки, ключи которых есть в левом датафрейме
- **“Right outer join”:** остаются строки, ключи которых есть в правом датафрейме
- **“Left semi join”:** остаются только строки левого датафрейма, ключи которых есть в правом датафрейме
- **“Left anti join”:** остаются только строки левого датафрейма, ключей которых нет в правом датафрейме
- **“Cross (or Cartesian) join”:** каждая строка левого датафрейма объединяется с каждой строкой правого датафрейма

Метод join() и его параметры

leftDf.join (rightDf, joinExpression, how)

- **“how”:** строка (inner, outer, full/full_outer, left/left_outer, right/right_outer, left_semi, left_anti, cross)
- **“joinExpression”:** выражение (например “person["graduate_program"]==graduateProgram['id']”)

На практике разберем виды “join” на примерах.

Union() и его использование

firstDf.union (secondDf)

- конкатенация данных двух датафреймов
- работает по "**локации**" (не по схеме)
- максимальная аккуратность, расположение колонок в нужном порядке
- **“unionByName()”** работает по именам колонок

Агрегаты и оконные функции, сортировка

- **агрегирование** - это получение итогового значения по некой группе
- "тип" итогового значения обычно определяется агрегирующей функцией (sum, max, ...)
- группировать (=агрегировать) можно разными способами
  - ✓ по всему датафрейму (нет группировки – **“df.select(sum("column"))”**)
  - ✓ **“groupBy()”** - по колонкам датафрейма
  - ✓ с использованием "оконных" функций
- агрегация – это трансформация

Сортировка

- два метода (синонимы) **“sort()”** и **“orderBy()”**
- сортируем **по значению** колонки/колонок
- задавать колонки можно **именами** или строковым **выражением**
- сортировать можно по **возрастанию** или **убыванию**

Вопросы?