> Конспект > 4 урок > Практика с DEEQU для Data Quality

> Подготовка

В этой практике будем работать в Jupyter Hub. Для начала импортируем все библиотеки:

```
import pydeequ
from pydeequ.profiles import *
from pydeequ.checks import *
from pydeequ.verification import *
from pyspark.sql import functions as F
from pyspark.sql import SparkSession
```

Создадим Spark-сессию, при этом добавляя некоторые пакеты и параметры для нашей библиотеки pydeequ:

```
spark = SparkSession\
    .builder\
    .config("spark.jars.packages", pydeequ.deequ_maven_coord)\
    .config("spark.jars.excludes", pydeequ.f2j_maven_coord)\
    .getOrCreate()
```

У нас есть некоторый датасет с данными о клиентах магазина, считаем его и посмотрим что в нем есть:

```
df = spark.read.parquet("shop-clients.parquet")
df.show()
df.printSchema()
```

Можем заметить, что датасет содержит базовую информацию о клиенте, а также один количественный показатель - количество заказов:

```
+--------------------+------+----------+----------+-----------+------------+---+------------+---------------+
|                  id|   sex|      name|   surname|second_name|passport_num|age|total_orders|           city|
+--------------------+------+----------+----------+-----------+------------+---+------------+---------------+
|0002baa3306b48f99...|female|      Соня|  Ермолова|  Сергеевич|  7070177511| 21|          23|         Липецк|
|00070f3a44b24c5da...|  male|  Анатолий|    Марков|Анатольевич|  8914742901| 22|          34|Санкт-Петербург|
|000c3230d39040749...|  male|     Павел|    Марков| Кириллович|  3800229978| 69|          45|           Омск|
|00160cd13a3047aaa...|female|Александра|  Дроздова| Кириллович|  6699696036| 48|          75|           Омск|
|00264b9c03c24d82a...|female|      Яна|  Равдеева|   Павлович|  5443568581| 46|          92|         Москва|
|0026f92043174b488...|female| Екатерина|    Дубина|   Олегович|  7731771480| 61|          69|       Новгород|
|00293705f1064490a...|female|      Алла|  Дроздова|   Олегович|  6870766131| 71|          93|         Москва|
|002ee035d9d344729...|  male|    Сергей|  Андропов| Дмитриевич|  6022142935| 55|          35|        Воронеж|
|00407f7c150942548...|female|     Дарья|  Равдеева| Дмитриевич|  8514458437| 48|          64|         Москва|
|00456600cec64ef49...|female|  Виктория|   Жмурина|  Денисович|  2185250116| 21|          46|          Казань|
|004b0a6930c4432db...|  male|   Дмитрий|  Свиридов|  Денисович|  4525230004| 49|          14|           Тула|
|00583a55f4874ca1a...|  male|     Денис| Коновалов| Кириллович|  4203119097| 27|          21|          Тверь|
|005a07bbc5d44d1c9...|female|     Дарья|   Жмурина|  Денисович|  4127441910| 65|          31|         Липецк|
|0060c7f62fcf43a69...|female|     Ольга|    Кирова| Максимович|  1532529212| 73|          11|          Томск|
|006bcd6c8b1a47b29...|  male|      Иван|Акульченко| Викторович|  5464246177| 32|          98|Санкт-Петербург|
|006e8416fc284232b...|female| Екатерина|  Кутузова|   Павлович|  6998786413| 52|           9|          Томск|
|007becc2dcde4096b...|  male|    Максим|  Свиридов|   Иванович|  6634231039| 37|          46|          Казань|
|0094065e055847118...|female|     Ольга|   Прошина|   Иванович|  3274871221| 33|           3|         Москва|
|009f4dd832534fcbb...|  male|      Олег|   Остапов| Максимович|  7541251930| 67|          13|Санкт-Петербург|
|00aba076028743eaa...|  male|   Дмитрий|    Марков| Максимович|  8351419654| 69|          16|          Казань|
+--------------------+------+----------+----------+-----------+------------+---+------------+---------------+
only showing top 20 rows
```

```
root
 |-- id: string (nullable = true)
 |-- sex: string (nullable = true)
 |-- name: string (nullable = true)
 |-- surname: string (nullable = true)
 |-- second_name: string (nullable = true)
 |-- passport_num: long (nullable = true)
 |-- age: integer (nullable = true)
 |-- total_orders: integer (nullable = true)
 |-- city: string (nullable = true)
```

> Analyzer

Первое, что мы хотим сделать с нашим датасетом - проанализировать. Для этого в библиотеке pydeequ есть AnylysisRunner:

```
from pydeequ.analyzers import *

analyzer = AnalysisRunner(spark) \
                    .onData(df) \
                    .addAnalyzer(Size()) \
                    .addAnalyzer(Completeness("name")) \
                    .addAnalyzer(Completeness("surname")) \
                    .addAnalyzer(Completeness("second_name")) \
                    .addAnalyzer(Completeness("passport_num")) \
                    .addAnalyzer(Completeness("city")) \
                    .addAnalyzer(Compliance("age less than 0", 'age<0')) \
                    .addAnalyzer(Compliance("age great than 100", 'age>100')) \
                    .addAnalyzer(Compliance("orders less than 0", 'total_orders<0')) \
```

```
                    .run()

analysisResult_df = AnalyzerContext.successMetricsAsDataFrame(spark, analyzer)
analysisResult_df.show()
```

- Size - размер датафрейма
- Completeness - насколько колонка полна (по сути поиск пробелов и null)
- Compliance - проверяет поле на определенное условие, которое мы задаем

Результатом выполнения анализатора будет:

```
+-------+------------------+------------+-------+
| entity|          instance|        name|  value|
+-------+------------------+------------+-------+
| Column|              city|Completeness| 0.9985|
| Column|      passport_num|Completeness|  0.999|
| Column|orders less than 0|  Compliance| 7.5E-4|
| Column|   age less than 0|  Compliance|    0.0|
| Column|       second_name|Completeness| 0.9995|
|Dataset|                 *|        Size|20000.0|
| Column|              name|Completeness|    1.0|
| Column|           surname|Completeness|    1.0|
| Column|age great than 100|  Compliance|  0.001|
+-------+------------------+------------+-------+
```

Можем заметить, что заполнены не все города, номера паспортов и фамилии. Также мы получили статистику по указанным нами условиям: мы проверили, что отсутствуют строки с возрастом < 0. При этом есть некоторые люди с возрастом > 100.

> Constraint Verification

Следующим этапом будет проверка ограничений, которые мы сами задаем для нашего датасета. Сначала мы создаем чекеры с помощью объекта Check.

```
check_age = Check(spark, CheckLevel.Warning, "Users Age Check")\
        .hasMax("age", lambda x: x <= 100.0)  \
        .isNonNegative("age") \
```

```
        .isComplete("age")
check_fio = Check(spark, CheckLevel.Warning, "Users FIO Check")\
        .isComplete("name") \
        .isComplete("surname") \
        .isComplete("second_name")
check_sex = Check(spark, CheckLevel.Warning, "Users Sex Check")\
        .isContainedIn("sex", ["male", "female"])
check_dataset = Check(spark, CheckLevel.Error, "Users Dataset Check")\
        .hasSize(lambda x: x >= 20000)\
        .isUnique("id")
```

Аргументы объекта Check:

- сессия Spark
- уровень проверки, в нашем случае Warning
- наименование чекера

Далее указываются условия

Первый чекер проверяет возраст на ограничение 100 лет, отрицательных значений и отсутствие пропусков.

Второй чекер проверяет наличие ФИО.

Третий чекер проверяет наличие только male и female в столбце age

Четвертый чекер проверяет, что размер датасета не менее 20к записей и уникальность столбца id.

Для просмотра результата выполним следующие строки:

```
checkResult = VerificationSuite(spark) \
    .onData(df) \
    .addCheck(check_age) \
    .addCheck(check_fio) \
    .addCheck(check_sex) \
    .addCheck(check_dataset) \
    .run()

checkResult_df = VerificationResult.checkResultsAsDataFrame(spark, checkResult)
checkResult_df.toPandas().head(20)
```

Результаты:

	check	check_level	check_status	constraint	constraint_status	constraint_message
0	Users Age Check	Warning	Warning	MaximumConstraint(Maximum(age,None))	Failure	Value: 18798.0 does not meet the constraint re...
1	Users Age Check	Warning	Warning	ComplianceConstraint(Compliance(age is non-neg...	Success	
2	Users Age Check	Warning	Warning	CompletenessConstraint(Completeness(age,None))	Success	
3	Users FIO Check	Warning	Warning	CompletenessConstraint(Completeness(name,None))	Success	
4	Users FIO Check	Warning	Warning	CompletenessConstraint(Completeness(surname,No...	Success	
5	Users FIO Check	Warning	Warning	CompletenessConstraint(Completeness(second_nam...	Failure	Value: 0.9995 does not meet the constraint req...
6	Users Sex Check	Warning	Success	ComplianceConstraint(Compliance(sex contained ...	Success	
7	Users Dataset Check	Error	Success	SizeConstraint(Size(None))	Success	
8	Users Dataset Check	Error	Success	UniquenessConstraint(Uniqueness(List(id),None))	Success	

```
VerificationResult.successMetricsAsDataFrame(spark, checkResult).toPandas().head(20)
```

	entity	instance	name	value
0	Column	age is non-negative	Compliance	1.0000
1	Column	second_name	Completeness	0.9995
2	Column	sex contained in male,female	Compliance	1.0000
3	Dataset	*	Size	20000.0000
4	Column	id	Uniqueness	1.0000
5	Column	name	Completeness	1.0000
6	Column	surname	Completeness	1.0000
7	Column	age	Maximum	18798.0000
8	Column	age	Completeness	1.0000