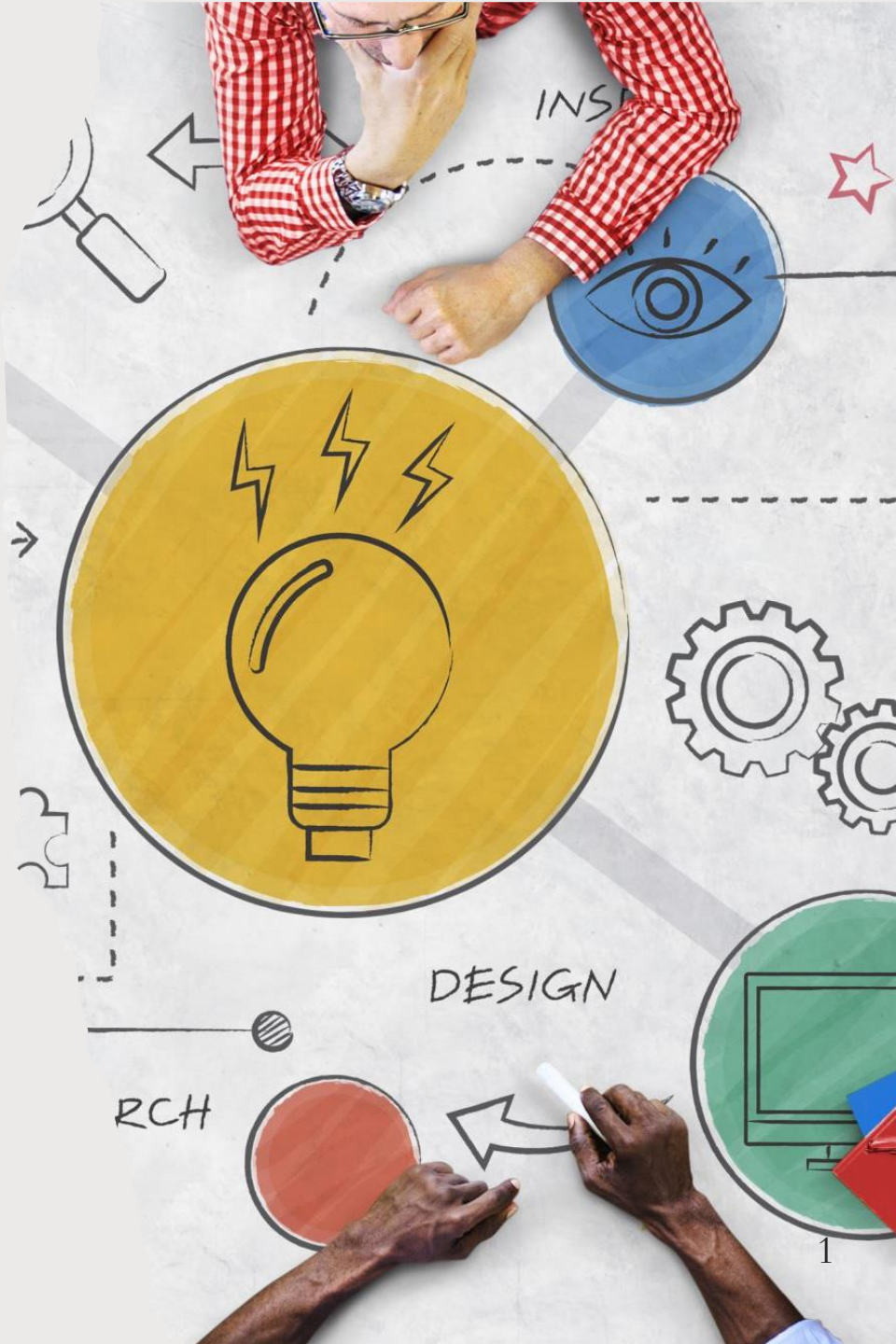
ACTIONS & BASIC TRANSFORMATIONS



CONTENT

- Introduction & Presentation ETL Process
- Actions & Basic Transformations
- Advanced Transformations
- Datavault

GOAL OF THE LEARNING SECTIONS

- Import Data
- Perform Basic Transformation
- Perform Conditional Selection of Rows
- Carry out basic Data Cleaning

IMPORTING DATA

- Importing Data with .spark.read.options()

```
df_fru = spark.read.options(header='True',
                            multiline='True',
                            inferSchema='True')
                            .csv('Fruits.csv')
```

```
df_veg = spark.read.option("multiline",True).json('Vegetables.json')
```

ACCESSING COLUMNS

- Accessing columns with .select()

crop	field	week	water_consumption	revenue	yield_per_sqm
strawberries	7	1	12	0	0
strawberries	7	2	10	0	0
strawberries	7	3	12	0	0
strawberries	7	4	14	0	0
strawberries	7	5	14	0	5
strawberries	7	6	18	30	10
strawberries	7	1	12	0	0
strawberries	7	2	10	0	0
strawberries	7	3	12	0	0
strawberries	7	7	20	60	25

```
df_fru.select('week').show(5)
```

```
+----+
|week|
+----+
|   1|
|   2|
|   3|
|   4|
|   5|
+----+
```

ACCESSING ROWS

- Accessing columns with workaround .collect() then access with print()

(and square brackets-operator for row)

crop	field	week	water_consumption	revenue	yield_per_sqm
strawberries	7	1	12	0	0
strawberries	7	2	10	0	0
strawberries	7	3	12	0	0
strawberries	7	4	14	0	0
strawberries	7	5	14	0	5
strawberries	7	6	18	30	10
strawberries	7	1	12	0	0
strawberries	7	2	10	0	0
strawberries	7	3	12	0	0
strawberries	7	7	20	60	25

```
df_fru_lokal = df_fru.collect()
print(f"Type of entries: {type(df_fru_lokal[0])}\n")
print(f"Entries: {df_fru_lokal[2]}")
```

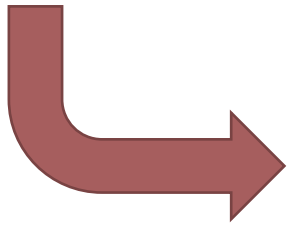
```
Type of entries: <class 'pyspark.sql.types.Row'>

Entries: Row(crop='strawberries', field=7, week=3, water_consumption=12, revenue=0, yield_per_sqm=0)
```

ADDING COLUMNS

- Adding columns with .withColumn()

```
+------------+-----+----+-----------------+-------+-------------+
|        crop|field|week|water_consumption|revenue|yield_per_sqm|
+------------+-----+----+-----------------+-------+-------------+
|strawberries|    7|   7|               20|     60|           25|
|strawberries|    7|   8|               26|    150|           25|
|strawberries|    7|   9|               24|    150|           25|
|strawberries|    7|  10|               10|    100|           25|
|strawberries|    7|  11|             null|    150|            0|
+------------+-----+----+-----------------+-------+-------------+
```



```
df_extraCol = df_fru.withColumn('newColumn', df_fru.yield_per_sqm * df_fru.revenue)
df_extraCol.show()
```

```
+------------+-----+----+-----------------+-------+-------------+---------+
|        crop|field|week|water_consumption|revenue|yield_per_sqm|newColumn|
+------------+-----+----+-----------------+-------+-------------+---------+
|strawberries|    7|   7|               20|     60|           25|     1500|
|strawberries|    7|   8|               26|    150|           25|     3750|
|strawberries|    7|   9|               24|    150|           25|     3750|
|strawberries|    7|  10|               10|    100|           25|     2500|
|strawberries|    7|  11|             null|    150|            0|        0|
+------------+-----+----+-----------------+-------+-------------+---------+
```

REMOVING COLUMNS

- Removing columns with .drop()

crop	field	week	water_consumption	revenue	yield_per_sqm	newColumn
strawberries	7	1	12	0	0	0
strawberries	7	2	10	0	0	0
strawberries	7	3	12	0	0	0
strawberries	7	4	14	0	0	0
strawberries	7	5	14	0	5	0

```
df_fru_2 = df_extraCol.drop(df_extraCol.newColumn)
df_fru_2.show(5)
```

crop	field	week	water_consumption	revenue	yield_per_sqm
strawberries	7	1	12	0	0
strawberries	7	2	10	0	0
strawberries	7	3	12	0	0
strawberries	7	4	14	0	0
strawberries	7	5	14	0	5

BASIC DATA CLEANING

- Removing NAs with `.dropna()`
- Removing duplicates with `.dropDuplicates()`

week	crop	water_consumption	field	revenue	yield_per_sqm
1	strawberries	12	7	0	0
1	strawberries	12	7	0	0
2	strawberries	10	7	0	0
2	strawberries	10	7	0	0
3	strawberries	12	7	0	0
3	strawberries	12	7	0	0
4	strawberries	14	7	0	0
5	strawberries	14	7	0	5
6	strawberries	18	7	30	10
7	strawberries	20	7	60	25
8	strawberries	26	7	150	25
9	strawberries	24	7	150	25
10	strawberries	10	7	100	25
11	strawberries	null	7	150	0

```
(df_fru.select('water_consumption')
       .dropna()
       .dropDuplicates()
       .show())
```

water_consumption
26
12
20
10
24
14
18

CONCATENATING DATAFRAMES

- Concatenating dataframes with .union()

crop	field	week	water_consumption	revenue	yield_per_sqm
strawberries	7	1	12	0	0
strawberries	7	2	10	0	0
strawberries	7	3	12	0	0
strawberries	7	4	14	0	0
strawberries	7	5	14	0	5

crop	field	week	water_consumption	revenue	yield_per_sqm
strawberries	7	6	18	30	10

```
df_fru1.union(df_fru2).show()
```

crop	field	week	water_consumption	revenue	yield_per_sqm
strawberries	7	1	12	0	0
strawberries	7	2	10	0	0
strawberries	7	3	12	0	0
strawberries	7	4	14	0	0
strawberries	7	5	14	0	5
strawberries	7	6	18	30	10

PERFORM CONDITIONAL SELECTION OF ROWS

- Combining aggregate functions, to achieve exact selection: .filter()
  .where()
  .withColumn()
  ...

week	crop	water_consumption	field	revenue	yield_per_sqm
1	strawberries	12	7	0	0
1	strawberries	12	7	0	0
2	strawberries	10	7	0	0
2	strawberries	10	7	0	0
3	strawberries	12	7	0	0
3	strawberries	12	7	0	0
4	strawberries	14	7	0	0
5	strawberries	14	7	0	5
6	strawberries	18	7	30	10
7	strawberries	20	7	60	25
8	strawberries	26	7	150	25
9	strawberries	24	7	150	25
10	strawberries	10	7	100	25
11	strawberries	null	7	150	0

```
(df_fru.select('week', 'water_consumption', 'revenue')
       .filter(df_fru.week > 5)
       .dropna()
       .show())
```

```
+----+----------------+-------+
|week|water_consumption|revenue|
+----+----------------+-------+
|   6|              18|     30|
|   7|              20|     60|
|   8|              26|    150|
|   9|              24|    150|
|  10|              10|    100|
+----+----------------+-------+
```

PERFORM CONDITIONAL SELECTION OF ROWS

```
(df_fru.select('*')
       .where((df_fru.revenue != 0) & (df_fru.week > 7))
       .withColumn('lucrativeness', df_fru.revenue * df_fru.yield_per_sqm)
       .dropna()
       .describe()
       .show())
```

```
+-------+-----------+-----+----+-----------------+------------------+------------+------------------+
|summary|       crop|field|week|water_consumption|           revenue|yield_per_sqm|     lucrativeness|
+-------+-----------+-----+----+-----------------+------------------+------------+------------------+
|  count|          3|    3|   3|                3|                 3|           3|                 3|
|   mean|       null|  7.0| 9.0|             20.0|133.33333333333334|        25.0|3333.3333333333335|
| stddev|       null|  0.0| 1.0|8.717797887081348|28.867513459481287|         0.0| 721.6878364870322|
|    min|strawberries|    7|   8|               10|               100|          25|              2500|
|    max|strawberries|    7|  10|               26|               150|          25|              3750|
+-------+-----------+-----+----+-----------------+------------------+------------+------------------+
```