Data tech test

Workflow

- Run each cell of this notebook by clicking on the play button ▸ on the top-right.
- Read carefully the questions and replace the *[INSERT YOUR CODE HERE]* placeholders with your answers.
- Explain with few words your answers in the *[DOUBLE-CLICK TO ADD COMMENTS HERE]* cells.
- Test and fix your code, then download the notebook as a PDF file from the menu option: *File → Download as → PDF (.pdf)*.
- **Send us the PDF file by email**.

You can write the answers in SQL (Hive SQL dialect. Documentation here https://cwiki.apache.org/confluence/display/Hive/Tutorial (https://cwiki.apache.org/confluence/display/Hive/Tutorial)) or in Scala/Spark (documentation here https://spark.apache.org/docs/latest/quick-start.html (https://spark.apache.org/docs/latest/quick-start.html)).

```
// Run the following setup code
val users = sparkSession.read.json("../notebooks/data-tech-test/users.jsonl")
val streams = sparkSession.read.json("../notebooks/data-tech-test/streams.jsonl")
val sqlContext = new org.apache.spark.sql.SQLContext(sparkContext)
users.createOrReplaceTempView("users")
streams.createOrReplaceTempView("streams")
// you can ignore the SQLContext warning
```

Data

There are two data tables:

- the **users** table that contains the information about the users of the service
- and the **streams** table where the streams (song played by the users) are stored.

> **Important**:
>
> the *users.registration* and *streams.stream_time* fields are dates formatted as Unix time stamps in milliseconds, for example `1559347200000` represents the following date *2019/06/01 00:00:00 UTC+0*

You can see extracts of both tables content bellow:

- Users table extract:

```
sqlContext.sql("""
select * from users limit 3
""")
```

- Streams table extract:

```
sqlContext.sql("""
select * from streams limit 3
""")
```

Question 1: Simple filtering

List the IDs of all the users from France (FR)

```
sqlContext.sql("""
[INSERT YOUR CODE HERE]
""")
```

...

[DOUBLE-CLICK TO ADD YOUR COMMENTS HERE]

...

Question 2: Multiple filters

List the IDs of all the users who match the following criteria:

- consent to receive messages (see *consent* field)
- don't use a @*gmail.com* email address
- have registered before 2019/06/15
- and never streamed

```
sqlContext.sql("""
[INSERT YOUR CODE HERE]
""")
```

...

[DOUBLE-CLICK TO ADD YOUR COMMENTS HERE]

...

Question 3: Aggregation

List the IDs of all the users who match the following criteria:

- have streamed at least two times a song from the artist with ID = 1
- have never streamed any song from the artist with ID = 9

```
sqlContext.sql("""
[INSERT YOUR CODE HERE]
""")
```

...

[DOUBLE-CLICK TO ADD YOUR COMMENTS HERE]

...

Question 4: Understand business requirements

The artists promotion team would like to send promotion messages for the artist with ID = 3.

Here are the business requirements:

- Users who don't consent to receive messages should not receive this promotion message.
- Users who already know well and stream often the artist will not find the message useful.
- The promoted artist has the same style as the artists with ID = 5 and ID = 7. Users who like them will be interested in this promotion message.
- The promotion message is only available in English and Arabic language, not in French.

Which criteria will you propose to select the users who will receive the promotion message ?

```
sqlContext.sql("""
[INSERT YOUR CODE HERE]
""")
```

...

[DOUBLE-CLICK TO ADD YOUR COMMENTS HERE]

...

Thank you!

Vars Errors

Terms defined

Name	Type

Build: | **buildTime**-*Tue Jul 24 18:21:18 UTC 2018* | **formattedShaVersion**-*0.8.3-da67206671e87656b41ba5aa03968334fb990f4a* | **sbtVersion**-*0.13.15* | **scalaVersion**-*2.11.8* | **sparkNotebookVersion**-*0.8.3* | **viewer**-*false* | **hadoopVersion**-*2.7.3* | **jets3tVersion**-*0.7.1* | **jlineDef**-*(jline,2.12)* | **sparkVersion**-*2.2.2* | **withHive**-*true* |.