databricks S22 HW6_template

Assignment 6

setup environment (must run first)

```
%python
courses_df = spark.read.format("json").load("/FileStore/tables/courses.json")
enrolledin_df = spark.read.format("json").load("/FileStore/tables/enrolledin.json")
meetings_df = spark.read.format("json").load("/FileStore/tables/meetings.json")
teaches_df = spark.read.format("json").load("/FileStore/tables/teaches.json")
thumbsup_df = spark.read.format("json").load("/FileStore/tables/thumbsup.json")
users_df = spark.read.format("json").load("/FileStore/tables/users.json")
posts_df = spark.read.format("json").load("/FileStore/tables/posts.json")
posts_df = spark.read.option("multiLine", True).json("/FileStore/tables/posts.json")

courses_df.createOrReplaceTempView("courses")
enrolledin_df.createOrReplaceTempView("enrolledin")
meetings_df.createOrReplaceTempView("meetings")
teaches_df.createOrReplaceTempView("teaches")
thumbsup_df.createOrReplaceTempView("thumbsup")
users_df.createOrReplaceTempView("users")
posts_df.createOrReplaceTempView("posts")
```

```
%sql
(select "courses", count(*) from courses)
union all
(select "enrolledin", count(*) from enrolledin)
union all
(select "meetings", count(*) from meetings)
union all
(select "teaches", count(*) from teaches)
union all
(select "thumbsup", count(*) from thumbsup)
union all
(select "users", count(*) from users)
union all
(select "posts", count(*) from posts)
```

	courses	count(1)
1	courses	90
2	enrolledin	1146
3	meetings	1506
4	teaches	290
5	thumbsup	519
6	users	500
7	posts	1000

Solution

```
#1.A
users_df.dtypes
#1.B name data type is: a struct consisting of 2 strings

Out[178]: [('education',
  'array<struct<degree:string,graduation_year:bigint,major:string,school:string>>'),
 ('email', 'string'),
 ('isinstructor', 'boolean'),
```

```
 ('isstudent', 'boolean'),
 ('name', 'struct<first_name:string,last_name:string>'),
 ('occupation', 'string'),
 ('user_id', 'string')]
```

```
%sql
--1.C
DESCRIBE posts;
--1.D The topic field is an array of strings
```

	col_name	data_type	comment
1	body	string	null
2	created_at	string	null
3	meeting_id	string	null
4	post_id	string	null
5	post_type	string	null
6	replied_to_post_id	string	null
7	topics	array<string>	null
8	user_id	string	null

```
#2.A DF
users_df.where("user_id = 4").show()
```

```
+---------+--------------------+------------+---------+----------------+--------------------+-------+
|education|               email|isinstructor|isstudent|            name|          occupation|user_id|
+---------+--------------------+------------+---------+----------------+--------------------+-------+
|     null|ryanhoward@penn s...|       false|     true|{Daniel, Duarte}|Building services...|      4|
+---------+--------------------+------------+---------+----------------+--------------------+-------+
```

```
%sql
--2.A SQL
SELECT *
FROM users
WHERE user_id = 4;
```

	education	email	isinstructor	isstudent	name	occupation	user_id
1	null	ryanhoward@penn state.edu	false	true	{"first_name": "Daniel", "last_name": "Duarte"}	Building services engineer	4

Showing all 1 rows.

```
#2.B DF
# Sort to display the same sequence of results as extra credits
enrolledin_df.where("user_id % 2 = 0").groupBy(enrolledin_df.user_id).agg({"course_id":
"count"}).orderBy("user_id").limit(10).show()
```

```
+-------+----------------+
|user_id|count(course_id)|
+-------+----------------+
|      0|               5|
|    102|               6|
|    104|               4|
|    106|               6|
|    108|               4|
|    114|               6|
|    118|               3|
|    122|               4|
|    124|               2|
|    126|               4|
+-------+----------------+
```

```sql
%sql
--2.B SQL
SELECT e.user_id, COUNT(e.course_id)
FROM enrolledin AS e
WHERE e.user_id % 2 = 0
GROUP BY e.user_id
ORDER BY e.user_id
LIMIT 10;
```

	user_id	count(course_id)
1	0	5
2	102	6
3	104	4
4	106	6
5	108	4
6	114	6
7	118	3
8	122	4
9	124	2
10	126	4

```
#2.C DF
teaches_df.groupBy("user_id").agg({"course_id": "count"}).orderBy(desc("COUNT(course_id)")).limit(5).show()
```

```
+-------+----------------+
|user_id|count(course_id)|
+-------+----------------+
|    486|              10|
|    366|               6|
|    492|               6|
|    100|               6|
|    347|               6|
+-------+----------------+
```

```sql
%sql
--2.C SQL
SELECT user_id, COUNT(course_id)
FROM teaches
GROUP BY user_id
ORDER BY COUNT(course_id) DESC
LIMIT 5;
```

	user_id	count(course_id)
1	486	10
2	366	6
3	492	6
4	100	6
5	347	6

```
#2.D DF
users_df.withColumn("indv_edu", explode(users_df.education)).groupBy("user_id").agg({"indv_edu":
"count"}).orderBy(desc("count(indv_edu)"), "user_id").limit(10).show()
```

```
+-------+---------------+
|user_id|count(indv_edu)|
+-------+---------------+
|    110|              5|
|    131|              5|
```

```
|    144|              5|
|    149|              5|
|    171|              5|
|    176|              5|
|    187|              5|
|    195|              5|
|     21|              5|
|    235|              5|
+-------+---------------+
```

```
%sql
--2.D SQL
SELECT user_id, SIZE(education)
FROM users
ORDER BY SIZE(education) DESC, user_id ASC
LIMIT 10;
```

	user_id	size(education)
1	110	5
2	131	5
3	144	5
4	149	5
5	171	5
6	176	5
7	187	5
8	195	5
9	21	5
10	235	5

```
#2.E DF
teaches_df.join(users_df, ["user_id"], 'inner').groupBy("user_id", "name.first_name", "name.last_name",
"email").agg({"course_id": "count"}).orderBy(desc("COUNT(course_id)")).limit(5).show()
```

```
+-------+----------+---------+--------------------+----------------+
|user_id|first_name|last_name|               email|count(course_id)|
+-------+----------+---------+--------------------+----------------+
|    486|  Nicholas|    Davis|  mchristian@uci.edu|              10|
|    347|   Lindsay|    Cooke|     tracy86@ucr.edu|               6|
|    100|    Denise|    Poole|rachelleach@umass...|               6|
|    492|      Mrs.|   Brandy| alexisperry@mit.edu|               6|
|    223|  Kathleen|   Fowler| steinjanet@ucsd.edu|               6|
+-------+----------+---------+--------------------+----------------+
```

```
%sql
--2.E SQL
SELECT t.user_id, u.name.first_name, u.name.last_name, u.email, COUNT(t.course_id)
FROM teaches AS t INNER JOIN users AS u ON t.user_id = u.user_id
GROUP BY t.user_id, u.name.first_name, u.name.last_name, u.email
ORDER BY COUNT(t.course_id) DESC
LIMIT 5;
```

	user_id	first_name	last_name	email	count(course_id)
1	486	Nicholas	Davis	mchristian@uci.edu	10
2	347	Lindsay	Cooke	tracy86@ucr.edu	6
3	100	Denise	Poole	rachelleach@umass.edu	6
4	492	Mrs.	Brandy	alexisperry@mit.edu	6
5	223	Kathleen	Fowler	steinjanet@ucsd.edu	6

Showing all 5 rows.

```
#2.F DF
topPoster = posts_df.groupBy("user_id").agg({"post_id":
"count"}).orderBy(desc("count(post_id)")).select("user_id", "count(post_id)").limit(5)
df2 = topPoster.join(users_df, ["user_id"], 'inner').select("user_id", "count(post_id)", "name.first_name",
"name.last_name").show()
```

```
+-------+--------------+----------+---------+
|user_id|count(post_id)|first_name|last_name|
+-------+--------------+----------+---------+
|     29|            18|       Amy|     Lamb|
|     97|            13|     Amber| Townsend|
|    191|            11|     Maria|  Ramirez|
|    213|            13| Dominique|   Harper|
|    491|            12|  Jennifer|    Evans|
+-------+--------------+----------+---------+
```

```
%sql
--2.F SQL
WITH topPoster (user_id, postCount) AS (
  SELECT p.user_id, COUNT(p.post_id)
  FROM posts AS p
  GROUP BY p.user_id
  ORDER BY COUNT(p.post_id) DESC
  LIMIT 5 )
SELECT tp.user_id, tp.postCount, u.name.first_name, u.name.last_name
FROM topPoster AS tp INNER JOIN users AS u ON tp.user_id = u.user_id
```

	user_id	postCount	first_name	last_name
1	29	18	Amy	Lamb
2	97	13	Amber	Townsend
3	191	11	Maria	Ramirez
4	213	13	Dominique	Harper
5	491	12	Jennifer	Evans

Showing all 5 rows.

```
%sql
--2.G SQL
SELECT m.course_id, c.course_name, AVG(m.duration)
FROM meetings AS m INNER JOIN courses AS c ON m.course_id = c.course_id
GROUP BY m.course_id, c.course_name
ORDER BY AVG(m.duration) DESC
LIMIT 5;
```

	course_id	course_name	avg(duration)
1	89	Calculus IV	120
2	68	Computer Science 101	120
3	22	Anatomy III	120
4	61	Macroeconomics	120
5	34	Financial Theory	120

Showing all 5 rows.

```
#2.H
from functools import reduce
from pyspark.sql import functions as F
from pyspark.sql.functions import *


data = enrolledin_rdd.collect()
data_even = enrolledin_rdd.filter(lambda x: int(x[2]) % 2 == 0)
data_map = data_even.map(lambda x: (x[2], 1))
data_reduce = data_map.reduceByKey(lambda a, b: a + b)
print(data_reduce.sortByKey(True).take(10))
```

```
[('0', 5), ('102', 6), ('104', 4), ('106', 6), ('108', 4), ('114', 6), ('118', 3), ('122', 4), ('124', 2), ('126', 4)]
```