1) Сделать таблицу artists в Hive и вставить туда значения, используя датасет.

```
0: jdbc:hive2://localhost:10000> CREATE TABLE artists(mbid STRING, artist_mb STRING, artist_lastfm STRING, country_mb STRING, country_lastf
m STRING, tags_mb STRING, tags_lastfm STRING, listeners_lastfm DOUBLE, scrobbles_lastfm DOUBLE, ambiguous_artist BOOLEAN) ROW FORMAT DELIMI
TED FIELDS TERMINATED BY ',' STORED AS TEXTFILE;
No rows affected (0.114 seconds)
0: jdbc:hive2://localhost:10000> LOAD DATA INPATH '/artists.csv' INTO TABLE artists;
No rows affected (0.553 seconds)
0: jdbc:hive2://localhost:10000>
```

2) Используя Hive найти (команды и результаты записать в файл и добавить в репозиторий):

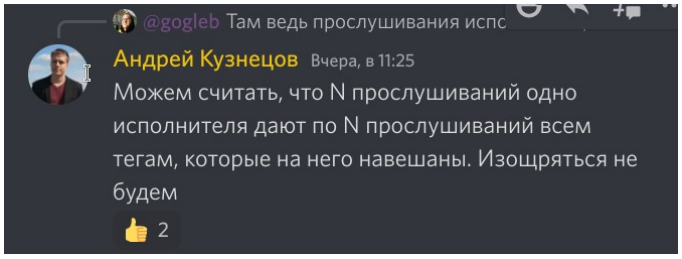
a) Исполнителя с максимальным числом скробблов

```
0: jdbc:hive2://localhost:10000> SELECT
. . . . . .>     artist_lastfm AS artist_most_popular
. . . . . .> FROM
. . . . . .>     artists
. . . . . .> WHERE
. . . . . .>     artists.scrobbles_lastfm IN (
. . . . . .>         SELECT MAX(scrobbles_lastfm) FROM artists
. . . . . .>     );
```

```
+----------------------+
| artist_most_popular  |
+----------------------+
| The Beatles          |
+----------------------+
1 row selected (11.266 seconds)
```

b) Самый популярный тэг на ластфм

Самый популярный тэг искал, исходя из количества тэгов и количества прослушиваний, в соответствии с комментарием преподавателя.




```
0: jdbc:hive2://localhost:10000> SELECT
. . . . . .> 	t2.tag AS tag_top1
. . . . . .> FROM (
. . . . . .> 	SELECT
. . . . . .> 		tag_array.tag AS tag,
. . . . . .> 		count(tag_array.tag) as tag_count,
. . . . . .> 		sum(t1.scrobbles_lastfm) as scrobbles_sum
. . . . . .> 	FROM (
. . . . . .> 		SELECT
. . . . . .> 			tags_lastfm,
. . . . . .> 			scrobbles_lastfm
. . . . . .> 		FROM
. . . . . .> 			artists
. . . . . .> 		WHERE
. . . . . .> 			tags_lastfm != ""
. . . . . .> 		) t1
. . . . . .> 	LATERAL VIEW EXPLODE (SPLIT(LOWER(t1.tags_lastfm), '; ')) tag_array AS tag
. . . . . .> 	GROUP BY
. . . . . .> 		tag
. . . . . .> 	ORDER BY
. . . . . .> 		scrobbles_sum DESC
. . . . . .> 	) t2
. . . . . .> LIMIT 1;
WARNING: Hive-on-MR is deprecated in Hive 2 and may not be available in the future versions. Consider using a di
.e. spark, tez) or using Hive 1.X releases.
+------------+
|  tag_top1  |
+------------+
| seen live  |
+------------+
1 row selected (11.585 seconds)
```

c) Самые популярные исполнители 10 самых популярных тегов ластфм баллов

```
0: jdbc:hive2://localhost:10000> SELECT DISTINCT
. . . . . .>     t3.artist_lastfm AS artist_top10,
. . . . . .>     t3.listeners_lastfm AS listeners
. . . . . .> FROM (
. . . . . .>     SELECT tag, artist_lastfm, listeners_lastfm
. . . . . .>     FROM artists
. . . . . .>     LATERAL VIEW EXPLODE (SPLIT(tags_lastfm, '; ')) tag_array AS tag
. . . . . .> ) t3
. . . . . .> WHERE t3.tag IN
. . . . . .> (
. . . . . .>     SELECT t2.tag AS tag_top10
. . . . . .>     FROM (
. . . . . .>         SELECT
. . . . . .>             tag_array.tag AS tag,
. . . . . .>             count(tag_array.tag) as tag_count,
. . . . . .>             sum(t1.scrobbles_lastfm) as scrobbles_sum
. . . . . .>         FROM (
. . . . . .>             SELECT tags_lastfm, scrobbles_lastfm FROM artists WHERE tags_lastfm != ""
. . . . . .>         ) t1
. . . . . .>         LATERAL VIEW EXPLODE (SPLIT(LOWER(t1.tags_lastfm), '; ')) tag_array AS tag
. . . . . .>         GROUP BY tag
. . . . . .>         ORDER BY scrobbles_sum DESC
. . . . . .>     ) t2
. . . . . .>     LIMIT 10
. . . . . .> )
. . . . . .> ORDER BY t3.listeners_lastfm DESC
. . . . . .> LIMIT 10;
```

artist_top10	listeners
Coldplay	5381567.0
Radiohead	4732528.0
Red Hot Chili Peppers	4620835.0
Rihanna	4558193.0
Eminem	4517997.0
The Killers	4428868.0
Kanye West	4390502.0
Nirvana	4272894.0
Muse	4089612.0
Queen	4023379.0

10 rows selected (27.778 seconds)

d) Любой другой инсайт на ваше усмотрение: найти топ 10 стран по количеству слушателей (*listeners_lastfm*), и в каждой найденной стране определить самый часто встречаемый тэг.

```
0: jdbc:hive2://localhost:10000> SELECT
. . . . . .> 	t4.country AS country, t4.tag AS tag
. . . . . .> FROM (
. . . . . .> 	SELECT
. . . . . .> 		*, row_number() OVER (PARTITION BY t3.country ORDER BY t3.tag_count DESC) AS tag_order
. . . . . .> 	FROM (
. . . . . .> 		SELECT
. . . . . .> 			t2.country_lastfm AS country, t2.tag AS tag, count(*) AS tag_count
. . . . . .> 		FROM (
. . . . . .> 			SELECT country_lastfm, tag
. . . . . .> 			FROM artists
. . . . . .> 			LATERAL VIEW EXPLODE (SPLIT(LOWER(tags_lastfm), '; ')) tag_array AS tag
. . . . . .> 			WHERE tags_lastfm != ""
. . . . . .> 		) t2
. . . . . .> 	WHERE
. . . . . .> 		t2.country_lastfm IN (
. . . . . .> 			SELECT t1.country_lastfm AS country_top10
. . . . . .> 			FROM (
. . . . . .> 				SELECT country_lastfm, sum(listeners_lastfm) AS listeners_sum
. . . . . .> 				FROM artists
. . . . . .> 				WHERE country_lastfm != ""
. . . . . .> 				GROUP BY country_lastfm
. . . . . .> 				ORDER BY listeners_sum DESC
. . . . . .> 				LIMIT 10
. . . . . .> 			) t1
. . . . . .> 		)
. . . . . .> 	GROUP BY t2.country_lastfm, t2.tag
. . . . . .> 	) t3
. . . . . .> ) t4
. . . . . .> WHERE
. . . . . .> 	t4.tag_order = 1;
```

Выглядит патриотично:)

```
+---------------------------------+-------------+
|             country             |     tag     |
+---------------------------------+-------------+
| Australia                       | australian  |
| Canada                          | canadian    |
| France                          | french      |
| Georgia; United States          | georgia     |
| Germany                         | german      |
| Japan                           | japanese    |
| Sweden                          | swedish     |
| United Kingdom                  | british     |
| United Kingdom; United States   | american    |
| United States                   | american    |
+---------------------------------+-------------+
10 rows selected (19.496 seconds)
```