EJADA

# Big Data Team

**Abdullah Galal**


Find Dataset

Load it into dataframe

Do some analysis

Visualize your results

# 1) Dataset :

- **Link** : https://www.kaggle.com/gagazet/path-of-exile-league-statistic

- **Content :**

The data contains statistics for 59,000 players who played path of exile.

One file with 12 data sections.

- **Inferred schema:**

```
root
 |-- rank: integer (nullable = true)
 |-- dead: boolean (nullable = true)
 |-- online: boolean (nullable = true)
 |-- name: string (nullable = true)
 |-- level: integer (nullable = true)
 |-- class: string (nullable = true)
 |-- id: string (nullable = true)
 |-- experience: double (nullable = true)
 |-- account: string (nullable = true)
 |-- challenges: integer (nullable = true)
 |-- twitch: string (nullable = true)
 |-- ladder: string (nullable = true)
```

- **Column name and description :**

| Column name | Description |
|---|---|
| Rank | Ranking of the player |
| Dead | If player dead or not |
| Online | Player playing online or offline |
| Name | Name of the player |
| Level | Level of the player |
| Class | Character who player chooses |

| | |
|---|---|
| Id | Id distinct for every player character |
| Experience | Points , player takes when end challenge |
| Account | Account name distinct for every player |
| Challenges | Number of challenges which player entered |
| Twitch | website designed for people to stream video games ( It shows who stream video games ) |
| Ladder | Description below |

One league - Harbinger, but 4 different types of divisions:

- Harbinger
- Hardcore Harbinger
- SSF Harbinger
- SSF Harbinger HC

Each division has own ladder with leaders

# 2) Preparing Dataset :

## Write Dataset in Hive parquet table :

**Step 1 :** Copy CSV to HDFS.

- Run following command in the shell :

```
hdfs dfs -put /tmp/idea-IU-183.6156.11/poe_stats.csv /user/hive/warehouse/
```

**Step 2 : Go to Hue, impala Query UI, and execute the following queries :**

```
// Create Hive table and Load CSV
CREATE TABLE hive_task_db.task_tb(
     `rank` INT,
     `dead` BOOLEAN,
     `online` BOOLEAN,
     `name` STRING,
     `level` INT,
     `class` STRING,
     `id` STRING,
      `experience` DOUBLE,
      `account` STRING,
      `challenges` INT,
      `twitch` STRING,
      `ladder` STRING)
    ROW FORMAT DELIMITED FIELDS TERMINATED BY ","
    LOCATION "hdfs:///user/hive/warehouse/"
    tblproperties("skip.header.line.count"="1" ,"serialization.null.format"="null");

```

```
////////////////////////////////NEXT STEP/////////////////////////////////////
////////////////////////////////////////////////////////////////////////////////

// Create Parquet table
CREATE TABLE hive_task_db.parquet_tb(
    `rank` INT,
    `dead` BOOLEAN,
    `online` BOOLEAN,
    `name` STRING,
    `level` INT,
    `class` STRING,
    `id` STRING,
    `experience` DOUBLE,
    `account` STRING,
    `challenges` INT,
    `twitch` STRING,
    `ladder` STRING)
  STORED AS PARQUET;

////////////////////////////////NEXT STEP/////////////////////////////////////
////////////////////////////////////////////////////////////////////////////////

 // Copy data to Parquet table
INSERT INTO TABLE hive_task_db.parquet_tb SELECT DISTINCT * FROM
hive_task_db.task_tb;
```

# 3) Reading Dataset :

**Method 1 :** Using spark.read.csv to read data ( ignore step 2 (Preparing Dataset except Create Parquet table part)) :

```
val spark = SparkSession
      .builder
      .appName("Hive task")
      .master("local[4]")
      .config("spark.sql.warehouse.dir","thrift://quickstart.cloudera:9083")
      .enableHiveSupport()
      .getOrCreate()
val customSchema = StructType(Array(
      StructField("rank", IntegerType, true),
       StructField("dead", BooleanType, true),
       StructField("online", BooleanType, true),
       StructField("name", StringType, true),
       StructField("level", IntegerType, true),
       StructField("class", StringType, true),
       StructField("id", StringType, true),
       StructField("experience", DoubleType, true),
       StructField("account", StringType, true),
       StructField("challenges", IntegerType, true),
       StructField("twitch", StringType, true),
       StructField("ladder", StringType, true)))
 val people = spark.read.option("header", "true").option("nullValue",
"null").schema(customSchema).csv("/tmp/idea-IU-183.6156.11/poe_stats.csv")
 // to solve overwrite mode issue
 spark.conf.set("spark.sql.legacy.allowCreatingManagedTableUsingNonemptyLocation","
true")
 people.write.mode("overwrite").saveAsTable("hive_task_db.parquet_tb")
 people = spark.read.table("hive_task_db.parquet_tb")
 people.createOrReplaceTempView("people_tb")
```

**Method 2 :** When data is read in Hue, impala query UI :

```
1. val spark = SparkSession
2.       .builder
3.       .appName("Hive task")
4.       .master("local[4]") /* running spark locally with 4 cores */
5.       /* connection to running Hive server metastore in cloudera */
6.       .config("spark.sql.warehouse.dir","thrift://quickstart.cloudera:9083")
7.       .enableHiveSupport()
8.       .getOrCreate()
9.
10.  people = spark.read.table("hive_task_db.parquet_tb")
11.  people.createOrReplaceTempView("people_tb")
```

# 4) Analysis and Visualization :

## - Count number of player in distinct ladder and sort :

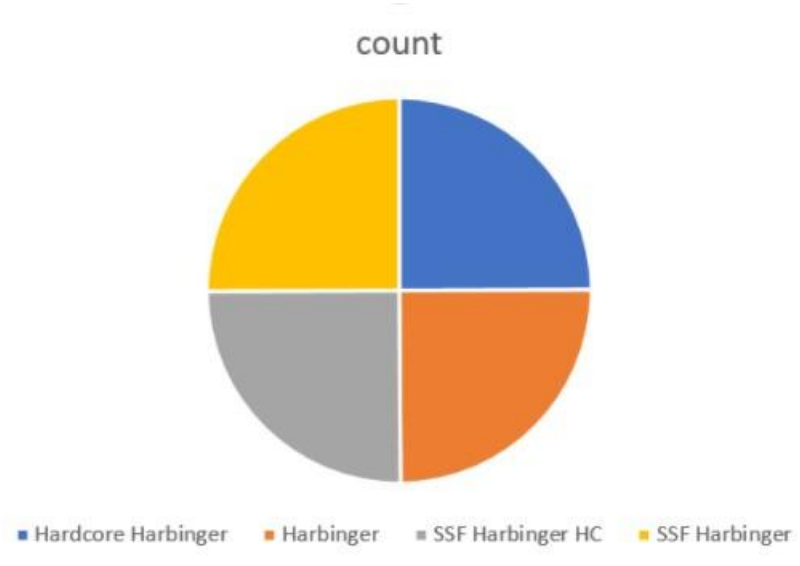
Code :

```
1. spark.sql("select ladder, count(*) As count From people_tb group by ladder Order by
   count").show()
```

Output :

```
+------------------+-----+
|            ladder|count|
+------------------+-----+
|Hardcore Harbinger|14905|
|         Harbinger|14918|
|  SSF Harbinger HC|14972|
|     SSF Harbinger|14981|
+------------------+-----+
```

Pie Chart :



## - Select the name and challenges columns where challenges > 35 :

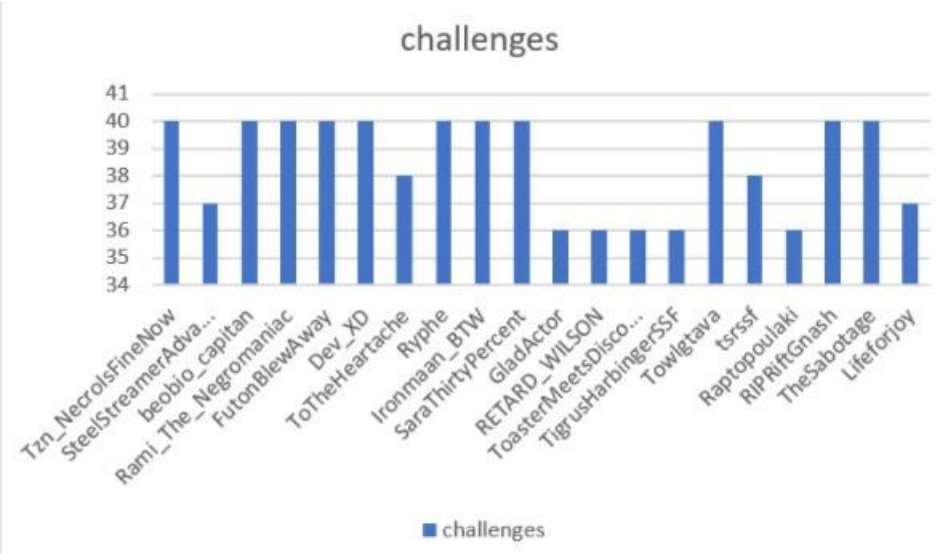
Code :

```
1. spark.sql("select name, challenges From people_tb where challenges > 35").show()
```

Output :

```
+--------------------+----------+
|                name|challenges|
+--------------------+----------+
|  Tzn_NecroIsFineNow|        40|
|SteelStreamerAdva...|        37|
|      beobio_capitan|        40|
|Rami_The_Negromaniac|        40|
|       FutonBlewAway|        40|
|              Dev_XD|        40|
|      ToTheHeartache|        38|
|              Ryphe|        40|
|        Ironmaan_BTW|        40|
|   SaraThirtyPercent|        40|
|           GladActor|        36|
|       RETARD_WILSON|        36|
|ToasterMeetsDisco...|        36|
|  TigrusHarbingerSSF|        36|
|           Towlgtava|        40|
|              tsrssf|        38|
|        Raptopoulaki|        36|
|        RIPRiftGnash|        40|
|         TheSabotage|        40|
|          Lifeforjoy|        37|
+--------------------+----------+
only showing top 20 rows
```

Bar Chart :



- Get classes name and count for the highest 5 in ladder = Harbinger and sort descending :

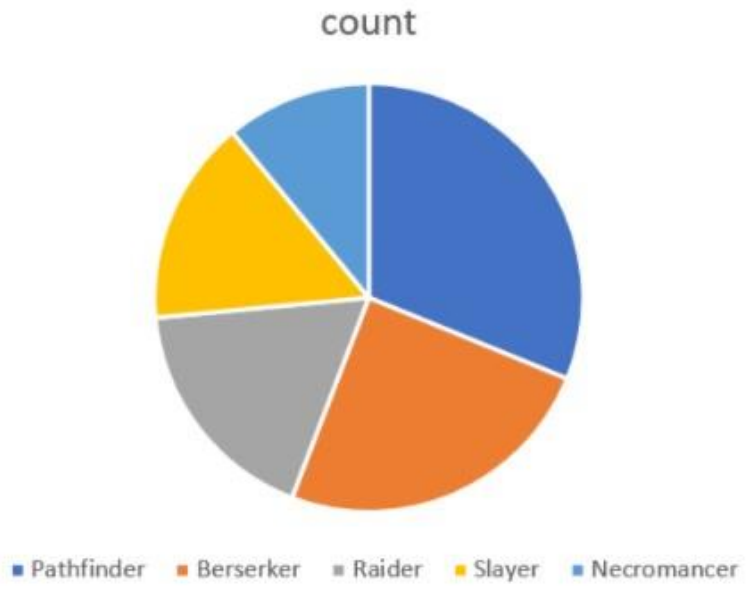
Code :

```
1. spark.sql("select class , count(*) as count from people_tb where ladder ==
   'Harbinger' group by class order by count desc ").show(5)
```

Output :

```
+-----------+-----+
|    class_p|count|
+-----------+-----+
| Pathfinder| 3428|
|  Berserker| 2713|
|     Raider| 1943|
|     Slayer| 1706|
|Necromancer| 1207|
+-----------+-----+
only showing top 5 rows
```

Pie Chart :



## - Count number of players dead :

Code :

```
1. spark.sql("select dead , count(*) as count from people_tb group by dead").show()
```

Output :

```
+-----+-----+
| dead|count|
+-----+-----+
| true|20581|
|false|39195|
+-----+-----+
```

Pie Chart :

## - Count number of finished challenges for each division :

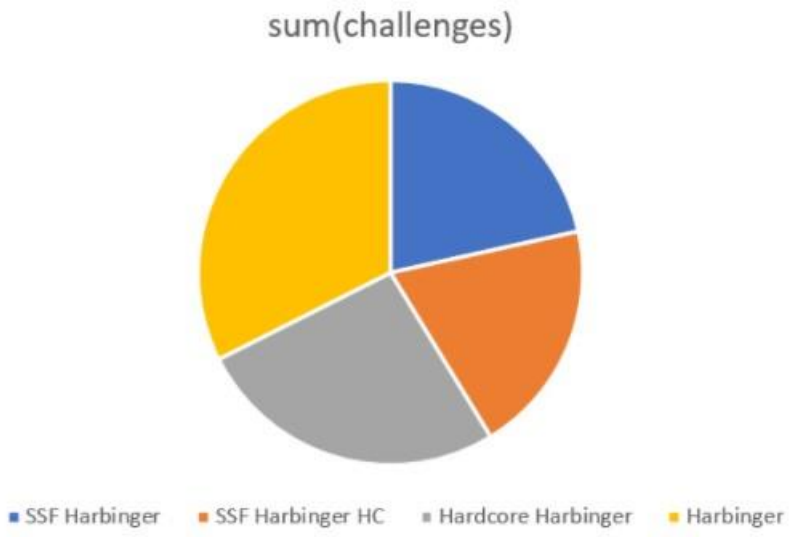
Code :

```
1. spark.sql("select ladder,sum(challenges) From people_tb group by ladder").show()
```

Output :

```
+------------------+---------------+
|            ladder|sum(challenges)|
+------------------+---------------+
|     SSF Harbinger|         331780|
|  SSF Harbinger HC|         303737|
|Hardcore Harbinger|         404949|
|         Harbinger|         499338|
+------------------+---------------+
```

Pie Chart :



- ## Show dependency between level and class of died characters. Only for SSF Harbinger HC divisions :

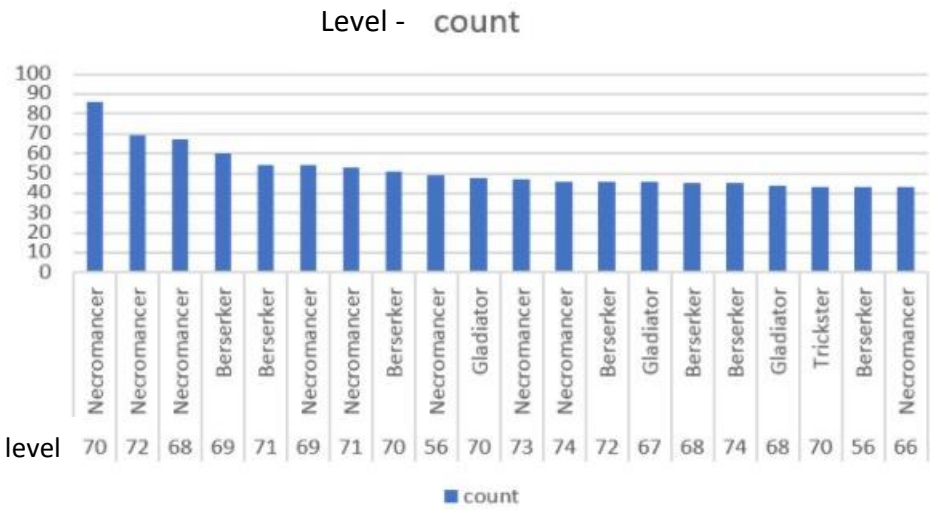
Code :

```
1. spark.sql("select level, class ,count(*) as count from people_tb where
   ladder == 'SSF Harbinger HC' and dead == true group by level, class order by count
   desc").show()
```

Output :

```
+-----+-----------+-----+
|level|    class_p|count|
+-----+-----------+-----+
|   70|Necromancer|   86|
|   72|Necromancer|   69|
|   68|Necromancer|   67|
|   69|  Berserker|   60|
|   71|  Berserker|   54|
|   69|Necromancer|   54|
|   71|Necromancer|   53|
|   70|  Berserker|   51|
|   56|Necromancer|   49|
|   70|  Gladiator|   48|
|   73|Necromancer|   47|
|   74|Necromancer|   46|
|   72|  Berserker|   46|
|   67|  Gladiator|   46|
|   68|  Berserker|   45|
|   74|  Berserker|   45|
|   68|  Gladiator|   44|
|   70|  Trickster|   43|
|   56|  Berserker|   43|
|   66|Necromancer|   43|
+-----+-----------+-----+
only showing top 20 rows
```

Bar Chart :



- Select the top 10 players they have the largest experience, twitch != null and ladder = SSF Harbinger :

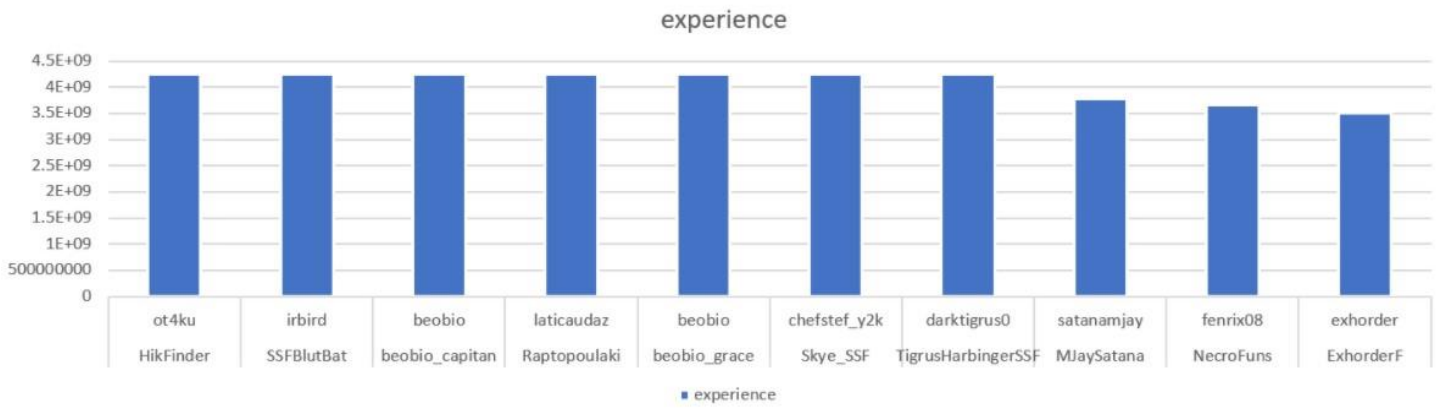
Code :

```
1. spark.sql("select name, twitch, experience From people_tb where ladder = 'SSF
   Harbinger' and twitch is not null Order by experience Desc ").show(10)
```

Output :

```
+------------------+------------+-------------+
|              name|      twitch|   experience|
+------------------+------------+-------------+
|        SSFBlutBat|      irbird|4.250334444E9|
|         HikFinder|       ot4ku|4.250334444E9|
|    beobio_capitan|      beobio|4.250334444E9|
|      beobio_grace|      beobio|4.250334444E9|
|      Raptopoulaki|  laticaudaz|4.250334444E9|
|          Skye_SSF|chefstef_y2k|4.250334444E9|
|TigrusHarbingerSSF| darktigrus0|4.250334444E9|
|        MJaySatana|  satanamjay|3.791654995E9|
|         NecroFuns|    fenrix08|3.660936043E9|
|         ExhorderF|    exhorder|3.511963871E9|
+------------------+------------+-------------+
only showing top 10 rows
```

Bar Chart :

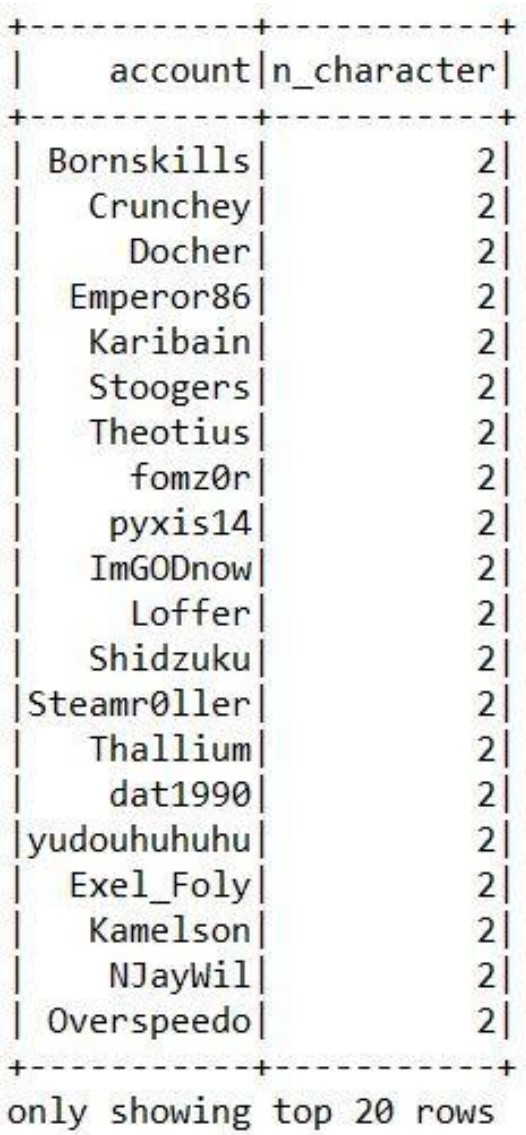


- Number of players with at least 1 character in 2 or more divisions :

Code :

```
1. spark.sql("select * from (select DISTINCT account, Count(count) over
   (partition by account) as n_character from (select account, ladder,
   count(*) as count from people_tb group by account, ladder)) where
   n_character > 1").show()
```

Output :



```
+-----------+-----------+
|    account|n_character|
+-----------+-----------+
| Bornskills|          2|
|   Crunchey|          2|
|     Docher|          2|
|  Emperor86|          2|
|   Karibain|          2|
|   Stoogers|          2|
|   Theotius|          2|
|     fomz0r|          2|
|    pyxis14|          2|
|   ImGODnow|          2|
|     Loffer|          2|
|   Shidzuku|          2|
|Steamr0ller|          2|
|   Thallium|          2|
|    dat1990|          2|
|yudouhuhuhu|          2|
|  Exel_Foly|          2|
|   Kamelson|          2|
|    NJayWil|          2|
| Overspeedo|          2|
+-----------+-----------+
only showing top 20 rows
```

## - Top 3 ranked players per class per level :

Code :

```
1. spark.sql("select * from (select name, rank, class, level, row_number() over
   (partition by class, level order by rank) as row_number from people_tb) ranks
   where row_number <= 3").show()
```

Output :

```
+--------------------+-----+----------+-----+----------+
|                name| rank|     class|level|row_number|
+--------------------+-----+----------+-----+----------+
|     FAIL_OF_ORIHATS| 2356| Berserker|   84|         1|
|   Unknown_Pleasures| 2361| Berserker|   84|         2|
|        Okeledokelie| 2362| Berserker|   84|         3|
|          SSFWarquin| 9610| Chieftain|   67|         1|
|             Anthrok| 9685| Chieftain|   67|         2|
|     SecondTimeSSFhc| 9695| Chieftain|   67|         3|
|    xxWolhaiksongSSF|14790| Gladiator|   53|         1|
|HailTheVictorious...|14811| Gladiator|   53|         2|
|     Cruos_Earthrend|14813| Gladiator|   53|         3|
|      PantslessBlast|14309|Hierophant|   54|         1|
|            Litusant|14337|Hierophant|   54|         2|
|      Solo_Self_Suck|14357|Hierophant|   54|         3|
|            КЙХъалко| 3977|Juggernaut|   79|         1|
|      MrUnbreakableX| 3981|Juggernaut|   79|         2|
|         solororbust| 3985|Juggernaut|   79|         3|
|         YzoofMaster| 7811| Occultist|   70|         1|
|Essence_Drain_Wen...| 7813| Occultist|   70|         2|
|            NauseaIV| 7864| Occultist|   70|         3|
|              Pardwn| 8386|    Raider|   69|         1|
|        SexistRanger| 8395|    Raider|   69|         2|
+--------------------+-----+----------+-----+----------+
only showing top 20 rows
```

## - Top 3 ranked players per division and are not twitch streamers :

Code :

```
1. spark.sql("select * from (select name, rank, ladder, row_number() over
   (partition by ladder order by rank) as row_number from people_tb where twitch
   is null) ranks where row_number <= 3").show()
```

Output :

Top 3 ranked players per division and are not twitch streamers

| name | rank | ladder | row_number |
|---|---|---|---|
| ChiroxPrime | 1 | SSF Harbinger | 1 |
| FutonBlewAway | 3 | SSF Harbinger | 2 |
| Kinther | 4 | SSF Harbinger | 3 |
| ShiruHBSSF | 3 | SSF Harbinger HC | 1 |
| Dev_XD | 4 | SSF Harbinger HC | 2 |
| midget_SPINNER_ | 7 | SSF Harbinger HC | 3 |
| Grosseplotee | 16 | Hardcore Harbinger | 1 |
| Rolllfer | 24 | Hardcore Harbinger | 2 |
| LauraPalmerRESURR... | 27 | Hardcore Harbinger | 3 |
| Cool_NecroIsFineNow | 2 | Harbinger | 1 |
| MISTER_EXECUTION_ | 11 | Harbinger | 2 |
| CaveReTienHarb | 12 | Harbinger | 3 |

- ## Top 3 class in death count for max leveled characters for each division :

Code :

```
1. spark.sql("select * from (select ladder, class, count , row_number() over
   (partition by ladder order by count DESC) as row_number from (select ladder, level,
   class, count(*) as count from people_tb where dead == true and   level == 100 group
   by ladder, class, level))ranks where row_number <= 3").show()
```

Output :

```
+------------------+-----------+-----+----------+
|            ladder|      class|count|row_number|
+------------------+-----------+-----+----------+
|  SSF Harbinger HC|Necromancer|    1|         1|
|  SSF Harbinger HC|     Raider|    1|         2|
|Hardcore Harbinger|   Guardian|    5|         1|
|Hardcore Harbinger|  Gladiator|    4|         2|
|Hardcore Harbinger|Necromancer|    3|         3|
+------------------+-----------+-----+----------+
```

- output shows there are two divisions only have players dead in dataset and in “SSF Harbinger HC” there are two only players have level 100 and dead.

## 5) Difference between the CSV file approach and the Hive Parquet file approach :

**Two approach have the same results and we can read into a Dataframe.**

**CSV file :** row based storage format

**Hive Parquet file :** column based storage format, query language built-in, accessing data faster than csv file, used with large data.

**In code :** When I use hive parquet table there are no changes in code except how we write data in table and how we read , there are no changes in Dataframe and how we deal with it.

## Task 1 full code Link :

- https://drive.google.com/file/d/1mQrcjq2A2_31p1G-EyP07shSFYIuZjzq/view?usp=sharing

## Task 2 full code Link :

- https://drive.google.com/file/d/1CEjjuk5fWt51GcNZVy33Gc2ZrzXRSN6f/view?usp=sharing