## Health Fitness Data:

```
from pyspark.sql import SparkSession

from pyspark.sql.functions import col

from pyspark.sql import functions as F


spark = SparkSession.builder.appName("HealthFitness").getOrCreate()


health_df =
spark.read.format("csv").option("header","true").option("inferSchema","true")
.load("/content/sample_data/health_fitness.csv")
```

**# 1. Find the Total Steps Taken by Each User**

```
steps_user_df =
health_df.groupBy("user_id").agg(F.sum("steps").alias("total_steps"))

steps_user_df.show()
```

**# 2. Filter Days with More Than 10,000 Steps**

```
high_steps = health_df.filter(col("steps")>10000).select("date", "user_id")

high_steps.show()
```

**# 3. Calculate the Average Calories Burned by Workout Type**

```
avg_calories =
health_df.groupBy("workout_type").agg(F.avg("calories_burned").alias("avg_c
alories"))
```

```
avg_calories.show()
```

**# 4. Identify the Day with the Most Steps for Each User**

```
max_user_steps =
health_df.groupBy("user_id").agg(F.max("steps").alias("max_steps"))


max_user_steps_dates =
max_user_steps.join(health_df,"user_id").filter(col("steps")==col("max_steps")
).select("user_id","date","steps")
max_user_steps_dates.show()
```

**# 5. Find Users Who Burned More Than 600 Calories on Any Day**

```
high_calories_burned = health_df.filter(col("calories_burned") > 600)
high_calories_burned.show()
```

**# 6. Calculate the Average Hours of Sleep per User**

```
avg_sleep_hrs =
health_df.groupBy("user_id").agg(F.avg("hours_of_sleep").alias("avg_sleep"))
avg_sleep_hrs.show()
```

**# 7. Find the Total Calories Burned per Day**

```
calories_burned_perday =
health_df.groupBy("date").agg(F.sum("calories_burned").alias("total_calories")
)
```

```
calories_burned_perday.show()
```

**# 8. Identify Users Who Did Different Types of Workouts**

```
different_workout = health_df.groupBy("user_id").agg(F.countDistinct("workout_type").alias("workout_types")).filter(col("workout_types") > 1)

different_workout.show()
```

**# 9. Calculate the Total Number of Workouts per User**

```
total_workout_peruser = health_df.groupBy("user_id").agg(F.count("workout_type").alias("total_workouts"))

total_workout_peruser.show()
```

**# 10. Create a New Column for "Active" Days**

```
health_df = health_df.withColumn("active_day", F.when(col("steps") > 10000, "Active").otherwise("Inactive"))

health_df.show()
```