

Federico Sabaj 261773

```
[1]: from pyspark.sql import SparkSession

from pyspark.sql.functions import *
from pyspark.sql.types import *

spark = SparkSession \
    .builder \
    .appName("Obligatorio") \
    .getOrCreate()

Setting default log level to "WARN".
To adjust logging level use sc.setLogLevel(newLevel). For SparkR, use setLogLevel(newLevel).
2024-11-30T23:57:09,715 WARN [Thread-4] org.apache.hadoop.util.NativeCodeLoader - Unable to load native-hadoop library for your platform... u
sing builtin-java classes where applicable

[2]: athletes = spark.read.csv("/user/ort/obligatorio/tables/athletes", header=True, inferSchema=True)
disciplines = spark.read.csv("/user/ort/obligatorio/tables/disciplines", header=True, inferSchema=True)
events = spark.read.csv("/user/ort/obligatorio/tables/events", header=True, inferSchema=True)
participations = spark.read.csv("/user/ort/obligatorio/tables/participations", header=True, inferSchema=True)
awards = spark.read.csv("/user/ort/obligatorio/tables/awards", header=True, inferSchema=True)
```

Tabla: Athletes

La tabla de atletas (Athletes) está compuesta por:

- athlete_id: string (PK) - ID del atleta dentro de la tabla
- name: string - Nombre del atleta
- country: string - País del atleta
- date_of_birth: string - Fecha de nacimiento del atleta en formato dd/mm/yyyy

```
[3]: athletes.describe()

[3]: DataFrame[summary: string, athlete_id: string, name: string, country: string, date_of_birth: string]

[4]: athletes.show(10)

+-----+-----+-----+-----+
|athlete_id|  name|country|date_of_birth|
+-----+-----+-----+-----+
|         1|   Brod|   Peru|    04/03/2005|
|         2| Adelle|  Brazil|    22/03/2003|
|         3|Clifford| Mexico|    08/08/2003|
|         4| Nadine|Colombia|   22/03/2006|
|         5|  Naoma|Paraguay|   30/09/1989|
|         6| Bondie|Colombia|   12/05/1990|
|         7|  Neale| Mexico|    05/02/2015|
|         8| Annice|  Brazil|   10/07/1988|
|         9|  Elisa|Colombia|   31/08/2024|
|        10| Krysta| Mexico|    01/09/1998|
+-----+-----+-----+-----+

only showing top 10 rows
```

Nos aseguraremos que las claves primarias (athlete_id) sean únicas de la siguiente manera:

```
[5]: amount = athletes.select('athlete_id').count()
distinct_amount = athletes.select('athlete_id').distinct().count()

print("Amount of athletes: ", amount)
print("Amount of distinct athletes: ", distinct_amount)

Amount of athletes: 200
Amount of distinct athletes: 200

Dado que ambas tablas tienen la misma cantidad de datos, corroboramos que las claves son únicas.


Ahora, modificaremos el formato de la fecha en date_of_birth para que este en formato ISO 8601, de tal manera que se estándar y más fácil manejarlas más adelante.


```

```
[6]: athletes = athletes.withColumn("date_aux", to_date(athletes["date_of_birth"], "dd/MM/yyyy"))

athletes = athletes.select('athlete_id', 'name', 'country', 'date_aux')
athletes = athletes.withColumnRenamed('date_aux', 'date_of_birth')

athletes.show(10)

+-----+-----+-----+-----+
|athlete_id|  name|country|date_of_birth|
+-----+-----+-----+-----+
|         1|   Brod|   Peru| 2005-03-04|
|         2| Adelle|  Brazil| 2003-03-22|
|         3|Clifford| Mexico| 2003-08-08|
|         4| Nadine|Colombia| 2006-03-22|
|         5|  Naoma|Paraguay| 1989-09-30|
|         6| Bondie|Colombia| 1990-05-12|
|         7|  Neale| Mexico| 2015-02-05|
|         8| Annice|  Brazil| 1988-07-10|
|         9|  Elisa|Colombia| 2024-08-31|
|        10| Krysta| Mexico| 1998-09-01|
+-----+-----+-----+-----+

only showing top 10 rows
```

Tabla: Disciplines

La tabla de disciplinas (Disciplines) está compuesta por:

- discipline_id: string (PK) - ID de la disciplina dentro de la tabla
- name: string - Nombre de la disciplina
- description: string - Breve descripción de la disciplina

```
[7]: disciplines.describe()

[7]: DataFrame[summary: string, discipline_id: string, name: string, description: string]

[8]: disciplines.show(10)

+-----+-----+-----+
|discipline_id|  name|description|
+-----+-----+-----+
|         1| football|A team sport play...|
|         2|basketball|A fast-paced game...|
|         3|  swimming|An individual or ...|
|         4|volleyball|A team sport play...|
|         5|   tennis|A racket sport pl...|
|         6|gymnastics|A discipline invo...|
|         7|  cycling|A competitive spo...|
|         8|   boxing|A combat sport wh...|
|         9|   golf|A precision sport...|
|        10|   rugby|A physical team s...|
+-----+-----+-----+

only showing top 10 rows
```

Nos aseguraremos que las claves primarias (discipline_id) sean únicas de la siguiente manera:

```
[9]: amount_disciplines = disciplines.select('discipline_id').count()
distinct_amount_disciplines = disciplines.select('discipline_id').distinct().count()

print("Amount of disciplines: ", amount_disciplines)
print("Amount of distinct disciplines: ", distinct_amount_disciplines)

Amount of disciplines: 10
Amount of distinct disciplines: 10

Dado que ambas tablas tienen la misma cantidad de datos, corroboramos que las claves son únicas
```

Tabla: Events

La tabla de eventos (Events) está compuesta por:

- event_id: string (PK) - ID del evento dentro de la tabla
- discipline_id: string (FK Disciplines) - ID de la disciplina para la cual tomará lugar el evento
- name: string - Nombre del evento
- date: string - Fecha en la que tomará lugar el evento en formato dd/mm/yyyy

```
[10]: events.describe()

[10]: DataFrame[summary: string, event_id: string, discipline_id: string, name: string, date: string]

[11]: events.show(10)

+-----+-----+-----+-----+
|event_id|discipline_id|  name|date|
+-----+-----+-----+-----+
|         1|         10|Midnight Madness|13/01/2024|
|         2|         2|Victory Celebration|25/03/2024|
|         3|         6|Champion's Challenge|03/04/2024|
|         4|         10|Golden Glory|20/05/2024|
|         5|         3|Epic Showdown|28/06/2024|
|         6|         5|Legends Unite|19/01/2024|
|         7|         5|Rising Stars Show...|21/06/2024|
|         8|         7|Ultimate Triumph|07/11/2024|
|         9|         9|Majestic Clash|24/11/2023|
|        10|         6|Dream Team Duel|06/01/2024|
+-----+-----+-----+-----+

only showing top 10 rows
```

Nos aseguraremos que las claves primarias (event_id) sean únicas de la siguiente manera:

```
[12]: amount_events = events.select('event_id').count()
distinct_amount_events = events.select('event_id').distinct().count()

print("Amount of events: ", amount_events)
print("Amount of distinct events: ", distinct_amount_events)

Amount of events: 100
Amount of distinct events: 100

Dado que ambas tablas tienen la misma cantidad de datos, corroboramos que las claves son únicas.


Al igual que hicimos con Athletes, modificaremos el formato de la fecha en date para que este en formato ISO 8601, de tal manera que se estándar y más fácil manejarlas más adelante.


```

```
[13]: events = events.withColumn("date_aux", to_date(events["date"], "dd/MM/yyyy"))

events = events.select('event_id', 'discipline_id', 'name', 'date_aux')
events = events.withColumnRenamed('date_aux', 'date')

events.show(10)

+-----+-----+-----+-----+
|event_id|discipline_id|  name|date|
+-----+-----+-----+-----+
|         1|         10|Midnight Madness|2024-01-13|
|         2|         2|Victory Celebration|2024-03-25|
|         3|         6|Champion's Challenge|2024-04-03|
|         4|         10|Golden Glory|2024-05-20|
|         5|         3|Epic Showdown|2024-06-28|
|         6|         5|Legends Unite|2024-01-19|
|         7|         5|Rising Stars Show...|2024-06-21|
|         8|         7|Ultimate Triumph|2024-11-07|
|         9|         9|Majestic Clash|2023-11-24|
|        10|         6|Dream Team Duel|2024-01-06|
+-----+-----+-----+-----+

only showing top 10 rows
```

Tabla: Participations

La tabla de participaciones (Participations) está compuesta por:

- participation_id: string (PK) - ID de la participación dentro de la tabla
- event_id: string (FK Events) - ID del evento para el cual el atleta participó
- athlete_id: string (FK Athletes) - ID del atleta que participó en el evento
- score: string - Puntaje que se sacó el atleta en la participación del evento. Es un número del 1 al 100, donde 100 es el máximo y 1 el mínimo

```
[14]: participations.describe()

[14]: DataFrame[summary: string, participation_id: string, event_id: string, athlete_id: string, score: string]

[15]: participations.show(10)

+-----+-----+-----+-----+
|participation_id|event_id|athlete_id|score|
+-----+-----+-----+-----+
|         1|         1|         41|    14|
|         2|         2|         78|    81|
|         3|         3|         87|     5|
|         4|         4|         94|    48|
|         5|         6|         62|    82|
|         6|         6|         33|    30|
|         7|         7|         48|    44|
|         8|         8|         88|    72|
|         9|         8|         89|    48|
|        10|        10|         44|    39|
+-----+-----+-----+-----+

only showing top 10 rows
```

Nos aseguraremos que las claves primarias (participation_id) sean únicas de la siguiente manera:

```
[16]: amount_participations = participations.select('participation_id').count()
distinct_amount_participations = participations.select('participation_id').distinct().count()

print("Amount of participations: ", amount_participations)
print("Amount of distinct participations: ", distinct_amount_participations)

Amount of participations: 300
Amount of distinct participations: 300

Dado que ambas tablas tienen la misma cantidad de datos, corroboramos que las claves son únicas
```

Tabla: Awards

La tabla de premios (Awards) está compuesta por:

- award_id: string (PK) - ID del premio dentro de la tabla
- participation_id: string (FK Participations) - ID de la participación que se ganó el premio
- medal: string - Medalla obtenida. Puede ser "Gold", "Silver" o "Bronze"
- award_date: string - Fecha que se obtuvo el premio en formato dd/mm/yyyy

```
[17]: awards.describe()

[17]: DataFrame[summary: string, award_id: string, participation_id: string, medal: string, award_date: string]

[18]: awards.show(10)

+-----+-----+-----+-----+
|award_id|participation_id| medal|award_date|
+-----+-----+-----+-----+
|         1|         122|   Gold|24/05/2024|
|         2|         220| Silver|19/11/2023|
|         3|         170| Bronze|09/03/2024|
|         4|         250|   Gold|15/10/2024|
|         5|         270| Silver|07/06/2024|
|         6|         146| Bronze|21/11/2023|
|         7|         162|   Gold|16/06/2024|
|         8|         295| Silver|24/07/2024|
|         9|         147| Bronze|28/05/2024|
|        10|         251|   Gold|02/12/2023|
+-----+-----+-----+-----+

only showing top 10 rows
```

Nos aseguraremos que las claves primarias (award_id) sean únicas de la siguiente manera:

```
[19]: amount_awards = awards.select('award_id').count()
distinct_amount_awards = awards.select('award_id').distinct().count()

print("Amount of awards: ", amount_awards)
print("Amount of distinct awards: ", distinct_amount_awards)

Amount of awards: 100
Amount of distinct awards: 100

Dado que ambas tablas tienen la misma cantidad de datos, corroboramos que las claves son únicas.
```

Finalmente, modificaremos el formato de la fecha en award_date para que este en formato ISO 8601, de tal manera que se estándar y más fácil manejarlas más adelante.

```
[20]: awards = awards.withColumn("date_aux", to_date(awards["award_date"], "dd/MM/yyyy"))

awards = awards.select('award_id', 'participation_id', 'medal', 'date_aux')
awards = awards.withColumnRenamed('date_aux', 'award_date')

awards.show(10)

+-----+-----+-----+-----+
|award_id|participation_id| medal|award_date|
+-----+-----+-----+-----+
|         1|         122|   Gold|2024-05-24|
|         2|         220| Silver|2023-11-19|
|         3|         170| Bronze|2024-03-09|
|         4|         250|   Gold|2024-10-15|
|         5|         270| Silver|2024-06-07|
|         6|         146| Bronze|2023-11-21|
|         7|         162|   Gold|2024-06-16|
|         8|         295| Silver|2024-07-24|
|         9|         147| Bronze|2024-05-28|
|        10|         251|   Gold|2023-12-02|
+-----+-----+-----+-----+

only showing top 10 rows
```

Persistimos los datos

Persistimos las tablas refinadas bajo las carpetas correspondientes en el directorio /user/ort/obligatorio/refined_tables

```
[21]: athletes.coalesce(1).write.csv('/user/ort/obligatorio/refined_tables/athletes', header=True, mode='overwrite')
disciplines.coalesce(1).write.csv('/user/ort/obligatorio/refined_tables/disciplines', header=True, mode='overwrite')
events.coalesce(1).write.csv('/user/ort/obligatorio/refined_tables/events', header=True, mode='overwrite')
participations.coalesce(1).write.csv('/user/ort/obligatorio/refined_tables/participations', header=True, mode='overwrite')
awards.coalesce(1).write.csv('/user/ort/obligatorio/refined_tables/awards', header=True, mode='overwrite')
```