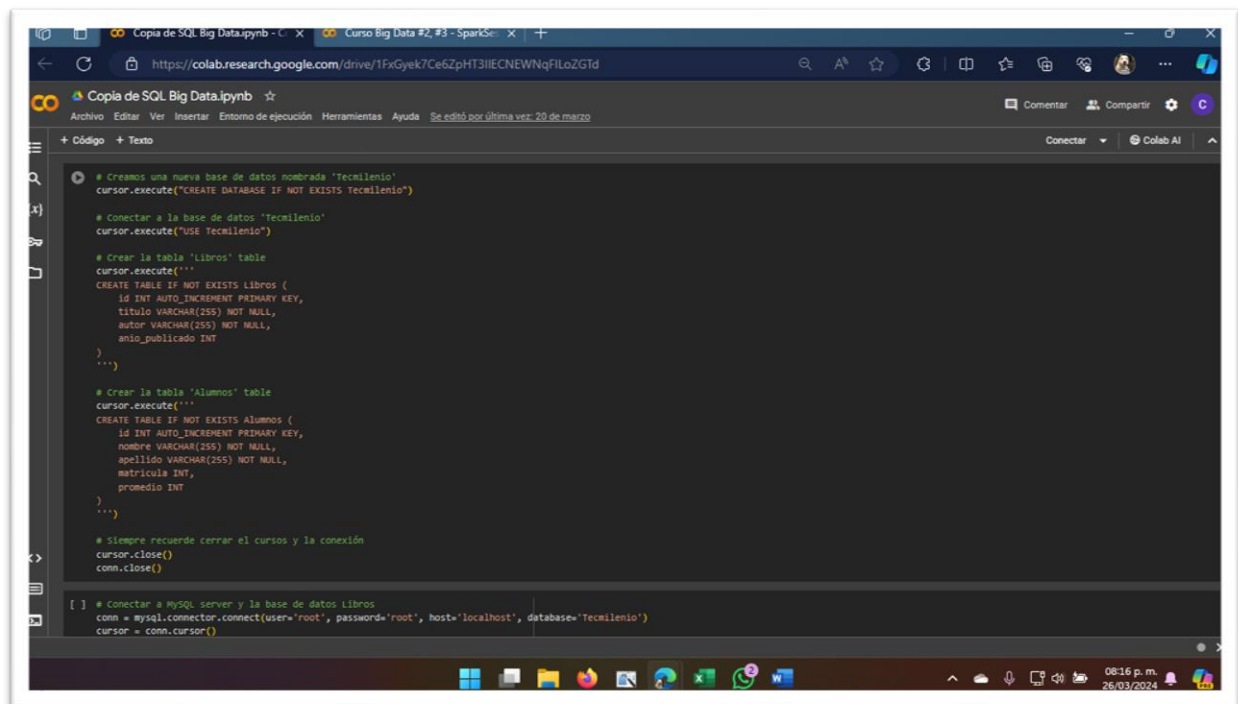


Nombre: Celic Gabriel Hernández Archundia.		Matrícula: 2877240
Nombre del curso: Infraestructura para Big Data		Nombre del profesor: Miguel de Jesús Martínez Felipe
Módulo II		Actividad: Actividad 8
Fecha: 26/03/2024		

Capturas SQL Big Data



```

# Creamos una nueva base de datos nombrada 'Tecmilenio'
cursor.execute("CREATE DATABASE IF NOT EXISTS Tecmilenio")

# Conectar a la base de datos 'Tecmilenio'
cursor.execute("USE Tecmilenio")

# Crear la tabla 'Libros' table
cursor.execute("""
CREATE TABLE IF NOT EXISTS Libros (
  id INT AUTO_INCREMENT PRIMARY KEY,
  titulo VARCHAR(255) NOT NULL,
  autor VARCHAR(255) NOT NULL,
  año_publicado INT
)
""")

# Crear la tabla 'Alumnos' table
cursor.execute("""
CREATE TABLE IF NOT EXISTS Alumnos (
  id INT AUTO_INCREMENT PRIMARY KEY,
  nombre VARCHAR(255) NOT NULL,
  apellido VARCHAR(255) NOT NULL,
  matricula INT,
  promedio INT
)
""")

# Siempre recuerde cerrar el cursor y la conexión
cursor.close()
conn.close()

[ ] # Conectar a MySQL server y la base de datos Libros
conn = mysql.connector.connect(user='root', password='root', host='localhost', database='Tecmilenio')
cursor = conn.cursor()

```

```
Copia de SQL Big Data.ipynb - C x Curso Big Data #2, #3 - SparkSe x +
https://colab.research.google.com/drive/1FxGyek7Ce6ZpHT3IECNEWNqFILOZGTd

Copia de SQL Big Data.ipynb
Archivo Editar Ver Insertar Entorno de ejecución Herramientas Ayuda Se editó por última vez: 20 de marzo

+ Código + Texto
Conectar Colab AI

# Conectar a MySQL server y la base de datos Libros
conn = mysql.connector.connect(user='root', password='root', host='localhost', database='tecnilenio')
cursor = conn.cursor()

books_data = [
    ("To Kill a Mockingbird", "Harper Lee", 1960),
    ("1984", "George Orwell", 1949),
    ("The Great Gatsby", "F. Scott Fitzgerald", 1925)
]

students_data = [
    ("Celio", "Hernández", 2877248, 99),
    ("Diego", "Malerva", 2483827, 97),
    ("Diego", "García", 2987328, 98),
    ("Luis", "Lezama", 2758499, 99),
    ("Karol", "G", 2349832, 100),
    ("Jade", "Voya", 2828128, 95),
    ("Max", "Steel", 2908938, 97),
    ("Jesús", "Atacho", 2877842, 97),
    ("Gabriel", "Uchiha", 2877242, 100),
    ("Mariana", "González", 2887345, 98),
]

# Insertar datos usando el cursor
cursor.executemany('''
INSERT INTO Libros (titulo, autor, año_publicado) VALUES (%s, %s, %s)
''', books_data)

# Insertar datos usando el cursor
cursor.executemany('''
INSERT INTO Alumnos (nombre, apellido, matricula, promedio) VALUES (%s, %s, %s, %s)
''', students_data)

# Commit cambios
conn.commit()
```

```
Copia de SQL Big Data.ipynb - C x Curso Big Data #2, #3 - SparkSe x +
https://colab.research.google.com/drive/1FxGyek7Ce6ZpHT3IECNEWNqFILOZGTd

Copia de SQL Big Data.ipynb
Archivo Editar Ver Insertar Entorno de ejecución Herramientas Ayuda Se editó por última vez: 20 de marzo

+ Código + Texto
Conectar Colab AI

[ ] cursor.close()
[ ] conn.close()

# Conectar a MySQL server y la base de datos Libros
conn = mysql.connector.connect(user='root', password='root', host='localhost', database='tecnilenio')
cursor = conn.cursor()

# Ejecución del query
#cursor.execute("SELECT * FROM Libros")
cursor.execute("SELECT * FROM Alumnos")

# Obtención de resultados
records = cursor.fetchall()

# Imprimir records
for record in records:
    print(record)

# Cerrar el cursos y la conexión
cursor.close()
conn.close()

[ ]
(1, 'Celio', 'Hernández', 2877248, 99)
(2, 'Diego', 'Malerva', 2483827, 97)
(3, 'Diego', 'García', 2987328, 98)
(4, 'Luis', 'Lezama', 2758499, 99)
(5, 'Karol', 'G', 2349832, 100)
(6, 'Jade', 'Voya', 2828128, 95)
(7, 'Max', 'Steel', 2908938, 97)
(8, 'Jesús', 'Atacho', 2877842, 97)
(9, 'Gabriel', 'Uchiha', 2877242, 100)
(10, 'Mariana', 'González', 2887345, 98)

[ ]
# Conectar a MySQL server y la base de datos Libros
```

Copia de SQL Big Data.ipynb

https://colab.research.google.com/drive/1FxGyek7Ce6ZpHT3IIECNEWNqFIoLoZGTd

Archivar Editar Ver Insertar Entorno de ejecución Herramientas Ayuda Se editó por última vez: 20 de marzo

+ Código + Texto Conectar Colab AI

Pandas

```
[ ] import pandas as pd

# Conectar a MySQL
conn = mysql.connector.connect(user='root', password='root', host='localhost', database='Tecmilenio')

# Obtención de los datos a pandas dataframe
query = "SELECT * FROM Alumnos"
df = pd.read_sql(query, conn)

<ipython-input-17-ba3ef991ee5>:8: UserWarning: pandas only supports SQLAlchemy connectable (engine/connection) or database string URI or sqlite3 DBAPI2 connection. Other DBAPI2 objects are not tested. Please
df = pd.read_sql(query, conn)

[ ] df.to_csv('alumnosDatos.csv')
files.download('alumnosDatos.csv')

print(df.head())
```

	id	nombre	apellido	matricula	promedio
0	1	Cellic	Hernández	2877240	99
1	2	Diego	Malerva	2483827	97
2	3	Diego	García	2987328	90
3	4	Luis	Lezama	2758499	99
4	5	Karol	G	2349832	100

```
[ ] import matplotlib.pyplot as plt

# Ploteo del histograma
plt.hist(df['promedio'], bins=10, edgecolor='black')
plt.title('Distribución de promedio')
plt.xlabel('promedio')
```

Copia de SQL Big Data.ipynb

https://colab.research.google.com/drive/1FxGyek7Ce6ZpHT3IIECNEWNqFIoLoZGTd

Archivar Editar Ver Insertar Entorno de ejecución Herramientas Ayuda Se editó por última vez: 20 de marzo

+ Código + Texto Conectar Colab AI

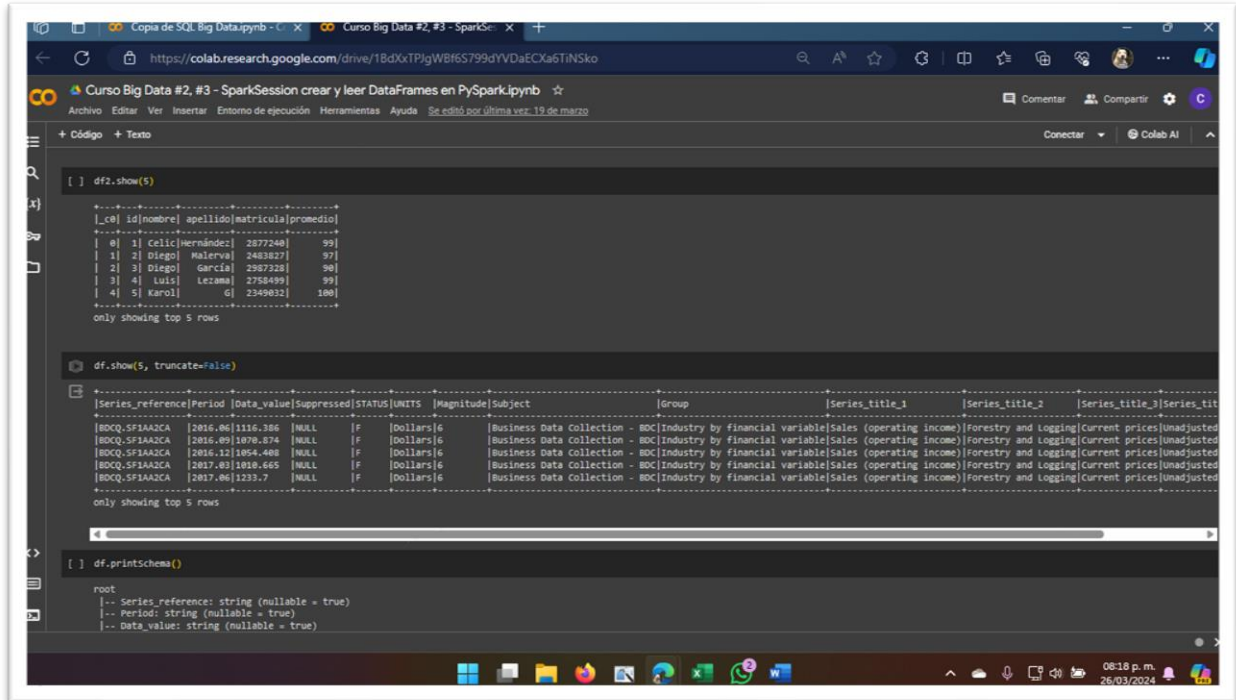
```
3 4 Luis Lezama 2758499 99
4 5 Karol G 2349832 100

import matplotlib.pyplot as plt

# Ploteo del histograma
plt.hist(df['promedio'], bins=10, edgecolor='black')
plt.title('Distribución de promedio')
plt.xlabel('promedio')
plt.ylabel('Número de Alumnos')
plt.show()
```

promedio	Número de Alumnos
90-92	2.0
92-94	0.0
94-96	1.0
96-98	3.0
98-100	4.0

Capturas Curso Big Data



The screenshot shows a Google Colab notebook titled "Curso Big Data #2, #3 - SparkSession crear y leer DataFrames en PySparklymb". The notebook is open to a code cell with the following content:

```
[ ] df2.show(5)
```

The output of the code cell shows the first 5 rows of the DataFrame:

id	nombre	apellido	matricula	promedio	
0	1	celic	hernandez	2877248	99
1	2	Diego	Malerva	2483827	97
2	3	Diego	Garcia	2387320	90
3	4	Luis	Lezama	2758499	99
4	5	Karol		2349832	100

only showing top 5 rows

```
[ ] df.show(5, truncate=False)
```

The output of the code cell shows the first 5 rows of the DataFrame with all columns:

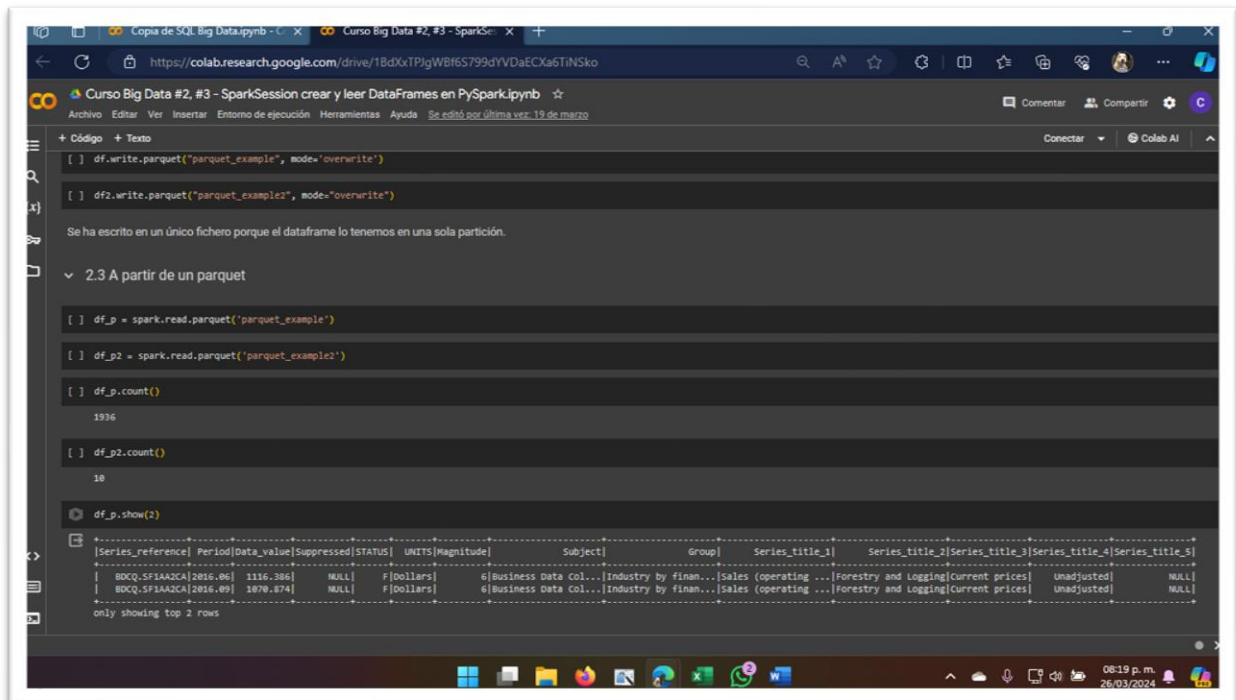
Series_reference	Period	Data_value	Suppressed	STATUS	UNITS	Magnitude	Subject	Group	Series_title_1	Series_title_2	Series_title_3	Series_title_4
BDCQ-SF1AA2CA	2016.06	1116.386	NULL	F	Dollars	6	Business Data Collection - BDC	Industry by financial variable	Sales (operating income)	Forestry and Logging	Current prices	Unadjusted
BDCQ-SF1AA2CA	2016.09	1078.874	NULL	F	Dollars	6	Business Data Collection - BDC	Industry by financial variable	Sales (operating income)	Forestry and Logging	Current prices	Unadjusted
BDCQ-SF1AA2CA	2016.12	1054.488	NULL	F	Dollars	6	Business Data Collection - BDC	Industry by financial variable	Sales (operating income)	Forestry and Logging	Current prices	Unadjusted
BDCQ-SF1AA2CA	2017.03	1018.665	NULL	F	Dollars	6	Business Data Collection - BDC	Industry by financial variable	Sales (operating income)	Forestry and Logging	Current prices	Unadjusted
BDCQ-SF1AA2CA	2017.06	1233.7	NULL	F	Dollars	6	Business Data Collection - BDC	Industry by financial variable	Sales (operating income)	Forestry and Logging	Current prices	Unadjusted

only showing top 5 rows

```
[ ] df.printschema()
```

The output of the code cell shows the schema of the DataFrame:

```
root
 |-- Series_reference: string (nullable = true)
 |-- Period: string (nullable = true)
 |-- Data_value: string (nullable = true)
```



The screenshot shows a Google Colab notebook titled "Curso Big Data #2, #3 - SparkSession crear y leer DataFrames en PySparklymb". The notebook is open to a code cell with the following content:

```
[ ] df.write.parquet("parquet_example", mode="overwrite")
```

```
[ ] df2.write.parquet("parquet_example2", mode="overwrite")
```

Se ha escrito en un único fichero porque el dataframe lo tenemos en una sola partición.

2.3 A partir de un parquet

```
[ ] df_p = spark.read.parquet("parquet_example")
```

```
[ ] df_p2 = spark.read.parquet("parquet_example2")
```

```
[ ] df_p.count()
```

1936

```
[ ] df_p2.count()
```

18

```
[ ] df_p.show(2)
```

The output of the code cell shows the first 2 rows of the DataFrame:

Series_reference	Period	Data_value	Suppressed	STATUS	UNITS	Magnitude	Subject	Group	Series_title_1	Series_title_2	Series_title_3	Series_title_4	Series_title_5
BDCQ-SF1AA2CA	2016.06	1116.386	NULL	F	Dollars	6	Business Data Collection - BDC	Industry by financial variable	Sales (operating income)	Forestry and Logging	Current prices	Unadjusted	NULL
BDCQ-SF1AA2CA	2016.09	1078.874	NULL	F	Dollars	6	Business Data Collection - BDC	Industry by financial variable	Sales (operating income)	Forestry and Logging	Current prices	Unadjusted	NULL

only showing top 2 rows

Curso Big Data #2, #3 - SparkSession crear y leer DataFrames en PySparklymb

Archivo Editar Ver Insertar Entorno de ejecución Herramientas Ayuda Se editó por última vez: 19 de marzo

+ Código + Texto

```
[ ] df_p.describe().show()
```

summary	Series_reference	Period	Data_value	Suppressed	STATUS	UNITS	Magnitude	Subject	Group	Series_title_1	Series_title_2	Series_title_3	Series
count	1936	1936	1936	0	1936	1936	1936	1936	1936	1936	1936	1936	
mean	NULL	2010.217975206615	2704.305560101053	NULL	NULL	NULL	6.0	NULL	NULL	NULL	NULL	NULL	
stddev	NULL	1.3594869192539778	4630.441460220322	NULL	NULL	NULL	0.0	NULL	NULL	NULL	NULL	NULL	
min	BDCQ_SF1AA2CA	2016.06	-398.194	NULL	F	Dollars	6	Business Data Col...	Industry by finan...	Operating profit	Accommodation and...	(Current prices)	Una
max	BDCQ_SF8RS2CA	2020.12	998.124	NULL	R	Dollars	6	Business Data Col...	Industry by finan...	Sales (operating ...)	Wood and Paper Pr...	(Current prices)	Una

df_p2.describe().show()

summary	_c0	id	nombre	apellido	matricula	promedio
count	10	10	10	10	10	10
mean	4.5	5.5	NULL	NULL	2701941.3	96.4
stddev	3.0276583540974917	3.0276583540974917	NULL	NULL	203607.63287321353	3.717824931626317
min	0	1	Cellic	Atacho	2349032	100
max	9	9	Maxi	Yoyo	2907228	99

```
[ ] df_pandas = df_p.toPandas()
```

df_pandas.head()

08:19 p.m.
26/03/2024