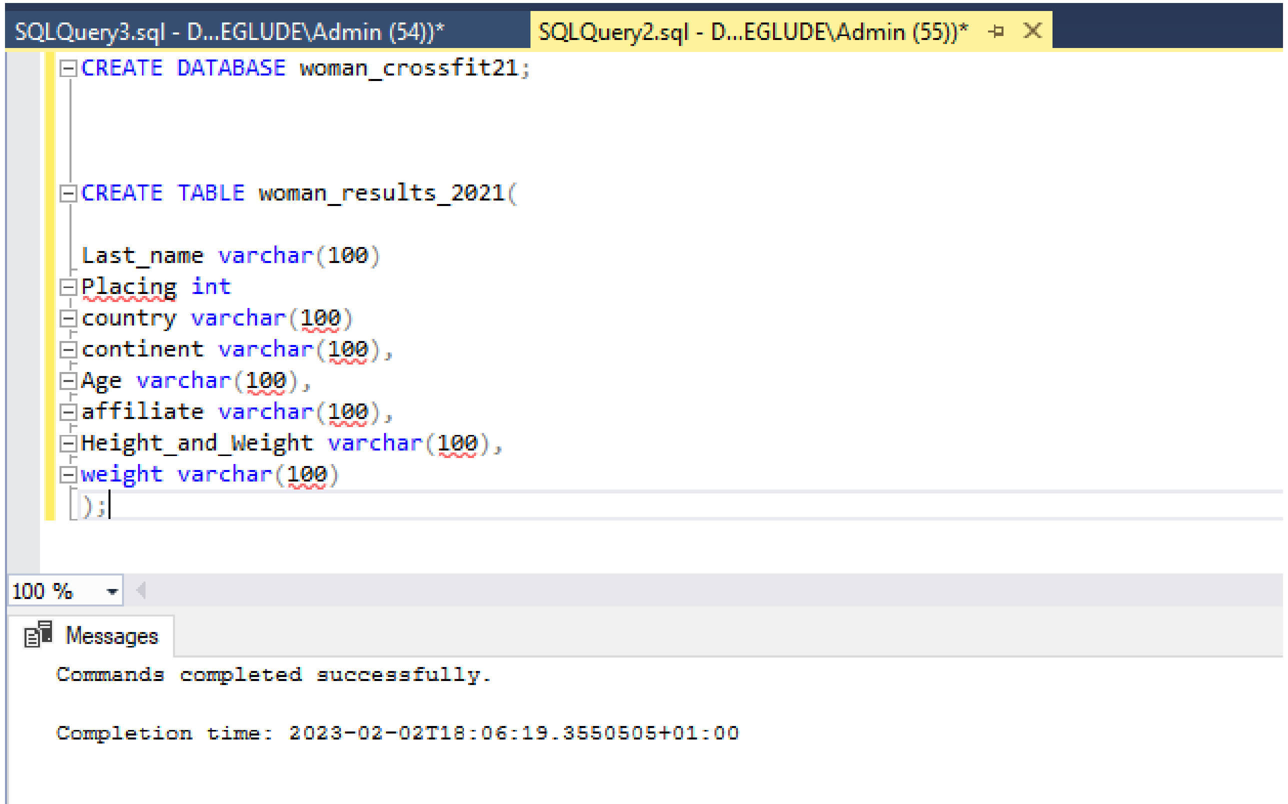
Cleaning Data in SQL:
Women Ranking in Crossfit Games 2021

In order to present this cleaning up job, I am going to use a dataset from the Crossfit games 2021, where we can find the women's ranking details.

- First, let's create the Database and Table

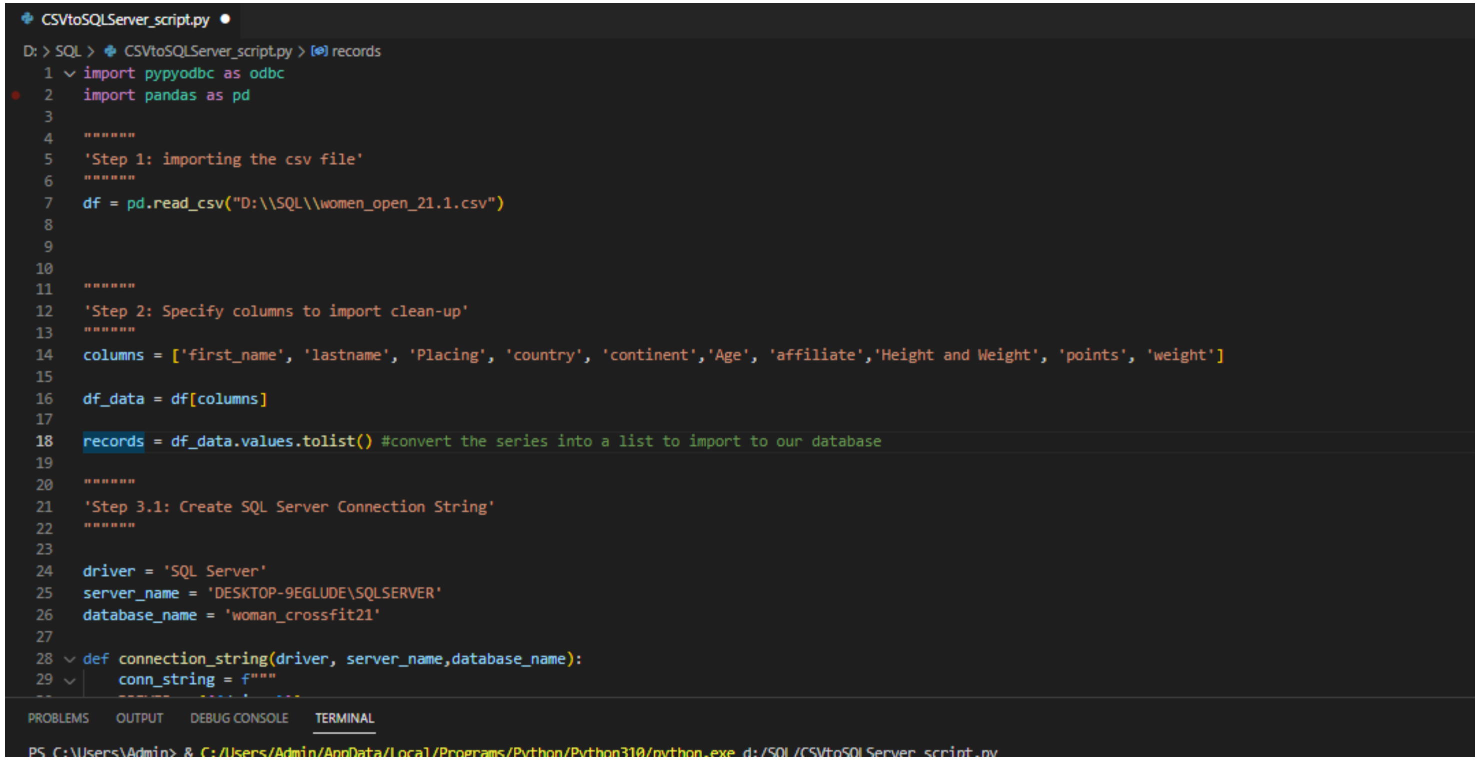

SQLQuery3.sql - D...EGLUDE\Admin (54))*
SQLQuery2.sql - D...EGLUDE\Admin (55))*
100 %
Messages

```sql
CREATE DATABASE woman_crossfit21;



CREATE TABLE woman_results_2021(

Last_name varchar(100)
Placing int
country varchar(100)
continent varchar(100),
Age varchar(100),
affiliate varchar(100),
Height_and_Weight varchar(100),
weight varchar(100)
);
```

```
Commands completed successfully.

Completion time: 2023-02-02T18:06:19.3550505+01:00
```

- # Import the data from .CSV to SQL Server

1) Running a python script

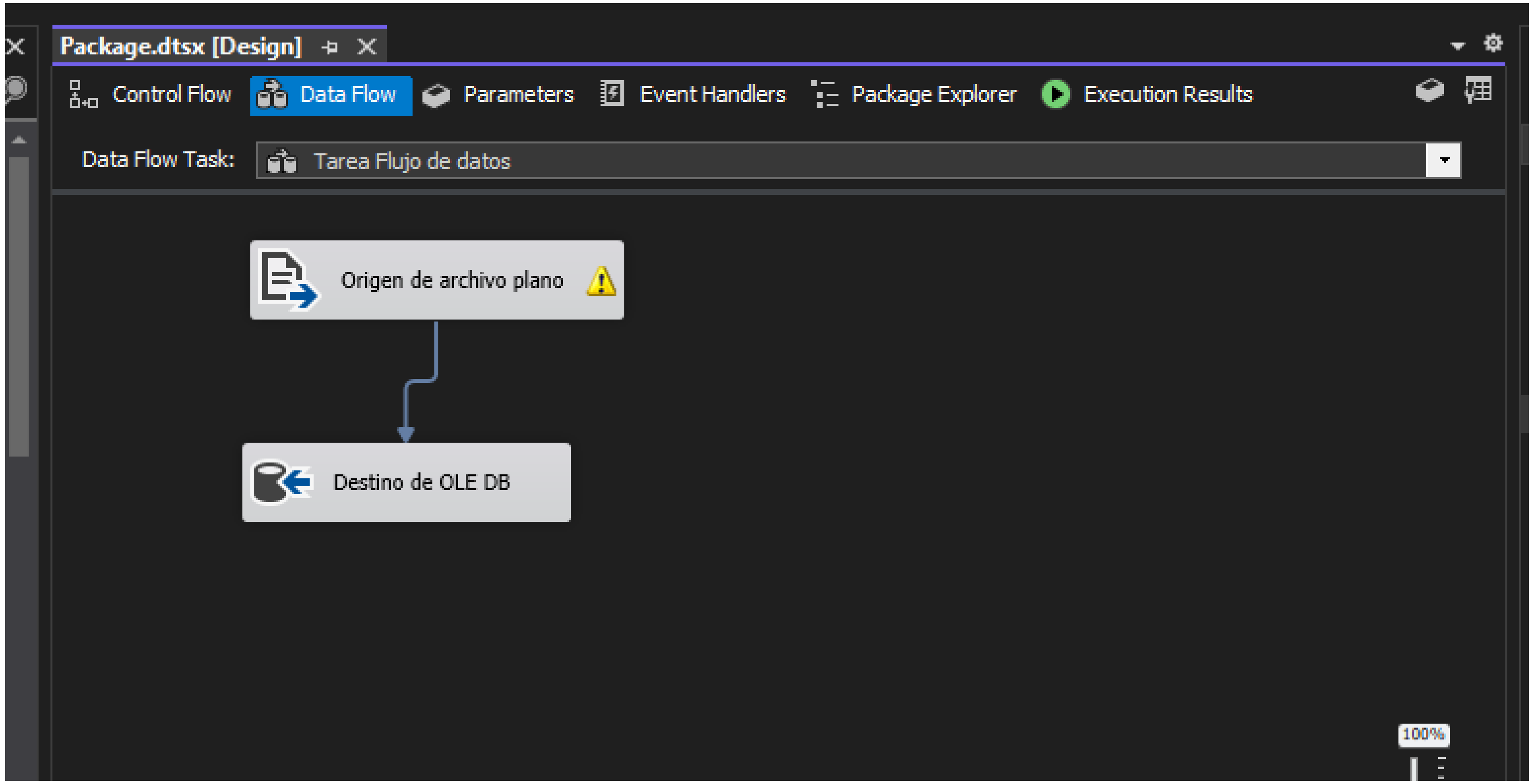


```python
import pypyodbc as odbc
import pandas as pd

""""""
'Step 1: importing the csv file'
""""""
df = pd.read_csv("D:\\SQL\\women_open_21.1.csv")



""""""
'Step 2: Specify columns to import clean-up'
""""""
columns = ['first_name', 'lastname', 'Placing', 'country', 'continent','Age', 'affiliate','Height and Weight', 'points', 'weight']

df_data = df[columns]

records = df_data.values.tolist() #convert the series into a list to import to our database

""""""
'Step 3.1: Create SQL Server Connection String'
""""""

driver = 'SQL Server'
server_name = 'DESKTOP-9EGLUDE\SQLSERVER'
database_name = 'woman_crossfit21'

def connection_string(driver, server_name,database_name):
    conn_string = f"""
```

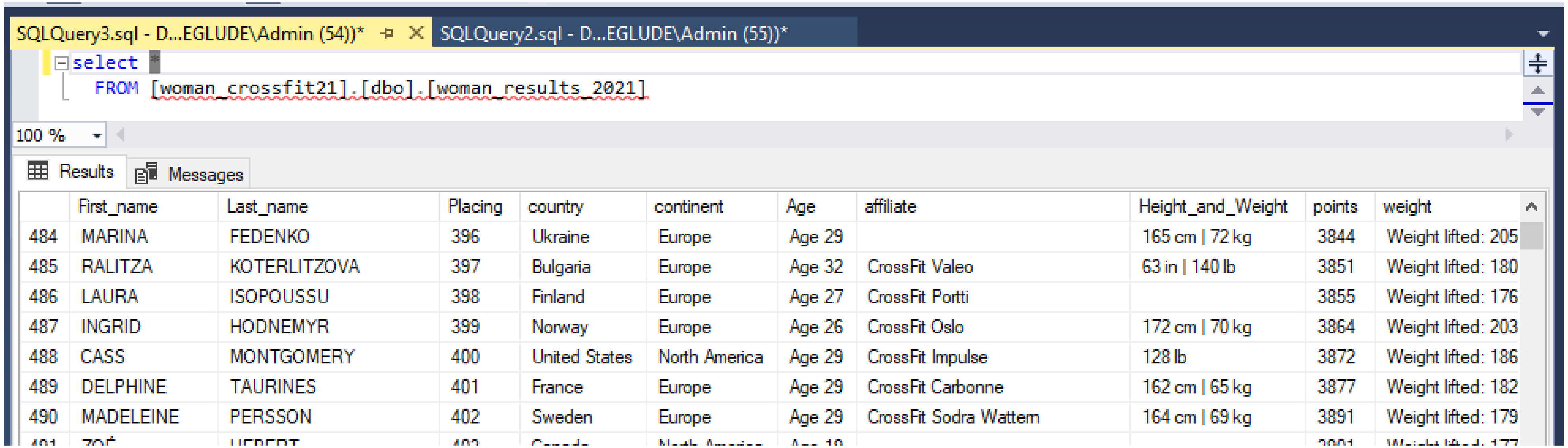
```
PS C:\Users\Admin> & C:/Users/Admin/AppData/Local/Programs/Python/Python310/python.exe d:/SQL/CSVtoSQLServer_script.py
```

- # Import the data from .CSV to SQL Server

2) Running a SSIS package



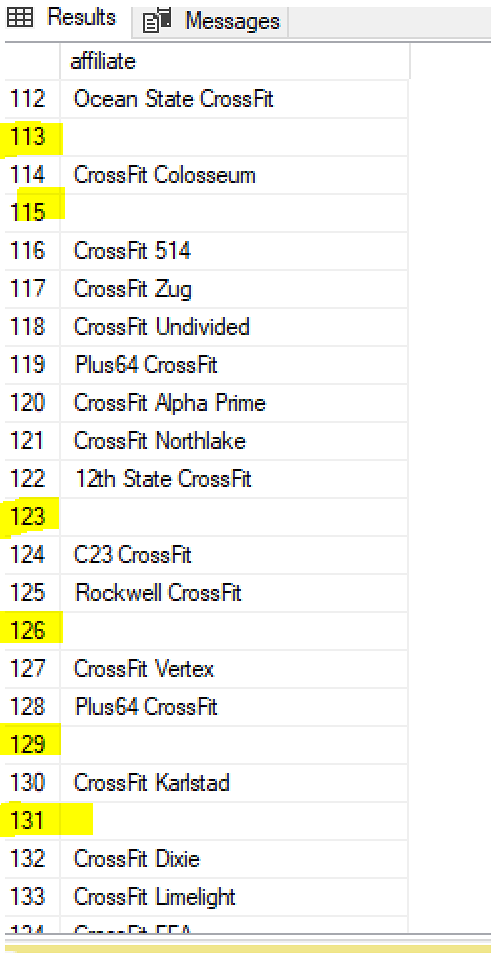
- **Verify the records loaded & start to clean the data**


SQLQuery3.sql - D...EGLUDE\Admin (54))*
SQLQuery2.sql - D...EGLUDE\Admin (55))*
100 %
Results
Messages

```
select *
  FROM [woman_crossfit21].[dbo].[woman_results_2021]
```

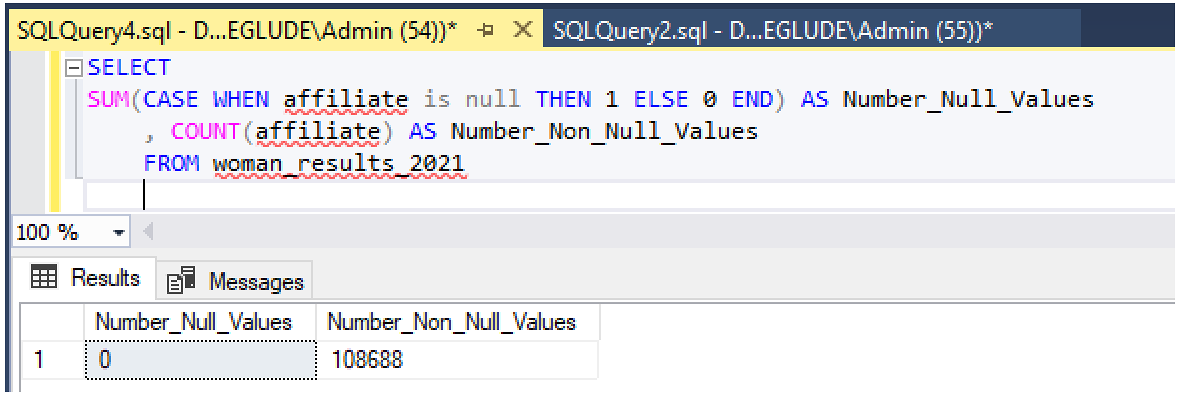
	First_name	Last_name	Placing	country	continent	Age	affiliate	Height_and_Weight	points	weight
484	MARINA	FEDENKO	396	Ukraine	Europe	Age 29		165 cm | 72 kg	3844	Weight lifted: 205
485	RALITZA	KOTERLITZOVA	397	Bulgaria	Europe	Age 32	CrossFit Valeo	63 in | 140 lb	3851	Weight lifted: 180
486	LAURA	ISOPOUSSU	398	Finland	Europe	Age 27	CrossFit Portti		3855	Weight lifted: 176
487	INGRID	HODNEMYR	399	Norway	Europe	Age 26	CrossFit Oslo	172 cm | 70 kg	3864	Weight lifted: 203
488	CASS	MONTGOMERY	400	United States	North America	Age 29	CrossFit Impulse	128 lb	3872	Weight lifted: 186
489	DELPHINE	TAURINES	401	France	Europe	Age 29	CrossFit Carbonne	162 cm | 65 kg	3877	Weight lifted: 182
490	MADELEINE	PERSSON	402	Sweden	Europe	Age 29	CrossFit Sodra Wattern	164 cm | 69 kg	3891	Weight lifted: 179
491	ZOÉ	HEBERT	403	Canada	North America	Age 19			3901	Weight lifted: 177

Missing Values



	affiliate
112	Ocean State CrossFit
113	
114	CrossFit Colosseum
115	
116	CrossFit 514
117	CrossFit Zug
118	CrossFit Undivided
119	Plus64 CrossFit
120	CrossFit Alpha Prime
121	CrossFit Northlake
122	12th State CrossFit
123	
124	C23 CrossFit
125	Rockwell CrossFit
126	
127	CrossFit Vertex
128	Plus64 CrossFit
129	
130	CrossFit Karlstad
131	
132	CrossFit Dixie
133	CrossFit Limelight
134	CrossFit EFA

- We have some missing records, for example in the "affiliate" column
- But we I count the null records, it gave me 0. So, it is not considering as a NULL.

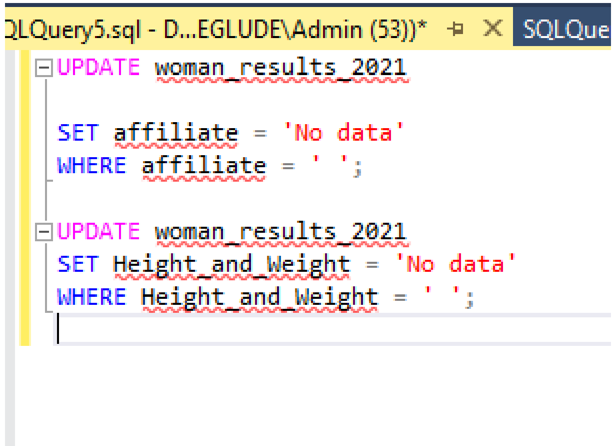


SQLQuery4.sql - D...EGLUDE\Admin (54))*
SQLQuery2.sql - D...EGLUDE\Admin (55))*
100 %
Results
Messages

```
SELECT
SUM(CASE WHEN affiliate is null THEN 1 ELSE 0 END) AS Number_Null_Values
    , COUNT(affiliate) AS Number_Non_Null_Values
    FROM woman_results_2021
```

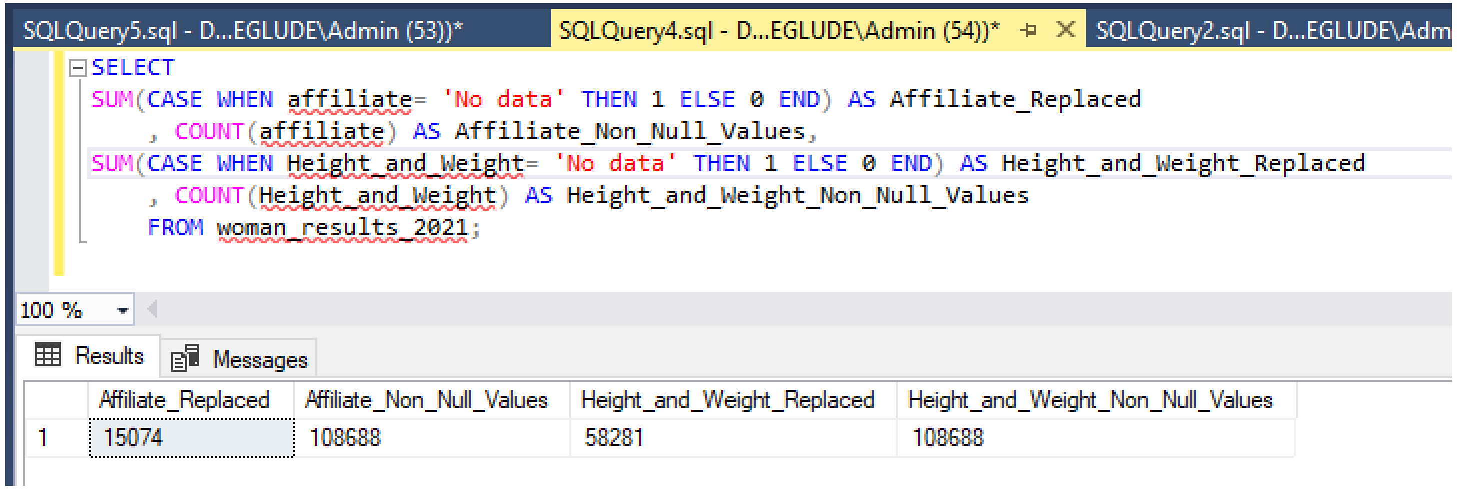
	Number_Null_Values	Number_Non_Null_Values
1	0	108688

- To avoid some troubles in “JOIN” statements for example, we can fill the blank (in this case) or null records with some useful description. In this case, I use UPDATE to set new record values:



```sql
UPDATE woman_results_2021

SET affiliate = 'No data'
WHERE affiliate = ' ';

UPDATE woman_results_2021
SET Height_and_Weight = 'No data'
WHERE Height_and_Weight = ' ';
```

- The outputs:



SQLQuery5.sql - D...EGLUDE\Admin (53))*
SQLQuery4.sql - D...EGLUDE\Admin (54))*
SQLQuery2.sql - D...EGLUDE\Adm
100 %
Results
Messages

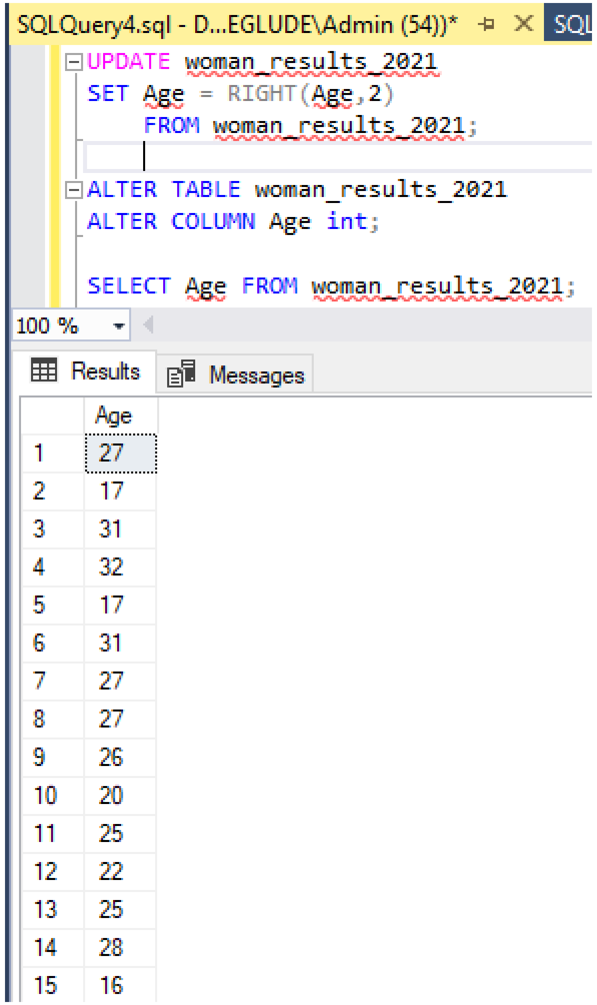
```sql
SELECT
SUM(CASE WHEN affiliate= 'No data' THEN 1 ELSE 0 END) AS Affiliate_Replaced
    , COUNT(affiliate) AS Affiliate_Non_Null_Values,
SUM(CASE WHEN Height_and_Weight= 'No data' THEN 1 ELSE 0 END) AS Height_and_Weight_Replaced
    , COUNT(Height_and_Weight) AS Height_and_Weight_Non_Null_Values
    FROM woman_results_2021;
```

	Affiliate_Replaced	Affiliate_Non_Null_Values	Height_and_Weight_Replaced	Height_and_Weight_Non_Null_Values
1	15074	108688	58281	108688

Now that we deal with missing data, we can review the content

- First, I going to work in the “Age” column because I am only interesting in the number of years. I’m going to use RIGHT statement to keep only the number and re-type the column into int.

- The output:



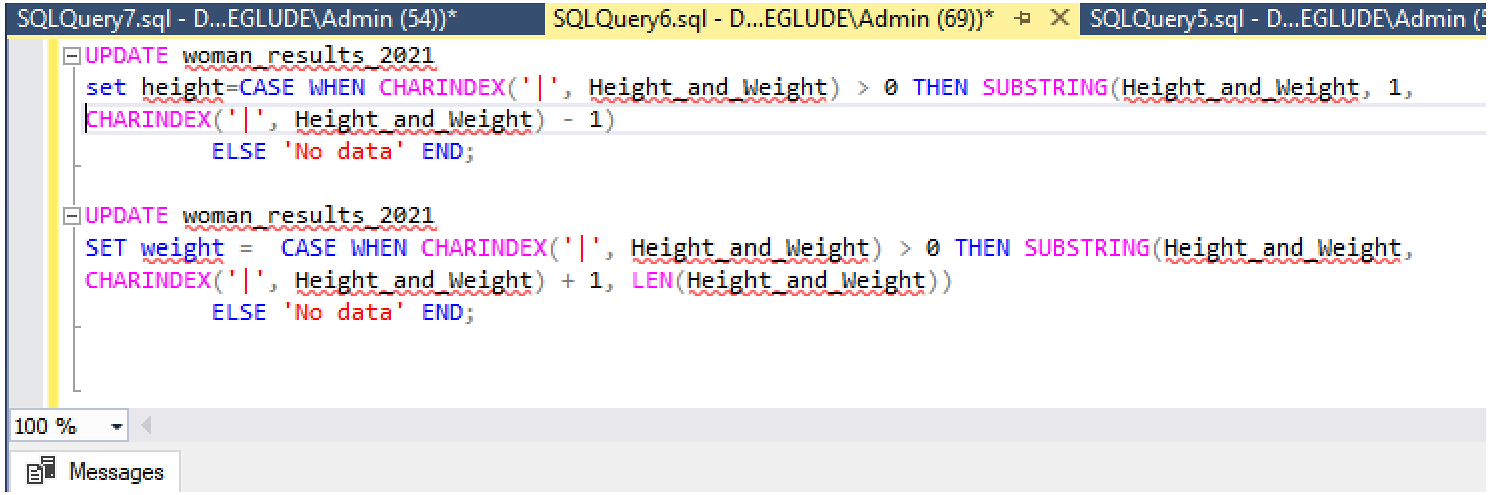
```
UPDATE woman_results_2021
SET Age = RIGHT(Age,2)
    FROM woman_results_2021;

ALTER TABLE woman_results_2021
ALTER COLUMN Age int;

SELECT Age FROM woman_results_2021;
```

	Age
1	27
2	17
3	31
4	32
5	17
6	31
7	27
8	27
9	26
10	20
11	25
12	22
13	25
14	28
15	16

In“Height_and_Weight” column we have to work a bit deeper.

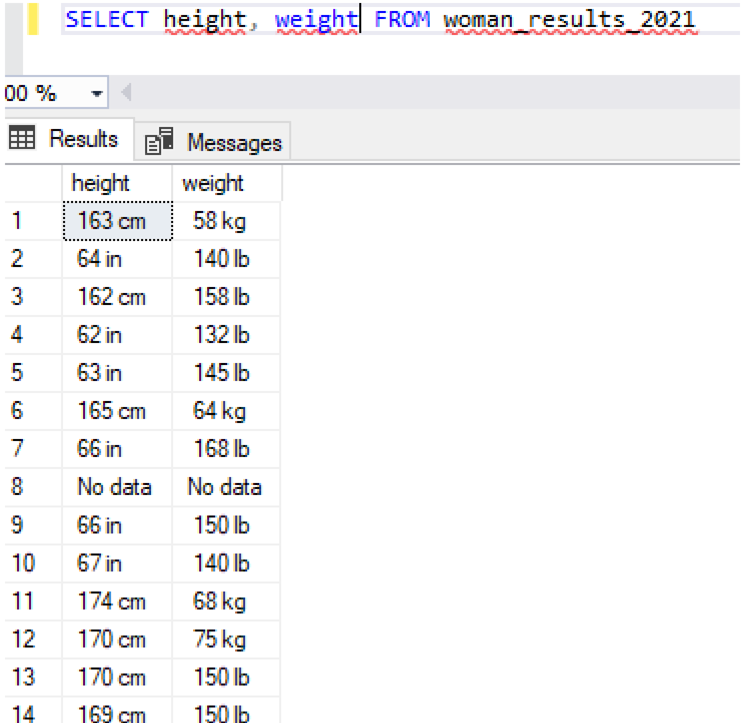
- First, let’s create the new columns called ‘height’ and ‘weight’. Previously, I renamed the original column ‘weight’ as ‘lbs_lifted’.
- Then, we want to split the ‘Height_and_Weight’ column in two. I worked out the splitting using *CHARINDEX* to map the “|” character and then used *SUBSTRING* to keep the data from the left side into height column and the data from the right into weight column.
- Finally, drop the old column.
- The Output:

1


SQLQuery7.sql - D...EGLUDE\Admin (54))*
SQLQuery6.sql - D...EGLUDE\Admin (69))*
SQLQuery5.sql - D...EGLUDE\Admin (5
100 %
Messages

```
UPDATE woman_results_2021
set height=CASE WHEN CHARINDEX('|', Height_and_Weight) > 0 THEN SUBSTRING(Height_and_Weight, 1,
CHARINDEX('|', Height_and_Weight) - 1)
         ELSE 'No data' END;

UPDATE woman_results_2021
SET weight =  CASE WHEN CHARINDEX('|', Height_and_Weight) > 0 THEN SUBSTRING(Height_and_Weight,
CHARINDEX('|', Height_and_Weight) + 1, LEN(Height_and_Weight))
         ELSE 'No data' END;
```

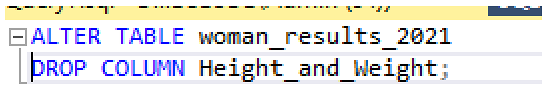
2



```
SELECT height, weight FROM woman_results_2021
```

	height	weight
1	163 cm	58 kg
2	64 in	140 lb
3	162 cm	158 lb
4	62 in	132 lb
5	63 in	145 lb
6	165 cm	64 kg
7	66 in	168 lb
8	No data	No data
9	66 in	150 lb
10	67 in	140 lb
11	174 cm	68 kg
12	170 cm	75 kg
13	170 cm	150 lb
14	169 cm	150 lb

3



```
ALTER TABLE woman_results_2021
DROP COLUMN Height_and_Weight;
```

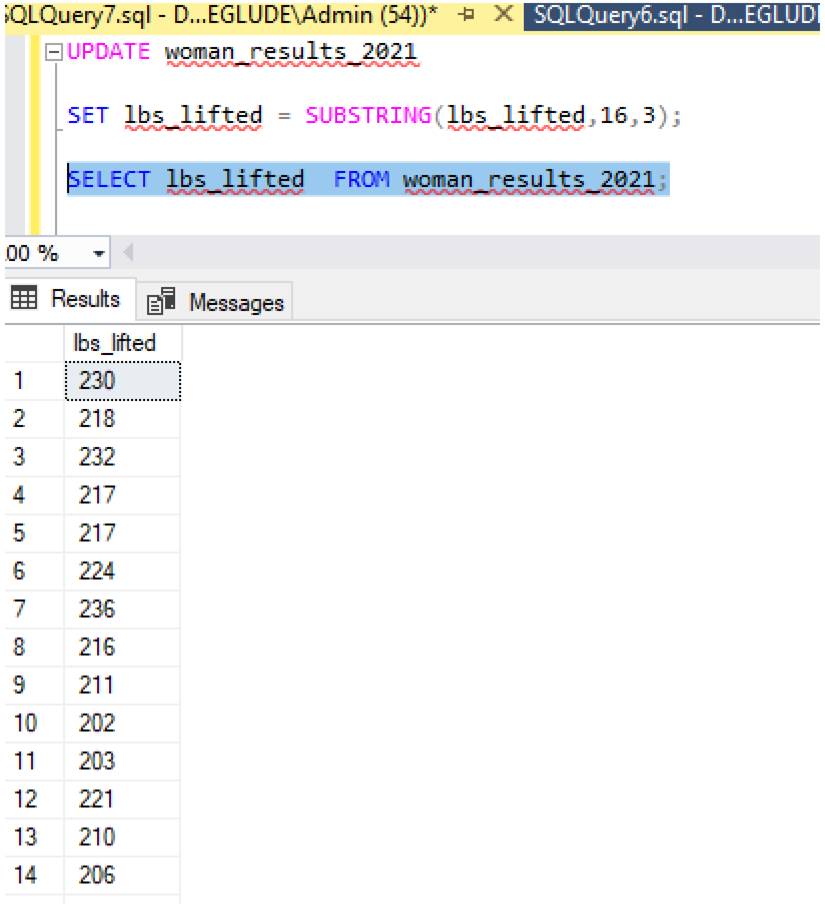
“Lbs_lifted” column

- Now, let’s move on into the “lbs_lifted” column and just maintain the actual weight lifted. To do that, I used the function *SUBSTRING.*

- The syntax is:

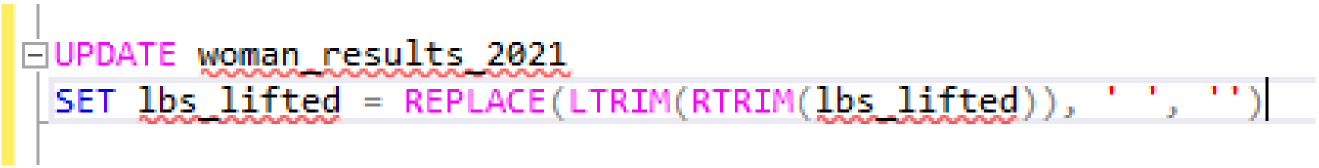
```
SUBSTRING(*string*, *starting character position*, *# of characters*)
```

- When I was trying to change the datatype into *int,* I noticed that there were some spaces in column. To remove them, use the *TRIM (or RTRIM, LTRIM)* function.


SQLQuery7.sql - D...EGLUDE\Admin (54))*
SQLQuery6.sql - D...EGLUD
00 %
Results
Messages

```
UPDATE woman_results_2021

SET lbs_lifted = SUBSTRING(lbs_lifted,16,3);

SELECT lbs_lifted  FROM woman_results_2021;
```

	lbs_lifted
1	230
2	218
3	232
4	217
5	217
6	224
7	236
8	216
9	211
10	202
11	203
12	221
13	210
14	206

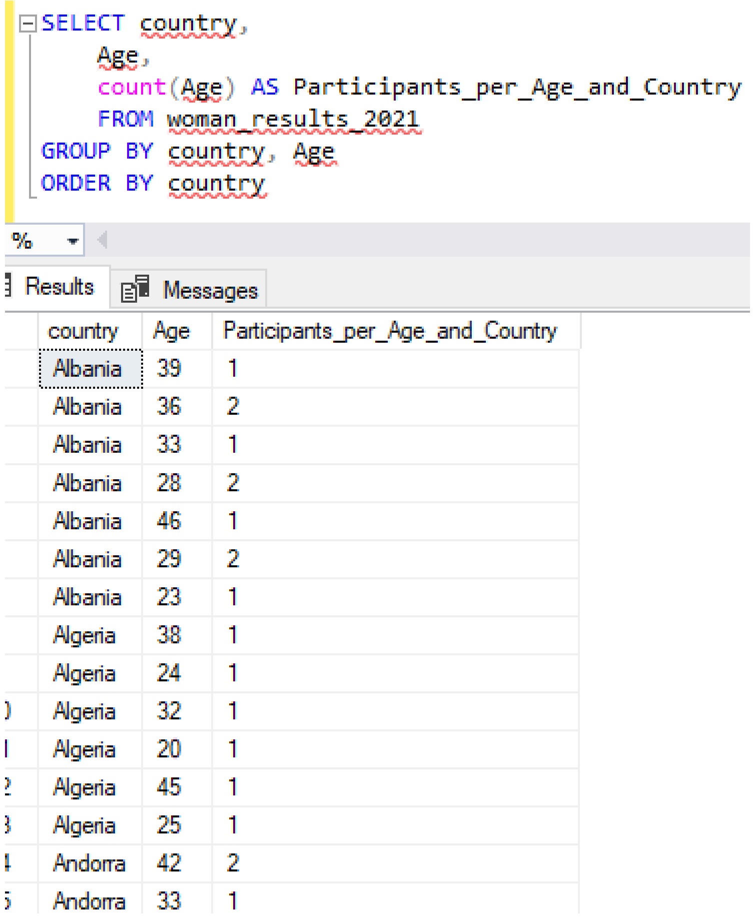


```
UPDATE woman_results_2021
SET lbs_lifted = REPLACE(LTRIM(RTRIM(lbs_lifted)), ' ', '')
```

Capturing Insights

- Now, we can querying to discover some insights as your requests.

- For this project, I will connect the SQL Server Database with Power BI in order to extract the data and gain insights. After analyzing the data, I will create visualizations to present in the final report.



```
SELECT country,
       Age,
       count(Age) AS Participants_per_Age_and_Country
       FROM woman_results_2021
GROUP BY country, Age
ORDER BY country
```

country	Age	Participants_per_Age_and_Country
Albania	39	1
Albania	36	2
Albania	33	1
Albania	28	2
Albania	46	1
Albania	29	2
Albania	23	1
Algeria	38	1
Algeria	24	1
Algeria	32	1
Algeria	20	1
Algeria	45	1
Algeria	25	1
Andorra	42	2
Andorra	33	1