Data Analysis Project with SQL

Luigi Mennella

Project Description

Purpose: Given a laboratory, analyze for each operator the value obtained from the experiments varied, before and after the date of the machine change (May 1, 2020).

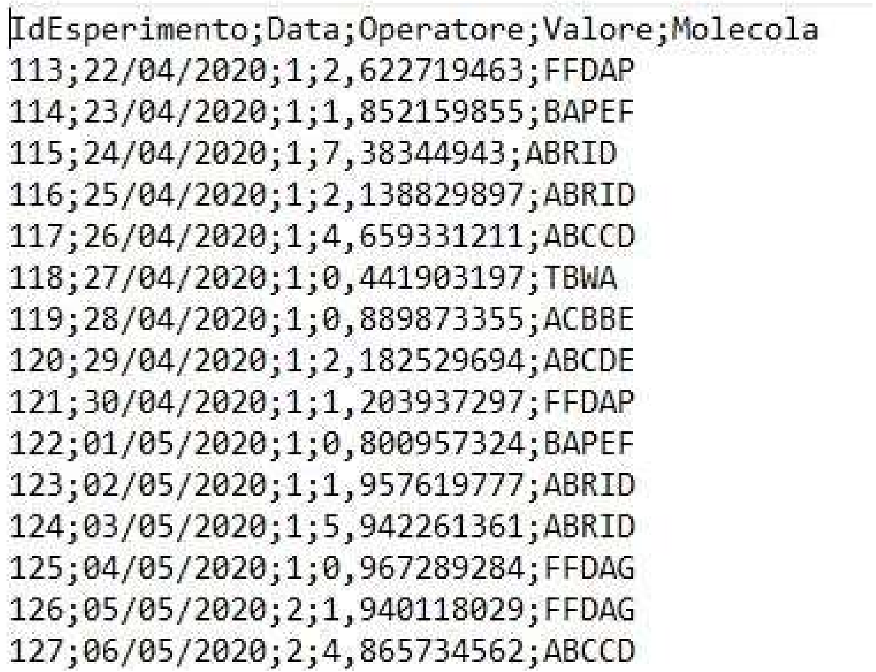
Starting data*: The following molecules are affected by the machinery:

- Whose name begins with “AB” and ends with “D”;
- Whose name starts with “F” and does NOT end with “P”.
- The data are available in the following files: *Experimenti_1.csv* , *Experimenti_2.csv*.

Analysis of input files

The input is represented by two **csv files** with:

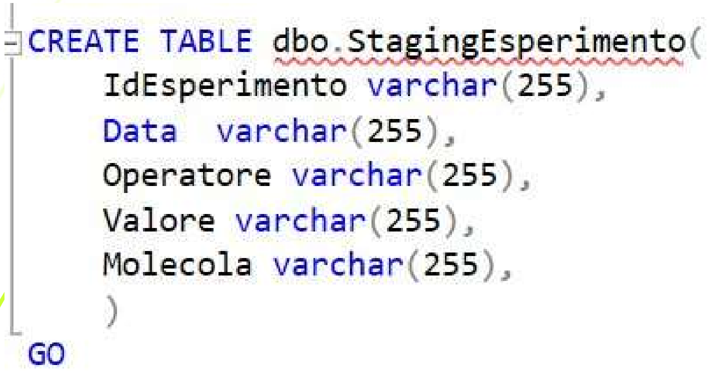
- *Semicolon* delimiter (;)
- Presence of header
- Presence of 5 columns
- Date in DD/MM/YYYY format
- Decimal separator *comma (,)*.



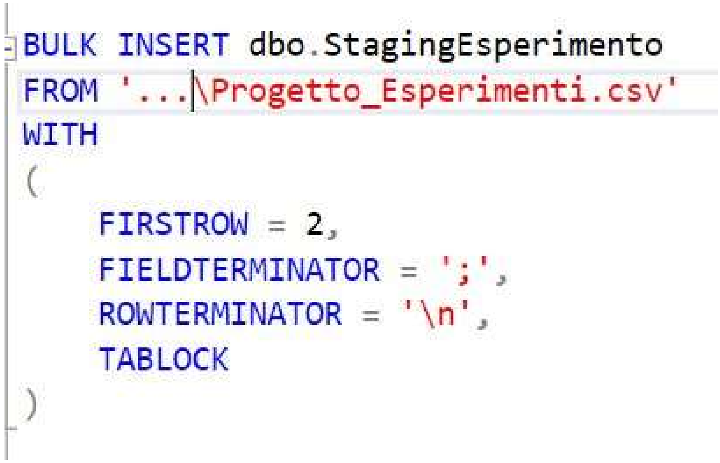
```
IdEsperimento;Data;Operatore;Valore;Molecola
113;22/04/2020;1;2,622719463;FFDAP
114;23/04/2020;1;1,852159855;BAPEF
115;24/04/2020;1;7,38344943;ABRID
116;25/04/2020;1;2,138829897;ABRID
117;26/04/2020;1;4,659331211;ABCCD
118;27/04/2020;1;0,441903197;TBWA
119;28/04/2020;1;0,889873355;ACBBE
120;29/04/2020;1;2,182529694;ABCDE
121;30/04/2020;1;1,203937297;FFDAP
122;01/05/2020;1;0,800957324;BAPEF
123;02/05/2020;1;1,957619777;ABRID
124;03/05/2020;1;5,942261361;ABRID
125;04/05/2020;1;0,967289284;FFDAG
126;05/05/2020;2;1,940118029;FFDAG
127;06/05/2020;2;4,865734562;ABCCD
```

Data Import

We initially load the data into a **staging table** without constraints.



```
CREATE TABLE dbo.StagingEsperimento(
    IdEsperimento varchar(255),
    Data  varchar(255),
    Operatore varchar(255),
    Valore varchar(255),
    Molecola varchar(255),
    )
GO
```

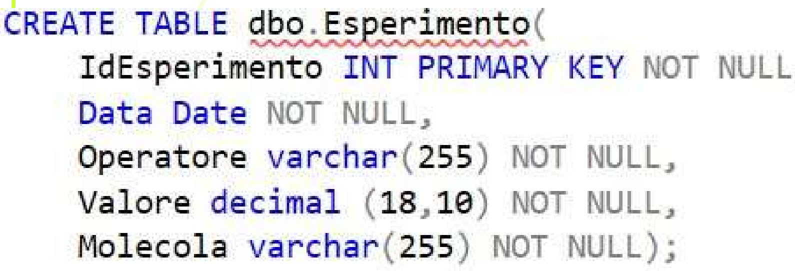


```
BULK INSERT dbo.StagingEsperimento
FROM '...\Progetto_Esperimenti.csv'
WITH
(
    FIRSTROW = 2,
    FIELDTERMINATOR = ';',
    ROWTERMINATOR = '\n',
    TABLOCK
)
```

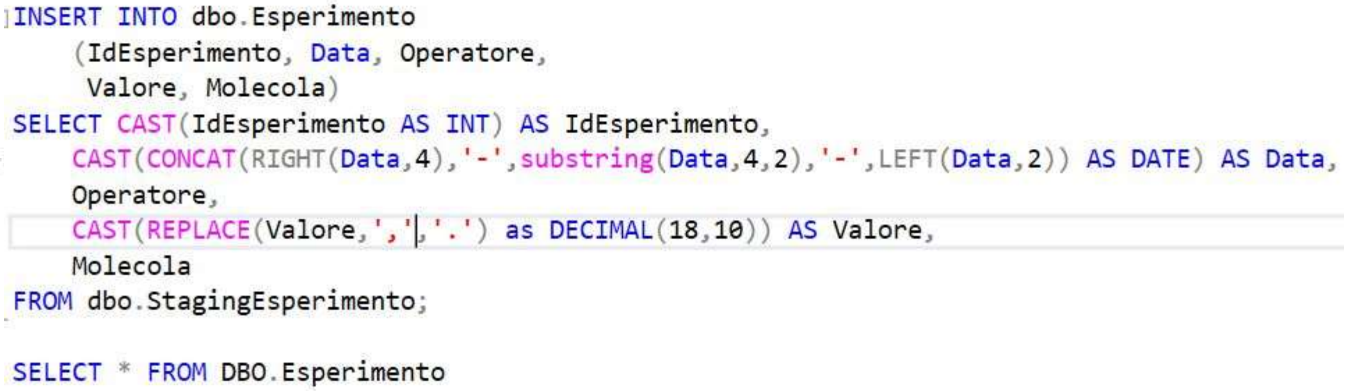
It should be remembered that the data import procedure is performed twice, once for each file to be imported.

Data Import - 2

We then transfer the data into the **target table**, with its ***primary key*** and ***non-nullity constraints.***



```
CREATE TABLE dbo.Esperimento(
    IdEsperimento INT PRIMARY KEY NOT NULL
    Data Date NOT NULL,
    Operatore varchar(255) NOT NULL,
    Valore decimal (18,10) NOT NULL,
    Molecola varchar(255) NOT NULL);
```



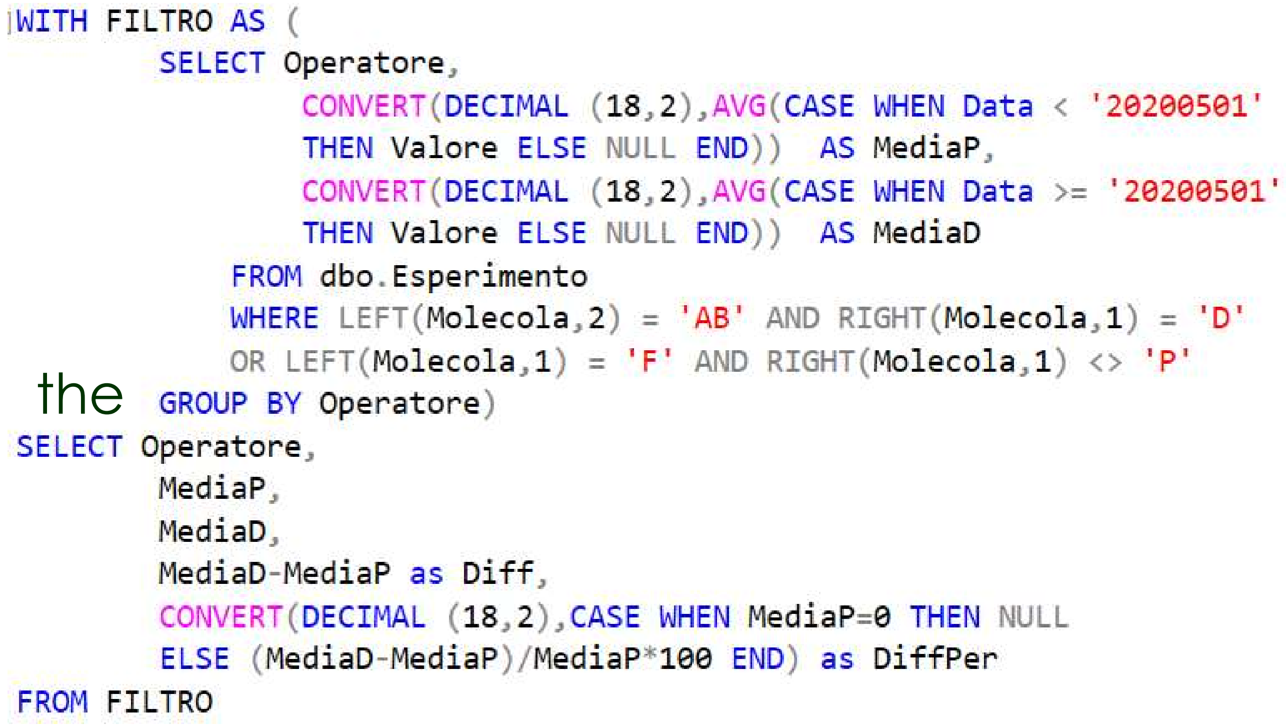
```
INSERT INTO dbo.Esperimento
    (IdEsperimento, Data, Operatore,
     Valore, Molecola)
SELECT CAST(IdEsperimento AS INT) AS IdEsperimento,
    CAST(CONCAT(RIGHT(Data,4),'-',substring(Data,4,2),'-',LEFT(Data,2)) AS DATE) AS Data,
    Operatore,
    CAST(REPLACE(Valore,',','.') as DECIMAL(18,10)) AS Valore,
    Molecola
FROM dbo.StagingEsperimento;

SELECT * FROM DBO.Esperimento
```

To change the formats of date and value fields, the commands: **CAST**, **CONCAT**, **REPLACE** and **SUBRSTRING are used**.

Writing the query in SQL

After studying several alternatives, we opt for one query, with:

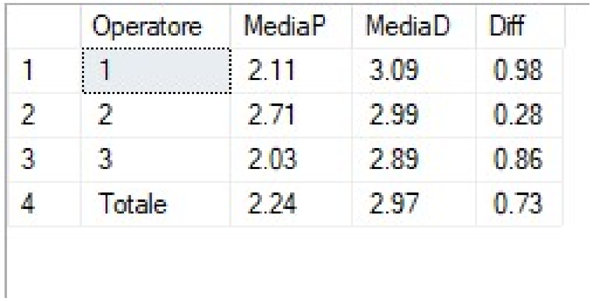
- **CASE WHEN** to differentiate the average as a function of the date
- **WHERE** to select the molecules
- Difference between means (**Diff**)
- Using a **CTE** for better readability



```
WITH FILTRO AS (
        SELECT Operatore,
                CONVERT(DECIMAL (18,2),AVG(CASE WHEN Data < '20200501'
                THEN Valore ELSE NULL END))  AS MediaP,
                CONVERT(DECIMAL (18,2),AVG(CASE WHEN Data >= '20200501'
                THEN Valore ELSE NULL END))  AS MediaD
            FROM dbo.Esperimento
            WHERE LEFT(Molecola,2) = 'AB' AND RIGHT(Molecola,1) = 'D'
            OR LEFT(Molecola,1) = 'F' AND RIGHT(Molecola,1) <> 'P'
        GROUP BY Operatore)
SELECT Operatore,
        MediaP,
        MediaD,
        MediaD-MediaP as Diff,
        CONVERT(DECIMAL (18,2),CASE WHEN MediaP=0 THEN NULL
        ELSE (MediaD-MediaP)/MediaP*100 END) as DiffPer
FROM FILTRO
```

Analysis of the results

The results show a significant variation in measurements for all operators, with a maximum value for operator no.1 and a minimum for no. 2.



	Operatore	MediaP	MediaD	Diff
1	1	2.11	3.09	0.98
2	2	2.71	2.99	0.28
3	3	2.03	2.89	0.86
4	Totale	2.24	2.97	0.73

It should also be noted that to obtain the summary table of the results it was necessary to use the following functions:

- **Temporary tables** to estimate separately for each operator and the total
- **UNION ALL** to merge the temporary tables results
- **ORDER BY** to ensure correct order.