



# SENTENCIAS MIB T-SQL

**MARIO JIMÉNEZ MARSET**

**ÍNDICE**

1. ENUNCIADO - OBJETIVOS.....	3
2. DESARROLLO – PROCEDIMIENTOS.....	3

## 1. ENUNCIADO - OBJETIVOS

Esta práctica consistía en realizar tareas de modificación, inserción y borrado, a partir de la tabla “VUREAVI” realizada en prácticas anteriores. Estas tareas eran:

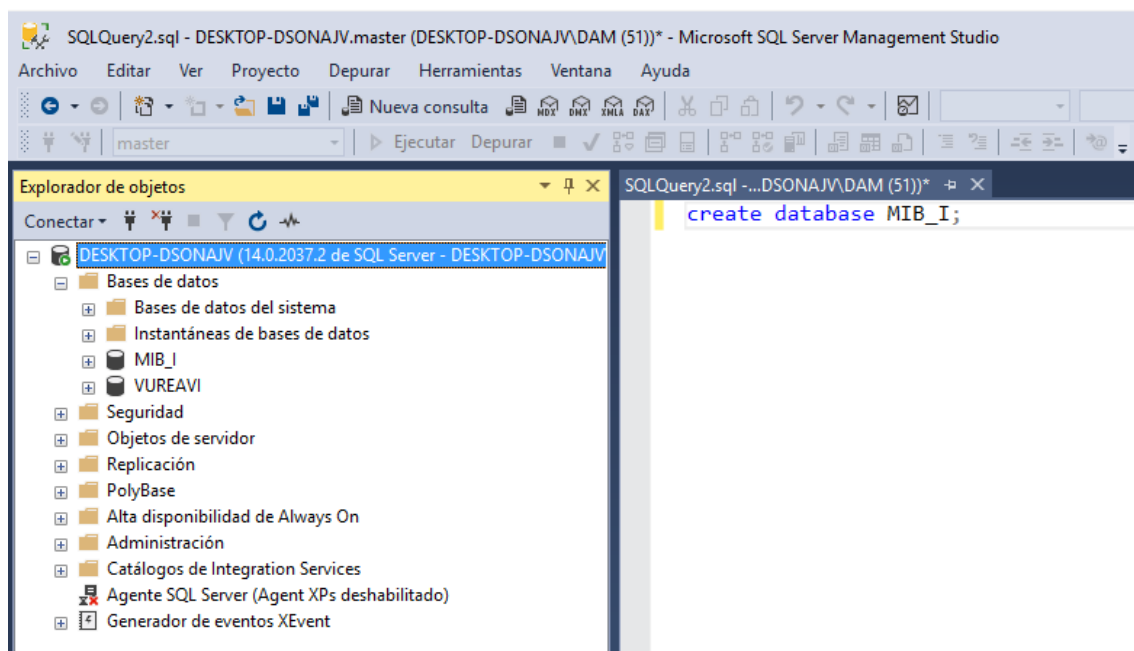
1. Crear una base de datos denominada MIB-I, sobre la cual se realizará el resto de ejercicios.
2. Crear una tabla con las columnas num\_vuelo, origen, destino y distancia, semejantes a las de la tabla vuelos.  
Añadir la columna tipo\_avion a la tabla creada.
3. Crear un índice único sobre la tabla creada, para la columna num\_vuelo.
4. Insertar en la tabla creada todas las informaciones que hay en la tabla vuelos, de los realizados desde una ciudad española.
5. Insertar en la tabla creada el vuelo IB8888 entre Nueva York y Londres.
6. Modificar el destino del vuelo IB8888, que pasar a ser el del vuelo IB3709.
7. Modificar las distancias en los vuelos en la tabla creada incrementándolos en un 5%.
8. De la tabla creada, borrar las filas cuya distancia sea mayor que la media de las distancias de los vuelos realizados por Iberia.
9. Modificar la columna distancia dejándola nula para todas las filas.
10. Borrar las filas con origen en Madrid o Barcelona.
11. Borra todas las filas.
12. Borrar el índice y la tabla creados.

## 2. DESARROLLO – PROCEDIMIENTOS

Se adjuntan las soluciones de cada uno de los apartados (con captura de pantalla):

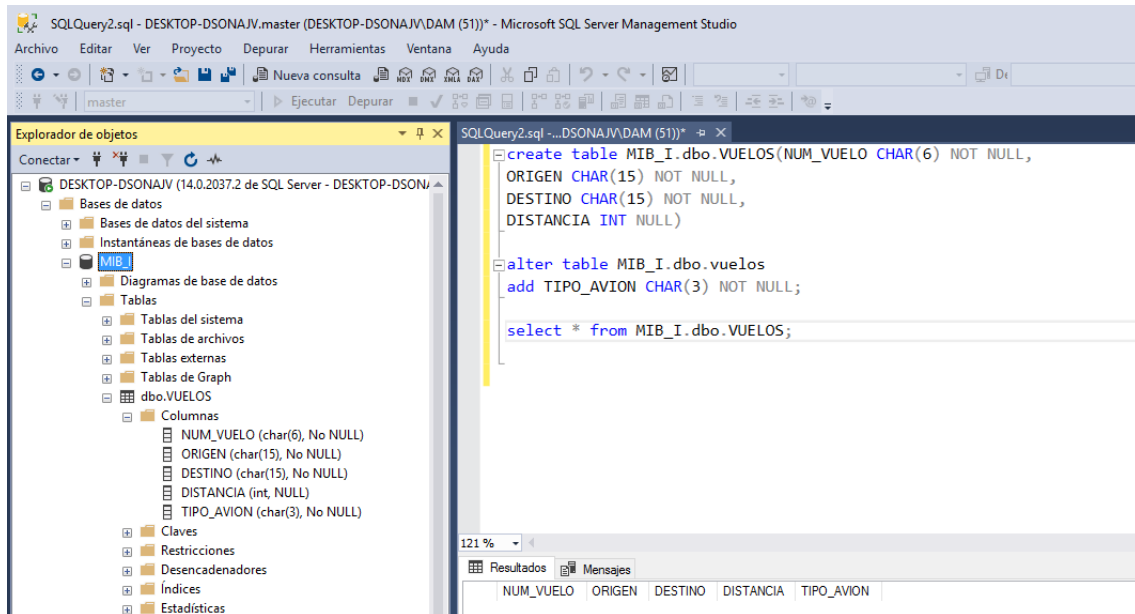
1. Crear una base de datos denominada MIB-I, sobre la cual se realizará el resto de ejercicios.

Simplemente se utiliza la sintaxis “create database”, además del nombre elegido.



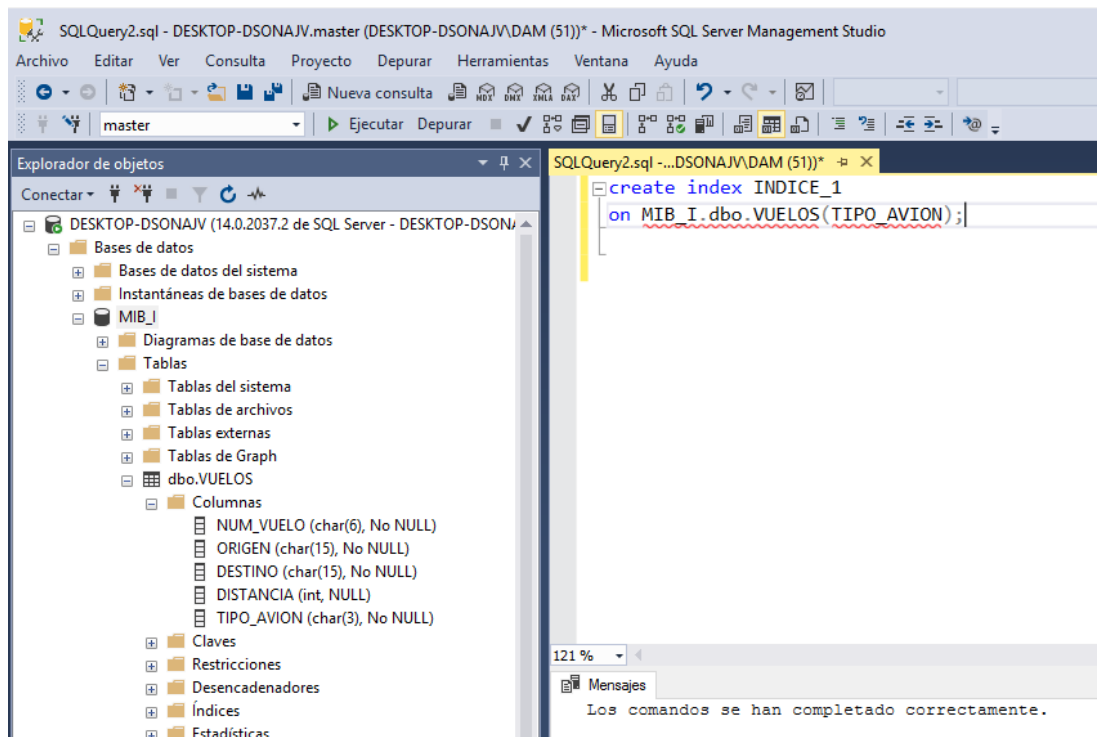
2. Crear una tabla con las columnas num\_vuelo, origen, destino y distancia, semejantes a las de la tabla vuelos. Añadir la columna tipo\_avion a la tabla creada.

Se crea la tabla al igual que en la práctica anterior, con la diferencia de que, para añadir otra columna, se utiliza “alter table” y con “add” la columna elegida.



3. Crear un índice único sobre la tabla creada, para la columna num\_vuelo.

Simplemente se crea el índice incluyendo su nombre en la tabla pertinente (“create index on”)



- Insertar en la tabla creada todas las informaciones que hay en la tabla vuelos, de los realizados desde una ciudad española.

A partir de “insert into”, se especifican el nombre de tabla y columnas, seleccionando esto mismo con “select” de otra tabla de la que se quiere información. Se utiliza “where” para establecer la condición (en este caso donde el origen sea igual a alguna de las ciudades especificadas).

The screenshot shows the Microsoft SQL Server Enterprise Manager interface. On the left, the 'Explorador de objetos' (Object Explorer) pane displays the database structure for 'DESKTOP-DSONAJV'. The 'dbo.VUELOS' table is selected, showing its columns: NUM\_VUELO (char(6), No NULL), ORIGEN (char(15), No NULL), DESTINO (char(15), No NULL), DISTANCIA (int, NULL), and TIPO\_AVION (char(3), No NULL). The main window displays a SQL query in the 'SQLQuery2.sql' file:

```
insert into MIB I.dbo.VUELOS(NUM_VUELO, ORIGEN, DESTINO, DISTANCIA, TIPO_AVION)
select NUM_VUELO, ORIGEN, DESTINO, DISTANCIA, TIPO_AVION from VUREAVI.dbo.VUELOS
where origen = 'MADRID' or origen = 'BARCELONA' or origen = 'BILBAO' or origen = 'SEVILLA' or origen = 'MALAGA';
select * from MIB I.dbo.VUELOS;
```

Below the query, the 'Resultados' (Results) pane shows the data from the 'VUELOS' table. The results are as follows:

	NUM_VUELO	ORIGEN	DESTINO	DISTANCIA	TIPO_AVION
1	IB600	MADRID	LONDRES	1800	320
2	BA467	MADRID	LONDRES	1800	73S
3	IB640	MADRID	BARCELONA	750	320
4	IB3742	MADRID	BARCELONA	750	72S
5	AF577	BILBAO	PARIS	800	737
6	IB778	BARCELONA	ROMA	1000	72S
7	IB721	BARCELONA	SEVILLA	950	72S
8	IB327	MADRID	SEVILLA	600	72S
9	IB023	MADRID	TENERIFE	1500	72S
10	IB368	MALAGA	BARCELONA	900	D9S
11	IB610	MALAGA	LONDRES	1900	73S
12	IB510	SEVILLA	MADRID	600	72S
13	IB318	SEVILLA	MADRID	600	72S

5. Insertar en la tabla creada el vuelo IB8888 entre Nueva York y Londres.

Con la sintaxis “insert into” se especifica el nombre de tabla y columnas, incluyendo en “values” los valores puestos en el enunciado, además de otros 2 valores inventados.

The screenshot shows the Microsoft SQL Server Management Studio interface. The left pane displays the 'Explorador de objetos' (Object Explorer) for the 'DESKTOP-DSONAJV' server. The 'Bases de datos' (Databases) folder is expanded, showing the 'MIB\_I' database. The 'Tablas' (Tables) folder is expanded, showing the 'dbo.VUELOS' table. The right pane shows the 'SQLQuery2.sql' script with the following SQL code:

```
insert into MIB_I.dbo.VUELOS(NUM_VUELO, ORIGEN, DESTINO, DISTANCIA, TIPO_AVION)
values ('IB8888', 'NUEVA YORK', 'LONDRES', 1230, '69A');

select * from MIB_I.dbo.VUELOS;
```

The bottom pane shows the 'Resultados' (Results) tab with a table of 14 rows. The first row is highlighted. The table columns are NUM\_VUELO, ORIGEN, DESTINO, DISTANCIA, and TIPO\_AVION.

	NUM_VUELO	ORIGEN	DESTINO	DISTANCIA	TIPO_AVION
1	IB600	MADRID	LONDRES	1800	320
2	BA467	MADRID	LONDRES	1800	73S
3	IB640	MADRID	BARCELONA	750	320
4	IB3742	MADRID	BARCELONA	750	72S
5	AF577	BILBAO	PARIS	800	737
6	IB778	BARCELONA	ROMA	1000	72S
7	IB721	BARCELONA	SEVILLA	950	72S
8	IB327	MADRID	SEVILLA	600	72S
9	IB023	MADRID	TENERIFE	1500	72S
10	IB368	MALAGA	BARCELONA	900	D9S
11	IB610	MALAGA	LONDRES	1900	73S
12	IB510	SEVILLA	MADRID	600	72S
13	IB318	SEVILLA	MADRID	600	72S
14	IB8888	NUEVA YORK	LONDRES	1230	69A

6. Modificar el destino del vuelo IB8888, que pasar a ser el del vuelo IB3709.

Con “update” se actualiza la tabla (se especifica el nombre de tabla), se establece cuál va a ser el destino (con “set”). Lo establecido es una consulta básica de “select”, donde se elige el destino de la tabla Vureavi (con “from”), cuya condición es la elección de un número de vuelo específico (con “where”).

Por último, se establece la condición (también con “where”) del número de vuelo que se quería modificar desde un principio.

The screenshot shows the Microsoft SQL Server Management Studio interface. The left pane displays the 'Explorador de objetos' (Object Explorer) for the 'DESKTOP-DSONAJV' server, showing the 'dbo.VUELOS' table. The right pane shows the 'SQLQuery1.sql' editor with the following SQL query:

```
update MIB_I.dbo.VUELOS
set DESTINO = (select DESTINO from VUREAVI.dbo.VUELOS where NUM_VUELO = 'IB3709')
where NUM_VUELO = 'IB8888'

select * from MIB_I.dbo.VUELOS;
```

Below the query editor, the 'Resultados' (Results) pane shows the output of the query, displaying a table with 14 rows and 5 columns: NUM\_VUELO, ORIGEN, DESTINO, DISTANCIA, and TIPO\_AVION. The first row is highlighted.

NUM_VUELO	ORIGEN	DESTINO	DISTANCIA	TIPO_AVION
IB600	MADRID	LONDRES	1800	320
BA467	MADRID	LONDRES	1800	73S
IB640	MADRID	BARCELONA	750	320
IB3742	MADRID	BARCELONA	750	72S
AF577	BILBAO	PARIS	800	737
IB778	BARCELONA	ROMA	1000	72S
IB721	BARCELONA	SEVILLA	950	72S
IB327	MADRID	SEVILLA	600	72S
IB023	MADRID	TENERIFE	1500	72S
IB368	MALAGA	BARCELONA	900	D9S
IB610	MALAGA	LONDRES	1900	73S
IB510	SEVILLA	MADRID	600	72S
IB318	SEVILLA	MADRID	600	72S
IB8888	NUEVA YORK	BARCELONA	1230	69A

## 7. Modificar las distancias en los vuelos en la tabla creada incrementándolos en un 5%.

Se actualiza la tabla con “update” (especificando el nombre de tabla) fijando la distancia realizando una operación para incrementarla en un 5% (esto se consigue sumando a la distancia la propia distancia multiplicada por 0.05).

The screenshot shows the Microsoft SQL Server Management Studio interface. The left pane displays the 'Explorador de objetos' (Object Explorer) for the 'DESKTOP-DSOIAJV' server. The 'dbo.VUELOS' table is selected, showing its columns: NUM\_VUELO (char(6), No NULL), ORIGEN (char(15), No NULL), DESTINO (char(15), No NULL), DISTANCIA (int, NULL), and TIPO\_AVION (char(3), No NULL). The right pane shows the SQL query editor with the following code:

```
update MIB_I.dbo.VUELOS
set distancia = distancia + distancia*0.05;

select * from MIB_I.dbo.VUELOS;
```

Below the query editor, the 'Resultados' (Results) pane displays the data from the 'VUELOS' table. The table has 14 rows and 6 columns: NUM\_VUELO, ORIGEN, DESTINO, DISTANCIA, and TIPO\_AVION. The first row is highlighted.

	NUM_VUELO	ORIGEN	DESTINO	DISTANCIA	TIPO_AVION
1	IB600	MADRID	LONDRES	1890	320
2	BA467	MADRID	LONDRES	1890	73S
3	IB640	MADRID	BARCELONA	787	320
4	IB3742	MADRID	BARCELONA	787	72S
5	AF577	BILBAO	PARIS	840	737
6	IB778	BARCELONA	ROMA	1050	72S
7	IB721	BARCELONA	SEVILLA	997	72S
8	IB327	MADRID	SEVILLA	630	72S
9	IB023	MADRID	TENERIFE	1575	72S
10	IB368	MALAGA	BARCELONA	945	D9S
11	IB610	MALAGA	LONDRES	1995	73S
12	IB510	SEVILLA	MADRID	630	72S
13	IB318	SEVILLA	MADRID	630	72S
14	IB8888	NUEVA YORK	BARCELONA	1291	69A



8. De la tabla creada, borrar las filas cuya distancia sea mayor que la media de las distancias de los vuelos realizados por Iberia.

Se hace una consulta con “select”. Esta incluye “avg”, lo cual permite realizar la media aritmética de la columna elegida (en este caso distancia), habiendo especificado (con “from”) la tabla de la que proviene la columna.

Además, con “delete from” (eligiendo la tabla) y con la condición “where” se borran los valores de la columna distancia especificados en la condición.

The screenshot shows the Microsoft SQL Server Management Studio interface. The left pane displays the 'Explorador de objetos' (Object Explorer) with the database structure of 'DESKTOP-DSONAJV'. The right pane shows a SQL query in the 'SQLQuery2.sql' window:

```
select avg(DISTANCIA) from MIB_I.dbo.VUELOS;
delete from MIB_I.dbo.VUELOS
where DISTANCIA > 1121;

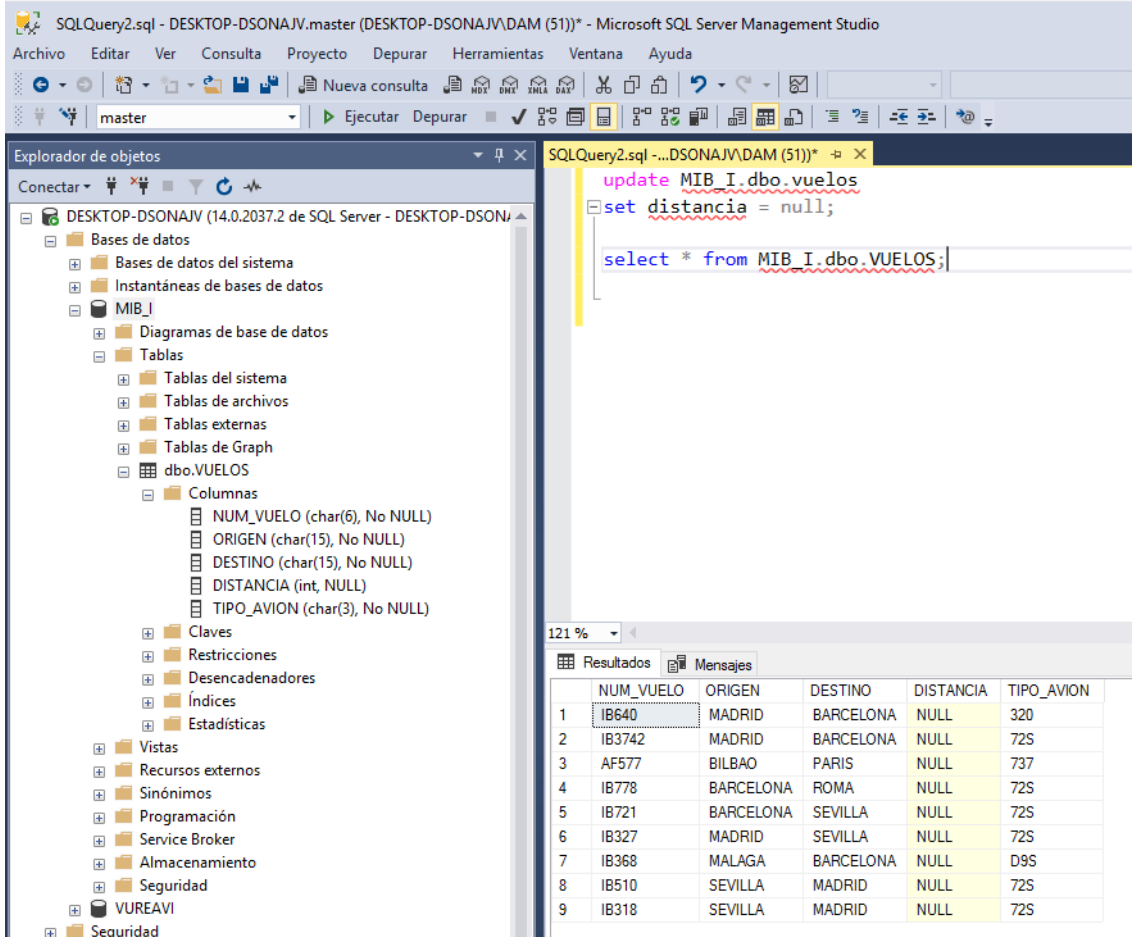
select * from MIB_I.dbo.VUELOS;
```

Below the query window, the 'Resultados' (Results) pane shows the output of the first query, which is the average distance: 1138. Below that, the 'Messages' pane shows the results of the second query, which is a list of flights with their distances. The table below is a representation of the data shown in the Messages pane:

	NUM_VUELO	ORIGEN	DESTINO	DISTANCIA	TIPO_AVION
1	IB640	MADRID	BARCELONA	787	320
2	IB3742	MADRID	BARCELONA	787	72S
3	AF577	BILBAO	PARIS	840	737
4	IB778	BARCELONA	ROMA	1050	72S
5	IB721	BARCELONA	SEVILLA	997	72S
6	IB327	MADRID	SEVILLA	630	72S
7	IB368	MALAGA	BARCELONA	945	D9S
8	IB510	SEVILLA	MADRID	630	72S
9	IB318	SEVILLA	MADRID	630	72S

## 9. Modificar la columna distancia dejándola nula para todas las filas.

Se actualiza con “update” la tabla elegida para fijar con “set” la columna distancia a valores nulos.



The screenshot shows the Microsoft SQL Server Management Studio interface. The left pane displays the 'Explorador de objetos' (Object Explorer) for the 'DESKTOP-DSONAJV' server. The 'dbo.VUELOS' table is selected, and its columns are listed: NUM\_VUELO (char(6), No NULL), ORIGEN (char(15), No NULL), DESTINO (char(15), No NULL), DISTANCIA (int, NULL), and TIPO\_AVION (char(3), No NULL). The right pane shows the SQL query editor with the following code:

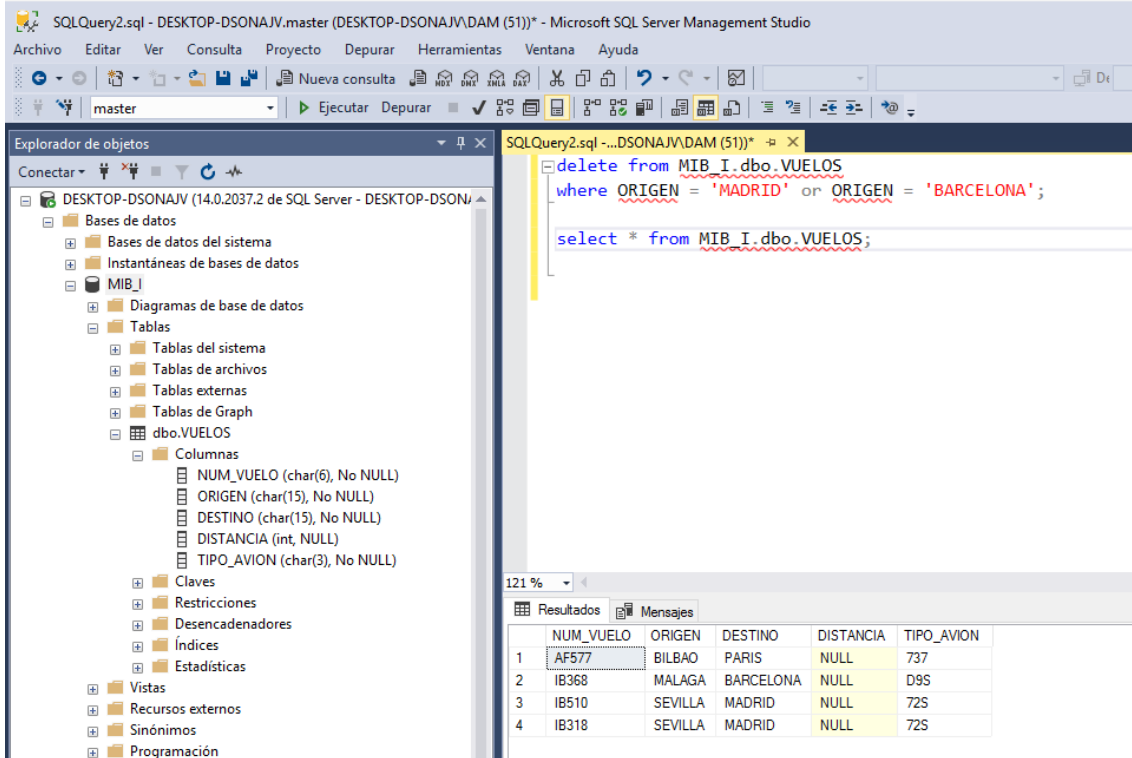
```
update MIB_I.dbo.vuelos  
set distancia = null;  
select * from MIB_I.dbo.VUELOS;
```

Below the query editor, the 'Resultados' (Results) pane displays the data from the 'VUELOS' table. The table has 9 rows and 6 columns: NUM\_VUELO, ORIGEN, DESTINO, DISTANCIA, and TIPO\_AVION. The 'DISTANCIA' column is highlighted in yellow for all rows, indicating it is being updated to NULL.

	NUM_VUELO	ORIGEN	DESTINO	DISTANCIA	TIPO_AVION
1	IB640	MADRID	BARCELONA	NULL	320
2	IB3742	MADRID	BARCELONA	NULL	72S
3	AF577	BILBAO	PARIS	NULL	737
4	IB778	BARCELONA	ROMA	NULL	72S
5	IB721	BARCELONA	SEVILLA	NULL	72S
6	IB327	MADRID	SEVILLA	NULL	72S
7	IB368	MALAGA	BARCELONA	NULL	D9S
8	IB510	SEVILLA	MADRID	NULL	72S
9	IB318	SEVILLA	MADRID	NULL	72S

## 10. Borrar las filas con origen en Madrid o Barcelona.

Con “delete from” y la condición “where”, se borran las filas de la tabla especificada con la condición especificada.



The screenshot shows the Microsoft SQL Server Management Studio interface. The left pane displays the 'Explorador de objetos' (Object Explorer) with the database structure of 'DESKTOP-DSONAJV'. The right pane shows the 'SQLQuery2.sql' file with the following SQL code:

```
delete from MIB_I.dbo.VUELOS
where ORIGEN = 'MADRID' or ORIGEN = 'BARCELONA';

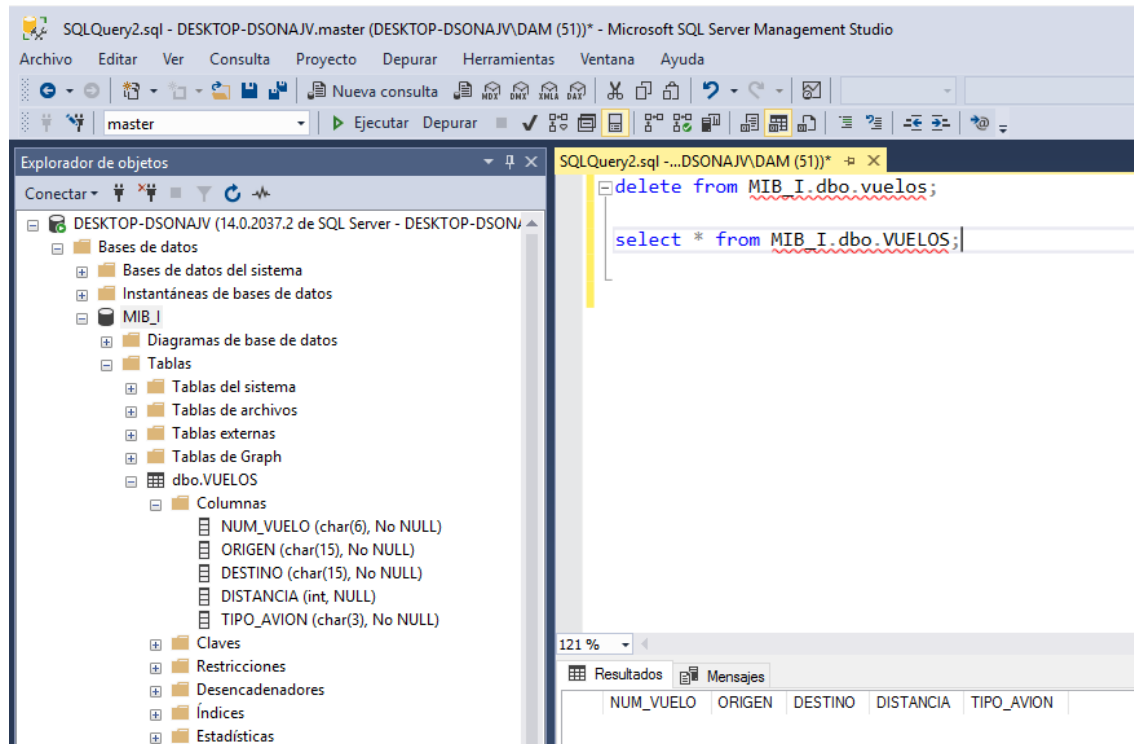
select * from MIB_I.dbo.VUELOS;
```

Below the query, the 'Resultados' (Results) pane shows the output of the select statement, displaying a table with 5 columns: NUM\_VUELO, ORIGEN, DESTINO, DISTANCIA, and TIPO\_AVION. The table contains 4 rows of data.

	NUM_VUELO	ORIGEN	DESTINO	DISTANCIA	TIPO_AVION
1	AF577	BILBAO	PARIS	NULL	737
2	IB368	MALAGA	BARCELONA	NULL	D9S
3	IB510	SEVILLA	MADRID	NULL	72S
4	IB318	SEVILLA	MADRID	NULL	72S

## 11. Borra todas las filas.

Con “delete from” simplemente se borran todos los valores de las filas de la tabla especificada, sin condición ninguna ya que se quieren eliminar absolutamente todos los valores.



## 12. Borrar el índice y la tabla creados.

Con “drop index” y “drop table” respectivamente, se borran el índice y la tabla especificados (para borrar el índice es necesario especificar la tabla en el que está creado el índice).

