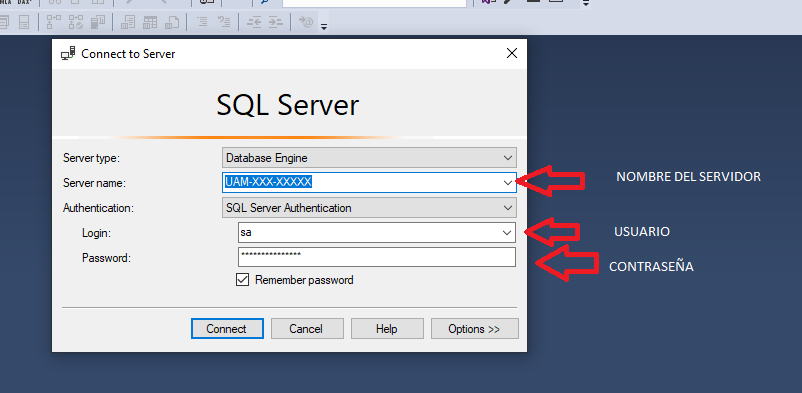
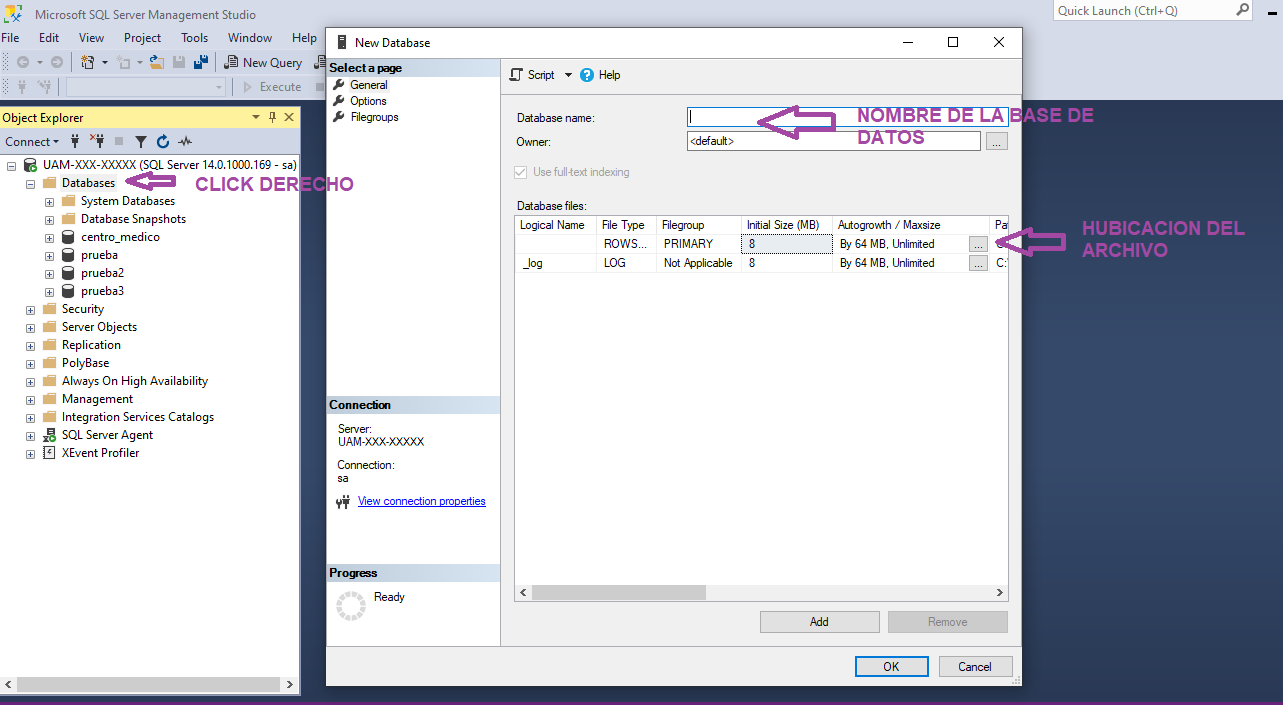
1. CREAR UN ARCHIVO DE BASES DE DATOS

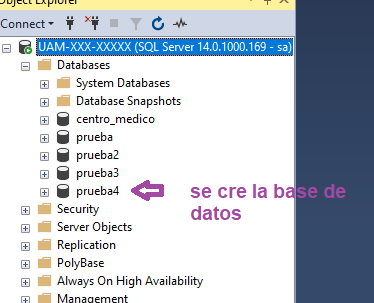
1.1 primero nos conectamos al motor de bases de datos.



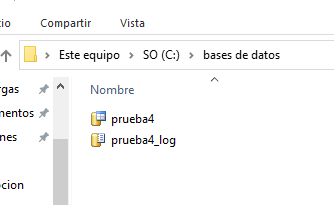
1.2 Hay 2 maneras de crear una base de datos

1.2.1 la primera

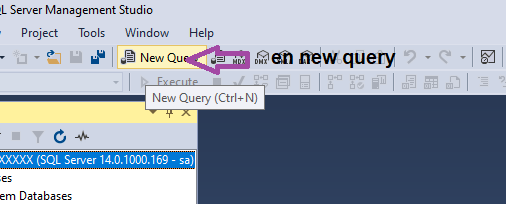




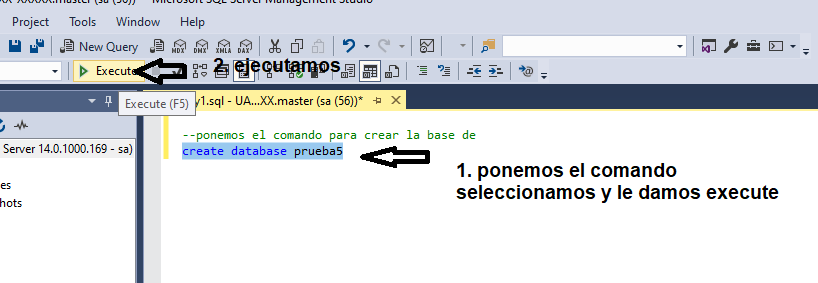
Me crea 2 archivos en la ubicación donde cree mi base de datos



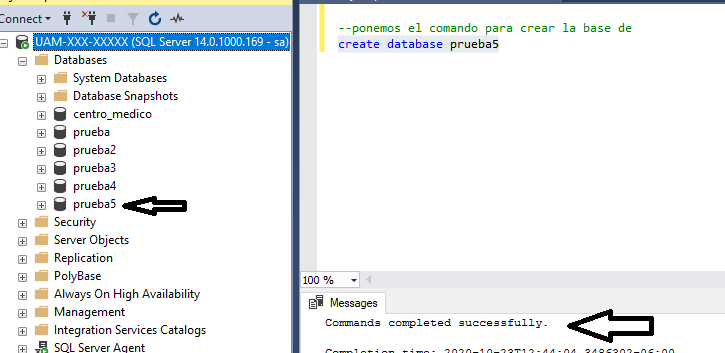
1.1.2 la segunda forma seria



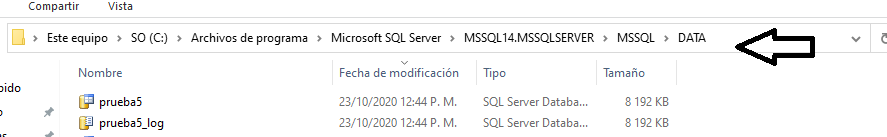
Ponemos: create database prueba5



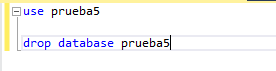
Aparece completado y se crea la base de datos llamada prueba 5



Se crea en



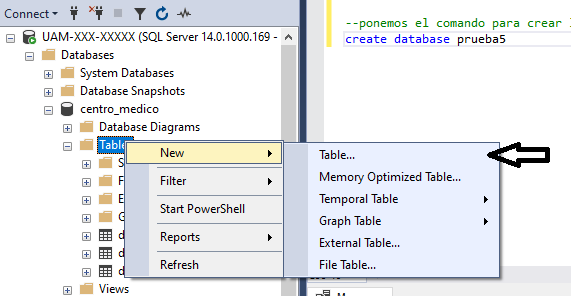
2, Para eliminar una base de datos seria:



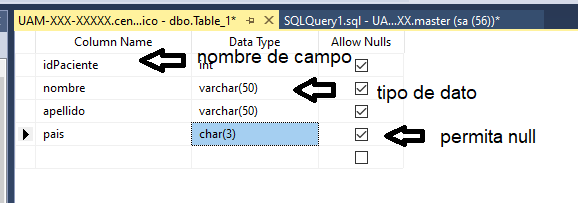
3. Crear una tabla hay 2 formas

* 1. Primera

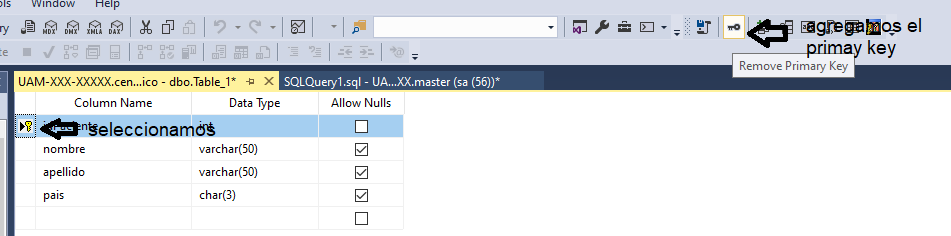
3.1.1 nos vamos a la base de datos y en tables click derecho new y table



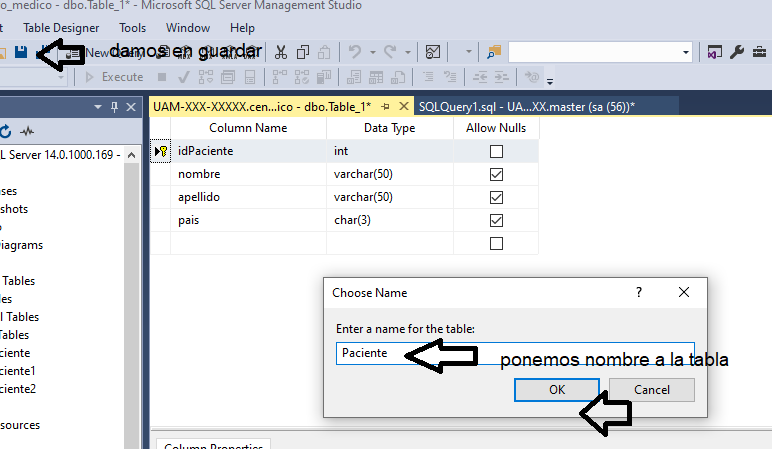
3.1.2 luego agregamos los campos, el tipo de dato y el valor de null o no null



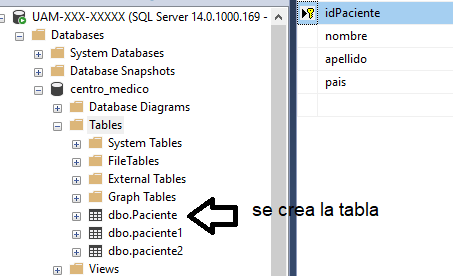
3.1.3 agregamos el primary key seleccionando el campo y tocando el botón de primary key.



* + 1. guardamos la tabla con un nombre en este caso paciente

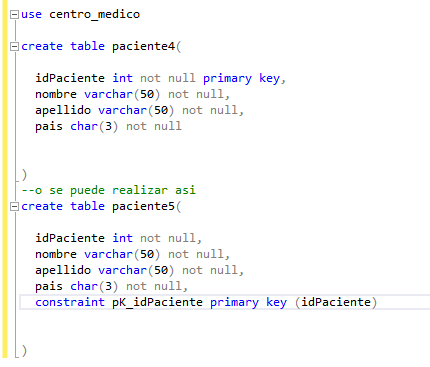


* + 1. se crea la tabla



3.2 Segunda forma de crear tabla

3.2.1 Se ingresa el siguiente código y hay 2 maneras de asignar el primary key



Código:

use centro\_medico

create table paciente4(

idPaciente int not null primary key,

nombre varchar(50) not null,

apellido varchar(50) not null,

pais char(3) not null

)

--o se puede realizar asi

create table paciente5(

idPaciente int not null,

nombre varchar(50) not null,

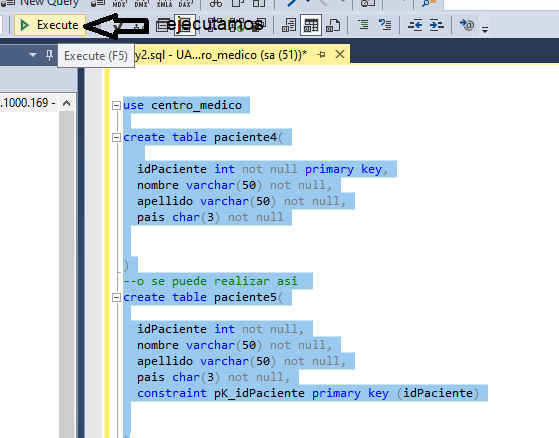
apellido varchar(50) not null,

pais char(3) not null,

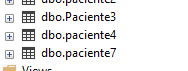
constraint pK\_idPaciente primary key (idPaciente)

)

3.2.2 seleccionamos el código y le damos en execute para crear la tabla



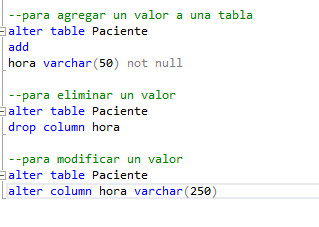
Se crea la tabla



4. Eliminar una tabla seria:



5 modificar los valores de una tabla seria:



Código:

--para agregar un valor a una tabla

alter table Paciente

add

hora varchar(50) not null

--para eliminar un valor

alter table Paciente

drop column hora

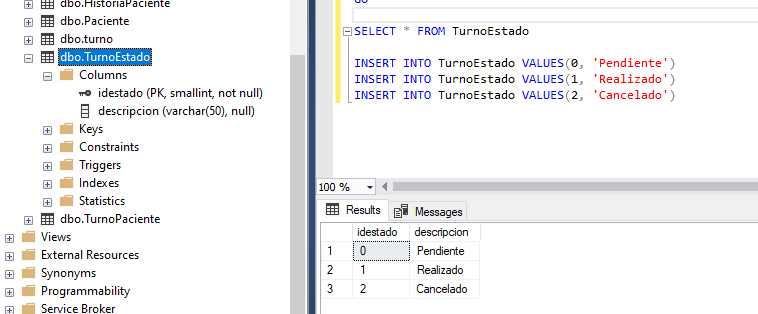
--para modificar un valor

alter table Paciente

alter column hora varchar(250)

X Insertar en una tabla:

x.1 Hacemos un select para ver los campos de la tabla he ingresamos datos en la tabla de la siguiente manera.



Código:

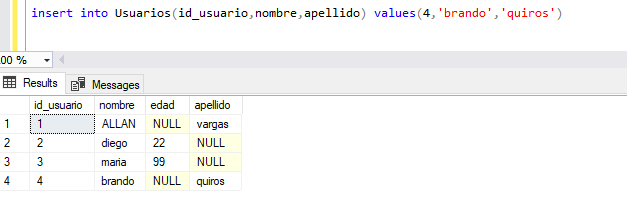
SELECT \* FROM TurnoEstado

INSERT INTO TurnoEstado VALUES(0, 'Pendiente')

INSERT INTO TurnoEstado VALUES(1, 'Realizado')

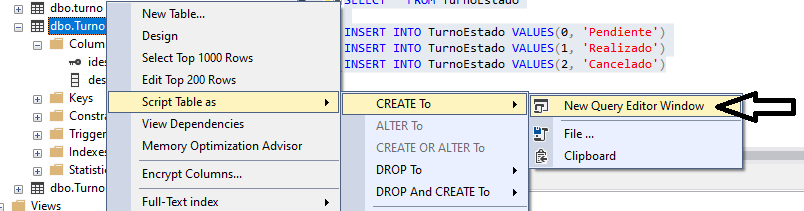
INSERT INTO TurnoEstado VALUES(2, 'Cancelado')

O

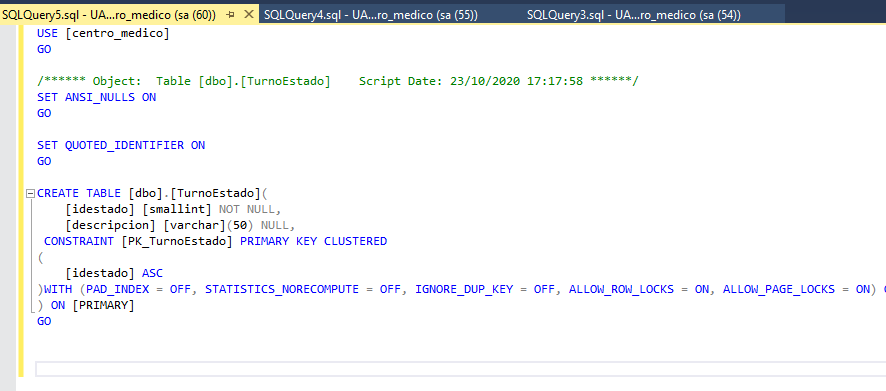


insert into Usuarios(id\_usuario,nombre,apellido) values(4,'brando','quiros')

x.2, para sacar un script con la creación de la tabla se hace de la siguiente manera.



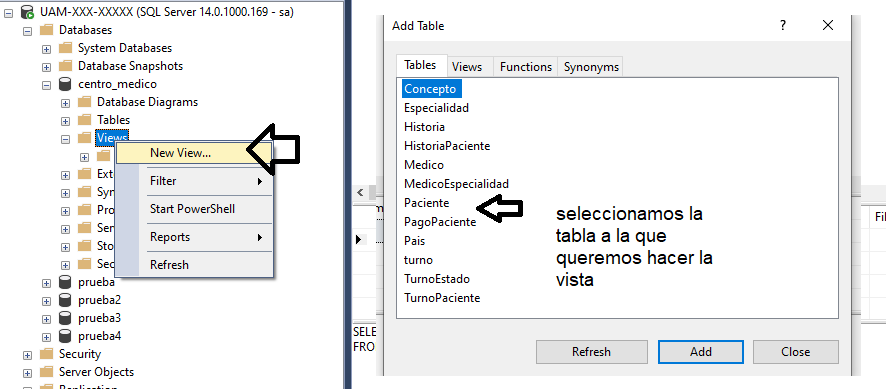
Genera el código de scrip de esa tabla



7. Select y vistas

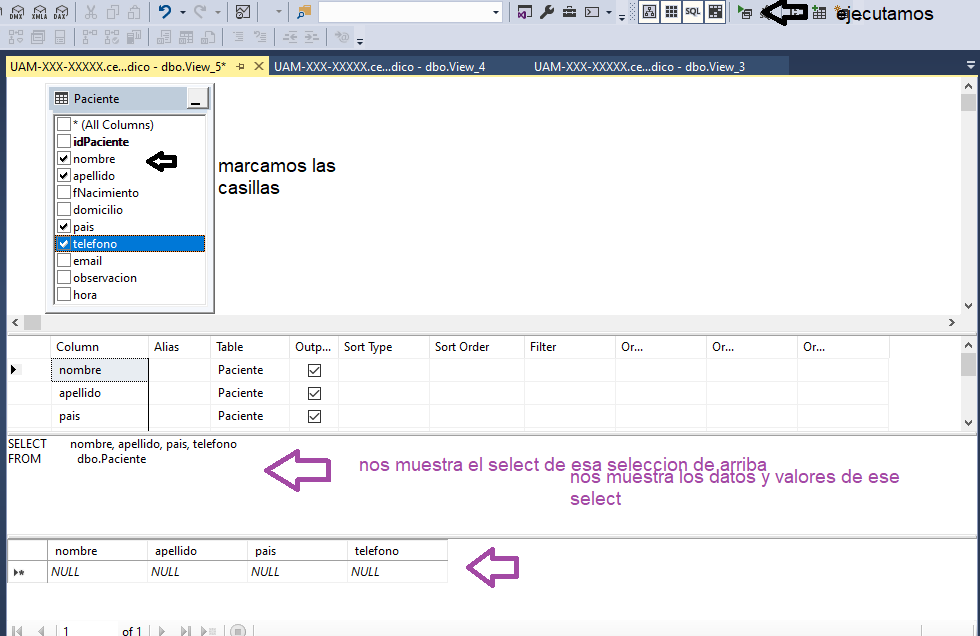
* 1. **vistas** para poder crear un select con características especificas

nos vamos a la base de datos y en vistas o views le damos click derecho new view y seleccionamos la tabla a la cual queremos crear un select

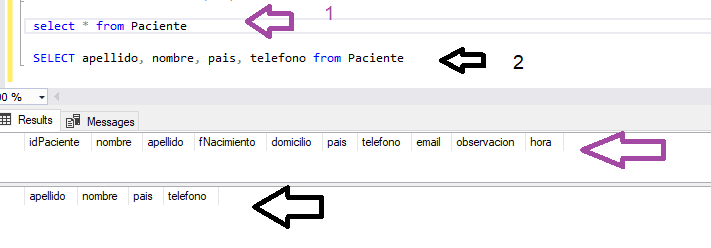


Luego seleccionamos los campos que queremos que tengan el select

Y le damos execute sql y nos mostrara lo siguiente



7.2 **Select** básico seria:

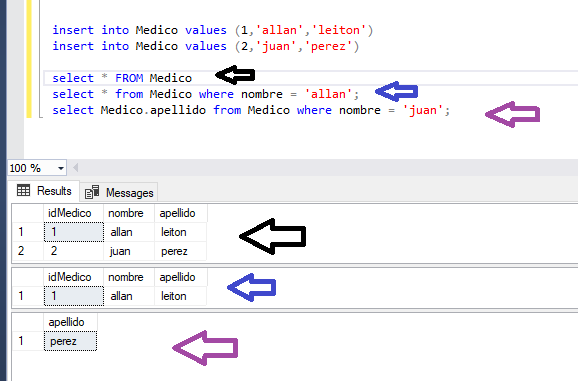


Código:

select \* from Paciente

SELECT apellido, nombre, pais, telefono from Paciente

8. Select con where



Código:

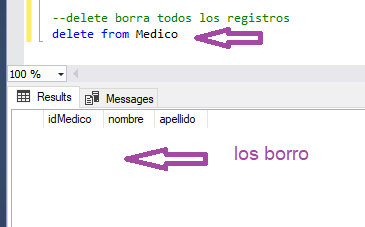
select \* FROM Medico

select \* from Medico where nombre = 'allan';

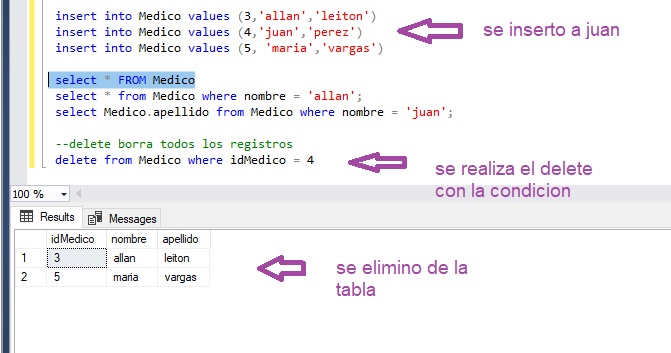
select Medico.apellido from Medico where nombre = 'juan';

9. Delete, drop y truncate :

9.1 delete borra todos los registros de una table. Si el primary fuera identity al ingresar otro valor sigue hasta donde quedo con la secuencia



O



Nota el idMedico es identity por lo que sigue la secuencia del identity

Código

delete from medico

delete from Medico where idMedico = 4

TRUNCATE

Si se borran todos los registros de una tabla y se vuelve a ingresar los datos donde la primary key en identity, vuelve a empezar el indentity desde 1



Código

truncate table Medico

DROP

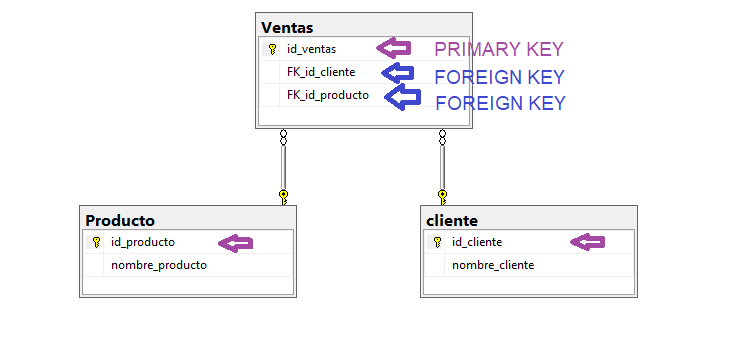
Borra toda la tabla

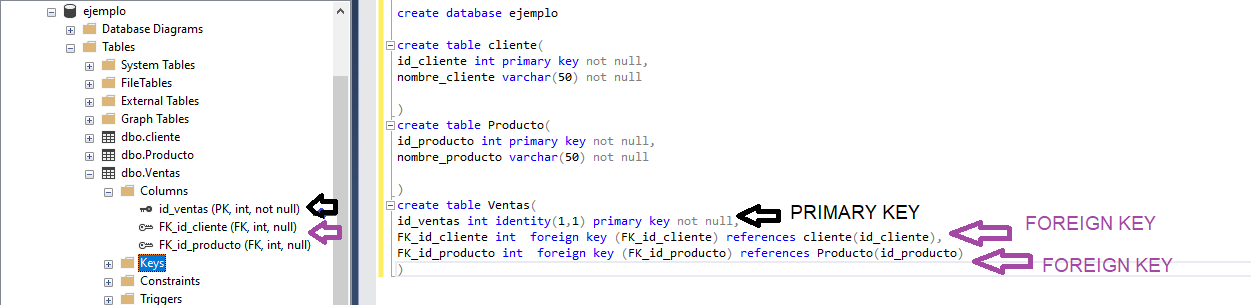


Código

drop table Medico

10. CREAR TABLA CON PRIMARY KEY Y FOREIGN KEY





CODIGO

use ejemplo

create database ejemplo

create table cliente(

id\_cliente int primary key not null,

nombre\_cliente varchar(50) not null

)

create table Producto(

id\_producto int primary key not null,

nombre\_producto varchar(50) not null

)

create table Ventas(

id\_ventas int identity(1,1) primary key not null,

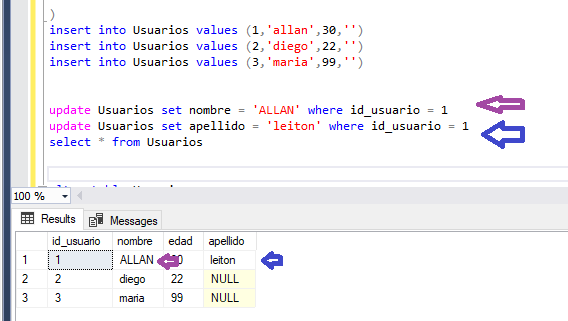
FK\_id\_cliente int foreign key (FK\_id\_cliente) references cliente(id\_cliente),

FK\_id\_producto int foreign key (FK\_id\_producto) references Producto(id\_producto)

)

11. UPDATE.

Lo utilizamos para actualizar registros de una tabla

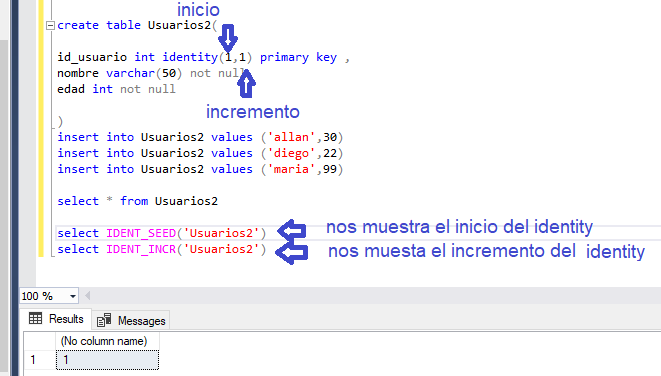


Codigo

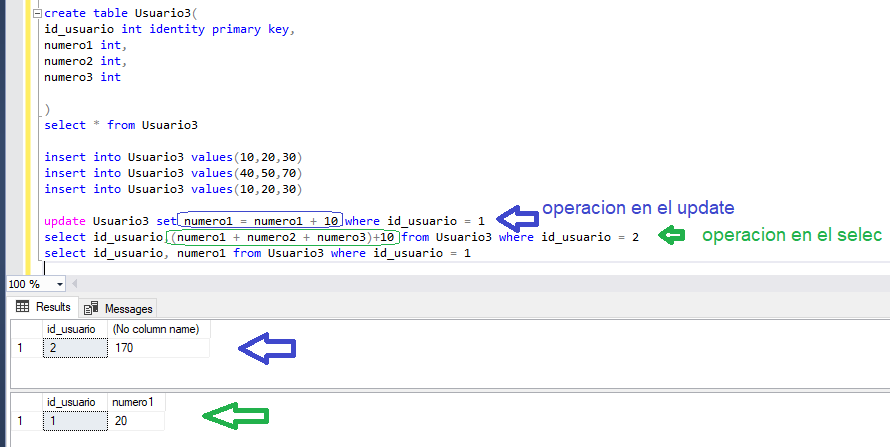
update Usuarios set nombre = 'ALLAN' where id\_usuario = 1

update Usuarios set apellido = 'leiton' where id\_usuario = 1

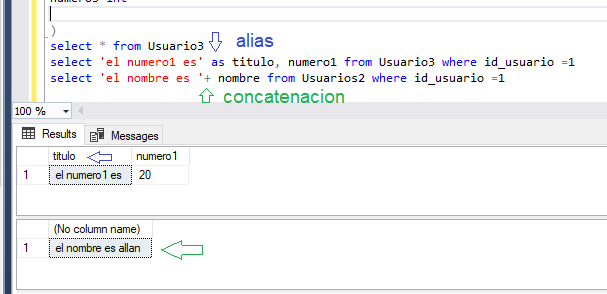
12. IDENTITY.



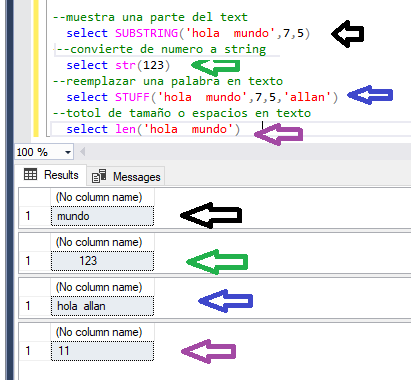
13. OPERADORES ARITMETRICOS.

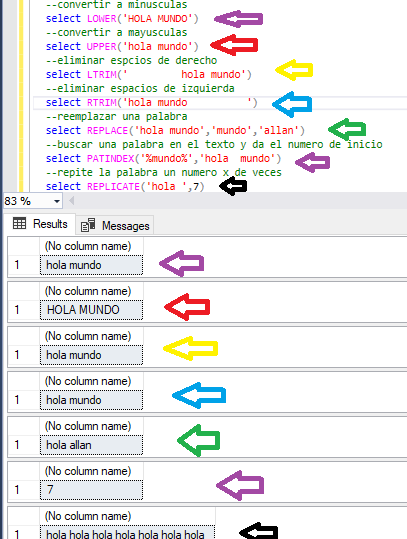


14 CONCATENACION Y ALIAS

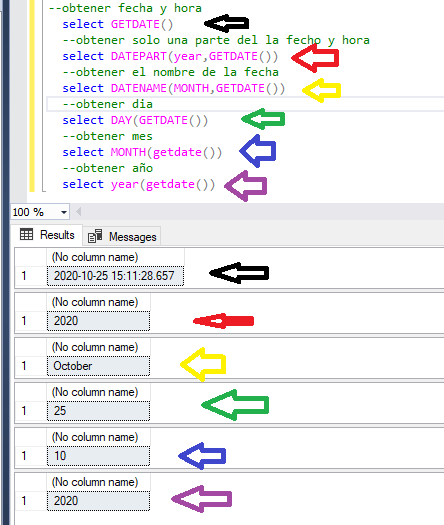


15 Manejo de cadenas

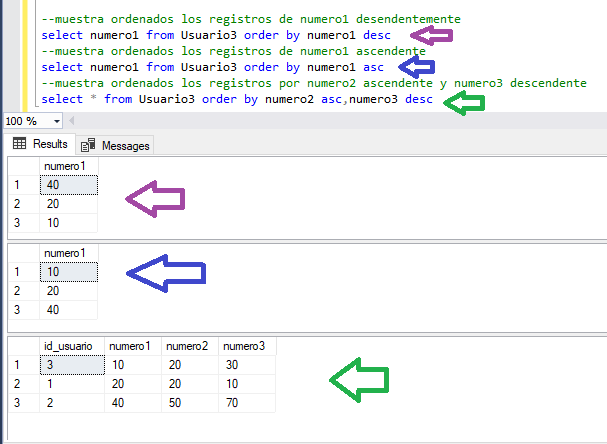




1X FECHA Y HORA



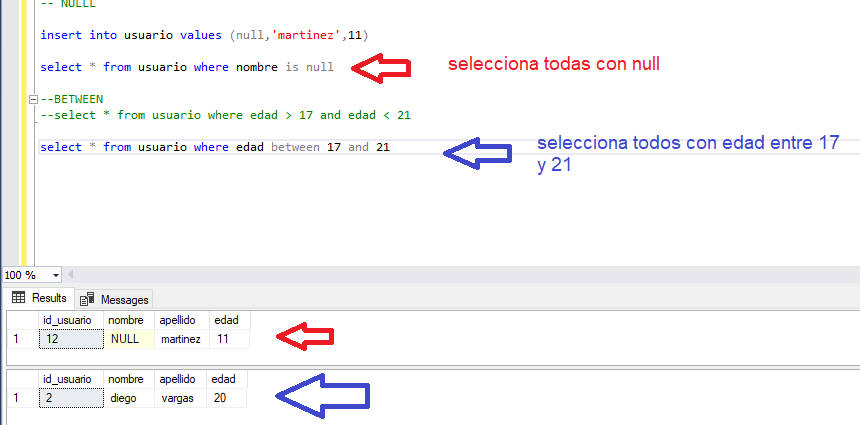
17 ORDENAR POR



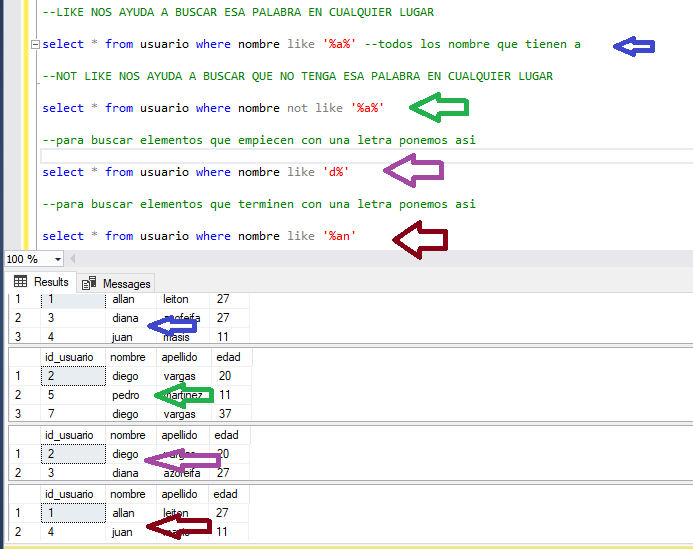
18 OPERADORES LOGICOS AND OR NOT

<https://www.youtube.com/watch?v=brBIaGk5X_A&list=PL6hPvfzEEMDaU4aiS389oXamdN8sip856&index=20>

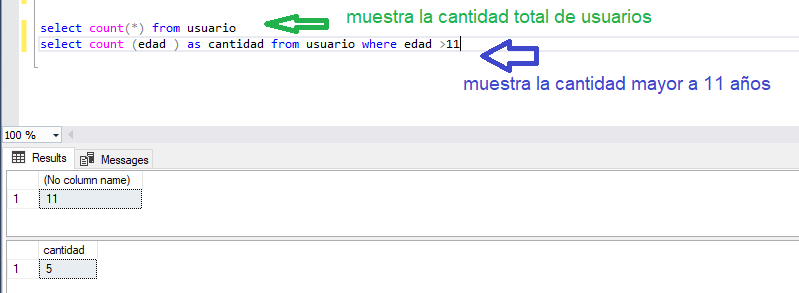
19 IS NULL - BETWEEN



20 LIKE, NOT LIKE



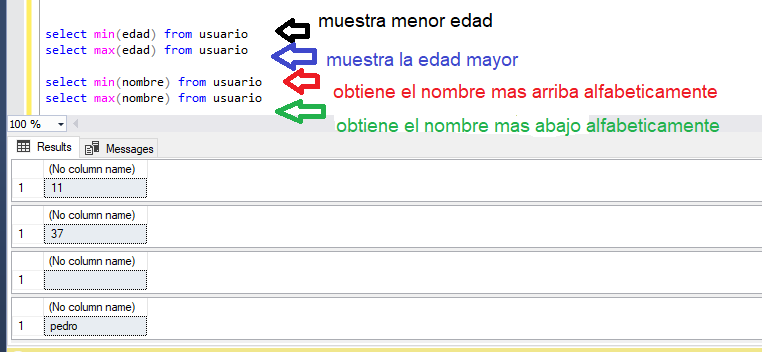
21 COUNT



22 SUM – AVG



23 MIN – MAX



24 GROUP BY / HAVING

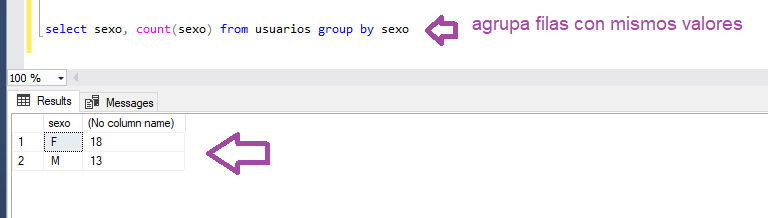
La instrucción GROUP BY de SQL

La declaración GROUP BY agrupa las filas que tienen los mismos valores en filas de resumen, como "encontrar el número de clientes en cada país".

La instrucción GROUP BY se usa a menudo con funciones agregadas (COUNT, MAX, MIN, SUM, AVG) para agrupar el conjunto de resultados por una o más columnas.

GRUPO POR Sintaxis

SELECT *nombre\_columna*  
FROM *nombre\_tabla*  
WHERE *condicion*  
GROUP BY *nombre\_columna*ORDER BY *nombre\_columna*



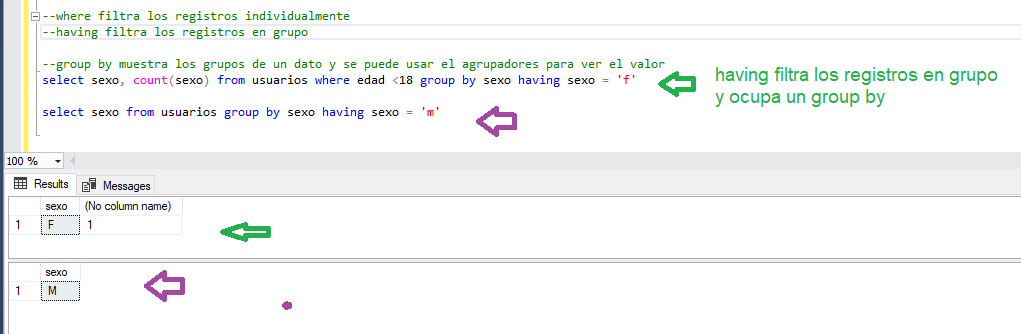
¿Qué hace la cláusula HAVING en una consulta?

La cláusula HAVING es como WHERE pero opera en registros agrupados devueltos por GROUP BY.  
HAVING se aplica a los registros de grupo resumidos, mientras que DONDE se aplica a los registros individuales.  
Solo se devolverán los grupos que cumplan con los criterios HAVING.

HAVING requiere que esté presente una cláusula GROUP BY.  
Tanto WHERE como HAVING se pueden utilizar en la misma consulta al mismo tiempo.

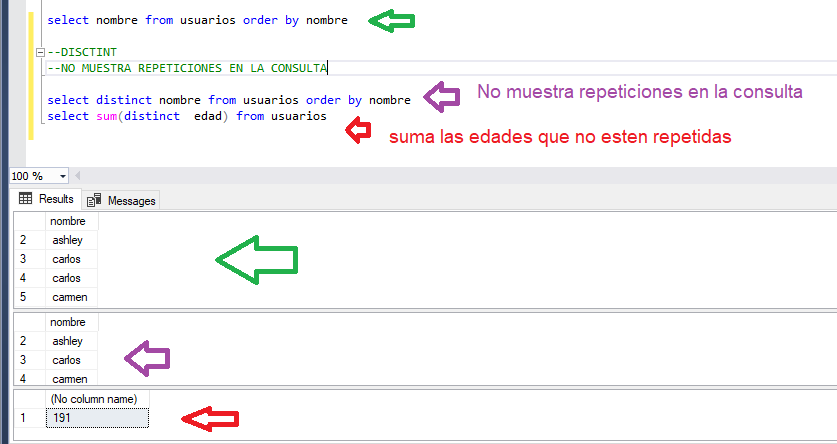
HAVING Syntax

SELECT *nombre\_columna*  
FROM *nombre\_tabla*  
WHERE *condicion*  
GROUP BY *nombre\_columna*HAVING *condicion*

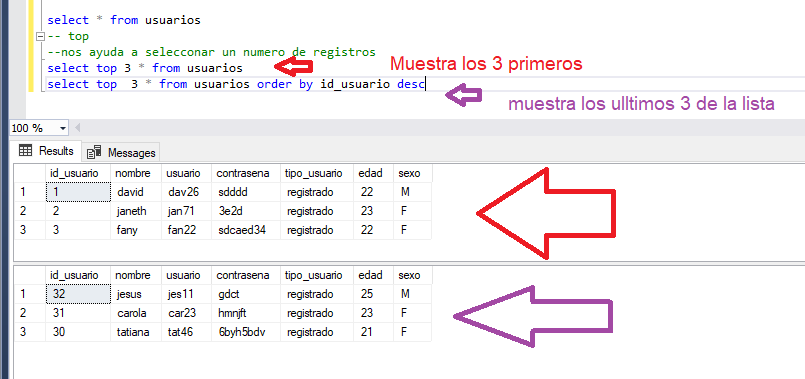


25 DISTINCT

No muestra repeticiones en una consulta



26 TOP



26 RESPALDO Y RESTAURACION DE BASES DE DATOS

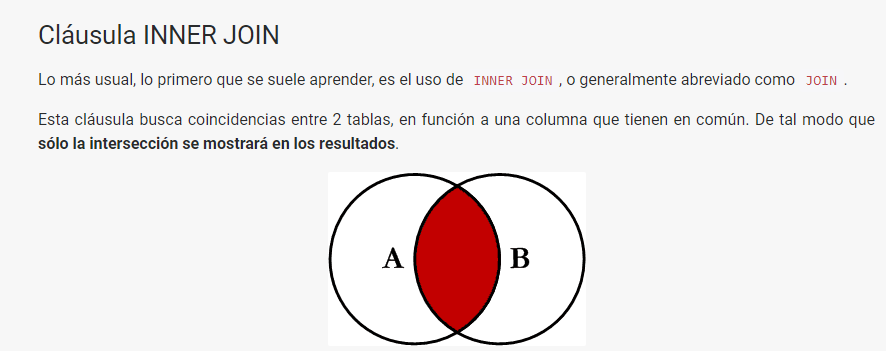
RESPALDO

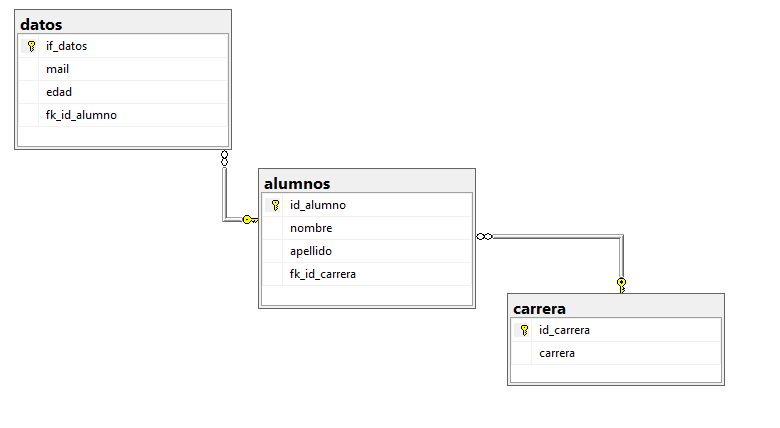
<https://www.youtube.com/watch?v=ve6dGwYycHc&list=PL6hPvfzEEMDaU4aiS389oXamdN8sip856&index=30>

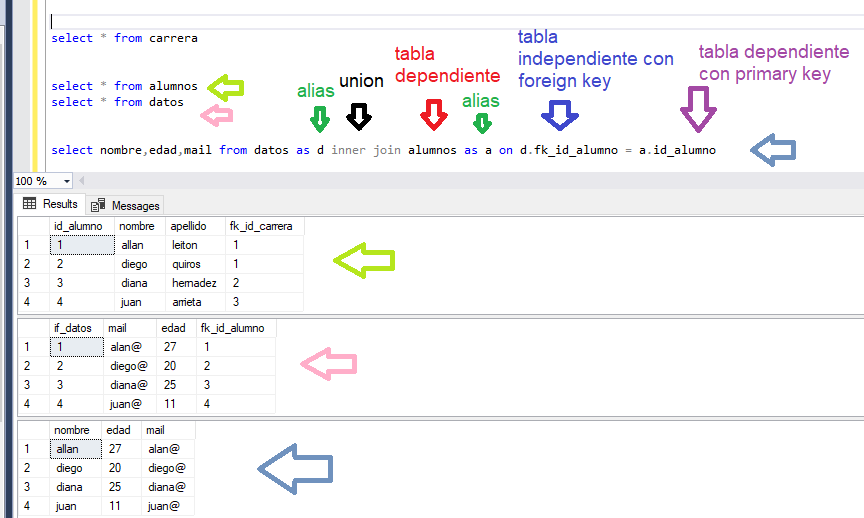
REVISAR VIDEO

27 INNER JOIN

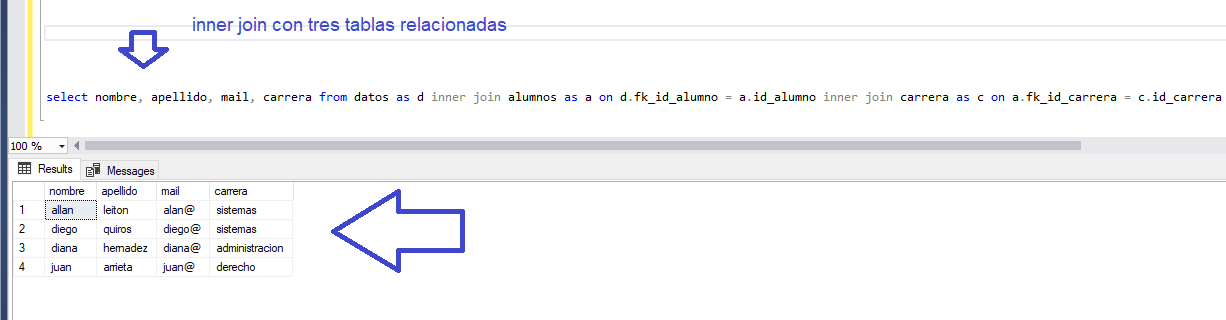
Muestra los datos de 2 o mas tablas relacionadas entre si





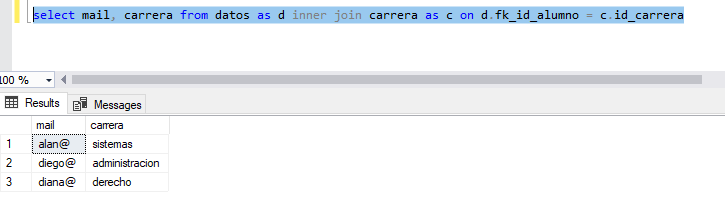


select nombre,edad,mail from datos as d inner join alumnos as a on d.fk\_id\_alumno = a.id\_alumno



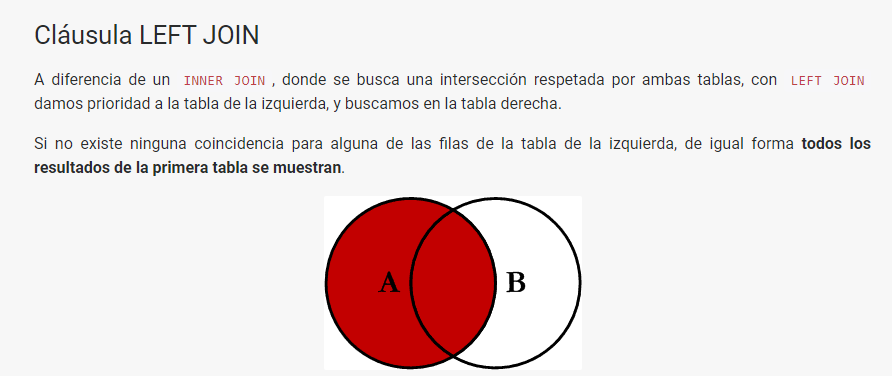
select nombre, apellido, mail, carrera from datos as d inner join alumnos as a on d.fk\_id\_alumno = a.id\_alumno inner join carrera as c on a.fk\_id\_carrera = c.id\_carrera

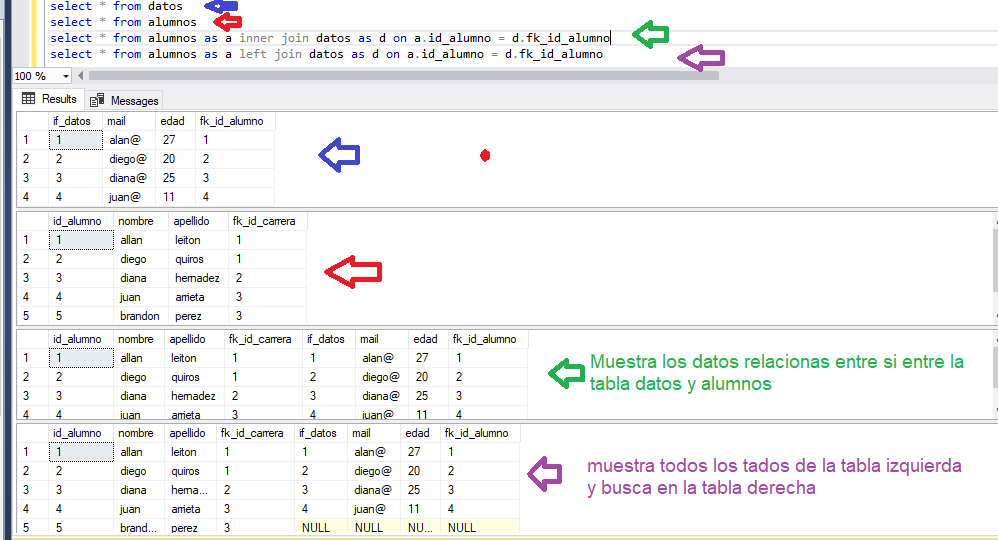
inner join entre la tabla datos y carrera:



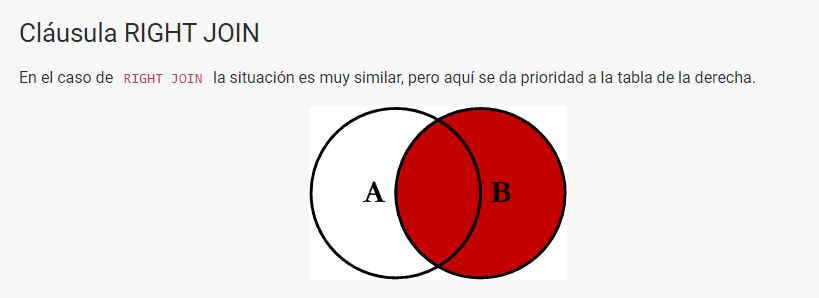
select mail, carrera from datos as d inner join carrera as c on d.fk\_id\_alumno = c.id\_carrera

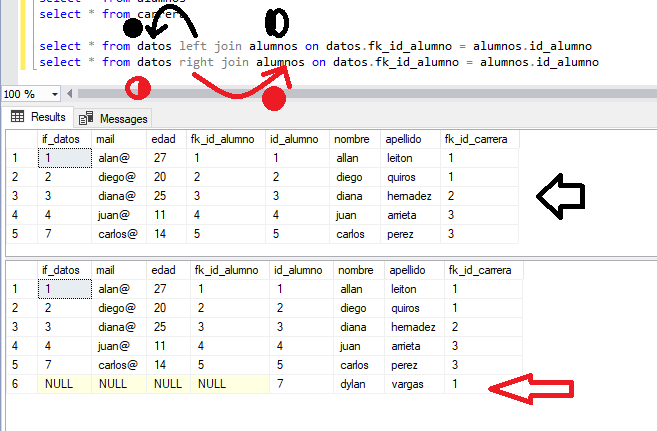
28 LEFT JOIN



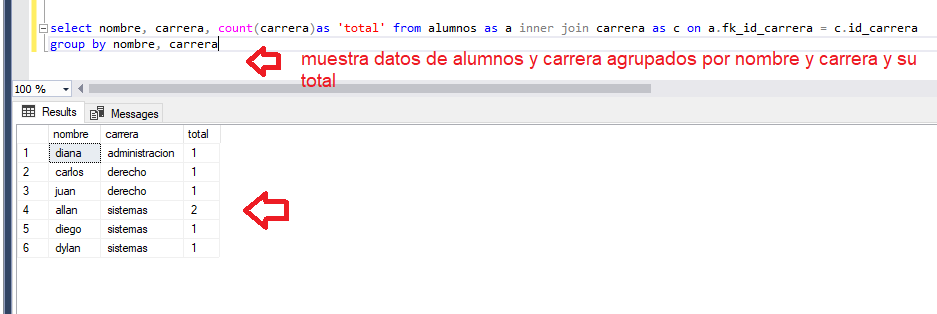


29 RIGHT JOIN



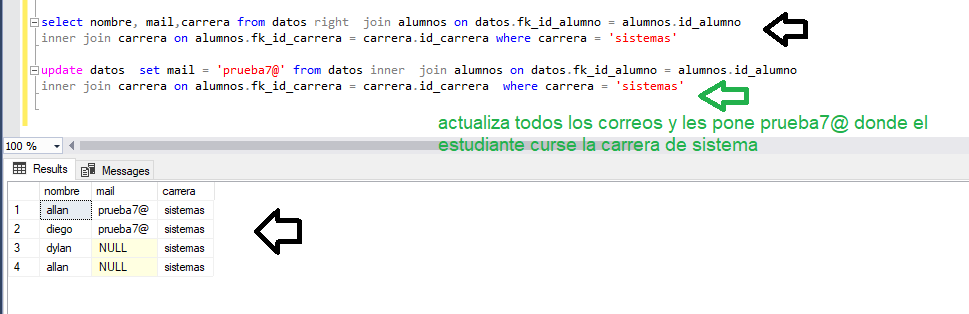


30 JOIN AND GROUP BY

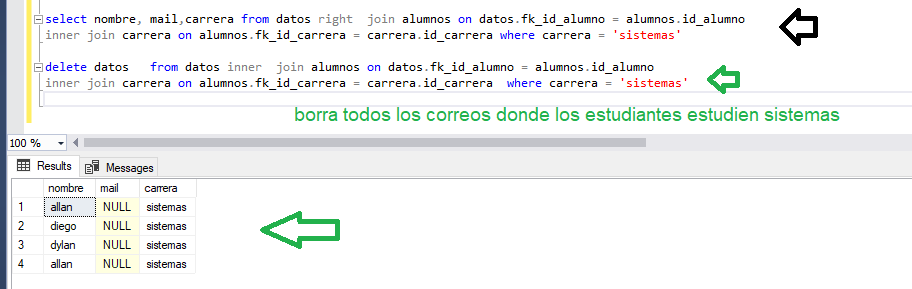


31 JOIN, UPDATE, AND DELETE

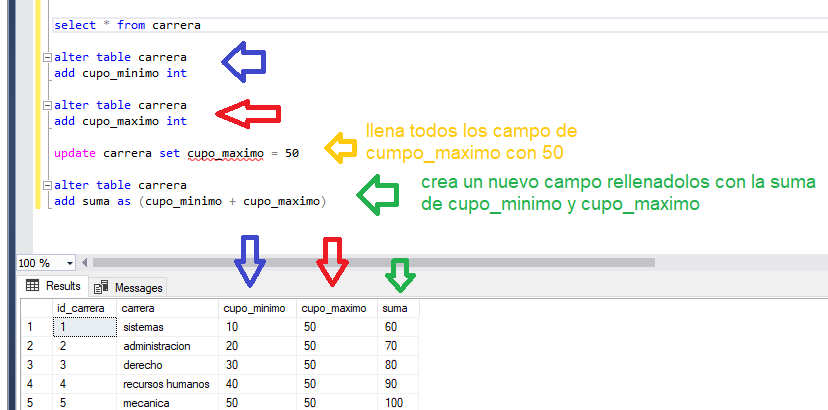
Update



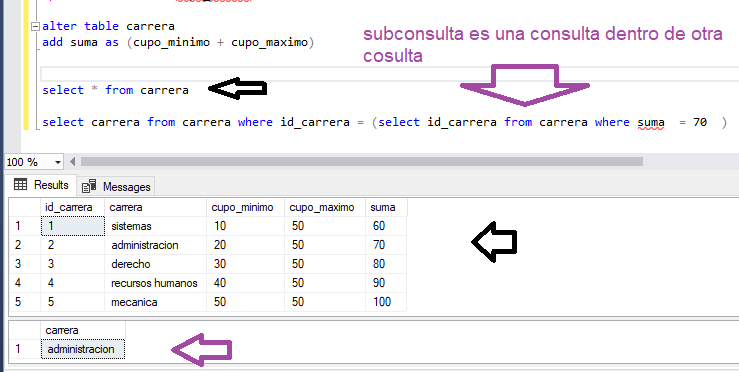
Delete



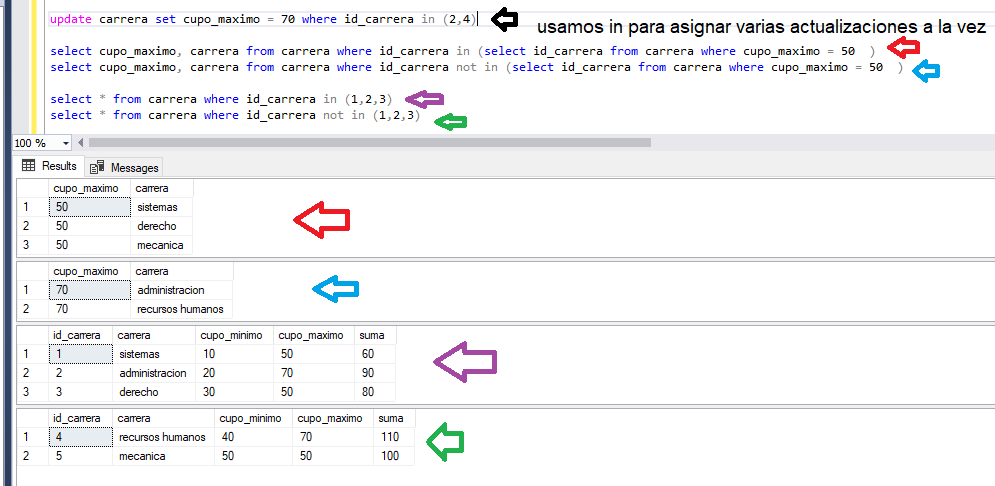
32 CAMPOS CALCULADOS



33 SUBCONSULTAS



34 SUBCONSULTAS CON IN & NOT IN

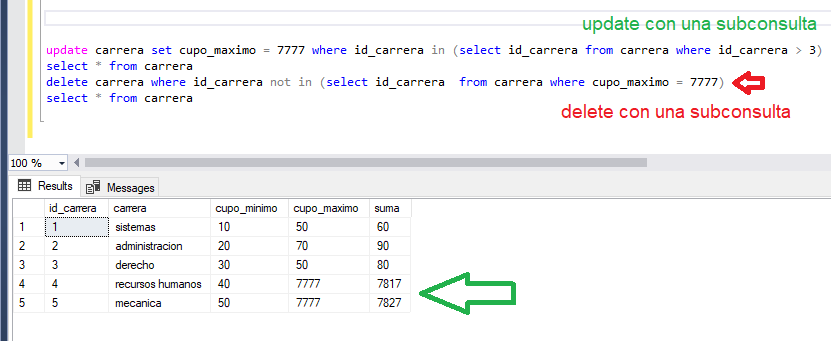


35 SUBCONSULTAS CON ANY & ALL

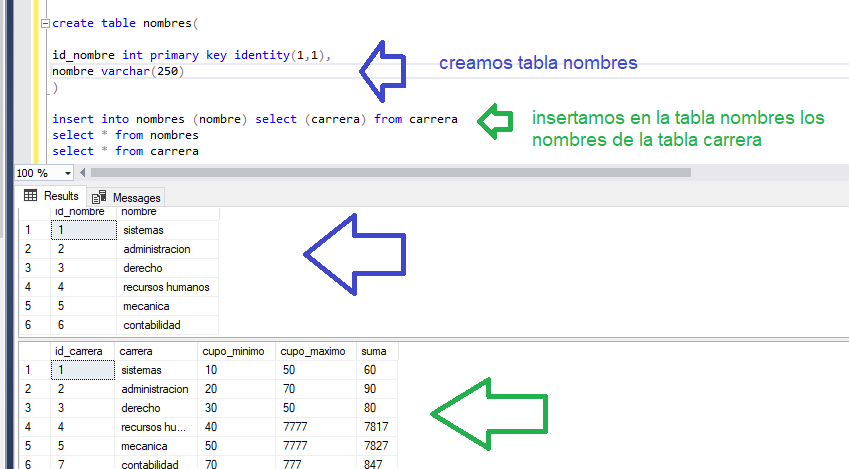
<https://www.youtube.com/watch?v=yLv1wPpWEGc&list=PL6hPvfzEEMDaU4aiS389oXamdN8sip856&index=41>

REVISAR VIDEO

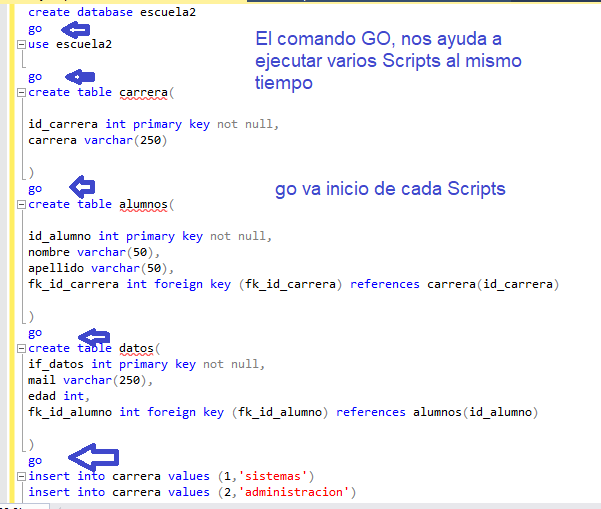
3X SUBCONSULTAS UPDATE & DELETE



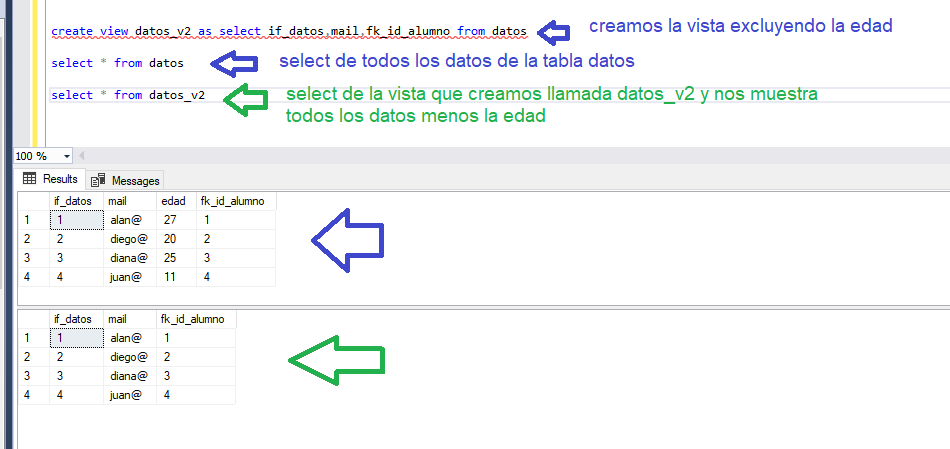
37 SUBCONSULTAS CON INSERT



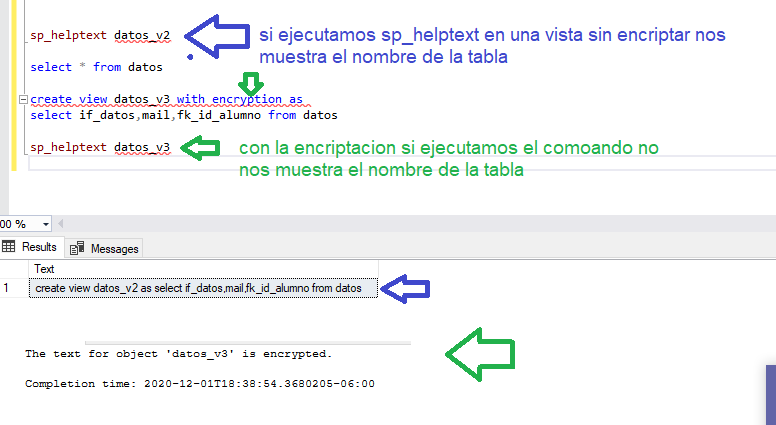
38 COMANDO GO



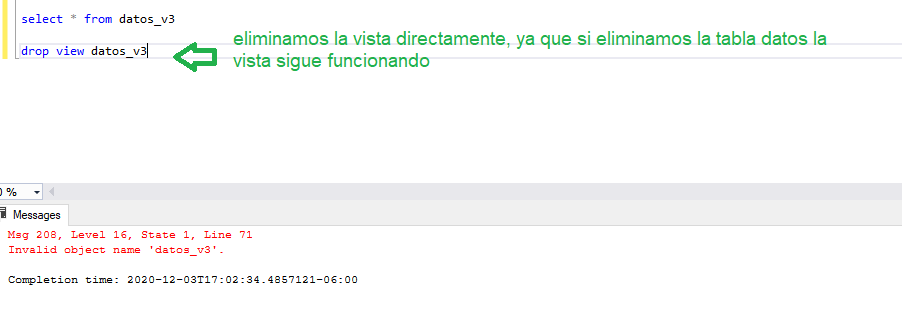
39 VISTAS



39 CIFRADO DE VISTA O ENCRIPTACION DE VISTAS

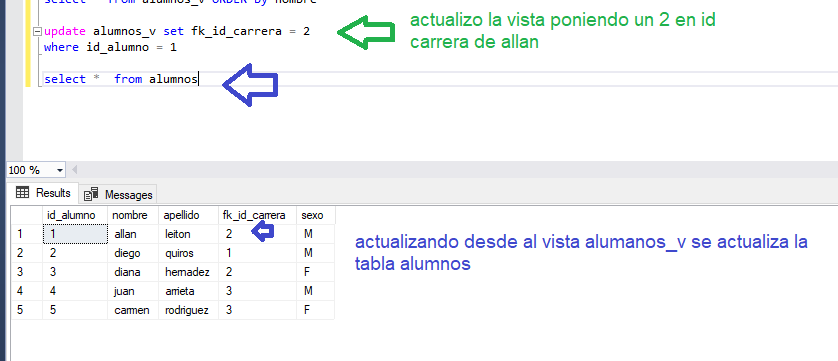


40 ELIMINAR VISTAS



41 UPDATE Y DELETE CON VISTAS

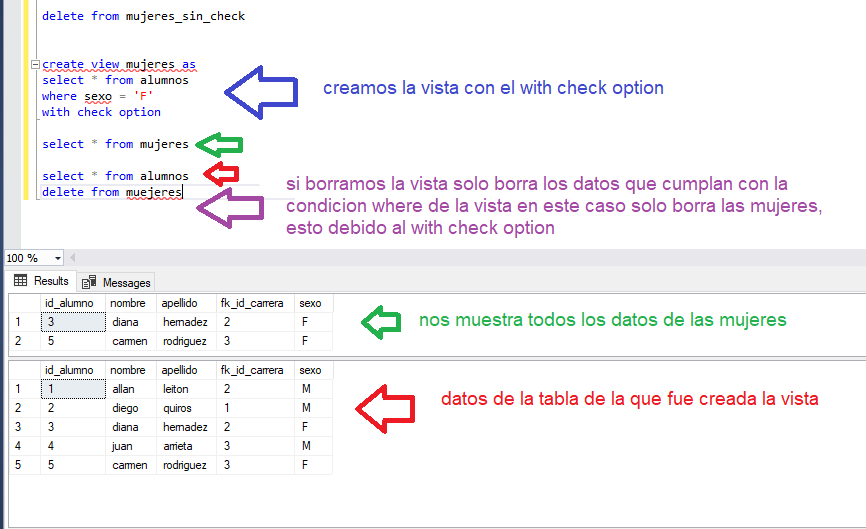
Update



Delete



42 sentencia witch check option en vistas



. Esa opcion "with check option" como su nombre lo dice, impide inserciòn o modificación si no se cumple con las condiciones where de la vista.

Por ejemplo,

create view copy as

select \* from usuarios

where edad > 5

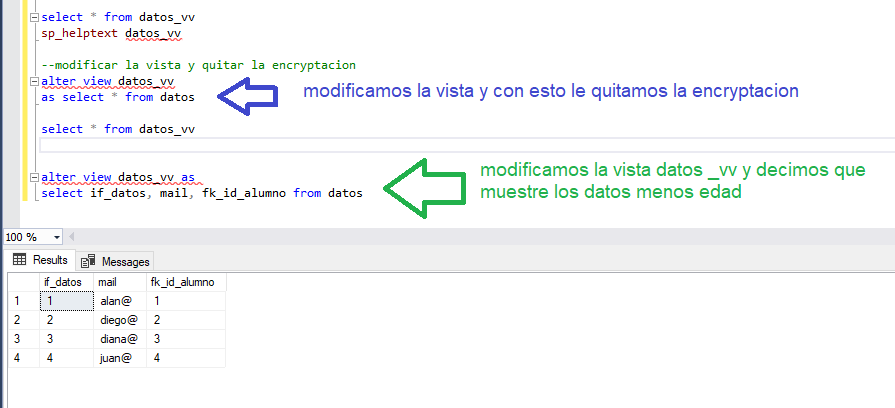
with check option

\*\*\*\*Si intentas cambiar la edad de un usuario de 8 años a 4 años, por ejemplo, te marca error, porque no cumple la condición edad > 5. \*\*\*\*

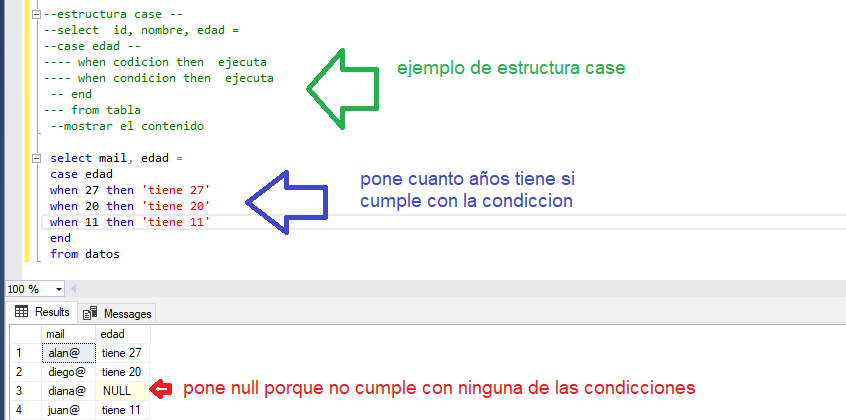
Si intentas insertar un usuario, con edad de 3 años, te marca error, porque no cumple la condición edad >5.

Sin la opción "with check option", podrias insertar o modificar sin importar la condicion where, estos cambios se ven en la tabla raiz, pero no se verían en la vista.

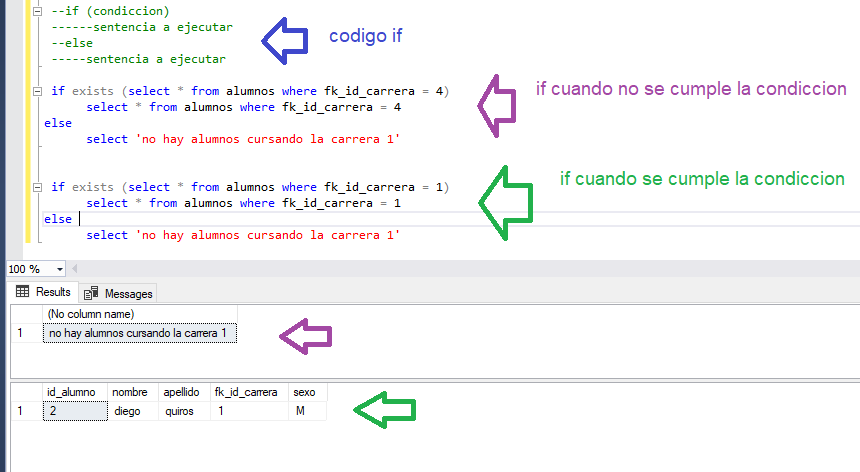
43 VISTAS MODIFICADA



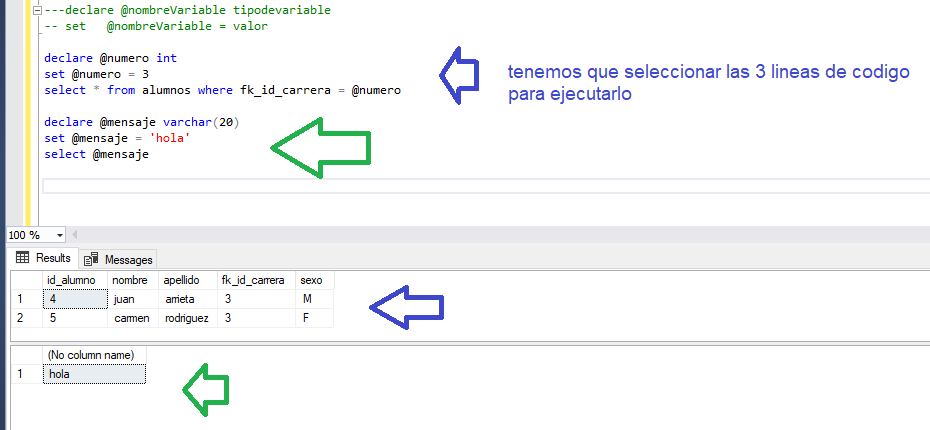
44 LENGUAJE CONTROL DE FLUJO CASE



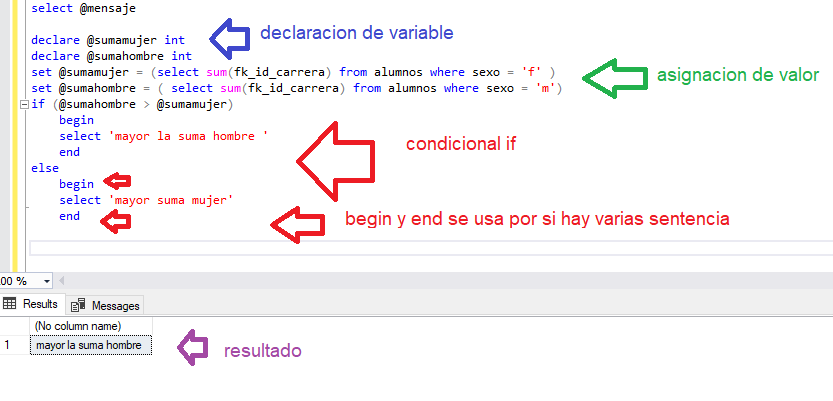
45 LENGUAJE CONTROL DE FLUJO IF



4x variables

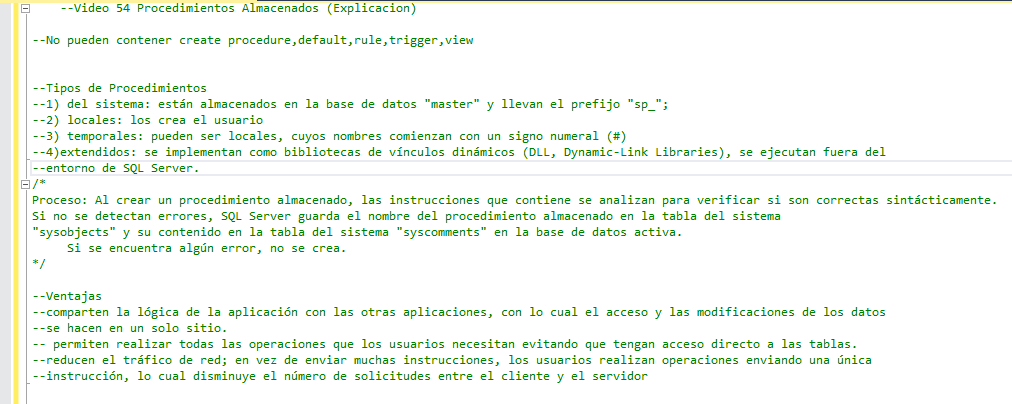


Ejemplo 2

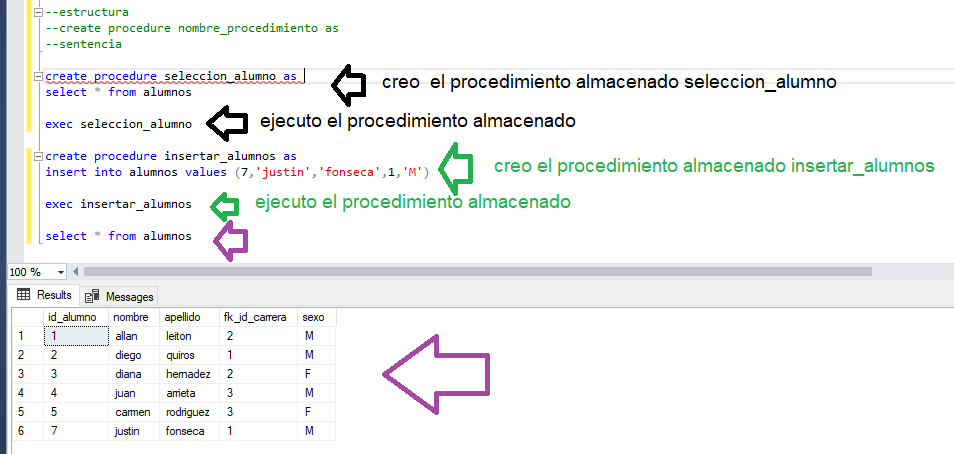


47 PROCEDIMIENTOS ALMACENADOS CREAR Y EJECUTAR

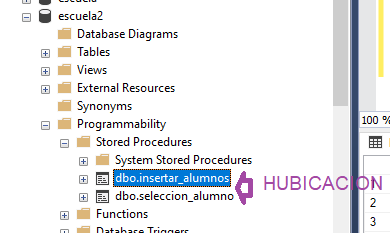
Es un conjunto de sentencias almacenadas en el servidor.

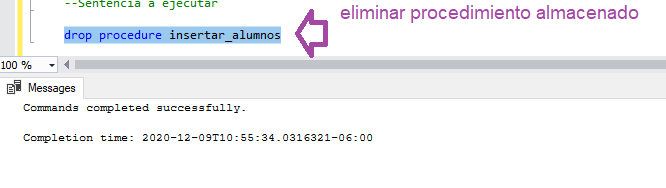


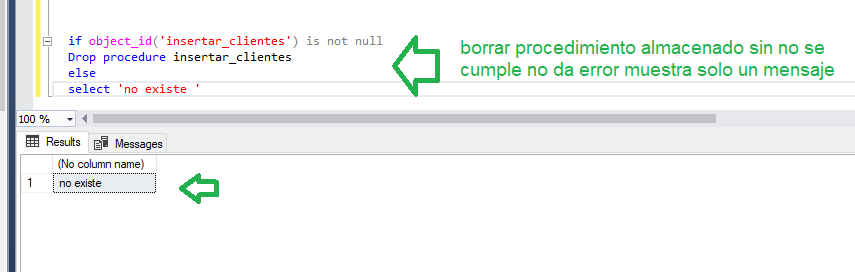
Crear un procedimiento almacenado



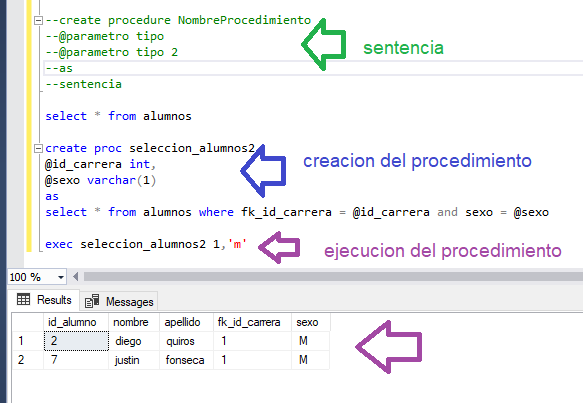
48 PROCEDIMIENTOS ALMACENADOS ELIMINAR



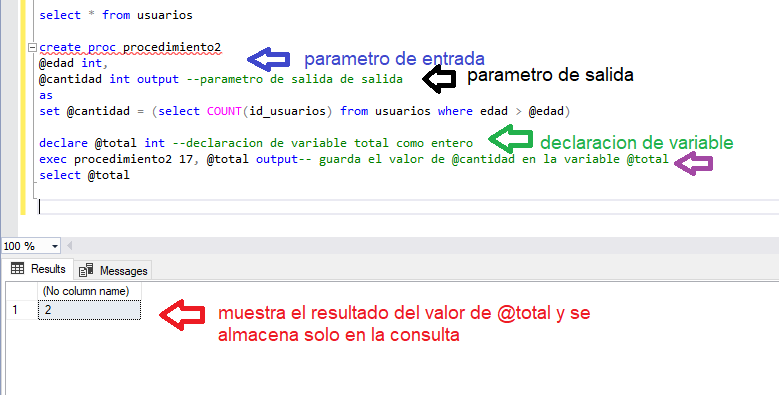




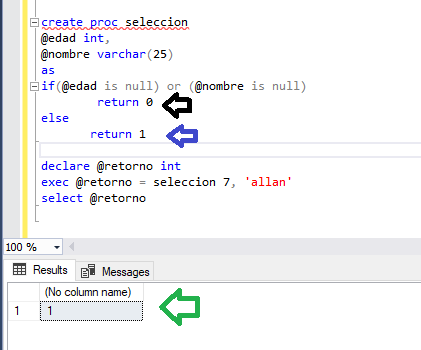
49 PROCEDIMIENTOS ALMACENADOS PARAMETROS DE ENTRADA.



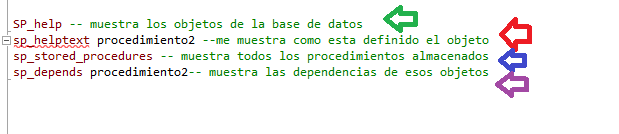
50 PROCEDIMIENTOS ALMACENADOS PARAMETROS DE SALIDA.

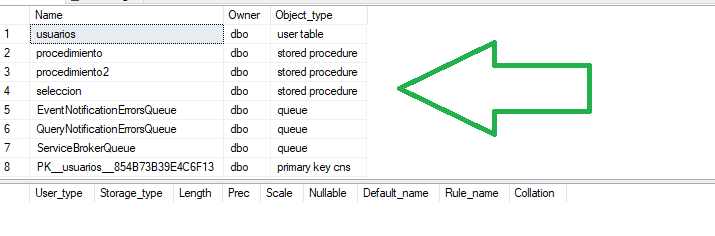


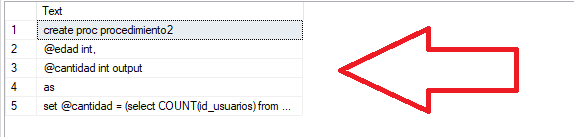
51 PROCEDIMIENTOS ALMACENADOS RETURN

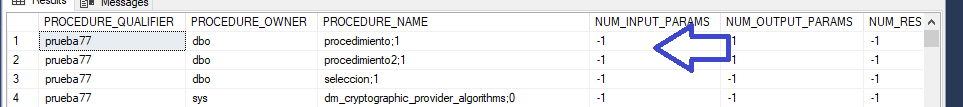


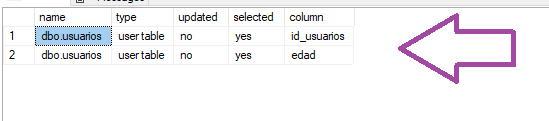
52 PROCEDIMIENTOS ALMACENADOS INFORMACION.



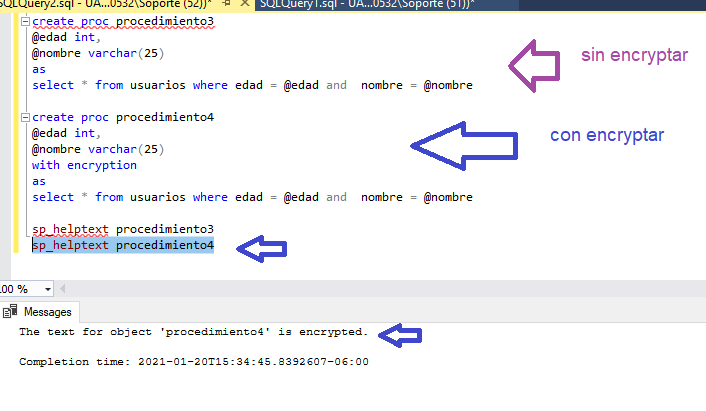








53 PROCEDIMIENTOS ALMACENADOS ENCRYPTACION .



FALTA TRIGGERS

https://www.youtube.com/watch?v=VNtmaJ2rJ3M&list=PL6hPvfzEEMDaU4aiS389oXamdN8sip856&index=64

y pendiente creación de base de datos relacionadas