***RESPUESTAS A LA TAREA DE LA SECCION UNIENDO DIFERENTES TABLAS***

1.- Revisa el video de la lectura llamada “Inner Join” para que prepares la base de datos que vamos a analizar, y, mediante la unión de las tablas “Datos\_Personales” y “Actividades\_Extra”, obtén una lista de los nombres, los correos y la Clase Extra que están tomando los alumnos que se encuentran registrados en alguna de las Actividades Extra-Curriculares. Ordena la lista de forma ascendente en base a la Actividad en la que se encuentran registrados los alumnos

***RESPUESTA:***

USE Alumnos;

SELECT Datos\_Personales.Nombre,

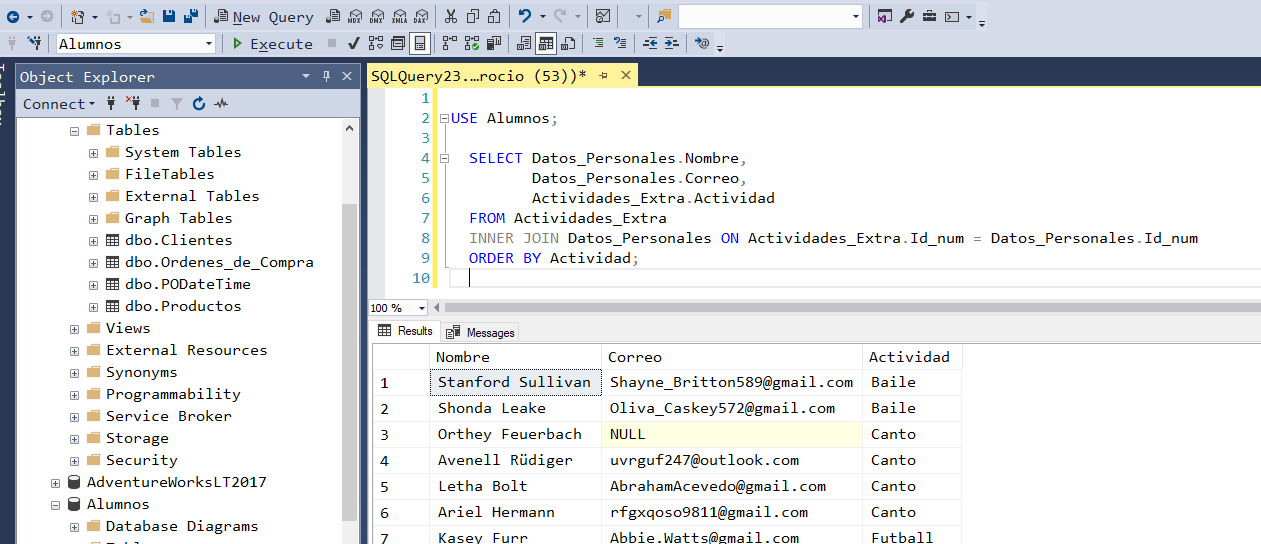
Datos\_Personales.Correo,

Actividades\_Extra.Actividad

FROM Actividades\_Extra

INNER JOIN Datos\_Personales ON Actividades\_Extra.Id\_num = Datos\_Personales.Id\_num

ORDER BY Actividad;



2.- Utilizando las Tablas Actividades\_Extra y Datos\_Personales, obtén una lista de los Números de Identificación de los alumnos que están incluidos en la lista de Actividades Extra-curriculares pero que no se encuentran incluidos en la lista de alumnos de la Tabla Datos\_personales

***RESPUESTA:***

USE Alumnos;

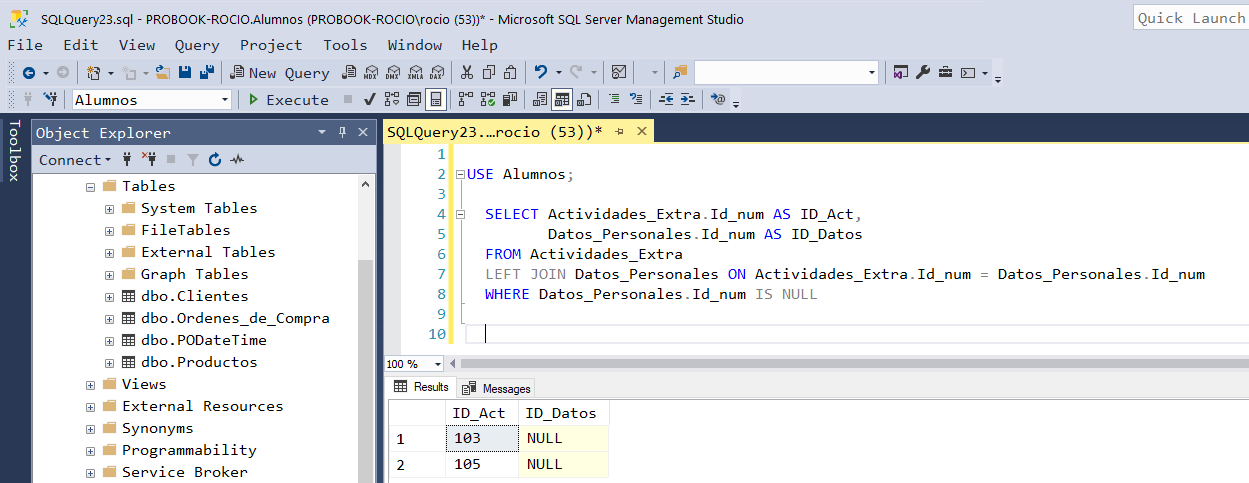
SELECT Actividades\_Extra.Id\_num AS ID\_Act,

Datos\_Personales.Id\_num AS ID\_Datos

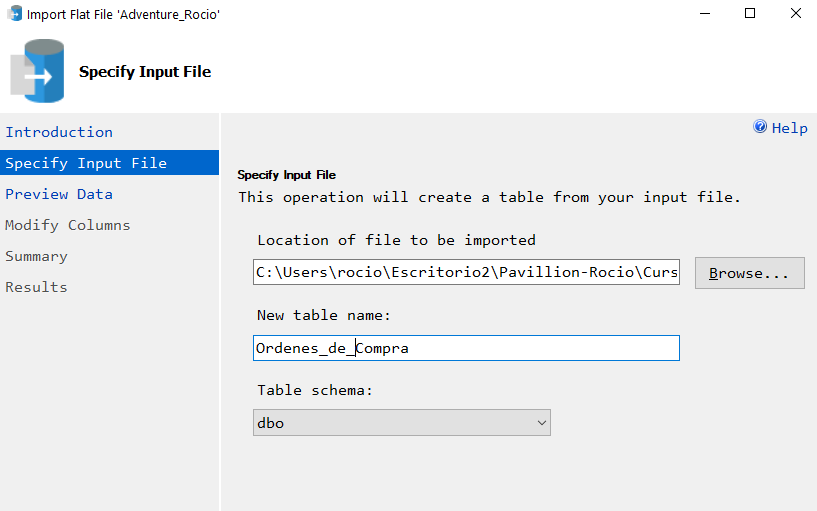
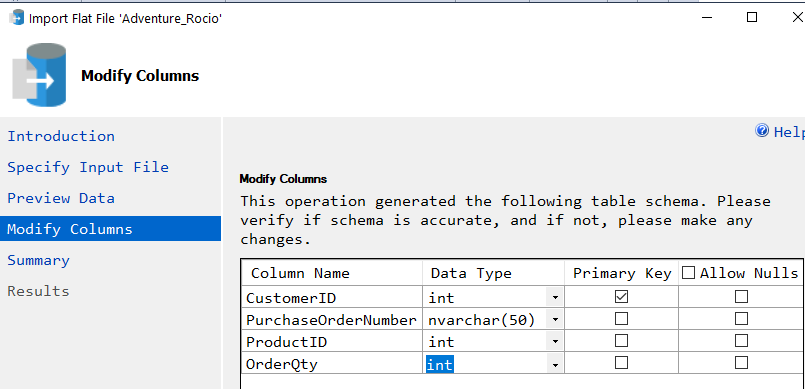
FROM Actividades\_Extra

LEFT JOIN Datos\_Personales ON Actividades\_Extra.Id\_num = Datos\_Personales.Id\_num

WHERE Datos\_Personales.Id\_num IS NULL



3.- Si todavía no has cargado la información de las tablas Clientes, Ordenes\_de\_Compra y Productos, de la base de datos que creamos en el video “Cargando un archivo csv en SQL Server”, es momento de hacerlo. Ten especial cuidado al momento de definir el formato de las columnas y asegúrate de que OrderQty se cargue como una variable numérica.

4.- Utilizando las tablas Clientes, Ordenes\_de\_Compra y Productos, de la base de datos que creamos en el video “Cargando un archivo csv en SQL Server”, obtén una lista de los productos adquiridos por cada uno de los Clientes, incluyendo sus descripciones y las cantidades ordenadas

***RESPUESTA:***

USE Adventure\_Rocio;

SELECT Clientes.CustomerID,

Ordenes\_de\_Compra.ProductID,

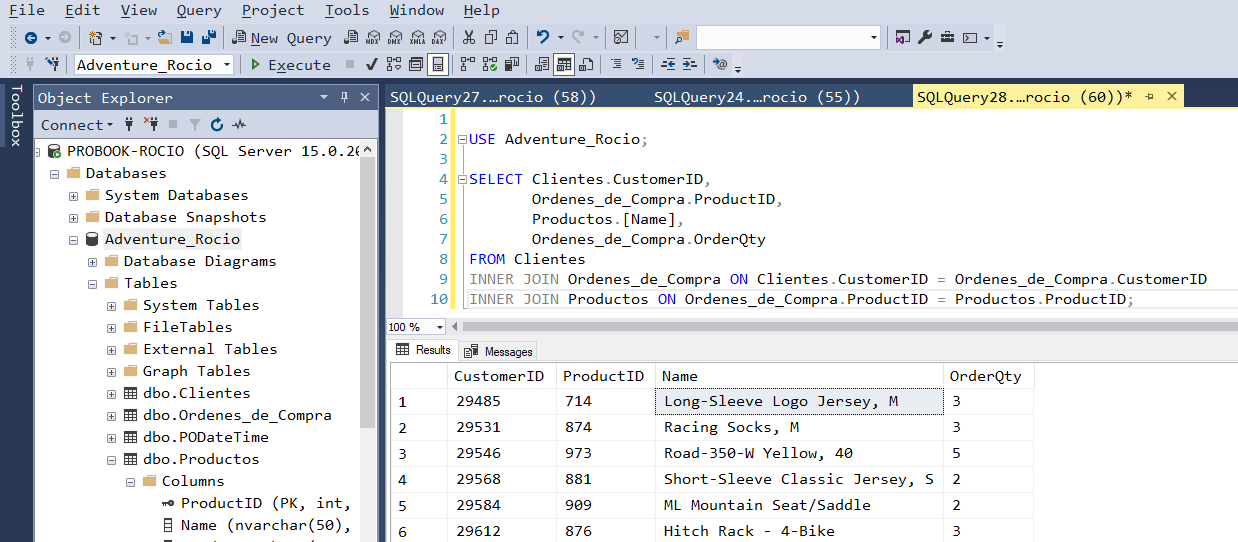
Productos.[Name],

Ordenes\_de\_Compra.OrderQty

FROM Clientes

INNER JOIN Ordenes\_de\_Compra ON Clientes.CustomerID = Ordenes\_de\_Compra.CustomerID

INNER JOIN Productos ON Ordenes\_de\_Compra.ProductID = Productos.ProductID;



5.- Utiliza la información obtenida en el punto anterior para saber cuantas unidades se vendieron de cada uno de los productos

***RESPUESTA:***

Crea una vista con la consulta del punto anterior y corre el código para que se forme la tabla temporal “Productos\_Vendidos”

CREATE VIEW Productos\_Vendidos AS

SELECT Clientes.CustomerID,

Ordenes\_de\_Compra.ProductID,

Productos.[Name] AS Descripcion,

Ordenes\_de\_Compra.OrderQty

FROM Clientes

INNER JOIN Ordenes\_de\_Compra ON Clientes.CustomerID = Ordenes\_de\_Compra.CustomerID

INNER JOIN Productos ON Ordenes\_de\_Compra.ProductID = Productos.ProductID;

Si así lo deseas, puedes ver los campos de la tabla temporal que acabas de crear

SELECT \*

FROM Productos\_Vendidos

Una vez creada la tabla, realiza otra consulta sumando la cantidad de productos vendidos

SELECT ProductID,

Descripcion,

SUM(OrderQty) AS Cant\_Total

FROM Productos\_Vendidos

GROUP BY ProductID, Descripcion

