**Hoja de trucos SQL**

**Introducción**

Aunque no es necesario tener conocimientos expertos de SQL para conectar Power BI a una base de datos SQL, resulta útil comprender cierta sintaxis esencial en las consultas SQL.

Esta lectura proporciona una visión general de algunos comandos SQL esenciales que debe conocer cuando trabaje con bases de datos SQL.

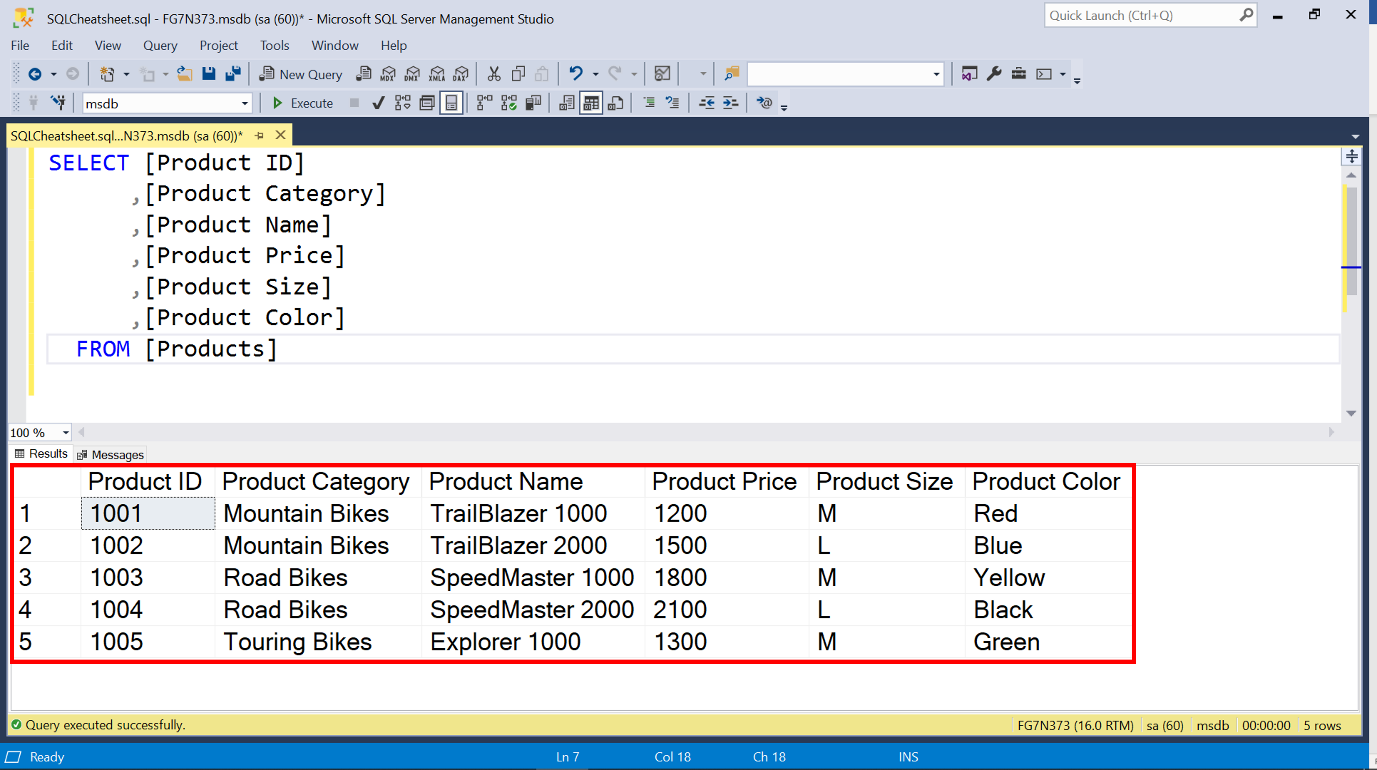
**Lenguaje de consulta estructurado (SQL)**

SQL (Lenguaje de consulta estructurado) es una potente herramienta utilizada para gestionar y manipular bases de datos relacionales. Permite recuperar, insertar, actualizar y eliminar datos, así como crear y modificar estructuras de bases de datos.

Esta lectura estructura los comandos básicos de SQL en tres categorías distintas:

* **Recuperación de datos:** Estos comandos se centran en consultar y recuperar información específica de una base de datos.
* **Manipulación de datos:** Esta sección abarca comandos para realizar cálculos, combinar datos y aplicar lógica condicional.
* **Modificación de datos:** Estos comandos se pueden utilizar para añadir, actualizar y eliminar registros en una base de datos.

Para ayudarle a comprender la funcionalidad de todos estos comandos, se utiliza como ejemplo una sencilla tabla de **Productos** . La tabla contiene cinco filas de productos, como se ve a continuación:



Identificaremos cómo afecta cada uno de estos comandos al resultado de la consulta en la ventana.

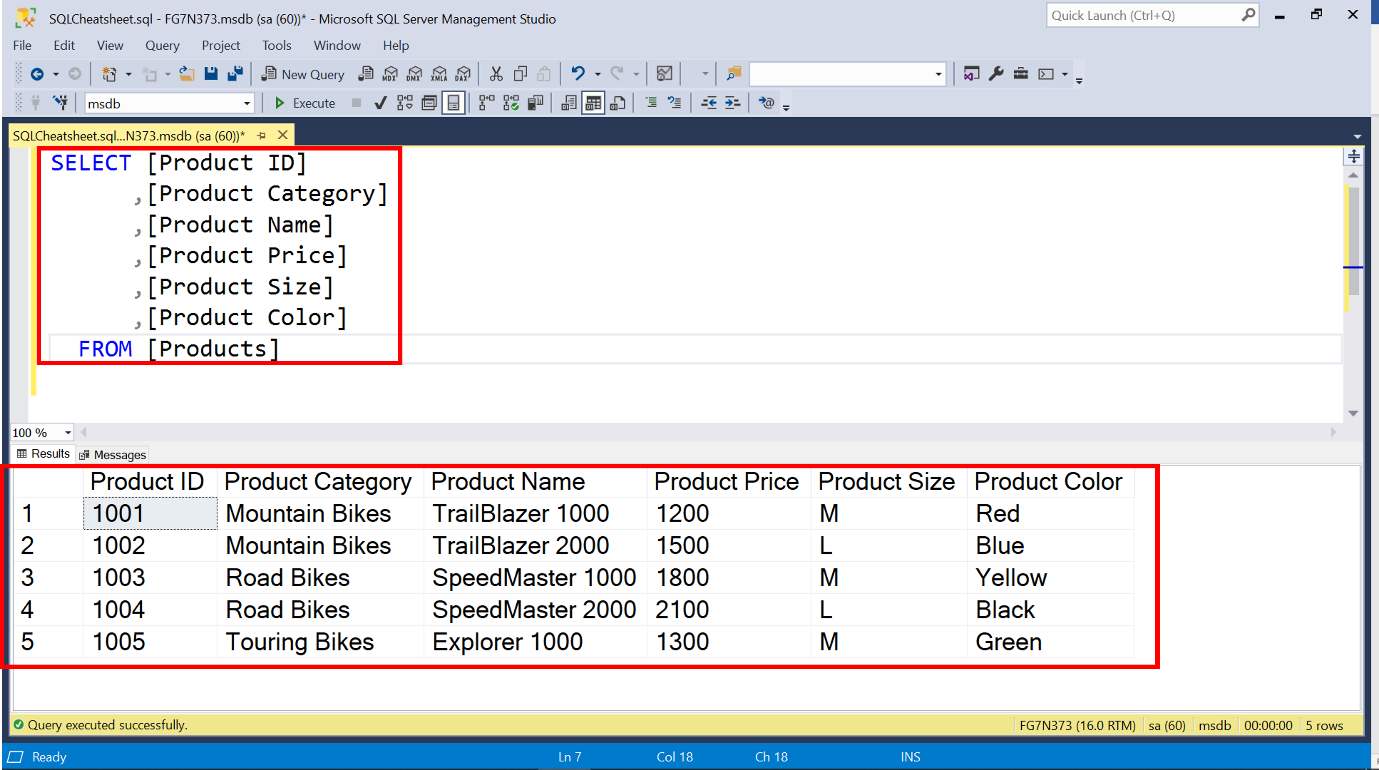
**Recuperación de datos**

A menudo tendrá que recuperar datos de bases de datos. Puede consultar y recuperar información específica de una base de datos utilizando varios comandos. Estos comandos se exploran en detalle a continuación.

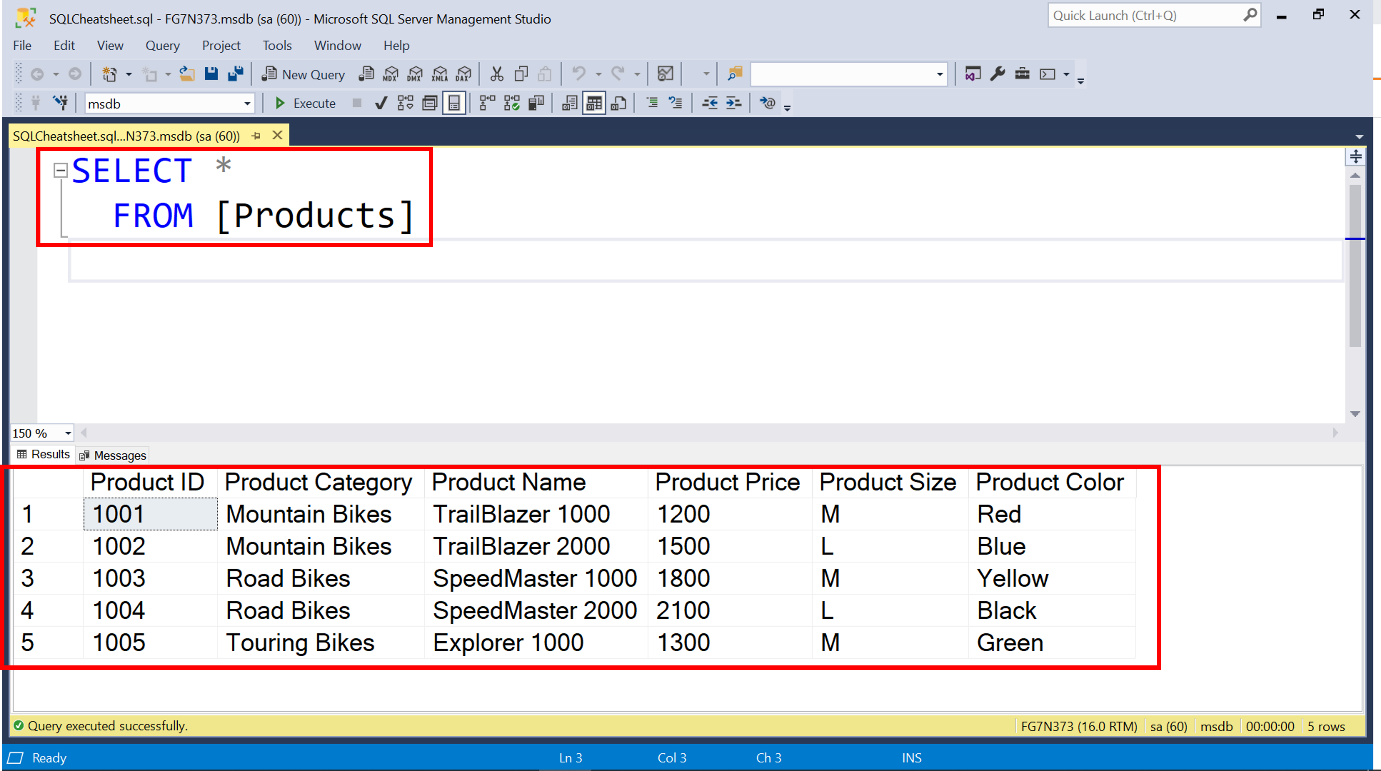
**Consulta SELECT**

Los datos se recuperan de una base de datos mediante una consulta **SELECT** de SQL. La consulta debe especificar los datos que se van a extraer, como los nombres de las columnas. Una cláusula **FROM** indica de qué tabla debe extraerse.

El siguiente ejemplo extrae todos los datos de la tabla **Producto**.



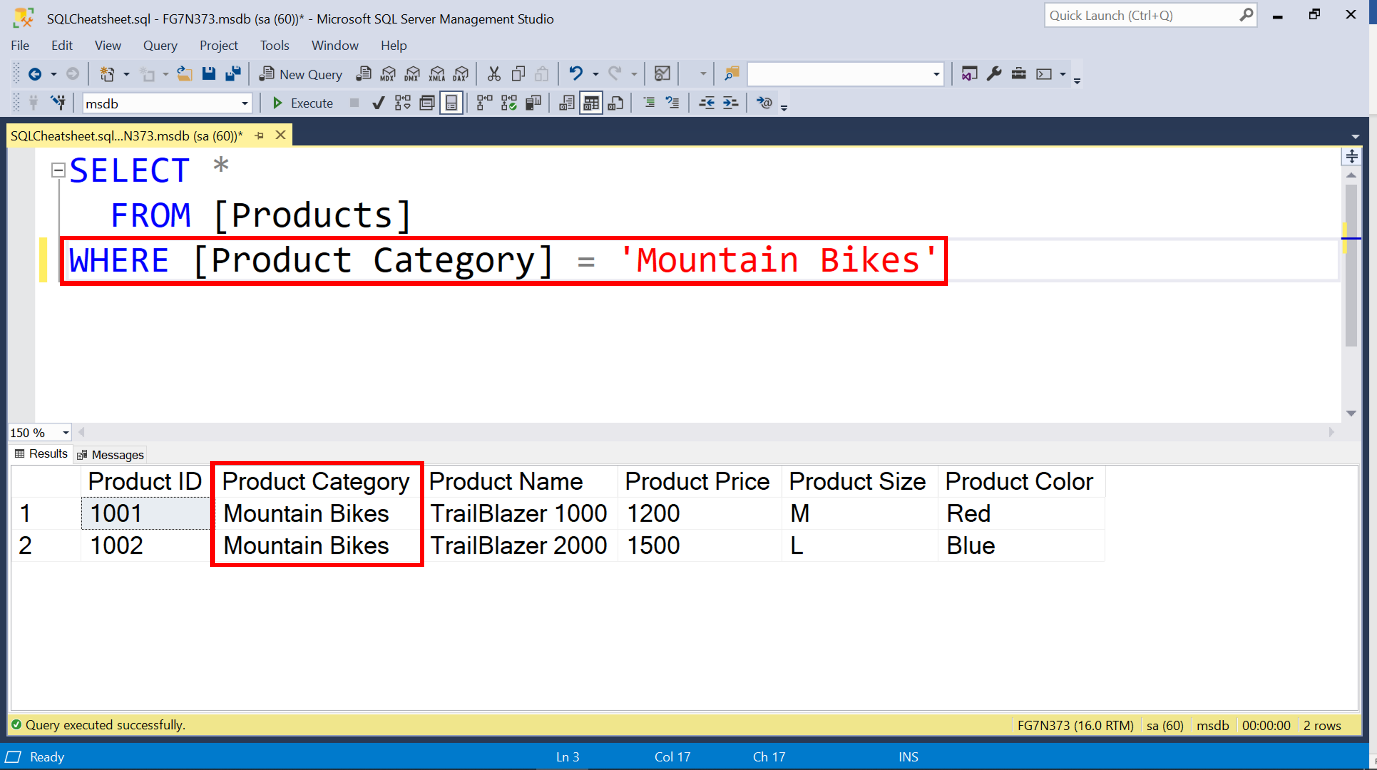
En lugar de enumerar todos los nombres de las columnas, también puede utilizar un asterisco para recuperar todas las columnas de una tabla, como en el ejemplo siguiente:



**Cláusulas de filtrado**

También puede aplicar filtros a sus consultas **SELECT** para que sólo se recupere información específica de la tabla en lugar de todos los registros. Un método habitual para aplicar una condición de filtro es utilizar la cláusula **WHERE**.

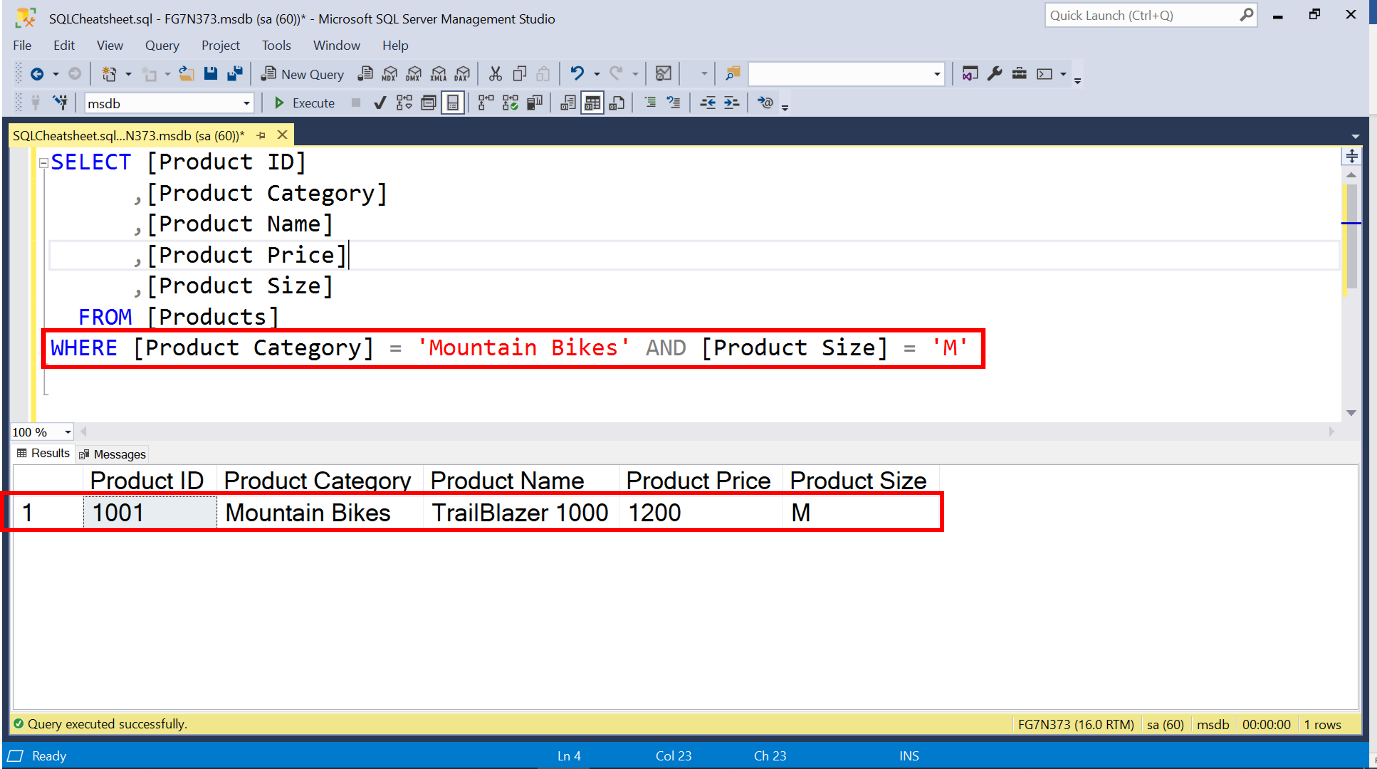
En el siguiente ejemplo, la consulta **SELECT** utiliza una cláusula **WHERE** para especificar que sólo debe recuperarse de la tabla la información de producto relacionada con el valor **Bicicletas de montaña**.



**Condiciones múltiples**

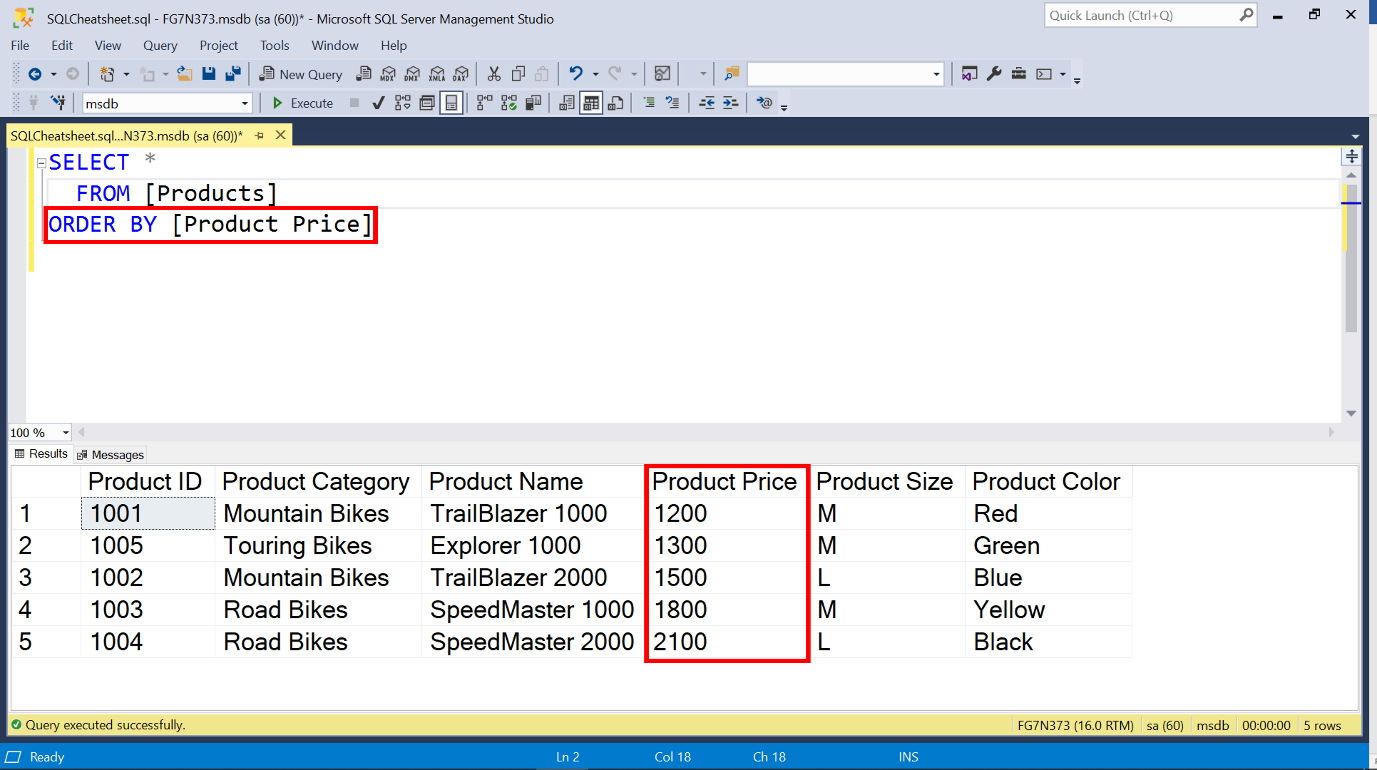
Puede ampliar el uso de la cláusula **WHERE** añadiendo determinadas condiciones mediante los comandos **AND** u **OR**. Estos comandos se utilizan para especificar más condiciones para el filtrado.

En el siguiente ejemplo, la cláusula **WHERE** debe filtrar todos los registros con el valor **Bicicleta de montaña** de la tabla **Categoría de producto** **Y** un valor **Mediano** de la columna **Tamaño del producto**.



**Ordenar resultados**

Puede utilizar el comando **ORDENAR POR** para ordenar los datos recuperados en un orden específico. En el ejemplo siguiente, los registros se ordenan por precio en orden ascendente. **ORDER** BY devuelve los registros en orden ascendente por defecto. Para devolver registros en orden descendente, utilice el comando ORDER BY **DESC**.



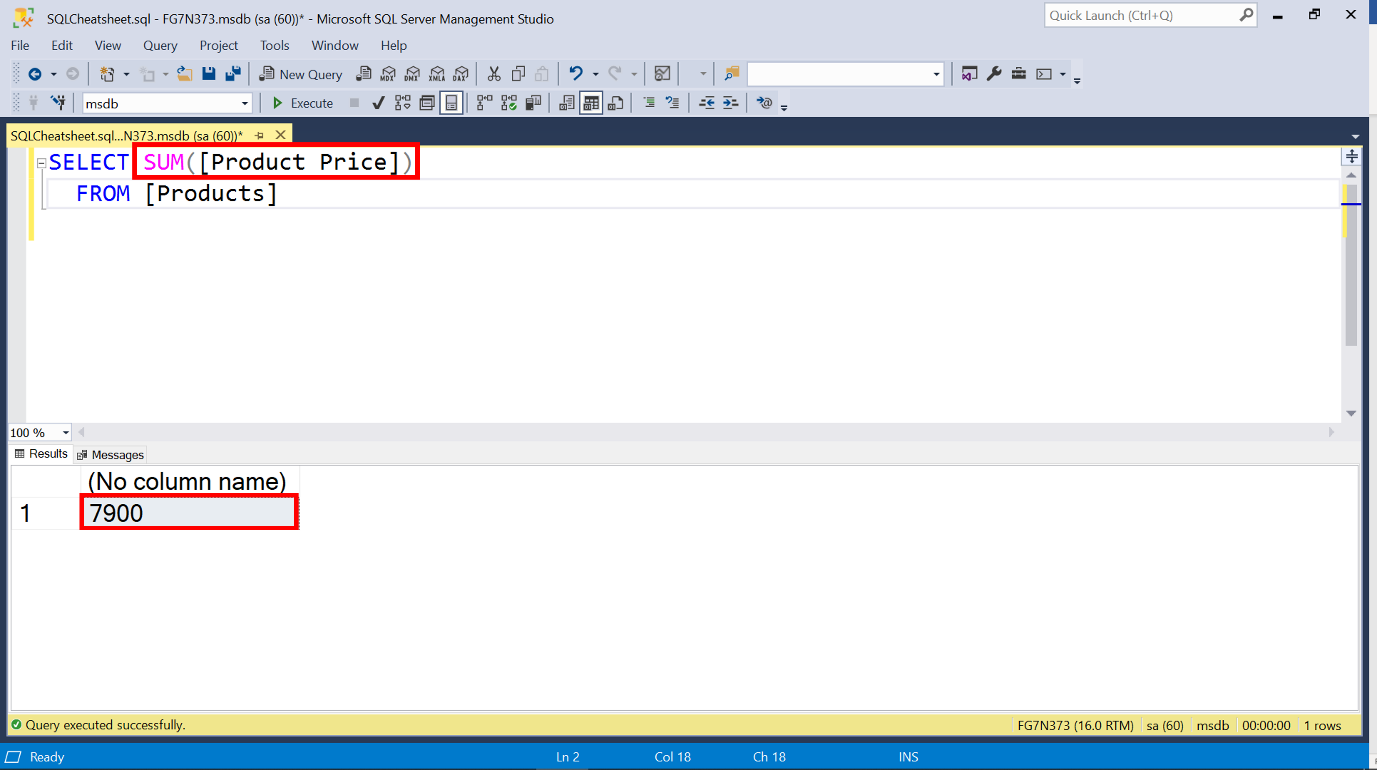
**Manipulación de datos**

SQL permite a los analistas de datos cambiar o manipular los datos de la base de datos. Puede utilizar técnicas como las funciones de agregación, los alias, la lógica condicional y los valores **NULL** para manipular los datos según sus necesidades.

**Funciones de agregación**

Las funciones de agregación se utilizan para calcular valores y devolver un único resultado. Algunos ejemplos de funciones de agregación son **SUM**, **COUNT**, **AVERAGE**, **MAX** y **MIN**.

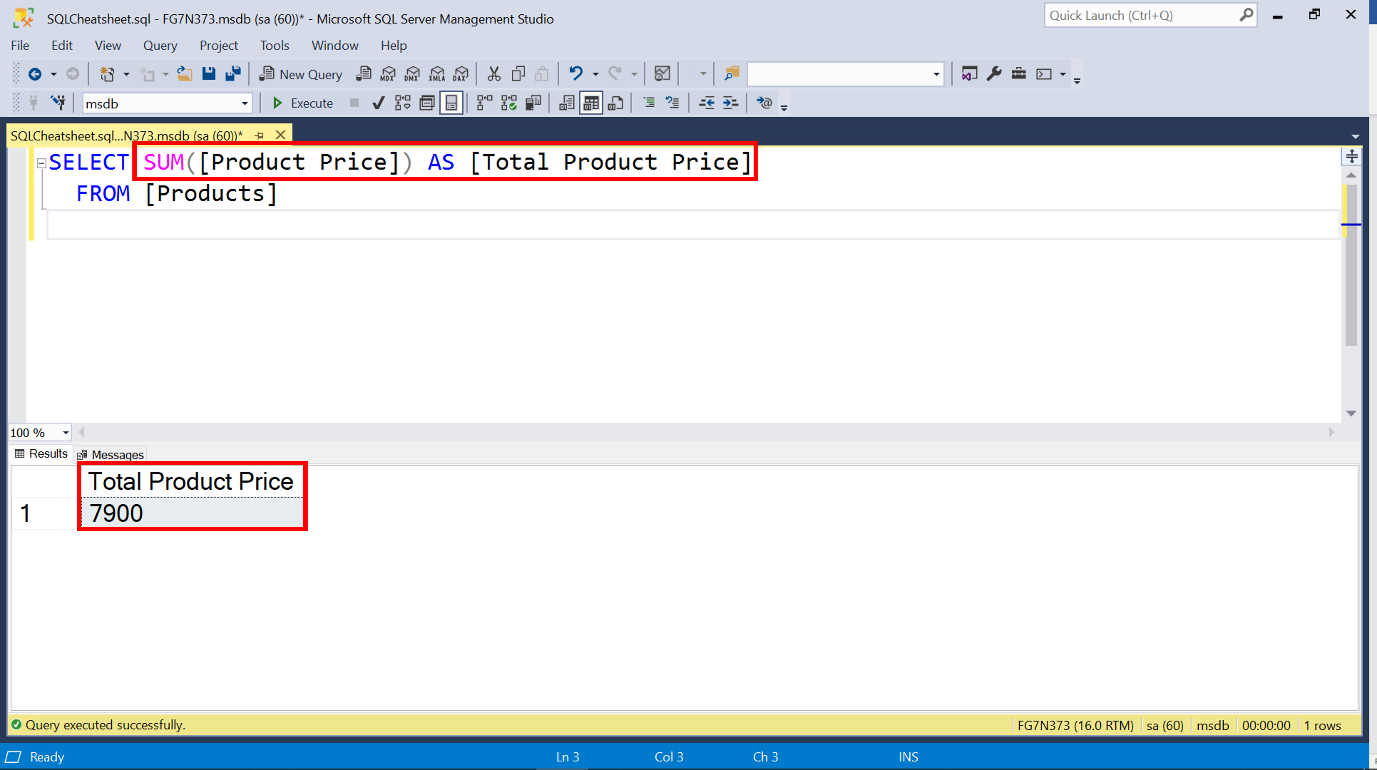
En el siguiente ejemplo, la función de agregación **SUMA** calcula el total de todos los precios de la columna **Precio del producto**.



**Alias**

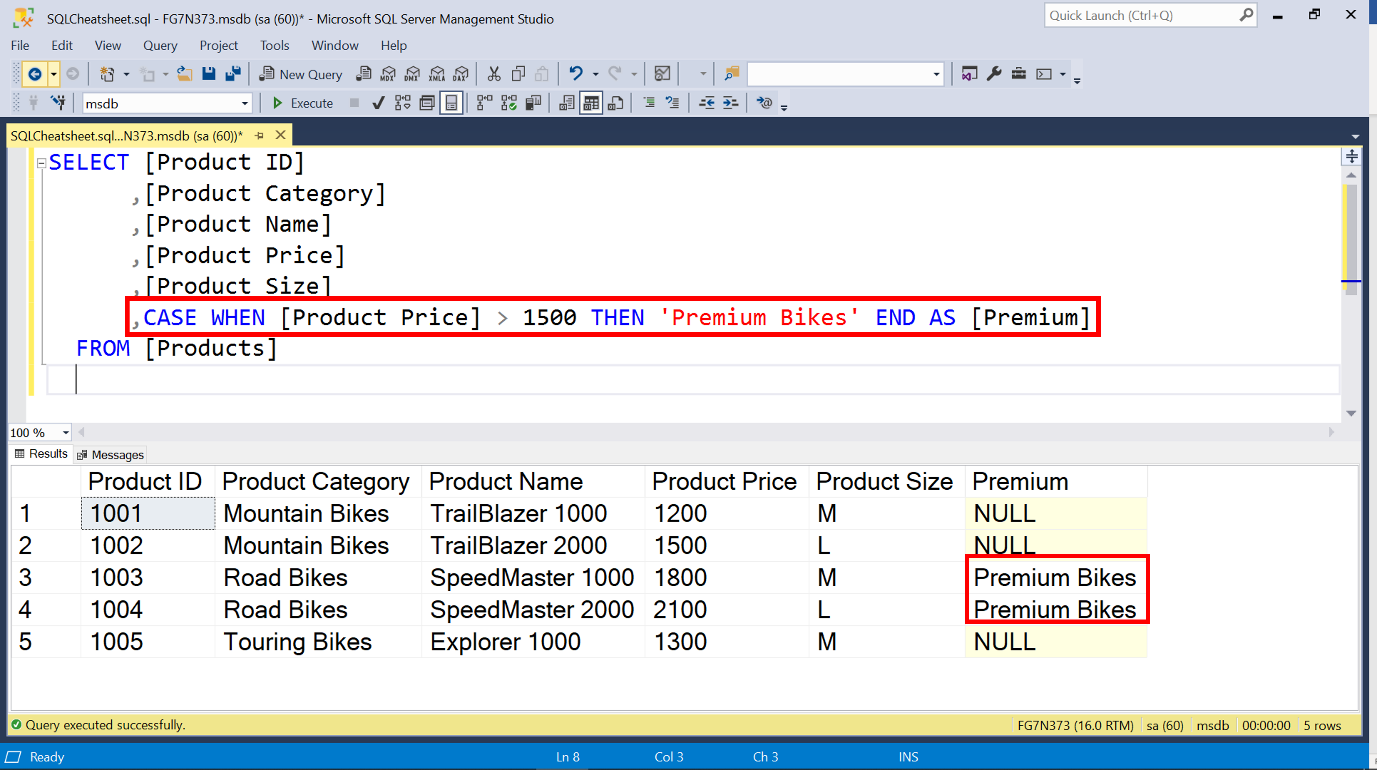
A menudo resulta útil devolver los resultados como una nueva columna. Con SQL, puede utilizar un alias para proporcionar un nombre a su nueva columna. Escriba el comando **AS** seguido del nombre de su nueva columna entre corchetes.

El siguiente ejemplo devuelve el precio total de todos los productos como una nueva columna llamada **Precio total del producto**.



**Lógica condicional**

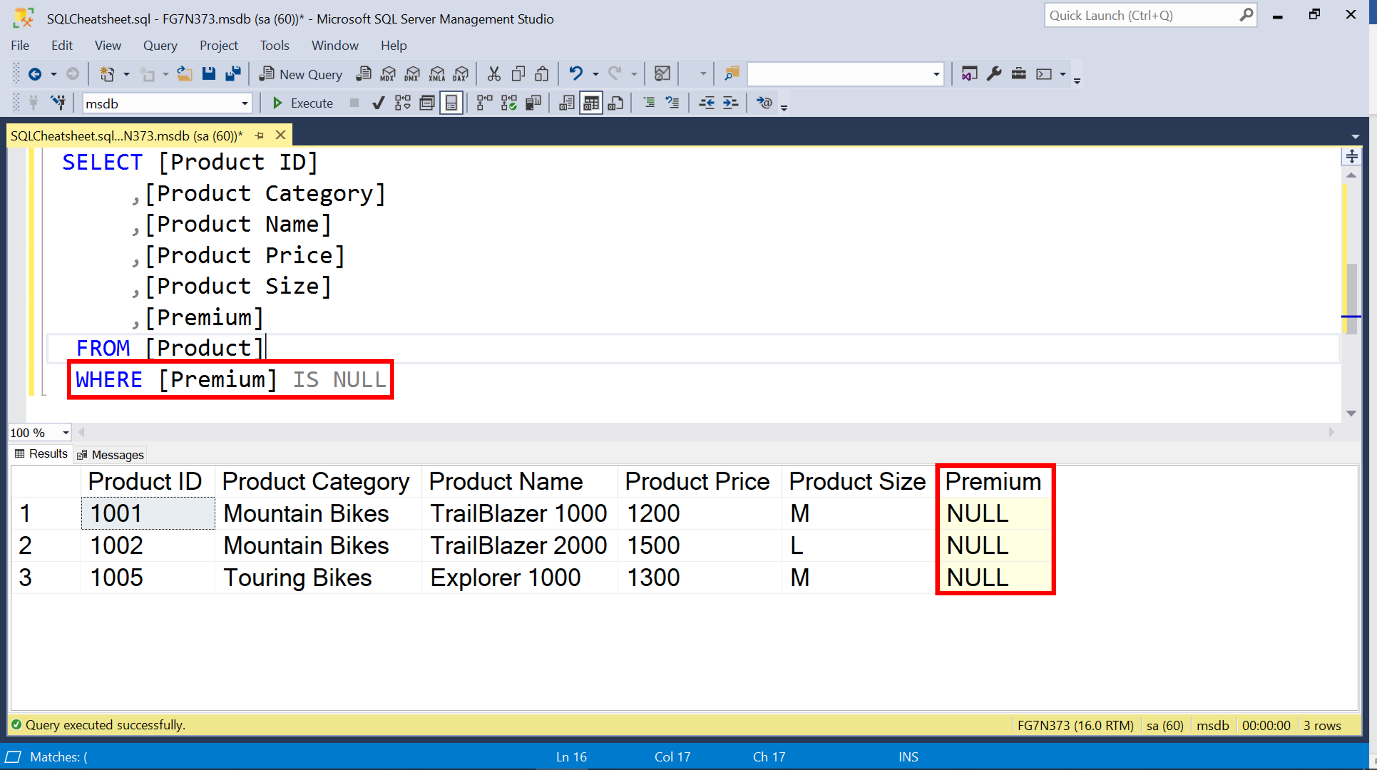
Es posible que a menudo tenga que crear consultas complicadas para devolver resultados específicos de una tabla. En estos casos, puede utilizar cláusulas de lógica condicional como CASE **WHEN** para proporcionar las instrucciones necesarias. CASE **WHEN** es una expresión condicional que le permite establecer un valor en un campo basándose en varias condiciones. El ejemplo siguiente indica a SQL que etiquete todos los valores de la columna **Precio del producto** con un valor superior a **1500** como **Bicicletas Premium**. Esto se demuestra en la sentencia SQL de ejemplo que aparece a continuación. En una instancia **CASE** o, **CUANDO** SQL encuentre un valor superior a **1500**, deberá etiquetarlo como **Bicicletas** Premium.



**Valores NULL**

En el ejemplo anterior, las filas que no cumplían la condición se marcaron con el valor **NULL**. Este es el valor en blanco en SQL. Para recuperar estas filas, puede utilizar una cláusula **WHERE** con la condición **IS NULL**.

En el siguiente ejemplo, SQL recupera todos los registros de la tabla **Producto** con un valor **NULL** en la columna **Prima**.



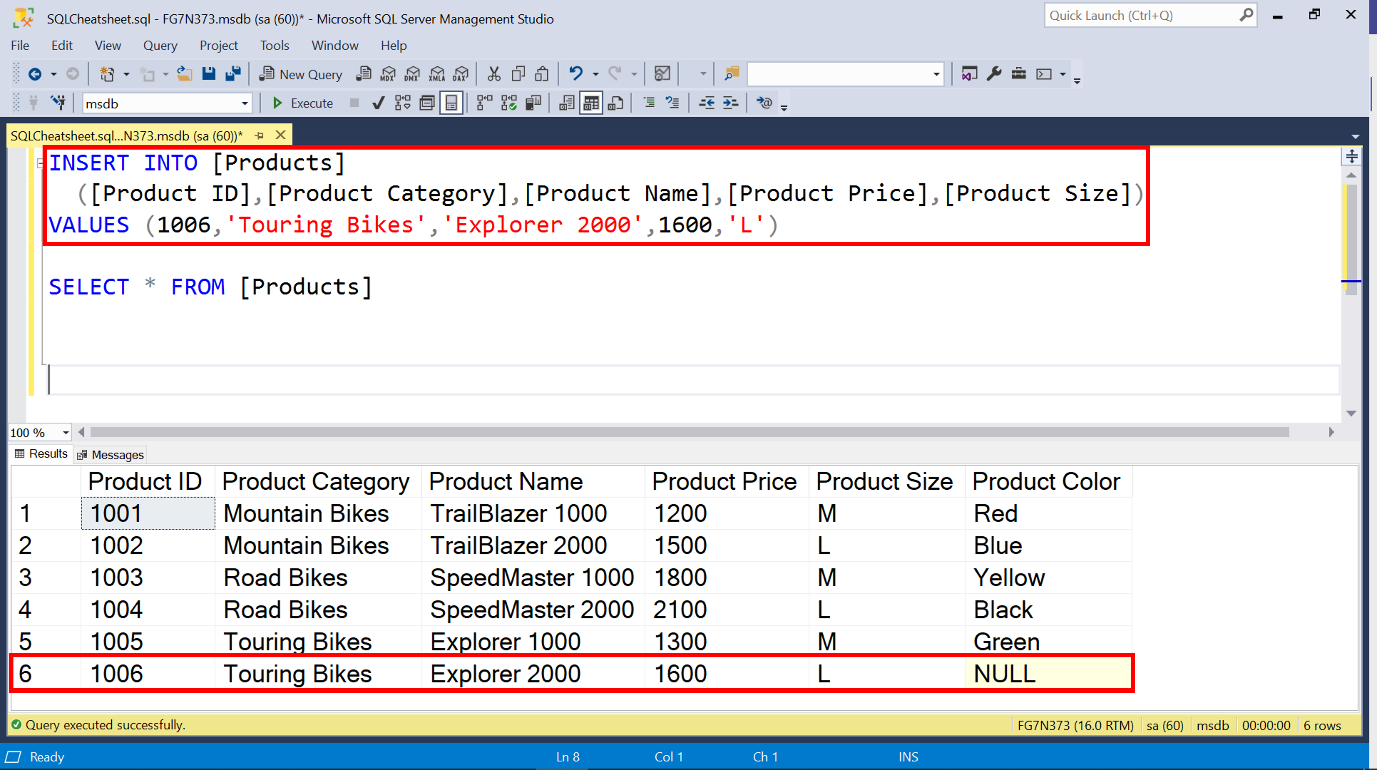
**Modificación de datos**

Con SQL, también es posible modificar los resultados de sus consultas utilizando funciones de modificación de datos. Algunos ejemplos de estas funciones son las cláusulas **INSERT**, **UPDATE** y **DELETE**. Puede obtener más información sobre las cláusulas en los ejemplos siguientes.

**La cláusula INSERT**

Puede añadir nuevos datos a sus tablas SQL utilizando la cláusula **INSERT** o **INSERT INTO**. Cuando utilice esta cláusula, especifique todas las columnas a las que deben asignarse valores. Asegúrese de asignar los valores en el orden correcto. Si omite un valor, la columna quedará en blanco.

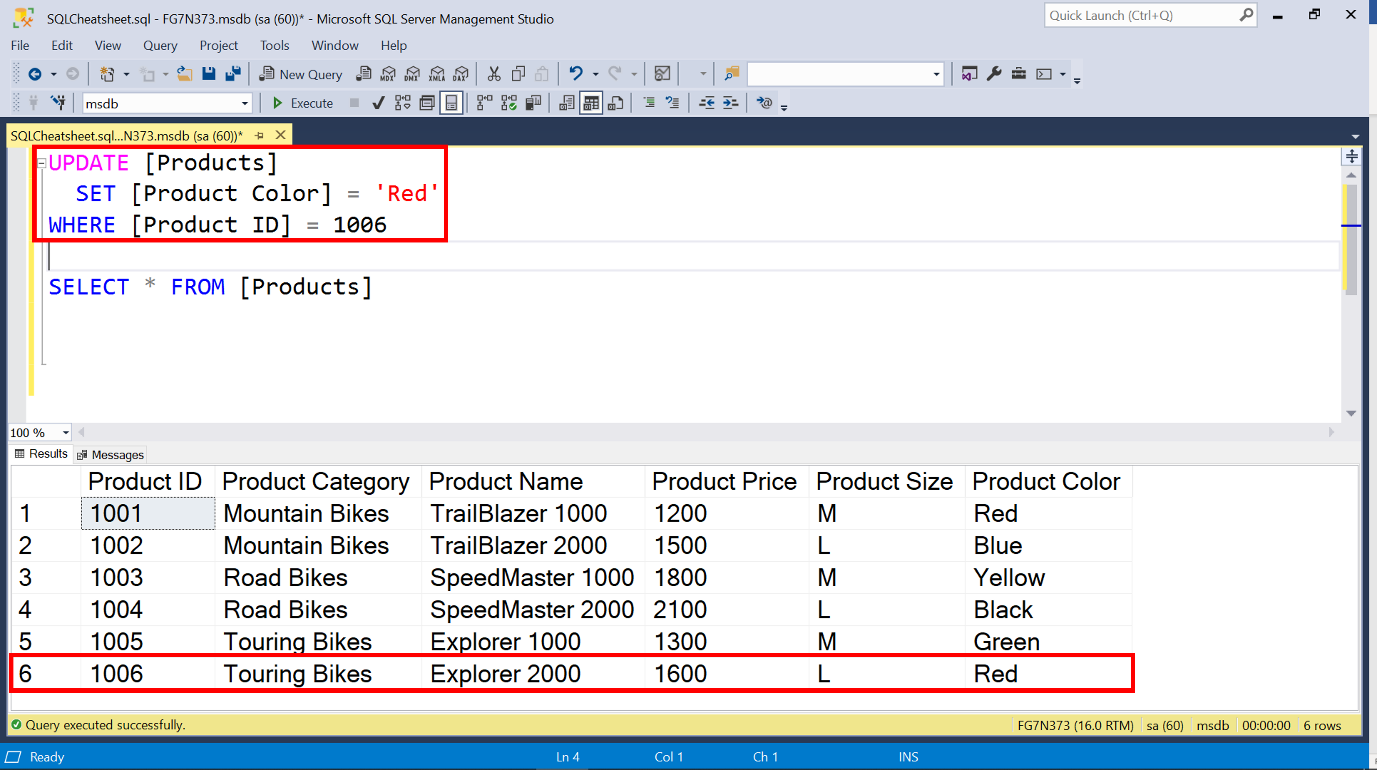
En el ejemplo siguiente, se añaden varios valores para un nuevo producto llamado **Explorer** 2000. Sin embargo, no se especifica ningún valor para la columna **Color del producto**, por lo que se deja en blanco o **NULL**.



**La cláusula UPDATE**

Puede actualizar los valores de una columna mediante la cláusula **UPDATE**. Sin embargo, debe utilizarse con la cláusula **WHERE** para especificar en qué filas debe producirse la actualización.

En el siguiente ejemplo, se añade el valor de **Rojo** a la columna **Color del producto** para el nuevo producto con un **ID de producto** de **1006**.

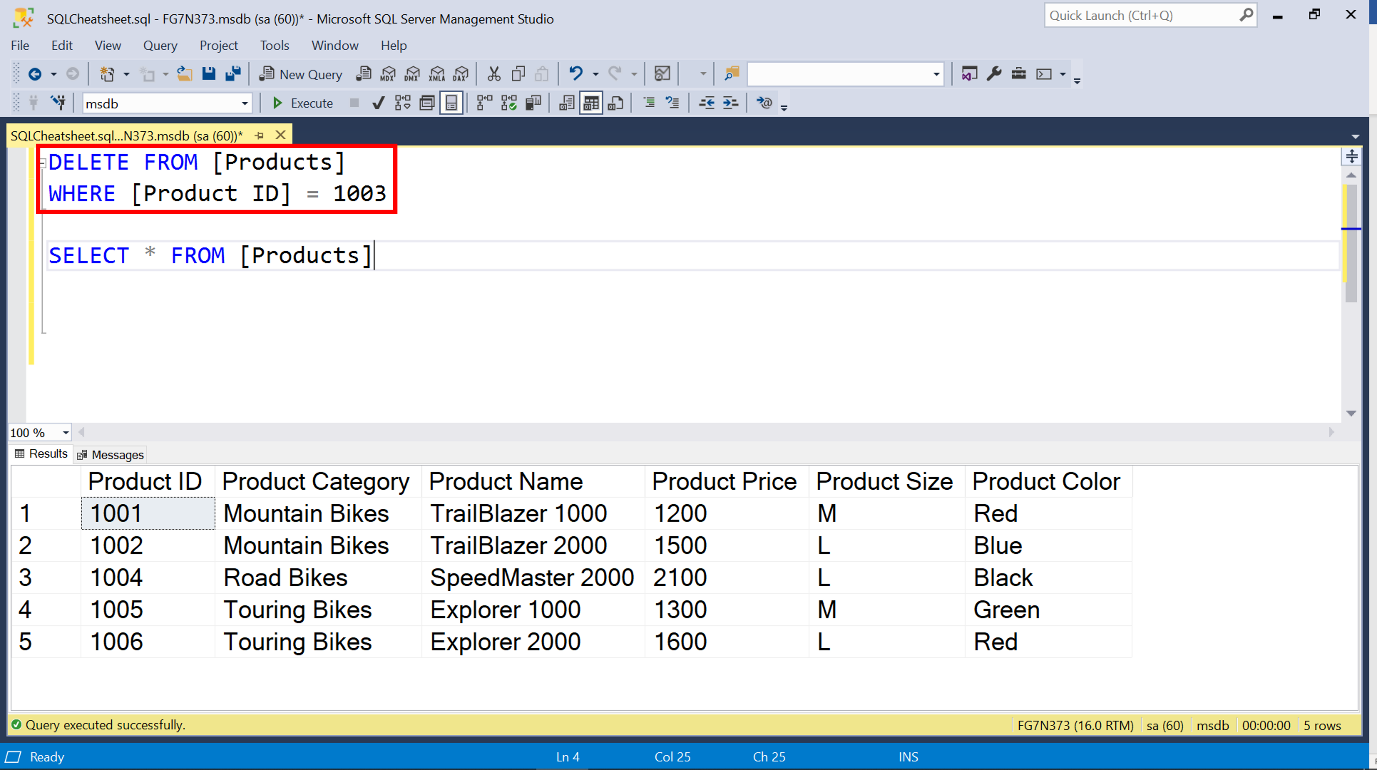


**El comando DELETE FROM**

Puede eliminar o borrar datos de una tabla SQL utilizando el comando **DELETE** FROM. Al igual que con **UPDATE**, también debe utilizar la cláusula **WHERE** para identificar las filas que se verán afectadas.

Tenga cuidado al utilizar **DELETE FROM** en una base de datos SQL. No hay botón de deshacer, por lo que cualquier cambio que realice será permanente.

En el ejemplo siguiente, el comando **DELETE** FROM ordena a SQL que **ELIMINE** todos los registros relacionados con el **ID de producto 1003 DE** la tabla **Productos**. Observe que **1003** ya no se encuentra en el conjunto de resultados.



**Conclusión**

Si domina SQL, podrá recuperar, manipular y modificar los datos según sus necesidades a partir de sus bases de datos SQL. Familiarícese con los comandos de esta lectura, continúe practicando su uso y rápidamente se convertirá en un experto en SQL El dominio de SQL es una habilidad muy valiosa para los analistas de datos y para cualquiera que intente desenvolverse en una empresa basada en datos.