**Normalización y desnormalización**

**Introducción**

Una de las tareas más importantes de un analista de datos en Microsoft Power BI es crear y gestionar modelos de datos. Si ejecuta esta tarea con eficacia, facilitará a su equipo la comprensión de los datos. Sin embargo, una de las preguntas a las que se enfrentará a menudo es "¿normalizar o desnormalizar?"

En esta lectura, explorará los conceptos de normalización y desnormalización, sus efectos en el modelo de datos y cómo ejecutar estas técnicas utilizando Power Query.

**El modelo de datos y sus ventajas**

Como ya sabrá, un modelo de datos es una representación conceptual de los elementos de datos. Es un resumen visual de sus tablas, sus columnas y las relaciones entre las tablas.

Un modelo de datos adecuadamente diseñado proporciona las siguientes ventajas:

* Exploración más rápida de los datos
* Creación de agregados más sencilla
* Informes más precisos
* Creación de informes más rápida
* Mantenimiento más sencillo de los informes

Sin embargo, también se pueden proporcionar más ventajas utilizando la normalización o la desnormalización.

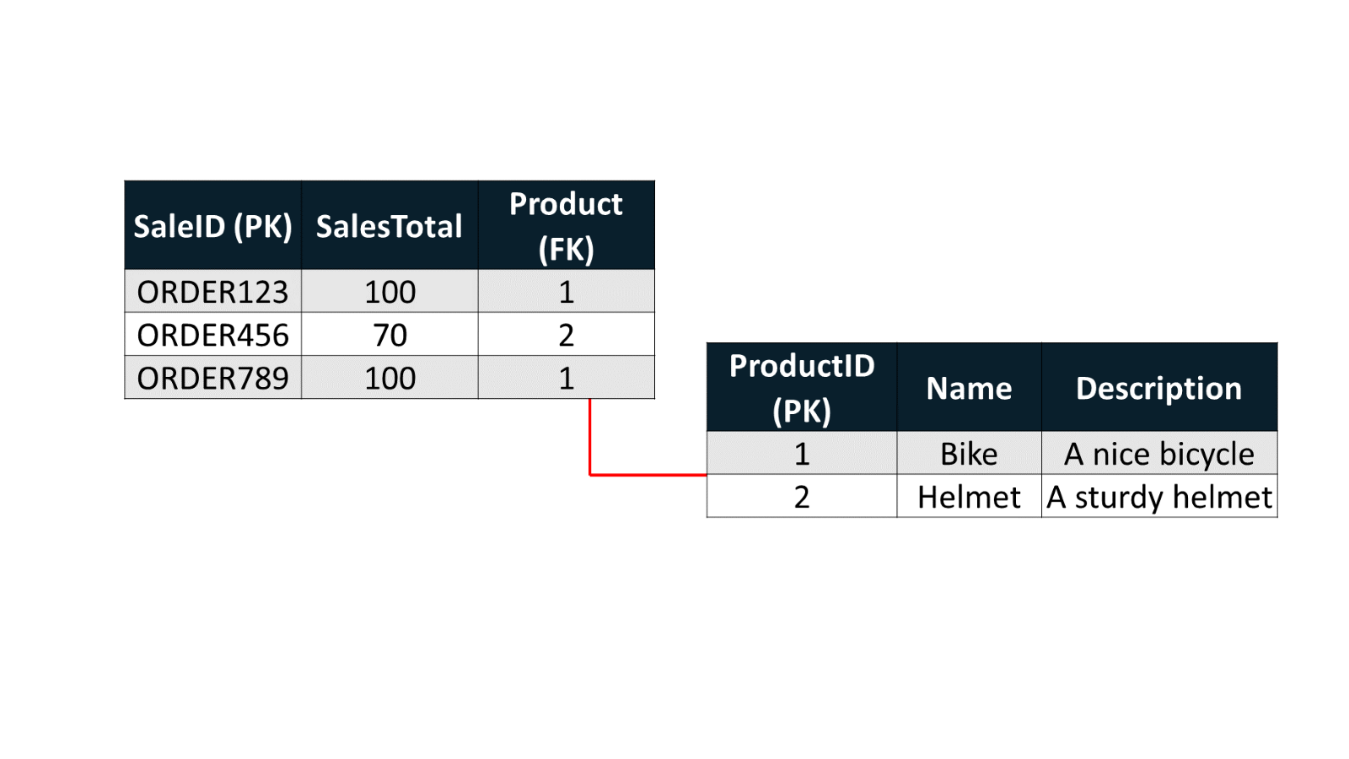
**Normalización**

La normalización es una técnica de diseño de modelos de datos que consiste en estructurar los datos para minimizar la redundancia y garantizar su integridad. Divide los datos en varias tablas relacionadas, cada una con un propósito específico. Este enfoque reduce la duplicación de datos. Sin embargo, a menudo requiere crear relaciones complejas entre las tablas.

La normalización ofrece las siguientes ventajas:

* Eliminación de datos redundantes.
* Mejora de la integridad de los datos.
* Mantenimiento más sencillo del modelo de datos.

Con la normalización, las relaciones se establecen entre tablas utilizando claves primarias y externas. Las claves primarias son columnas que identifican de forma exclusiva cada fila de datos.



Por ejemplo, la tabla **Productos** de Adventure Works tiene una columna de clave única llamada **productid**. Las columnas adicionales representan atributos del producto como **el nombre** y la **descripción**.

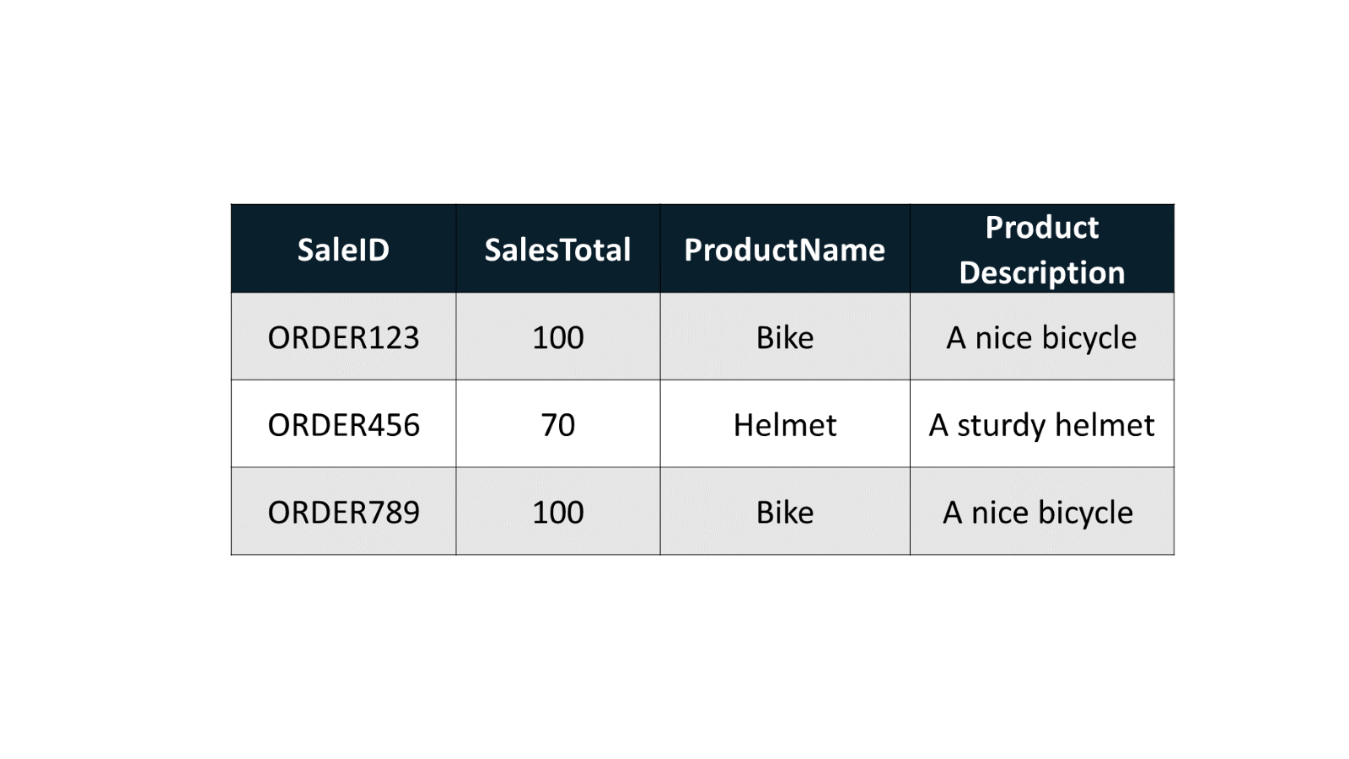
La segunda tabla, **Ventas**, tiene una columna de clave única denominada **SaleId** y una columna **SalesTotal**. También tiene una columna de clave foránea denominada **Producto**. Cada campo de esta columna representa un **ID** en la tabla **Producto**. La relación entre la tabla **Ventas** y la tabla **Producto** es un ejemplo de relación muchos-a-uno. En otras palabras, puede haber muchas filas en la tabla **Ventas** relacionadas con una fila en la tabla **Producto**.

Este esquema está normalizado porque la tabla **Ventas** utiliza una clave foránea para asociar cada fila con un producto específico.

**Desnormalización**

La desnormalización es el proceso inverso a la normalización. Consiste en convertir el esquema normalizado en un esquema con información redundante. Implementar la desnormalización ayuda a evitar consultas costosas entre las tablas, pero a costa de crear datos redundantes o duplicados.

Para demostrarlo, volvamos al ejemplo anterior. Para desnormalizar el esquema de Adventure Works, la tabla **Ventas** debe contener las columnas **SaleId**, **SalesTotal, ProductName** y **ProductDescription**.



Observe cómo la información del producto está ahora duplicada en varias filas.

Por otro lado, la desnormalización ofrece algunas ventajas:

* En lugar de cargar varias tablas de datos en su modelo de datos, sólo tiene una tabla grande, lo que podría ser más eficiente en cuanto a rendimiento. Al no existir relaciones entre las tablas, se reducen las consultas para unir los datos.
* La propagación de filtros entre las tablas puede ser a veces menos eficiente que los filtros aplicados a una sola tabla.
* Establecer una jerarquía a partir de campos dentro de una tabla es más sencillo.

Sin embargo, a pesar de estas ventajas, no tiene por qué crear un modelo de datos desnormalizado para cada conjunto de datos. Durante la fase de diseño, debe tener en cuenta otros factores que también pueden afectar a su modelo, como los requisitos analíticos y empresariales, sus fuentes de datos y el tamaño de su modelo.

**Diferencia entre normalización y desnormalización**

Existen varias diferencias entre los modelos normalizados y desnormalizados, que es importante conocer:

**Integridad de los datos**

La integridad de los datos se mantiene durante el proceso de normalización. En cambio, la integridad de los datos es más difícil de mantener con un modelo desnormalizado. Esto se debe a que puede ser necesario actualizar varias filas cuando cambian los datos.

**Datos redundantes**

Los datos redundantes se eliminan cuando se realiza la normalización. Sin embargo, la desnormalización aumenta los datos redundantes.

**Tamaño del modelo**

La normalización aumenta el número de tablas y de consultas join (debido a las relaciones entre tablas). Por el contrario, la desnormalización reduce el número de tablas y relaciones.

**Memoria**

El espacio en disco se utiliza en exceso en un esquema desnormalizado porque los mismos datos se almacenan en otro lugar. Por el contrario, el espacio en disco se optimiza en un esquema normalizado.

**Normalización y desnormalización de datos en Power Query**

Ahora que ya está familiarizado con la normalización y la desnormalización, repasemos cómo se implementan estas técnicas.

**Cargar los datos**

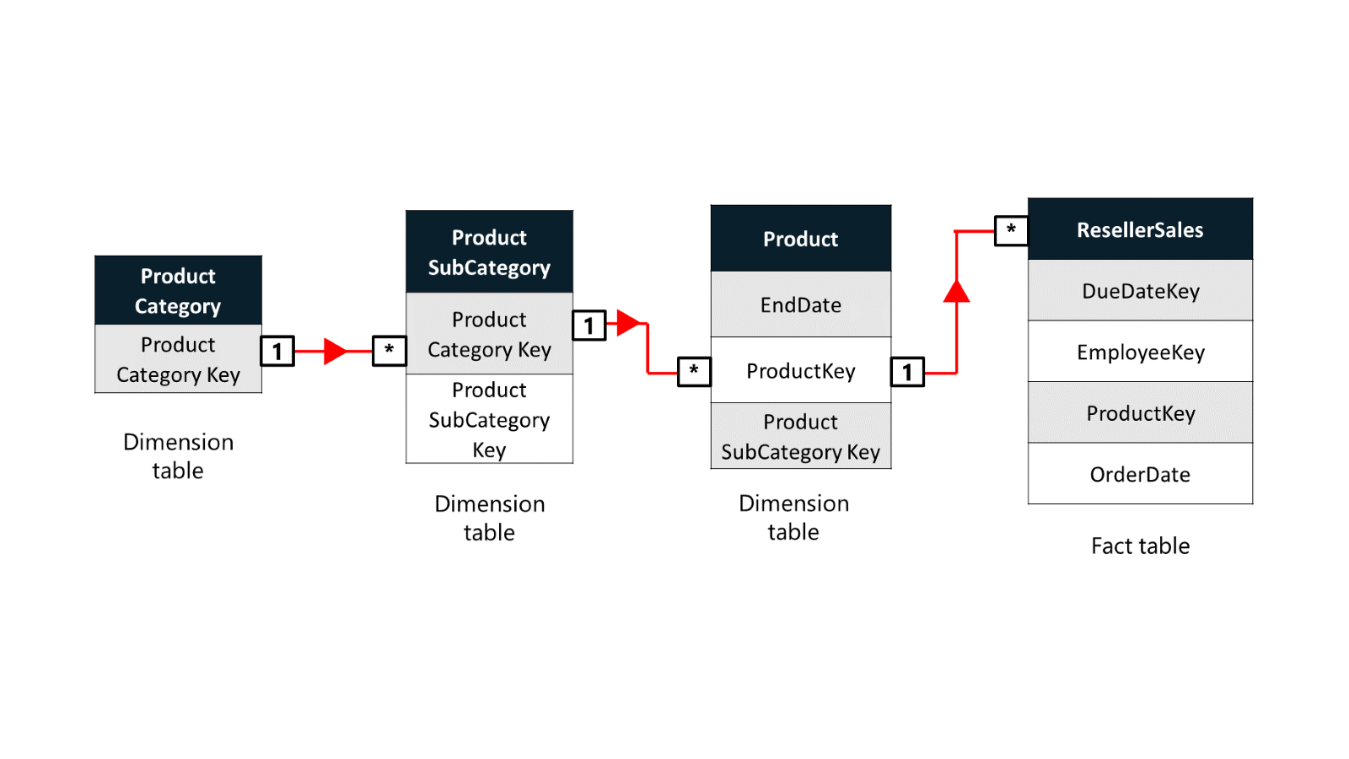
Cuando los datos se cargan en Power BI desde una única fuente de datos, es probable que representen una estructura de datos desnormalizada. En este caso, puede utilizar el editor de Power Query para transformar y dar forma a sus datos de origen en varias tablas normalizadas. Una vez que haya realizado la normalización en Power Query, podrá construir relaciones entre sus tablas de datos en la vista de modelado del escritorio de Power BI.

**Diseñe su modelo**

Su responsabilidad es diseñar un modelo de datos optimizado con tablas que representen los requisitos empresariales. Cualquier decisión en torno a la normalización o desnormalización debe estar alineada con estos requisitos.

En el siguiente ejemplo, Adventure Works clasifica los productos por **categoría** y **subcategoría**. Los productos se asignan a **subcategorías**. Mientras que las **subcategorías** se asignan a **categorías**. En este modelo, la tabla de dimensiones **Producto** se normaliza y almacena en tres tablas separadas:

* **Categoría**,
* **Subcategoría**
* y **Producto**.



**Desnormalice su modelo**

A Adventure Works le interesa desnormalizar este modelo. Sus analistas de datos pueden utilizar Power Query para transformar estas tablas en una única tabla de productos utilizando la función **Fusionar consultas**.

**Conclusión**

El debate sobre normalización o desnormalización consiste en encontrar el equilibrio adecuado entre la integridad de los datos, el rendimiento de las consultas y la facilidad de mantenimiento. No existe una solución única para todos los casos, y la elección depende de los requisitos empresariales y analíticos específicos de su proyecto. Sin embargo, si comprende los fundamentos, las diferencias y los casos de uso de cada estructura de datos normalizada y desnormalizada, podrá adaptar su modelo de datos para satisfacer sus necesidades específicas.