

Dimensionalidad: Star & Snowflake Schema

1. Normalización:

La normalización es un proceso en el diseño de bases de datos que consiste en organizar los datos de manera eficiente para minimizar la redundancia y garantizar la integridad de los mismos. Se basa en reglas específicas, conocidas como formas normales, que guían la estructura de las tablas y las relaciones entre ellas. La normalización ayuda a eliminar la duplicación innecesaria de datos y mejora la eficiencia en el almacenamiento y recuperación de información.

2. Dimensión:

En el contexto de los almacenes de datos, una dimensión representa una categoría o aspecto descriptivo de los datos. Las dimensiones proporcionan el contexto en el que se analizan las métricas o medidas cuantitativas de interés. Por ejemplo, en un almacén de datos de ventas, las dimensiones comunes pueden ser tiempo, producto, ubicación o cliente. Las dimensiones suelen tener jerarquías y contienen atributos descriptivos que se utilizan para agrupar, filtrar y analizar los datos en la tabla de hechos.

3. Tabla de dimensiones:

Las tablas de dimensiones contienen los atributos descriptivos que proporcionan contexto a los datos en la tabla de hechos. Cada tabla de dimensiones representa una dimensión específica y contiene información detallada sobre los atributos relacionados. Por ejemplo, una tabla de dimensiones de productos puede contener información como el nombre del producto, la categoría, el precio, etc. Las tablas de dimensiones se unen a la tabla de hechos mediante claves primarias(PK) y extranjeras(FK) para proporcionar un análisis más completo y detallado de los datos.

4. Hecho:

Un hecho es una medida cuantitativa o métrica de interés en un almacén de datos. Representa los datos numéricos que se analizan y se resumen en función de las dimensiones. Por ejemplo, en un almacén de datos(DW) de ventas, los hechos pueden ser el monto total de ventas, la cantidad de productos vendidos o el

beneficio obtenido. Los hechos se almacenan en la tabla de hechos y están relacionados con las dimensiones a través de claves primarias y extranjeras.

5. Tabla de hechos:

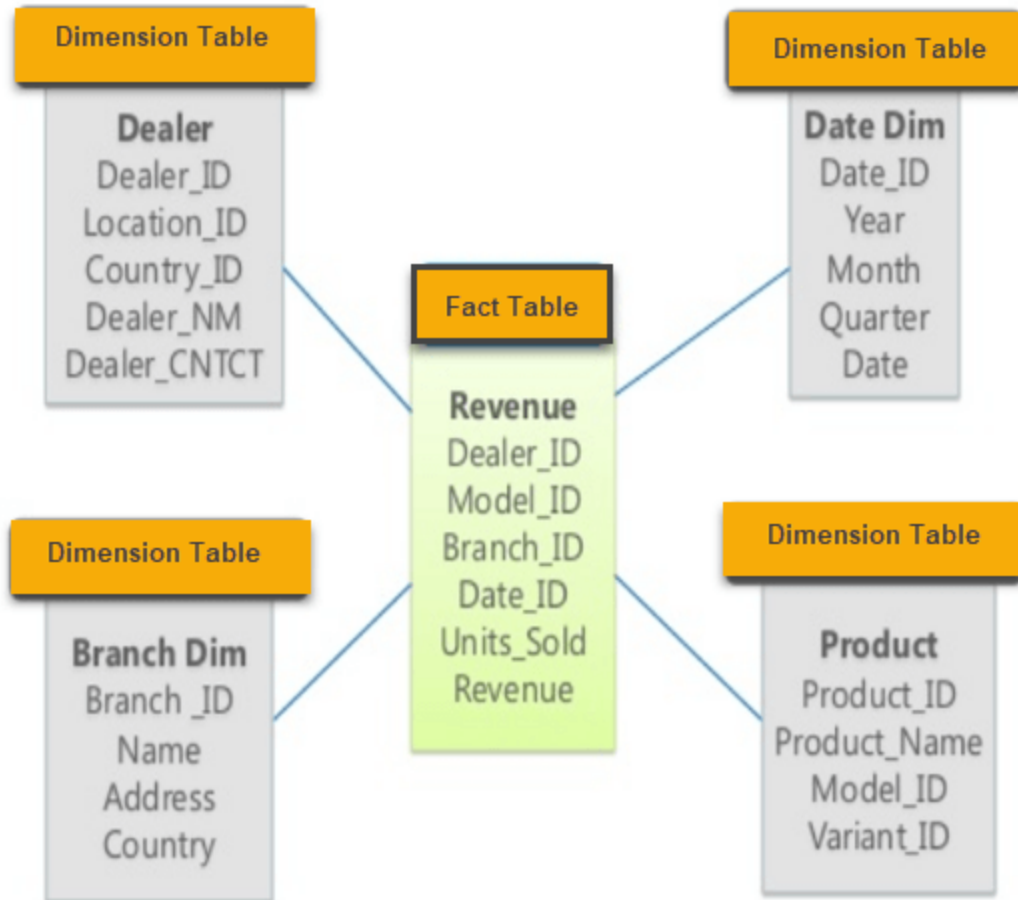
La tabla de hechos es la tabla central en un esquema estrella o copo de nieve. Contiene las medidas cuantitativas o métricas de interés, es decir, los hechos. Cada fila en la tabla de hechos representa una instancia o evento que se desea analizar. La tabla de hechos suele tener columnas numéricas para las medidas y claves extranjeras que establecen las relaciones con las tablas de dimensiones.

Star Schema(Esquema estrella):

El esquema estrella es un modelo de diseño de base de datos que consta de una tabla de hechos central que se conecta a varias tablas de dimensiones. En este esquema, la tabla de hechos contiene las medidas numéricas o cuantitativas de interés, mientras que las tablas de dimensiones contienen los atributos o características relacionados con esas medidas. La tabla de hechos se conecta a las tablas de dimensiones a través de claves foráneas.

La tabla de hechos en un esquema estrella está diseñada para ser el punto central de análisis y contiene claves primarias y claves foráneas que se utilizan para establecer relaciones con las tablas de dimensiones. Las tablas de dimensiones contienen información descriptiva, como nombres, ubicaciones, fechas y otros atributos relacionados.

El esquema estrella es conocido por su estructura simple y de fácil comprensión. Permite consultas rápidas y sencillas, ya que todas las dimensiones están directamente conectadas a la tabla de hechos. Sin embargo, este enfoque puede llevar a una redundancia de datos, ya que los atributos repetidos pueden existir en diferentes tablas de dimensiones.



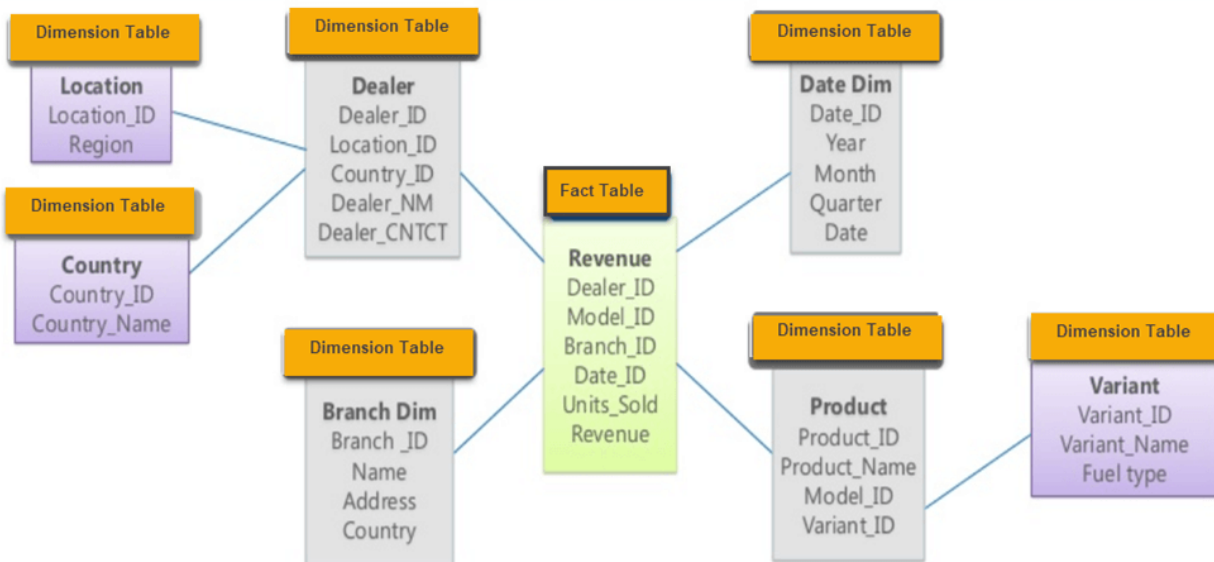
Snowflake Schema(Esquema copo de nieve):

Por otro lado, el esquema copo de nieve es una variación del esquema estrella que busca normalizar aún más las tablas de dimensiones para reducir la redundancia. En el esquema copo de nieve, las tablas de dimensiones se dividen en subtablas más pequeñas y normalizadas, creando una estructura de copo de nieve en lugar de una estructura estrella.

En un esquema copo de nieve, una tabla de dimensión puede estar dividida en múltiples tablas relacionadas, cada una representando un nivel diferente de detalle. Estas subtablas están conectadas entre sí a través de claves foráneas, formando una estructura de copo de nieve. Esta división y normalización adicional permiten reducir la redundancia, ahorrar espacio de almacenamiento y mejorar el rendimiento de las consultas complejas.

Sin embargo, el esquema copo de nieve puede ser más complejo de entender y utilizar en comparación con el esquema estrella. Las consultas pueden requerir más joins

entre tablas, lo que puede afectar el rendimiento(mayor capacidad de computo para realizar la query o consulta) en algunas situaciones.



Ambos esquemas tienen sus ventajas y desventajas, y la elección entre ellos dependerá de los requisitos y características específicas de tu aplicación o proyecto. Es importante considerar el volumen de datos, los requisitos de consulta, el rendimiento esperado y la capacidad de mantenimiento al decidir qué esquema utilizar.

Diferencias:

El esquema estrella es el tipo más simple de esquema de almacén de datos. Recibe su nombre de su estructura, que se asemeja a una estrella.

En comparación, el esquema copo de nieve (snowflake schema) es una extensión del esquema estrella que agrega dimensiones adicionales. Se llama copo de nieve porque su diagrama se asemeja a un copo de nieve.

En un esquema estrella, solo se requiere una única unión (join) para establecer la relación entre la tabla de hechos y las tablas de dimensiones.

El esquema estrella contiene una tabla de hechos rodeada de tablas de dimensiones.

En cambio, el esquema copo de nieve está rodeado por tablas de dimensiones, las cuales a su vez están rodeadas por otras tablas de dimensiones.

Un esquema copo de nieve requiere múltiples uniones (joins) para recuperar los datos. En cuanto a la comparación entre el esquema estrella y el esquema copo de nieve, el **esquema estrella** tiene un diseño de base de datos más simple, mientras que el esquema **copo de nieve** tiene un diseño de base de datos mucho más complejo.

| Star Schema | Snowflake Schema |
|---|---|
| Las jerarquías de las dimensiones se almacenan en la tabla dimensional. | Las jerarquías se dividen en tablas separadas. |
| Contiene una tabla de hechos rodeada de tablas de dimensiones. | Una tabla de hechos rodeada de tablas de dimensiones, las cuales a su vez están rodeadas por otras tablas de dimensiones. |
| En un esquema estrella, solo se requiere una única unión (join) para establecer la relación entre la tabla de hechos y las tablas de dimensiones. | Un esquema copo de nieve requiere múltiples uniones (joins) para recuperar los datos. |
| Diseño de base de datos simple. | Diseño de base de datos muy complejo. |
| Estructura de datos desnormalizada y las consultas también se ejecutan más rápido. | Estructura de datos normalizada. |
| Alto nivel de redundancia de datos. | Muy baja redundancia de datos. |
| Una sola tabla de dimensión contiene datos agregados. | Datos separados en diferentes tablas de dimensión. |
| Procesamiento de cubos más rápido. | El procesamiento de cubos puede ser lento debido a las uniones complejas. |
| Ofrece consultas de alto rendimiento utilizando la optimización de consultas de unión estrella. | El esquema copo de nieve se representa mediante una tabla de hechos centralizada que no está conectada con múltiples dimensiones. Las tablas pueden estar conectadas con múltiples dimensiones. |

Ejemplo de esquema estrella:

Supongamos que tenemos una base de datos de una tienda en línea y queremos realizar consultas sobre las ventas. Aquí tienes una estructura básica de esquema estrella:

1. Tabla de hechos (fact table): "Ventas"

- Columnas:

```
CREATE TABLE IF NOT EXISTS(
venta_id (clave primaria) INT NOT NULL
fecha_venta DATETIME
producto_id
cantidad_vendida
monto_total
)
```

2. Tablas de dimensiones:

a) Tabla "Productos":

- Columnas: producto_id (clave primaria), nombre_producto, categoría, precio.

b) Tabla "Clientes":

- Columnas: cliente_id (clave primaria), nombre_cliente, dirección, ciudad.

c) Tabla "Tiendas":

- Columnas: tienda_id (clave primaria), nombre_tienda, ubicación.

Consulta de ejemplo en SQL en el esquema estrella:

```
SELECT p.nombre_producto, SUM(v.cantidad_vendida) AS total_vendido
FROM ventas v
JOIN productos p
ON v.producto_id = p.producto_id
WHERE v.fecha_venta BETWEEN '2023-01-01' AND '2023-03-31'
GROUP BY p.nombre_producto
```

Hardcoding = poner un valor en específico

Esta consulta obtiene el total vendido por cada producto durante el primer trimestre de 2023.

MER para el **esquema estrella**:

| Ventas | Productos |
|--------|-----------|
| | |



Ejemplo de esquema copo de nieve:

Utilicemos el mismo escenario de la tienda en línea, pero ahora con una estructura de esquema copo de nieve más normalizada:

1. Tablas de dimensiones:

a) Tabla "Productos":

- Columnas: producto_id (clave primaria), nombre_producto, categoría_id.

b) Tabla "Categorías":

- Columnas: categoría_id (clave primaria), nombre_categoria.

c) Tabla "Clientes":

- Columnas: cliente_id (clave primaria), nombre_cliente, dirección, ciudad.

d) Tabla "Tiendas":

- Columnas: tienda_id (clave primaria), nombre_tienda, ubicación.

Consulta de ejemplo en SQL en el esquema copo de nieve:

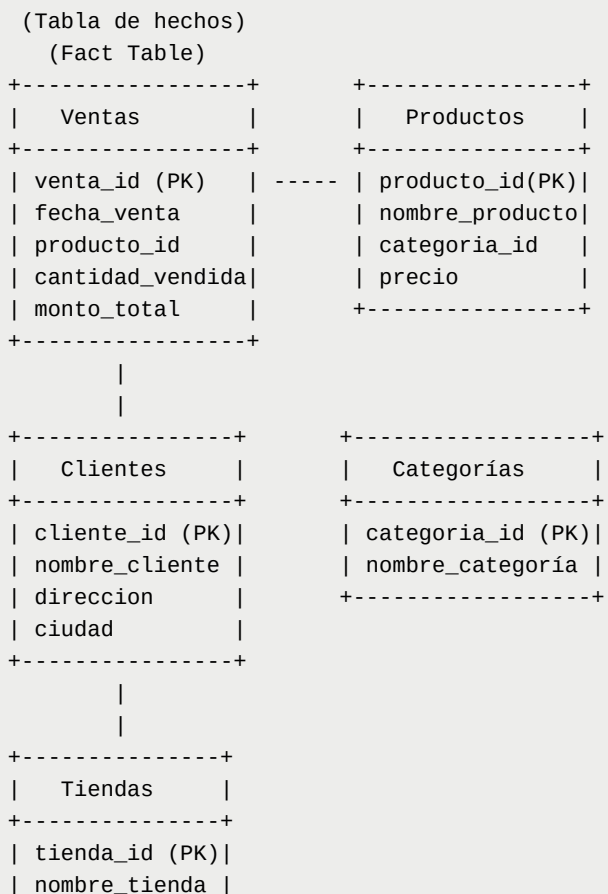
```
SELECT c.nombre_categoria, SUM(v.cantidad_vendida) AS total_vendido
FROM Ventas v
JOIN Productos p ON v.producto_id = p.producto_id
JOIN Categorías c ON p.categoria_id = c.categoria_id
WHERE v.fecha_venta BETWEEN '2023-01-01' AND '2023-03-31'
GROUP BY c.nombre_categoria;
```

Esta consulta obtiene el total vendido por cada categoría de productos durante el primer trimestre de 2023, utilizando las tablas de dimensiones adicionales en el esquema copo de nieve.

MER para el esquema copo de nieve:

1. Tabla de hechos (fact table):

"Ventas" (misma estructura que en el ejemplo del esquema estrella).




```
| ubicacion      |
+-----+
```

MER para el esquema **copo de nieve**:

```
+-----+      +-----+
| Ventas      |      | Productos      |
+-----+      +-----+
| venta_id (PK) |      | producto_id (PK) |
| fecha_venta   |      | nombre_producto |
| producto_id   |      | categoria_id   |
| cantidad_vendida |      | precio           |
| monto_total   |      +-----+
+-----+

      |
      |

+-----+
| Clientes     |
+-----+
| cliente_id (PK) |
| nombre_cliente |
| direccion      |
| ciudad         |
+-----+

      |
      |

+-----+
| Tiendas      |
+-----+
| tienda_id (PK) |
| nombre_tienda |
| ubicacion      |
+-----+
```