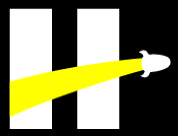
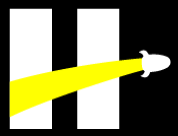
Query Optimization



Query Optimization

La optimización de consultas es de gran importancia para el rendimiento de una base de datos relacional, especialmente para la ejecución de sentencias SQL complejas. Un optimizador de consultas decide los mejores métodos para implementar cada consulta.

El optimizador de consultas selecciona, por ejemplo, si desea o no usar índices para una consulta determinada y qué métodos de unión usar al unir varias tablas. Estas decisiones tienen un tremendo efecto en el rendimiento de SQL, y la optimización de consultas es una tecnología clave para cada aplicación, desde sistemas operativos hasta sistemas de almacenamiento de datos y sistemas analíticos hasta sistemas de gestión de contenido.



Query Optimization

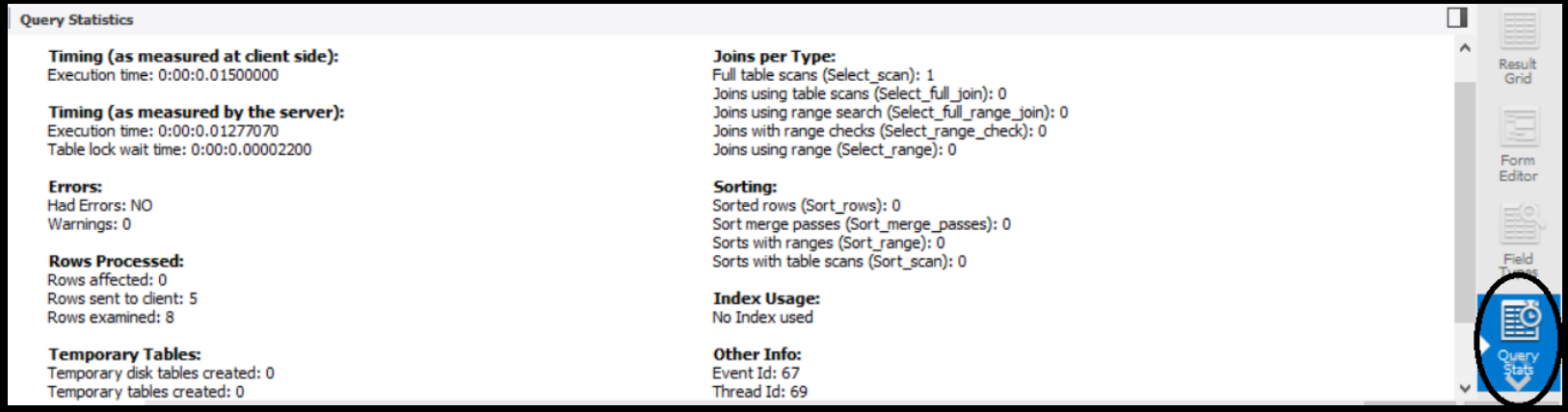
Las principales consideraciones para optimizar las consultas son:

- Configurar índices en las columnas utilizadas en la cláusula, acelera la evaluación, el filtrado y la recuperación final de los resultados.
- Minimizar el número de análisis de tablas completas en sus consultas, especialmente para tablas grandes. Consultar cuando sea posible a partir de criterios específicos.
- La investigación de la documentación del SGBD para poder utilizar funciones y procedimientos que performen de mejor que otras opciones.

Por lo general LEFT(Nombre) = 'Rod' es menos performante que LIKE 'Rod%'

Estadísticas de consultas

La ficha Resultados del editor SQL posee Estadísticas de consulta que utiliza datos del esquema de rendimiento para recopilar métricas clave para la consulta ejecutada, como la sincronización, las tablas temporales, los índices, las combinaciones y mucho más. Las estadísticas en MySQL se pueden consultar en el margen derecho de la pantalla de resultados, mediante la opción "Query Stats".

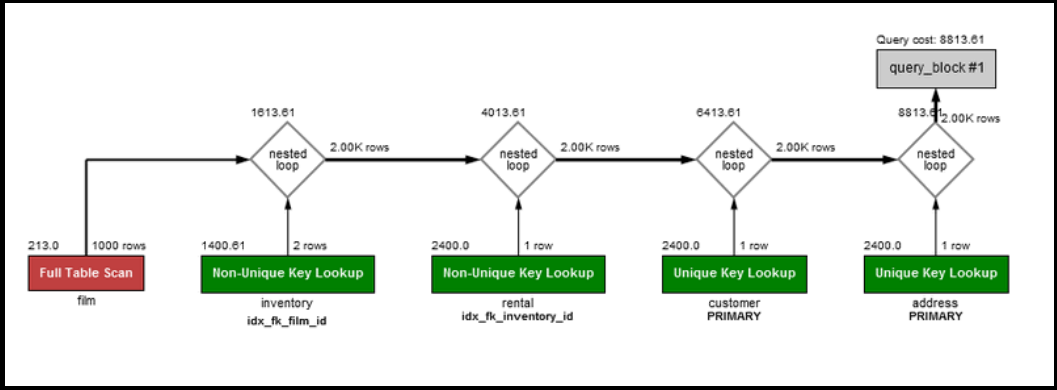


Plan de explicación visual

La característica de explicación visual genera y muestra una representación visual de la instrucción MySQL EXPLAIN mediante el uso de información extendida disponible en el formato JSON extendido.

MySQL Workbench proporciona todos los formatos para las consultas ejecutadas, incluido el JSON extendido sin procesar, el formato tradicional y el plan de consulta visual.

Este es un complemento que debe ser instalado en Workbench.

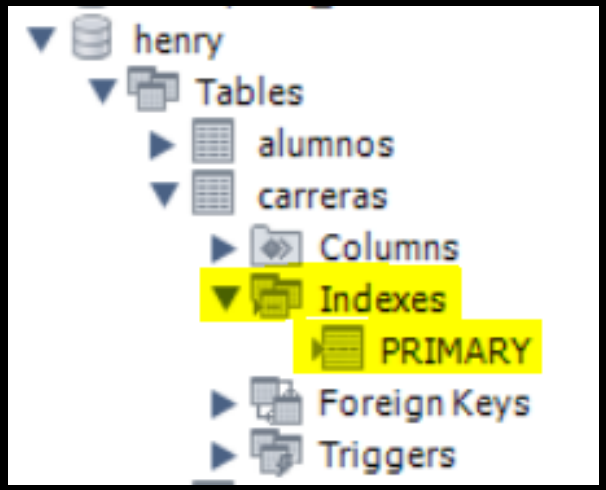


SQL – Índices

Los índices de SQL son la principal herramienta de rendimiento, por lo que generalmente se aplican si una base de datos se incrementa. SQL Server reconoce varios tipos de índices, pero uno de los más comunes es el índice agrupado. Esta clase de índice se crea automáticamente con una PK. En el momento en el que se ejecuta la consulta, SQL Server creará automáticamente un índice agrupado en la columna especificada.

```
CREATE TABLE carrera (
        idCarrera INT NOT NULL AUTO_INCREMENT,
        nombre VARCHAR (20) NOT NULL,
        PRIMARY KEY (idCarrera) --Aquí al crear una PK, SQL además crea un índice agrupado.
);
```

SQL – Índices


henry
Tables
alumnos
carreras
Columns
Indexes
PRIMARY
Foreign Keys
Triggers

SQL – Índices

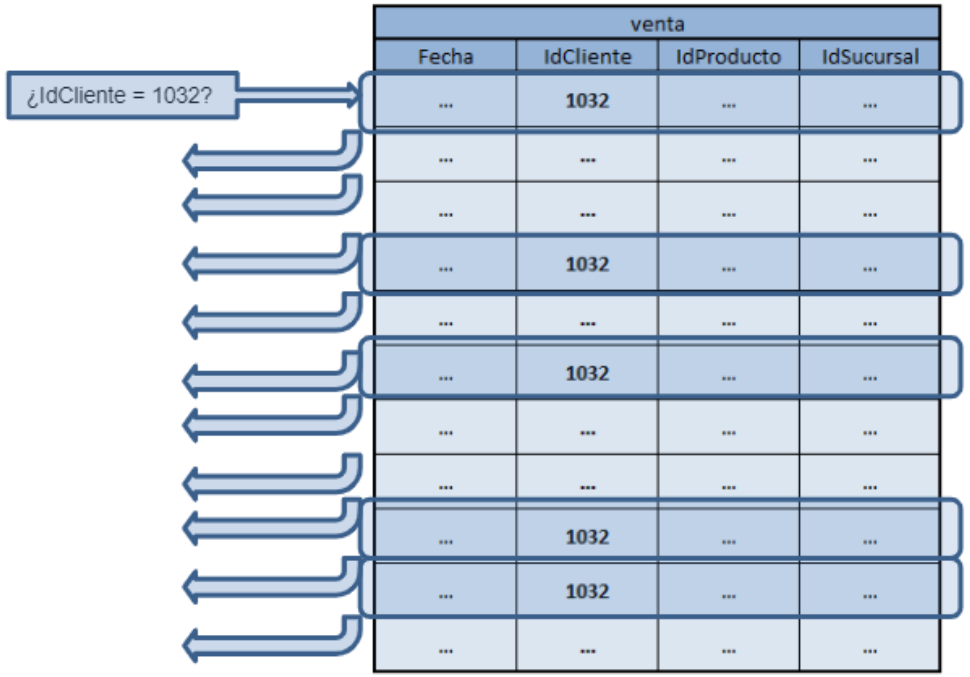
Los índices de las tablas ayudan a indexar el contenido de diversas columnas para facilitar la búsquedas de contenido de cuando se ejecutan consultas sobre esas tablas.
De ahí que la creación de índices optimiza el rendimiento de las consultas y a su vez el de la BBDD, pueden agregarse índices en caso de tablas puentes donde no se ha solucionado el problema de indexación aplicado claves concatenadas.

En MySQL puede utilizarse CREATE INDEX para crear o añadir índices en las tablas de una base de datos.

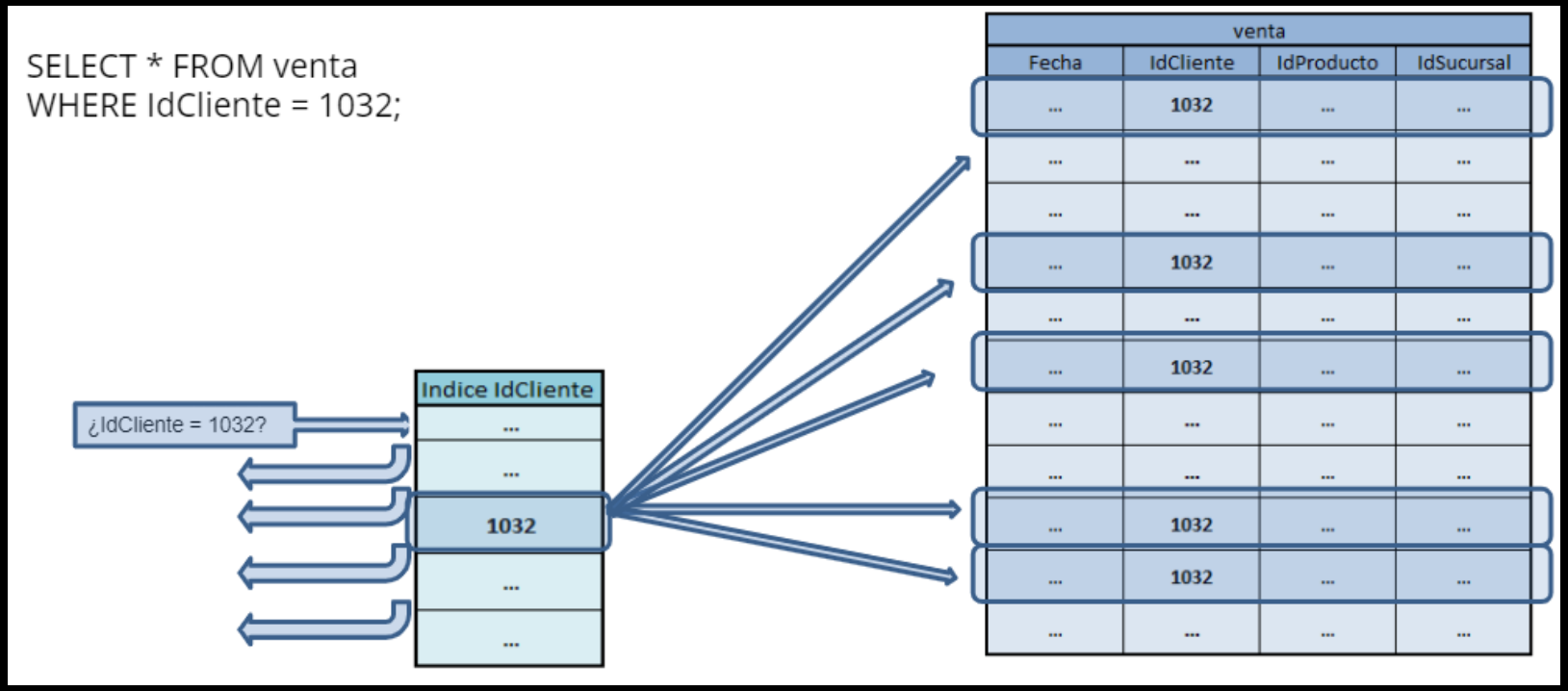
```
MySQL
1 CREATE INDEX nombre_indice ON nombrede_tabla(columna [columna2...]);
```

SQL – Índices

```
SELECT * FROM venta
WHERE IdCliente = 1032;
```



SQL – Índices


SELECT * FROM venta
WHERE IdCliente = 1032;
¿IdCliente = 1032?
Indice IdCliente
...
...
1032
...
...
venta
Fecha
IdCliente
IdProducto
IdSucursal
1032
1032
1032
1032
1032

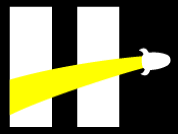
SQL – Índices

Podemos tener los siguientes tipos de índices en una tabla de MySQL:

- Únicos.
- Primarios.
- Ordinarios.
- De texto completo.
- Parte de campos o columnas.

También se pueden eliminar índices mediante la sentencia DROP INDEX, siempre teniendo presente que la utilización de la sentencia DROP puede llevar a consecuencias indeseadas.

```mysql
DROP INDEX 'segundo_apellido' ON clientes
DROP INDEX 'PRIMARY' ON clientes;
```



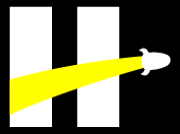
Optimización en WHERE

Un tema en la optimización es un enfoque constante en la cláusula WHERE. ¡Cuanto más rápido podamos dividir nuestro conjunto de datos a solo las filas que necesitamos, más eficiente será la ejecución de la consulta!

Ejemplo:

```
#Utilizamos IN para crear un conjunto en la segmentación
SELECT *
FROM instructores
WHERE idInstructor IN (1,2,3,4,5)

#Utilizamos OR para crear el mismo conjuntos
SELECT *
FROM instructores
WHERE idInstructor = 1 OR idInstructor = 2 OR idInstructor = 3
OR idInstructor = 4 OR idInstructor = 5
```



Formas Normales

Las formas normales son conjuntos de criterios que utilizamos para diseñar la estructura de las bases de datos. Para mejorar el desempeño de una base de datos, así como evitar redundancia en la información que contiene y, en consecuencia, generar condiciones para un mejor diseño, se deben conocer las formas de normalización y condiciones en las que la desnormalización es recomendable.

FN 1

Una relación se encuentra en 1FN sólo si cada uno de sus atributos contiene un único valor para un registro determinado.

Codigo	Nombre	Apellido	Sucursal
3467	Daniel Arnoldo	Abib	Avellaneda, Caseros y Córdoba Quiroz
3243	Hugo Briz	Abilio	Quilmes y Velez
3124	Jaime Gaston	Acevedo Salinas	Caballito, Caseros y Quilmes

Codigo	Nombre	Apellido
3467	Daniel Arnoldo	Abib
3243	Hugo Briz	Abilio
3124	Jaime Gaston	Acevedo Salinas

Codigo	Sucursal
3467	Avellaneda
3467	Caseros
3467	Córdoba Quiroz
3243	Quilmes
3243	Velez
3124	Caballito
3124	Cabildo
3124	Mendoza
3124	Moron

FN 2

Una relación se encuentra en 2FN sólo si se cumple 1FN y todos sus atributos no clave dependen en forma completa de la clave.

Codigo_Cliente	Codigo_Sucursal	Apellido_Nombre	Sucursal	Ventas
3467	101	Abib, Daniel Arnoldo	Avellaneda	3
3467	103	Abib, Daniel Arnoldo	Caseros	2
3467	106	Abib, Daniel Arnoldo	Córdoba Quiroz	5
3243	107	Abilio, Hugo Briz	Quilmes	6
3243	110	Abilio, Hugo Briz	Velez	2
3124	109	Acevedo Salinas, Jaime Gaston	Caballito	3
3124	103	Acevedo Salinas, Jaime Gaston	Caseros	1
3124	107	Acevedo Salinas, Jaime Gaston	Quilmes	2

Codigo_Cliente	Codigo_Sucursal	Ventas
3467	101	3
3467	103	2
3467	106	5
3243	107	6
3243	110	2
3124	109	3
3124	103	1
3124	107	2

Codigo_Cliente	Apellido_Nombre
3467	Abib, Daniel Arnoldo
3243	Abilio, Hugo Briz
3124	Acevedo Salinas, Jaime Gaston

Codigo_Sucursal	Sucursal
101	Avellaneda
103	Caseros
106	Córdoba Quiroz
107	Quilmes
110	Velez
109	Caballito
103	Caseros
107	Quilmes

FN 3

Una relación se encuentra en 3FN sólo si se cumple 2FN y los campos no clave dependen únicamente de la clave o los campos no clave no dependen unos de otros.

Codigo_Cliente	Apellido_Nombre	Localidad	Provincia
3467	Abib, Daniel Arnoldo	Cordoba	Cordoba
3243	Abilio, Hugo Briz	Quilmes	Buenos Aires
3124	Acevedo Salinas, Jaime Gaston	CABA	Buenos Aires

Codigo_Cliente	Apellido_Nombre	Localidad
3467	Abib, Daniel Arnoldo	Cordoba
3243	Abilio, Hugo Briz	Quilmes
3124	Acevedo Salinas, Jaime Gaston	CABA

Localidad	Provincia
Cordoba	Cordoba
Quilmes	Buenos Aires
CABA	Buenos Aires

FN 4

Una relación se encuentra en 4FN sólo si se cumple 3FN y no posee dependencias multivaluadas no triviales.

Sucursal	Canal de Venta	Producto
Avellaneda	OnLine	Parlante Jbl Go Blue Bluetooth
Avellaneda	Presencial	Parlante Jbl Go Blue Bluetooth
Caseros	Presencial	Parlante Kingta So-101 Bluetooth/ Waterproof Red
Caseros	Presencial	Parlante Jbl Go Blue Bluetooth
Caballito	OnLine	Parlante Jbl Go Blue Bluetooth
Caballito	OnLine	Parlante Kingta So-101 Bluetooth/ Waterproof Red
Caballito	OnLine	Parlante Jbl Flip 4 Gray Bluetooth

Sucursal	Canal de Venta
Avellaneda	OnLine
Avellaneda	Presencial
Caseros	Presencial
Caballito	OnLine

Sucursal	Producto
Avellaneda	Parlante Jbl Go Blue Bluetooth
Caseros	Parlante Kingta So-101 Bluetooth/ Waterproof Red
Caseros	Parlante Jbl Go Blue Bluetooth
Caballito	Parlante Jbl Go Blue Bluetooth
Caballito	Parlante Kingta So-101 Bluetooth/ Waterproof Red
Caballito	Parlante Jbl Flip 4 Gray Bluetooth

FN 5

Una relación se encuentra en 5FN sólo si se cumple 4FN y cada dependencia de unión en ella es implicada por las claves candidatas.

Sucursal	Canal de Venta	Producto
Avellaneda	OnLine	Parlante Jbl Go Blue Bluetooth
Avellaneda	Presencial	Parlante Jbl Go Blue Bluetooth
Caseros	Presencial	Parlante Kingta So-101 Bluetooth/ Waterproof Red
Caseros	Presencial	Parlante Jbl Go Blue Bluetooth
Caballito	OnLine	Parlante Jbl Go Blue Bluetooth
Caballito	OnLine	Parlante Kingta So-101 Bluetooth/ Waterproof Red
Caballito	OnLine	Parlante Jbl Flip 4 Gray Bluetooth

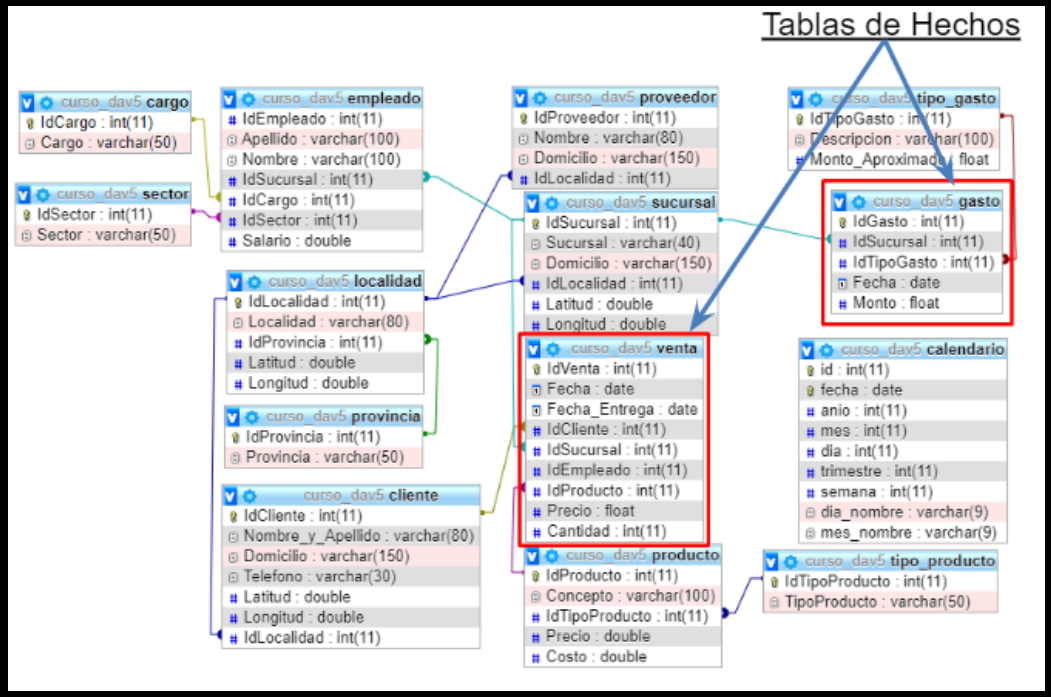
Sucursal	Producto
Avellaneda	Parlante Jbl Go Blue Bluetooth
Caseros	Parlante Kingta So-101 Bluetooth/ Waterproof Red
Caseros	Parlante Jbl Go Blue Bluetooth
Caballito	Parlante Jbl Go Blue Bluetooth
Caballito	Parlante Kingta So-101 Bluetooth/ Waterproof Red
Caballito	Parlante Jbl Flip 4 Gray Bluetooth

Sucursal	Canal de Venta
Avellaneda	OnLine
Avellaneda	Presencial
Caseros	Presencial
Caballito	OnLine

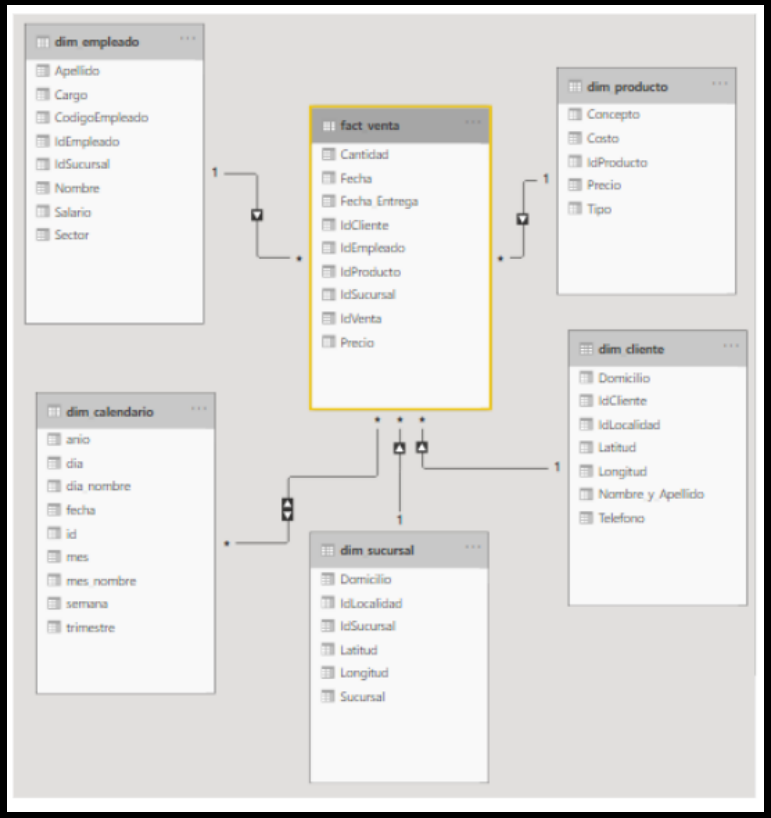
Canal de Venta	Producto
OnLine	Parlante Jbl Go Blue Bluetooth
Presencial	Parlante Jbl Go Blue Bluetooth
Presencial	Parlante Kingta So-101 Bluetooth/ Waterproof Red
OnLine	Parlante Kingta So-101 Bluetooth/ Waterproof Red
OnLine	Parlante Jbl Flip 4 Gray Bluetooth

Modelos de Datos

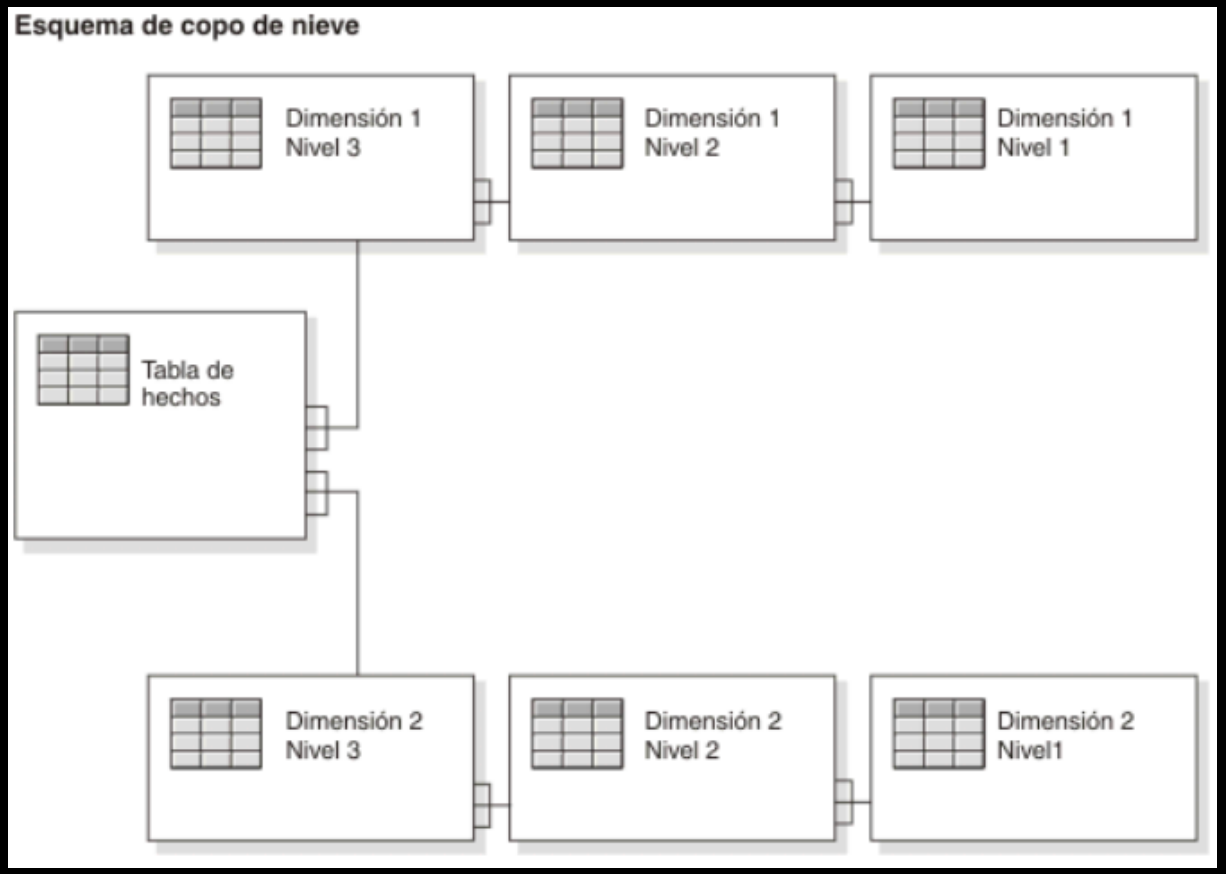
Los modelos de datos buscan representar un realidad que es posible representar mediante las entidades que la conforman. Esas entidades quedan representadas en tablas, y pueden ser de dos tipos:



Modelos de Estrella



Modelos de Copo de Nieve



Modelos de Copo de Nieve