

SESIÓN 6 – BUENAS PRÁCTICAS EN EL MODELADO MULTIDIMENSIONAL

El modelado multidimensional es fundamental para sistemas de Business Intelligence, permitiendo análisis ágiles y toma de decisiones basadas en datos. Esta sesión explora las mejores prácticas para diseñar modelos efectivos que balancean rendimiento y usabilidad.

 por Kibernum Capacitación S.A.



Normalización vs. Desnormalización

Normalización

Busca evitar la redundancia dividiendo los datos en múltiples tablas (modelo relacional clásico).

Desnormalización

Prioriza la lectura rápida y simplifica las relaciones, aunque repita datos.

Aplicación en modelos multidimensionales:

Esquema	Enfoque	Ideal para...
Estrella	Desnormalizado	Análisis ágil, dashboards, usuarios no técnicos
Copo de nieve	Parcialmente normalizado	Escenarios más complejos, integridad de datos

Referencia: The Data Warehouse Toolkit – Ralph Kimball:

¿Qué priorizarías en un sistema BI para una empresa retail: rendimiento o consistencia de datos?

Selección de atributos



Incluir solo los atributos que aportan al análisis

Mantener el modelo limpio y enfocado en lo relevante



Usar nombres claros y legibles para los usuarios finales

Facilitar la comprensión y adopción del modelo

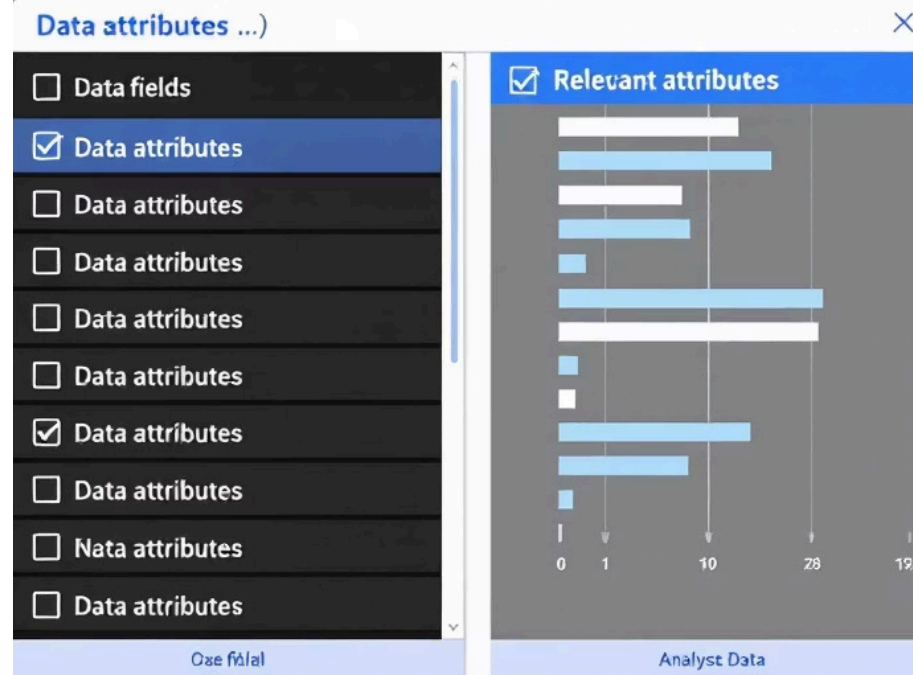


Evitar campos técnicos irrelevantes

Como fecha_mod, id_usuario_log que no aportan al análisis

Consejo: puedes enriquecer atributos para facilitar filtros.

Ejemplo: en la dimensión "cliente", agregar un campo rango_edad.

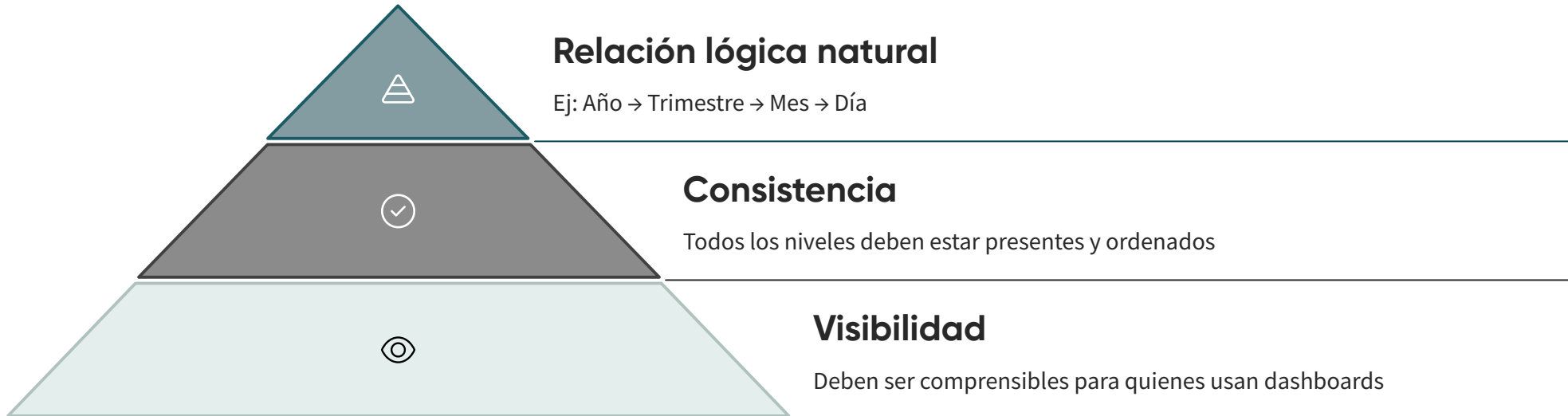


Analys Data

Diseño de jerarquías

Por qué importan:

Las jerarquías permiten navegar entre niveles de agregación, usando operaciones como drill-down y roll-up.



Design and query tabular and multi-dimensional models
using Microsoft's SQL Server Analysis Services

Herramientas para modelado multidimensional

Herramientas clave:

SQL Server Analysis Services (SSAS)

Creación de cubos OLAP, jerarquías, KPIs

Power BI

Visualización de modelos analíticos

Tableau

Exploración interactiva de dimensiones

MySQL Workbench / DBeaver

Diseño de esquemas tipo estrella

Divídanse en grupos, asignar una herramienta a cada grupo buscar que busquen 1 captura de pantalla o video corto mostrando cómo se usa en un modelo OLAP, investiga de forma, usa todas las fuentes disponibles.

Casos de uso prácticos



Retail

Análisis por categoría y canal de ventas.



Educación

Seguimiento del rendimiento por cohorte y asignatura.



Salud

Indicadores por centro, especialidad y período.

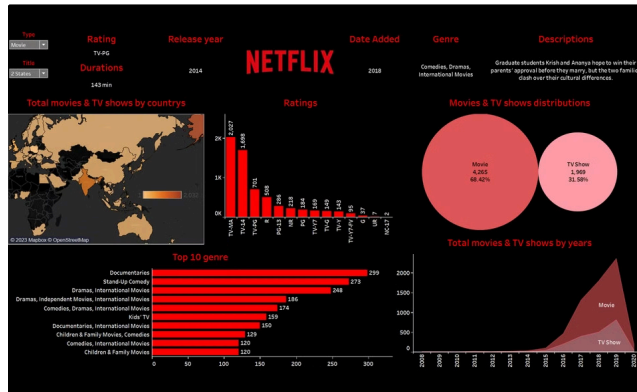
A lo largo del modulo pudimos comentar algunos de estos casos, por ejemplo

Retail: análisis por categoría y canal de ventas.

Educación: seguimiento del rendimiento por cohorte y asignatura.

Salud: indicadores por centro, especialidad y período.

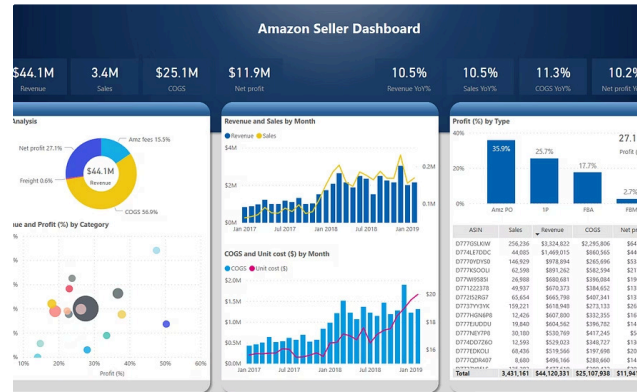
Casos de estudio reales de modelamiento multidimensional



Netflix

Analiza comportamiento de visualización cruzando dimensiones como género, perfil, tiempo y tipo de dispositivo.

Que ejemplos pueden agregar, discutan en clase estos ejemplos y agreguen mas ejemplos de modelamiento multidimensional.



Amazon

Explora ventas históricas por producto, ubicación, día, tipo de usuario.



Ministerios de salud

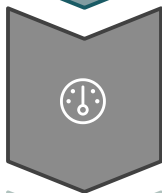
Seguimiento de campañas por comuna, mes, categoría de riesgo.

Aplicaciones prácticas



Reportes ejecutivos semanales

Imagina que trabajas como ingeniero/a de datos en una empresa. ¿Cómo aplicarías un modelo multidimensional en este caso?



Dashboards de autoservicio para clientes internos

Imagina que trabajas como ingeniero/a de datos en una empresa. ¿Cómo aplicarías un modelo multidimensional en este caso?



Análisis de campañas pasadas por canal y temporada

Imagina que trabajas como ingeniero/a de datos en una empresa. ¿Cómo aplicarías un modelo multidimensional en este caso?

Tengan en cuenta que:

Un buen modelo no es solo técnico: es comprensible, útil y alineado al negocio.

Jerarquías, medidas, dimensiones y hechos bien definidos = análisis poderoso.

Las herramientas son importantes, pero las preguntas de negocio son la guía principal.

"Un gráfico sin un buen modelo detrás puede ser bonito, pero vacío. Un buen modelo transforma los datos en decisiones."

SESIÓN ACOMPAÑADA – ACTIVIDAD PREPARATORIA AL EXAMEN FINAL

Crear la base de datos para análisis

Primero creamos las BBDD y luego nos conectamos a la BBDD creada usando en comando "USE {nombre_bbdd};" también puedes comprobar que la BBDD esta creada correctamente con el comando "SHOW DATABASES;"

Crear las dimensiones

Producto, Sucursal, Tiempo. "Cada dimensión da contexto: el qué (producto), el dónde (sucursal), y el cuándo (fecha)."

Definir la tabla de hechos

"Esta tabla representa los hechos del negocio: movimientos logísticos con cantidades y montos. Lo conectamos con dimensiones clave."

Insertar datos y consultar

Insertar datos simulados y realizar consultas de agregación para análisis.

Dibujar el esquema estrella

Utilizar herramientas como dbdiagram.io para visualizar la estructura en estrella del modelo.

Introducción del docente Imaginemos que estamos en el área de logística de un supermercado mediano con múltiples sucursales. Este equipo debe hacer seguimiento de productos, entradas y salidas del almacén, así como analizar el comportamiento de las ventas por canal y ubicación. La empresa ya cuenta con un sistema transaccional que guarda los movimientos en una base OLTP (como MySQL), pero la gerencia necesita reportes históricos, visualizaciones por categoría, producto o sucursal... Y aquí es donde entra el ingeniero de datos. Nuestra tarea será extraer esa información desde la base de datos operativa (simulada), y diseñar un modelo OLAP tipo estrella que nos permita realizar análisis agregados. A futuro, este modelo podría incluso conectarse con Python y Pandas para reportes automáticos o dashboards en Power BI.

Paso a paso guiado En este paso a paso usaremos MySQL, (Terminal de MySQL Shell y MySQL WorkBench),

1. Crear la base de datos para análisis

```
MySQL localhost:33060+ ssl ventas_olap_directa SQL > CREATE DATABASE supermercado_olap;
Query OK, 1 row affected (0.0150 sec)
MySQL localhost:33060+ ssl ventas_olap_directa SQL > USE supermercado_olap;
Default schema set to 'supermercado_olap'.
Fetching global names, object names from 'supermercado_olap' for auto-completion... Press ^C to stop.
MySQL localhost:33060+ ssl supermercado_olap SQL > |
```

- 2.

Imagen: primero creamos las BBDD y luego nos conectamos a la BBDD creada usando en comando "USE {nombre_bbdd};" también puedes comprobar que la BBDD esta creada correctamente con el comando "SHOW DATABASES;" observa:

```
Query OK, 0 rows affected (0.0789 sec)
MySQL localhost:33060+ ssl supermercado_olap SQL > show databases;
+-----+
| Database |
+-----+
| information_schema |
| mysql |
| performance_schema |
| sakila |
| supermercado_olap |
| sys |
| ventas_olap |
| ventas_olap_directa |
| world |
+-----+
9 rows in set (0.0061 sec)
MySQL localhost:33060+ ssl supermercado_olap SQL > |
```

Ten en cuenta creamos una base separada para no afectar la operativa, como haríamos en un entorno real con Data Warehouse o un Data Mart temático.

1. Crear las dimensiones Las siguientes imágenes muestran sentencias SQL que serán introducidas en la terminal. Para mejorar la visibilidad del código y facilitar su comprensión, los pantallazos serán capturados desde Visual Studio Code (VSC), donde la sintaxis resaltada por colores mejora significativamente la experiencia de aprendizaje.

Producto:

```
-- Crear dimensiones
CREATE TABLE producto (
  id_producto INT PRIMARY KEY,
  nombre TEXT,
  categoria TEXT,
  proveedor TEXT
);
```

Sucursal:

```
CREATE TABLE sucursal (
  id_sucursal INT PRIMARY KEY,
  ciudad TEXT,
  zona TEXT
);
```

Tiempo:

```
CREATE TABLE fecha (
  id_fecha INT PRIMARY KEY,
  fecha DATE,
  mes TEXT,
  año INT
);
```

"Cada dimensión da contexto: el qué (producto), el dónde (sucursal), y el cuándo (fecha)."

1. Definir la tabla de hechos

```
-- Definir la tabla de hechos
CREATE TABLE movimientos (
  id BIGINT PRIMARY KEY AUTO_INCREMENT,
  id_producto INT,
  id_sucursal INT,
  id_fecha INT,
  tipo_movimiento TEXT, -- entrada o salida
  cantidad INT,
  total DECIMAL(10,2),
  CONSTRAINT fk_prod FOREIGN KEY (id_producto) REFERENCES producto(id_producto),
  CONSTRAINT fk_suc FOREIGN KEY (id_sucursal) REFERENCES sucursal(id_sucursal),
  CONSTRAINT fk_fecha FOREIGN KEY (id_fecha) REFERENCES fecha(id_fecha)
);
```

Imagen: "Esta tabla representa los hechos del negocio: movimientos logísticos con cantidades y montos. Lo conectamos con dimensiones clave. "Cada dimensión da contexto: el qué (producto), el dónde (sucursal), y el cuándo (fecha)."

1. Insertar datos simulados (mínimo 3 registros por tabla) Aquí puedes guiar al grupo con ejemplos, puedes crear los tuyos y subirlos a la BBDD.

```
INSERT INTO producto VALUES
(1, 'Arroz 1kg', 'Alimentos', 'Distribuidora Sur'),
(2, 'Leche entera', 'Lácteos', 'Lacteos Chile');

INSERT INTO sucursal VALUES
(1, 'Valparaíso', 'Costa'),
(2, 'Santiago', 'Centro');

INSERT INTO fecha VALUES
(101, '2024-01-10', 'Enero', 2024),
(102, '2024-01-15', 'Enero', 2024);

INSERT INTO movimientos VALUES
(NULL, 1, 1, 101, 'entrada', 100, 75000),
(NULL, 2, 2, 102, 'salida', 30, 29970);
```

Una vez creadas y pobladas las tablas puedes verificar que todo este ok con las siguientes consultas:

Para ver que tablas están creadas en la BBDD:

```
MySQL localhost:33060+ ssl supermercado_olap SQL > show tables;
+-----+
| Tables_in_supermercado_olap |
+-----+
| fecha |
| movimientos |
| producto |
| sucursal |
+-----+
```

Para seleccionar una tabla y ver su contenido:

```
MySQL localhost:33060+ ssl supermercado_olap SQL > select * from producto;
+-----+-----+-----+-----+
| id_producto | nombre | categoria | proveedor |
+-----+-----+-----+-----+
| 1 | Arroz 1kg | Alimentos | Distribuidora Sur |
| 2 | Leche entera | Lácteos | Lacteos Chile |
+-----+-----+-----+-----+
```

Consultar el modelo con una agregación

```
SELECT
  f.mes,
  p.categoria,
  SUM(m.cantidad) AS total_items,
  SUM(m.total) AS total_monto
FROM movimientos m
JOIN producto p ON m.id_producto = p.id_producto
JOIN fecha f ON m.id_fecha = f.id_fecha
GROUP BY f.mes, p.categoria
ORDER BY f.mes;
```

Output:

mes	categoria	total_items	total_monto
Enero	Alimentos	100	75000.00
Enero	Lácteos	30	29970.00

Este tipo de consulta es fundamental para los equipos comerciales, ya que les permite identificar qué categorías de productos tienen mayor movimiento mes a mes. Al utilizar técnicas avanzadas de SQL, como las funciones de agregación junto con cláusulas JOIN y GROUP BY, es posible responder a preguntas clave del negocio, tales como: ¿Cuántas unidades se vendieron por categoría en cada mes? ¿Cuál es el ingreso total generado por cada categoría de producto mensualmente? ¿Cómo varían las ventas de diferentes categorías a lo largo del tiempo? Estas consultas proporcionan información valiosa para la toma de decisiones estratégicas y la planificación de inventarios. Anímate y crea consultas SQL que respondan a esas interrogantes. Fuente: LearnSQL

Dibujar el esquema estrella

Utilicen la herramienta dbdiagram.io para **crear** el diagrama del **esquema en estrella** correspondiente al modelo de datos que hemos desarrollado, pueden dibujarlo a mano en una hoja de papel. Al finalizar, observen la estructura creada *¿logran visualizar la formación de una estrella?* Este ejercicio les permitirá comprender cómo la tabla de hechos central se conecta directamente con las tablas de dimensiones, formando la característica estructura estelar.

¿Qué viene ahora?

La actividad que acabamos de realizar no solo es un ejercicio práctico, sino que también sirve como preparación directa para el **examen final del módulo**. En esa evaluación, abordaremos un caso similar, pero más amplio, que incluirá nuevas dimensiones y tipos de análisis.

Es fundamental que aprovechen esta oportunidad para practicar y comprender a fondo el diseño de modelos de datos. Tomarse el tiempo necesario para entender y modelar correctamente es crucial, ya que este ejercicio introductorio nos permite explorar los conceptos en un entorno controlado.

En un escenario real, el proceso de modelado de datos sigue una secuencia lógica:

Modelado conceptual: Se identifican las entidades y relaciones principales del negocio, generalmente representadas en diagramas Entidad-Relación (ER).

Modelado lógico: Se detallan las estructuras de datos sin considerar aspectos físicos, definiendo tablas, campos y relaciones.

Modelado físico: Se implementa el modelo lógico en un sistema de gestión de bases de datos (SGBD), considerando aspectos como tipos de datos específicos, índices y restricciones.

Fuente: [Learn R, Python & Data Science Online+2Incentro+2Bismart Blog+2](#)

Este enfoque estructurado garantiza que el modelo de datos sea coherente, escalable y alineado con los objetivos del negocio.