



Modelado dimensional: técnicas avanzadas

Buenas prácticas para enriquecer modelos
analíticos sin comprometer la usabilidad



¿Qué vamos a ver?

En esta presentación exploramos cuatro tipos de dimensiones que permiten resolver necesidades comunes en el modelado dimensional sin sobrecargar el diseño:

1

Dimensiones degeneradas

2

Dimensiones con múltiples roles (role-playing)

3

Junk dimensions

4

Mini-dimensiones (tipo 4)

Degenerate dimensions

Son claves de negocio que se mantienen en la tabla de hechos, **sin atributos adicionales** y **sin crear una dimensión separada**. Se utilizan para trazabilidad, auditoría o drill-through, pero no para descripciones o agrupamientos.

Este tipo de dimensiones surgen cuando ciertos datos tienen valor analítico limitado pero son esenciales para vincular el hecho con un documento, ticket o transacción específica. Es una técnica que evita joins innecesarios.

Ejemplo de dimensión degenerada

Tabla de hechos: fact_ordenes

| order_id | customer_id | fecha_key | total_orden | nro_factura |
|----------|-------------|-----------|-------------|-------------|
| 1001 | C001 | 20240101 | 150.00 | FAC-0001 |
| 1002 | C002 | 20240102 | 300.00 | FAC-0002 |

- nro_factura es una **dimensión degenerada**: no tiene atributos, pero es útil para trazabilidad o navegación en reportes.

Cuándo conviene

**Cuando la clave de negocio
tiene valor operativo**

**Cuando no existen atributos
adicionales**

**Cuando se necesita acceso
directo a identificadores
únicos**

Role-playing dimensions

Una misma dimensión puede cumplir **múltiples roles** en una tabla de hechos. Esto sucede cuando diferentes columnas del hecho se refieren al mismo conjunto de atributos (por ejemplo, fechas), pero con significados distintos.

Es común en tablas que contienen varios hitos temporales, como fecha de orden, de envío o de entrega. Todas se refieren a dim_fecha, pero representan roles diferentes.

Implementación con vistas

```
create view fecha_orden as
select date_key as fecha_orden_key,
       day_of_week as dia_orden,
       month as mes_orden,
       ...
from dim_fecha;
```

```
create view fecha_envio as
select date_key as fecha_envio_key,
       day_of_week as dia_envio,
       month as mes_envio,
       ...
from dim_fecha;
```

Esto permite reutilizar dim_fecha en múltiples roles sin duplicar estructura, y con **nombres de columnas únicos** para evitar ambigüedad en herramientas de BI.

Tabla de hechos con role-playing

Tabla: fact_envios

| envio_id | producto_id | fecha_orden_key | fecha_envio_key |
|----------|-------------|-----------------|-----------------|
| E001 | P001 | 20240101 | 20240103 |

Cuándo conviene

Cuando una misma dimensión representa múltiples momentos o roles

Cuando se necesita evitar duplicación de estructuras

Cuando el modelo se integra con herramientas de BI que requieren nombres claros y únicos

Junk dimensions

Agrupar atributos de **baja cardinalidad** o flags que por sí solos no justifican una dimensión. Evitan llenar la tabla de hechos con columnas sueltas, mejoran la limpieza del modelo y facilitan filtros combinados.

Estos valores muchas veces provienen del sistema transaccional como selecciones de catálogo, campos de tipo o indicadores booleanos.

Tablas de origen

Tabla: order_headers

| order_id | payment_type | order_type | commission_flag |
|----------|--------------|------------|-----------------|
| 1001 | Cash | Inbound | Y |
| 1002 | Visa | Outbound | N |

Tabla: payment_types

| payment_type | payment_group |
|--------------|---------------|
| Cash | Cash |
| Visa | Credit |
| MasterCard | Credit |

Mapeo hacia junk dimension

| Atributo en junk dimension | Campo origen | Tabla origen |
|----------------------------|-----------------|---------------|
| tipo_pago | payment_type | order_headers |
| grupo_pago | payment_group | payment_types |
| tipo_orden | order_type | order_headers |
| indicador_comision | commission_flag | order_headers |

Dimensión: dim_indicadores_orden

| id_indicador | tipo_pago | grupo_pago | tipo_orden | indicador_comision |
|--------------|-----------|------------|------------|--------------------|
| 1 | Efectivo | Efectivo | Entrada | Con comisión |
| 2 | Visa | Crédito | Salida | Sin comisión |
| ... | ... | ... | ... | ... |

Ventajas y consideraciones

Ventajas



Simplifica la tabla de hechos



Centraliza múltiples indicadores



Facilita filtros combinados

Consideraciones



Cuidado con la **explosión de combinaciones** si los atributos son muy variados o cambian mucho.

Mini-dimensiones (Tipo 4)

Cuando una dimensión es muy grande y contiene atributos que cambian seguido, el uso de SCD tipo 2 puede ser costoso. En esos casos, conviene separar los atributos más volátiles en una dimensión más pequeña y discretizada.

Estas mini-dimensiones contienen combinaciones de atributos como edad, ingresos, puntuaciones de uso, etc., en formato de "perfiles". La cardinalidad se reduce agrupando valores continuos en rangos.

Caso típico

Atributos originales en dim_cliente:

| cliente_id | edad | score_frecuencia | ingreso |
|------------|------|------------------|---------|
| C001 | 25 | Bajo | 25,000 |

Se transforman en una **mini-dimensión** por perfil demográfico:

dim_demografica_cliente

| demographics_key | franja_etaria | score_frecuencia | nivel_ingresos |
|------------------|---------------|------------------|-------------------|
| 1 | 21-25 | Bajo | < \$30.000 |
| 2 | 26-30 | Bajo | \$30.000-\$39.999 |

¿Cómo se usa?

La tabla de hechos incluye **dos claves foráneas**:

1

FK a dim_cliente

2

FK a dim_demografica_cliente (estado vigente al momento del hecho)

Esto permite **trackear los cambios de perfil** sin modificar registros anteriores ni duplicar datos en la dimensión principal.

Ventajas y compromisos

Ventajas



Mejora el rendimiento de consultas



Permite tracking sin SCD completa



Reduce cardinalidad en la dimensión principal

Consideraciones



Requiere **discretización** (bandas de edad, ingresos, etc.)

Si no se limitan los valores posibles, puede crecer tanto como la dimensión original

Comparativa final

| Tipo de dimensión | Cuándo se usa | Ventaja principal | Riesgo común |
|-------------------|--|--------------------------------|--|
| Degenerada | Clave sin atributos | Permite trazabilidad sin joins | No útil si requiere atributos |
| Role-playing | Múltiples roles de una misma dimensión | Evita duplicación | Ambigüedad si no se renombran campos |
| Junk | Muchos flags simples dispersos | Reduce columnas sueltas | Combinaciones excesivas |
| Mini (Tipo 4) | Atributos volátiles en dimensiones grandes | Trackeo + performance | Complejidad + necesidad de discretizar |

Tablas puente

En el modelado dimensional, cada dimensión debería tener un único valor por fila en la tabla de hechos.

Sin embargo, en algunos casos hay dimensiones multivaluadas:

- 1 Un cliente puede tener varios hobbies**
- 2 Un préstamo puede tener múltiples garantías**
- 3 Un producto puede tener varios atributos opcionales**

Estas relaciones de muchos a muchos no pueden resolverse con una única clave foránea.

¿Qué es una tabla puente?

Una tabla puente (o "bridge table") permite conectar una tabla de hechos con una dimensión que puede tener múltiples valores por fila, resolviendo relaciones de muchos a muchos.



Clave de la Tabla de Hechos

Identifica un registro único en la tabla de hechos.



Clave de la Dimensión Multivaluada

Identifica un registro único en la dimensión que tiene múltiples valores.



Peso o Proporción (Opcional)

Permite realizar cálculos ponderados cuando un hecho se relaciona con múltiples dimensiones.

Ejemplo simple: hobbies por cliente

Supongamos que almacenamos compras por cliente, pero queremos saber cuáles son sus hobbies:

Tabla: fact_compras

| compra_id | cliente_id | producto_id | fecha_key | total |
|-----------|------------|-------------|-----------|-------|
| 101 | C001 | P01 | 20240101 | 120 |
| 102 | C002 | P03 | 20240102 | 80 |

Tabla: dim_cliente

| cliente_id | nombre | ... |
|------------|-----------|-----|
| C001 | Ana López | ... |
| C002 | Juan Ruiz | ... |

Tabla: dim_hobby

| hobby_id | nombre |
|----------|------------|
| H01 | Fotografía |
| H02 | Ciclismo |
| H03 | Lectura |

Tabla puente: cliente-hobby

Tabla: bridge_cliente_hobby

| cliente_id | hobby_id |
|------------|----------|
| C001 | H01 |
| C001 | H03 |
| C002 | H02 |

Así, Ana López tiene Fotografía y Lectura como hobbies, y Juan Ruiz practica Ciclismo.

¿Cómo consultar con una tabla bridge?

Para analizar las compras en relación con los hobbies de los clientes, es necesario realizar una serie de uniones entre las tablas:

1. fact_compras a dim_cliente

Conecta los hechos de compra con los datos del cliente.

2. dim_cliente a bridge_cliente_hobby

Vincula cada cliente con sus respectivos hobbies a través de la tabla puente.

3. bridge_cliente_hobby a dim_hobby

Asocia las claves de los hobbies con sus descripciones completas en la dimensión de hobbies.

Este enfoque permite generar métricas y análisis complejos, como:

- Total de ventas por hobby.
- Distribución de hobbies entre los diferentes grupos de compradores.

Ventajas y Riesgos

Ventajas

Permite representar relaciones multivaluadas de forma eficaz.

Es escalable, facilitando la adición de nuevos valores sin modificar el modelo.

Mejora la capacidad de análisis sin duplicar los hechos.

Riesgos

⊗ Las consultas se vuelven **más complejas**, requiriendo más joins.

Algunos motores de BI podrían no manejar bien los joins con tablas bridge.

Existe el riesgo de **duplicidad de hechos** si los joins no se controlan correctamente.

Cuándo conviene usar tablas bridge

Usá una tabla bridge si:

La relación es de muchos a muchos y no cambia el grano del hecho.

El conjunto de valores posibles es dinámico o muy amplio.

No se puede representar eficientemente con columnas fijas.

Ejemplos de uso:

- Diagnósticos de un paciente
- Opciones contratadas en un plan de servicios
- Habilidades de un postulante