



Diseño Lógico

Diseño Lógico Relacional

■ Temas:

- Introducción
- Diseño a partir de esquema conceptual de las fuentes
- Diseño a partir de esquema lógico de las fuentes
- Refinamientos al esquema
- Práctico



Donde estamos?

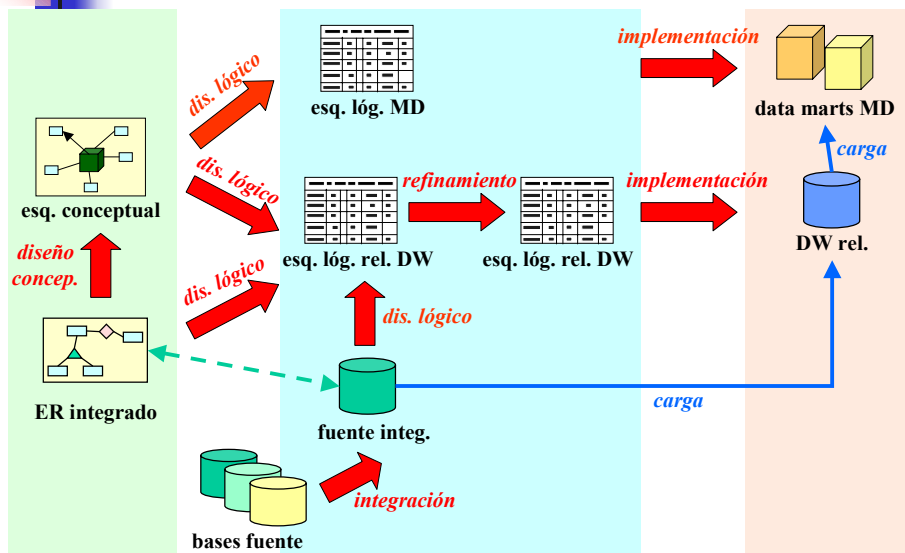
■ Hasta ahora:

- Diseño conceptual
- Diseño lógico
 - Mecanismos básicos para poder construir base de datos OLAP y utilizarla con una herramienta de consultas OLAP.

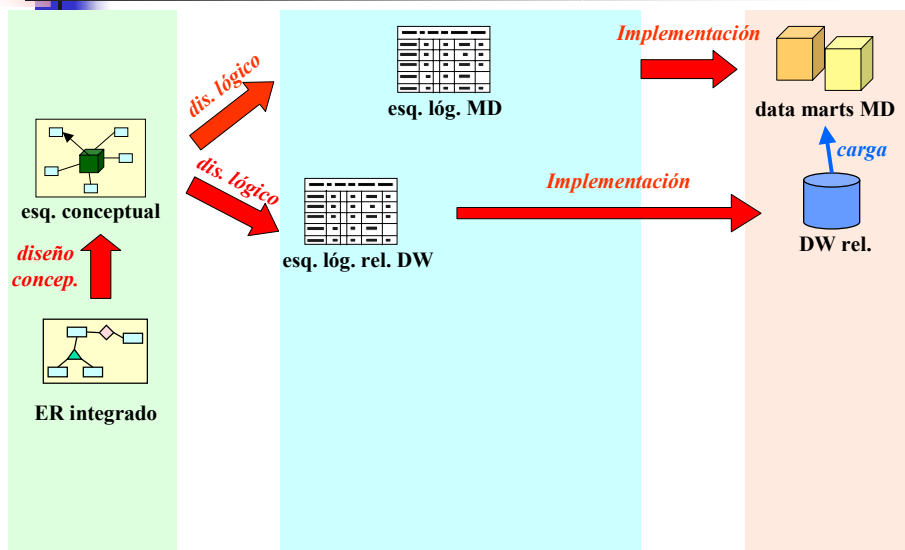
■ A partir de ahora:

- Diseño lógico relacional del DW
- Carga y actualización del DW

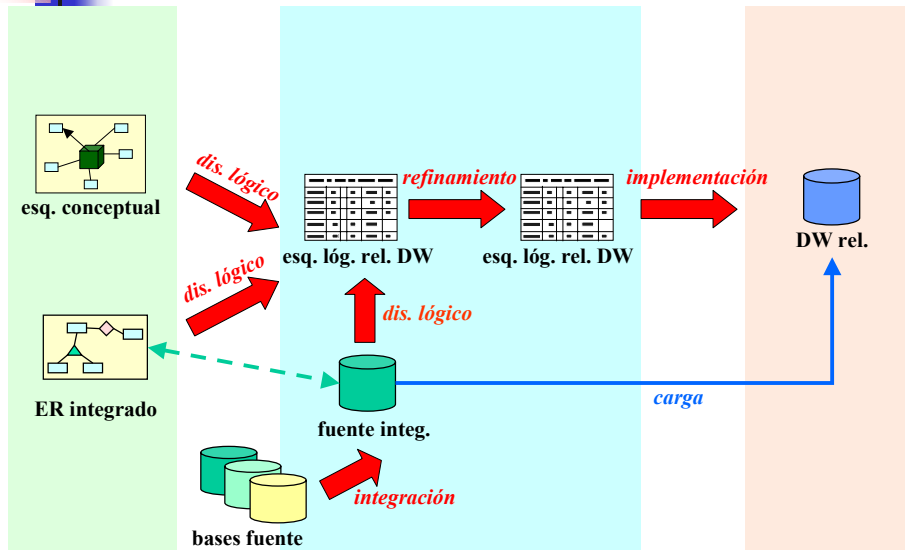
Proceso de Diseño



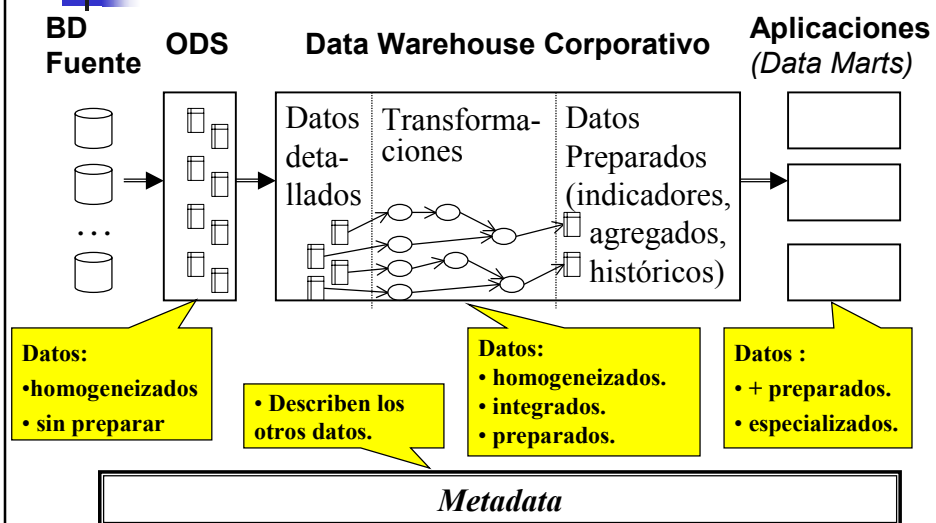
Hasta ahora



A partir de ahora



Estructura del Data Warehouse





Diseño lógico relacional de DW

■ Metodologías

- Pasaje / traducción del esquema conceptual multidimensional al esquema lógico (ya visto)
- Diseño a partir de las fuentes
 - Partiendo del esquema **conceptual** integrado de las fuentes (MER)
 - Partiendo del esquema **lógico** integrado de las fuentes (Modelo Relacional)

■ Refinamientos

- Estudio de problemas frecuentes y sus soluciones



Diseño lógico relacional de DW

■ Metodologías

- Pasaje / traducción del esquema conceptual multidimensional al esquema lógico (ya visto)
- Diseño a partir de las fuentes
 - Partiendo del esquema **conceptual** integrado de las fuentes (MER)
 - Partiendo del esquema **lógico** integrado de las fuentes (Modelo Relacional)

■ Refinamientos

- Estudio de problemas frecuentes y sus soluciones



Repaso

- Un esquema MD puede ser representado en el modelo relacional adoptando diferentes esq.
- Representación directa: *Esquema estrella (Star schema)*.
- Afinamiento orientado a facilitar mantenimiento de dimensiones: *Esquema en copos de nieve (Snowflake schema)*.
- Los llamaremos *esquemas dimensionales relacionales*.



De EER fuentes a Relacional DW

- **Kortink & Moody, 2000.**
Moody, D. Kortink, M.: "From Enterprise Models to Dimensional Models: A Methodology for Data Warehouse and Data Mart Design". DMDW'00, Sweden, 2000.
- **Problemas enfoque Star Schema derivado de los requerimientos**
 - No todos los datos se adecuan naturalmente a esta estructura
 - Requerimientos de usuario: base inestable para diseño
 - Si el diseñador no entiende las relaciones subyacentes entre los datos, puede llevar a diseños incorrectos.



De EER fuentes a Relacional DW

■ Objetivos

- Producir estructuras de BD fáciles de entender para el usuario.
- Maximizar la eficiencia de las consultas.

■ Los alcanza

- Minimizando el número de tablas y las relaciones entre ellas.

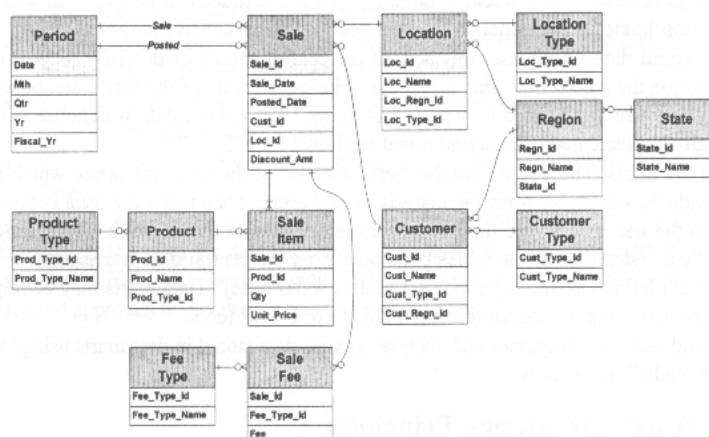
■ Pasos

- Clasificar entidades
- Identificar jerarquías
- Producir modelo dimensional



Ejemplo

■ Notación de [Hitchman, 1995] para ER.





Clasificar entidades

■ 3 categorías:

■ *Transaction entity*

- Describe un evento que sucede en un punto en el tiempo.
- Contiene medidas o cantidades.

■ *Component entity*

- Directamente relacionado a una *transaction entity* via una relación 1:N.
- Contestan “quién”, “qué”, “cuándo”, “dónde”, “cómo” y “porqué” del evento.



Clasificar entidades (1)

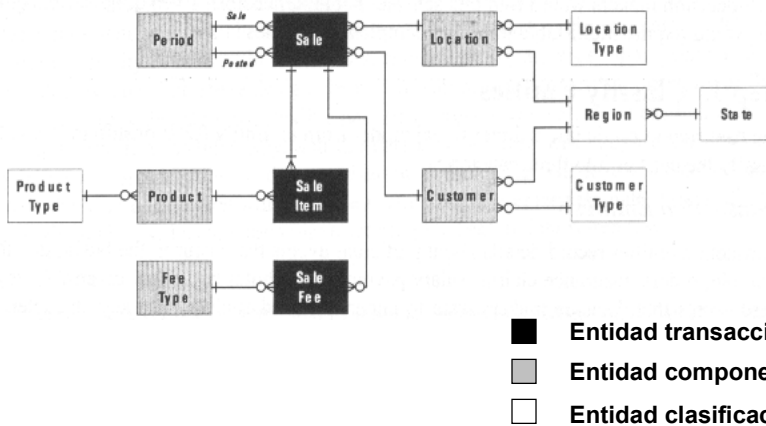
■ 3 categorías (cont.):

■ *Classification entity*

- Son entidades que están relacionadas a las *component* por una cadena de relaciones 1:N.
- Representan jerarquías embebidas en el modelo de datos, que podrían ser colapsadas en las *component entities* para formar las dimensiones en un *star schema*.

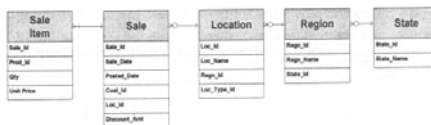


Ejemplo/ Clasificación de entidades

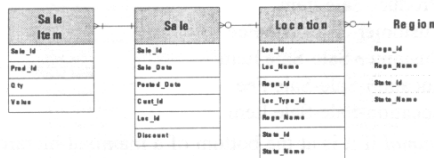


Identificar jerarquías

- Una jerarquía en un MER puede ser identificada como secuencia de entidades unidas a través de relaciones 1:N, alineadas en la misma dirección.
- Ejemplo



- “Colapsar” una jerarquía (desnormalizar)



colapso



Producir Modelo Dimensional

- Hay una gama de opciones para el modelo dimensional
 - Flat Schema
 - Terraced Schema
 - Star Schema
 - Constellation Schema
 - Snowflake Schema
 - Star Cluster Schema



Flat Schema

- Se colapsan todas las entidades hasta las entidades minimales (entidades “hoja” de la jerarquia).
- Queda una tabla para cada entidad minimal del modelo original.
- Redundancia en dependencias parciales y transitivas.
- No involucra ninguna agregacion.
- Pueden haber errores al realizar agregaciones cuando existen relaciones entre transaction entities (Ej.: Sale y Sale Item)
- Minimiza nro. de tablas, incrementa mucho la complejidad de cada tabla.

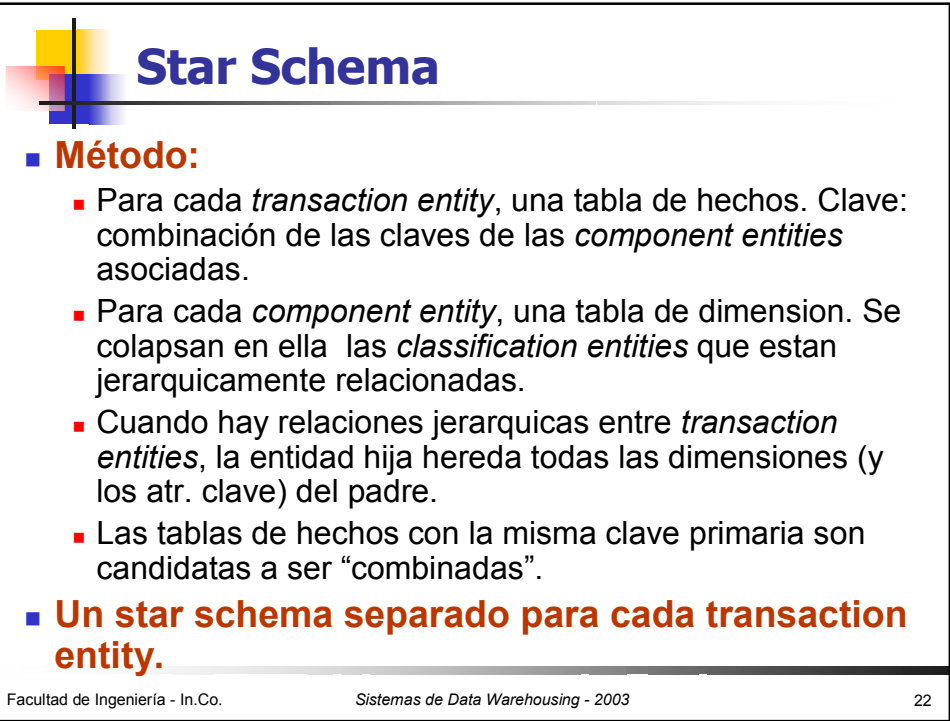
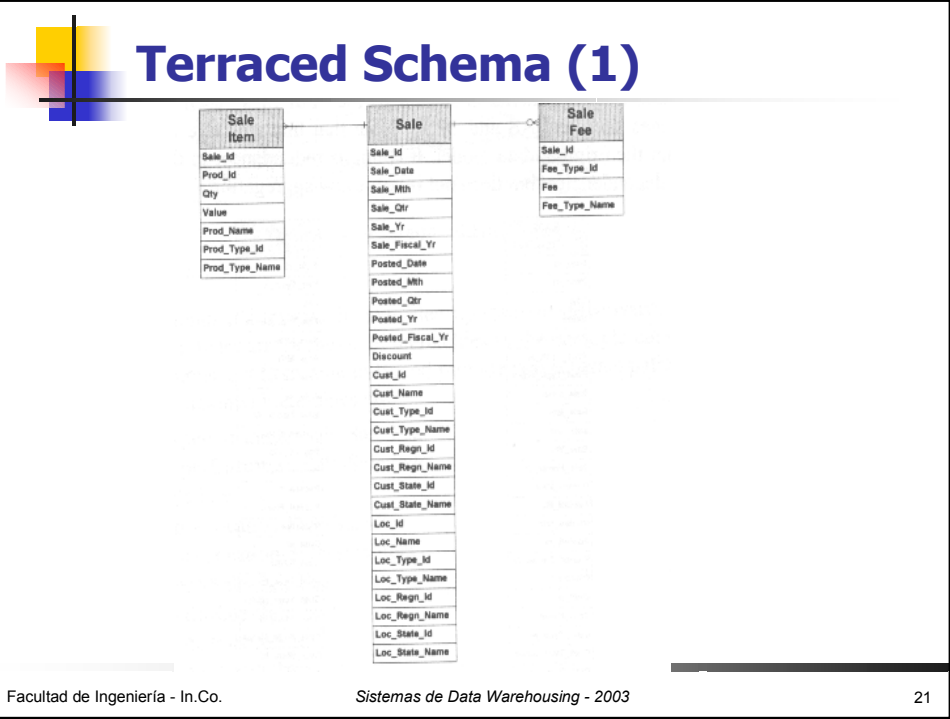


Flat Schema (1)



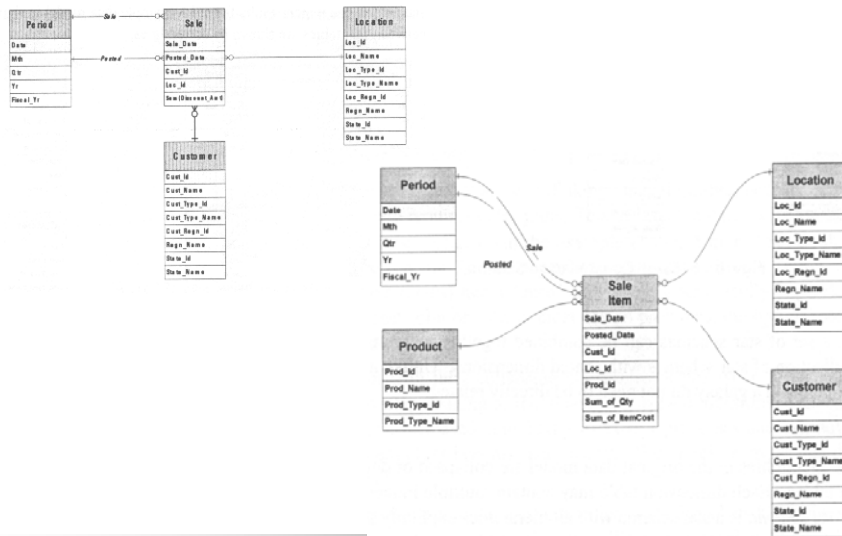
Terraced Schema

- Se forma colapsando entidades en las jerarquías maximales, pero deteniéndose cuando se alcanza una *transaction entity*.
- Queda una tabla para cada *transaction entity*.





Star Schema (1)

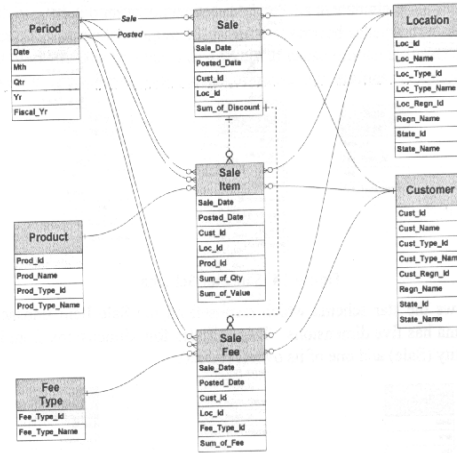


Constellation Schema y Galaxy

- **Un esquema constelación consiste de:**
 - Un conjunto de esquemas estrella cuyas tablas de hecho están relacionadas jerárquicamente (1-N).
- **Galaxy**
 - Es un conjunto de esquemas estrella que comparten las dimensiones.



Constelación y Galaxia

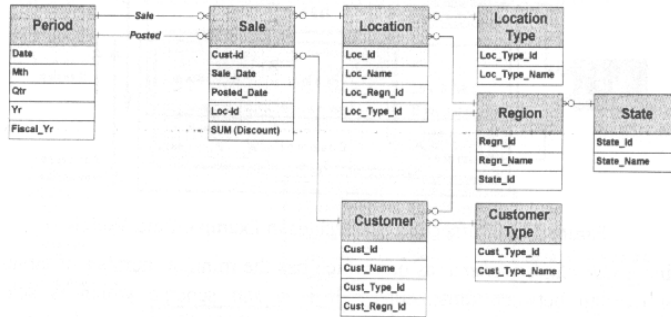


Snowflake Schema

- Es un Star Schema con las jerarquías mostradas explícitamente.
- Se puede construir a partir del Star Schema (normalizando las jerarquías en cada dimensión) o directamente de la forma:
 - Una tabla de hechos para cada *transaction entity*.
 - Cada *component entity* se transforma en una tabla de dimensión.
 - Las *classification entity* quedan iguales.
 - Cuando hay relaciones jerárquicas entre *transaction entities*, la entidad hija hereda todas las dimensiones (y los atr. clave) del padre.
 - Las tablas de hechos con la misma clave primaria son candidatas a ser “combinadas”.

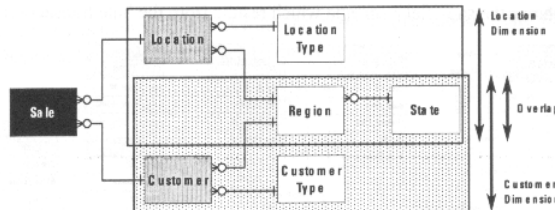


Snowflake Schema (1)



Star Cluster Schema

- Ni un Star puro ni un Snowflake puro.
- Esquema resultado de balance entre los 2 extremos.
- Problema en colapsar totalmente las jerarquías: cuando las jerarquías en diferentes dimensiones se “solapan”. Hay redundancia y puede haber inconsistencias.
 - Se identifica cuando hay “fork entities”. Son entidades clasificación que tienen multiples relaciones 1:N



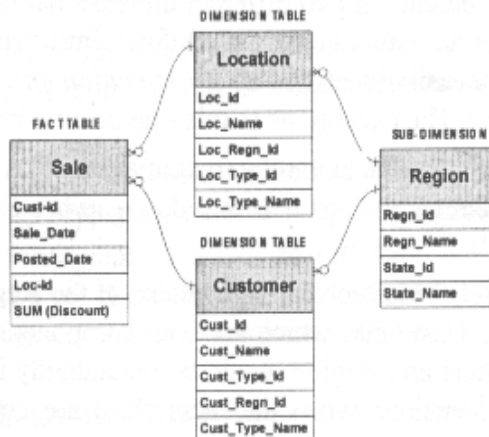


Star Cluster Schema (1)

- Es un esquema que disminuye el número de tablas y a la vez evita “solapamiento” entre dimensiones.
- Es “snowflaked” selectivamente. Separa fragmentos de jerarquías que son compartidos entre diferentes dimensiones.



Star Cluster Schema (3)





Star Cluster Schema (2)

■ Método:

- Una tabla de hechos para cada *transaction entity*.
- Las *classification entities* se colapsan con sus jerarquias hasta que alcanzan una "fork entity" o una *component entity*.
 - Si se alcanzo una "fork entity", se forma una tabla de sub-dimension. Consistira de la "fork entity" mas todos sus ancestros.
 - Si se alcanzo una *component entity*, se forma una tabla de dimension.
- Cuando hay relaciones jerarquicas entre *transaction entities*, la entidad hija hereda todas las dimensiones (y los atr. clave) del padre.
- Las tablas de hechos con la misma clave primaria son candidatas a ser "combinadas".



Refinamiento

- **Combinar tablas de hechos**
 - Combinar tablas de hechos que tienen las mismas dimensiones. Facilita comparacion entre medidas relacionadas.
- **Relaciones N:N**
 - Representan una ruptura en la cadena jerarquica y no pueden ser colapsadas.
 - Posibles decisiones a tomar:
 - Ignorar la relacion
 - Convertirla en 1:N
 - Incluirla como una relacion N:N
- **Manejo de sub-tipos**
 - Se pueden transformar a nivel del MER en una rel. 1:N



Resumen

■ Pasos


- Desarrollar modelo de datos de la empresa.
 - Modelo integrado ER
- Clasificar entidades.
- Diseñar esquemas dimensionales lógicos.
 - Desarrollar star-cluster schemas para cada *transaction entity* del modelo.
 - Estos se pueden combinar formando *constellations* o *galaxies*.



Resumen

■ Ventajas

- Encare estructural para desarrollar modelos dimensionales
- Los esquemas dimensionales reflejan las relaciones subyacentes en los datos.
- Sugerencia: para flexibilizar soporte a requerimientos de analisis
 - DW central guarda datos al nivel de transacciones individuales. (No recomiendan agregación de datos a este nivel)




Resumen – Tipos de esquemas

- Flat Schema
- Terraced Schema
- Star Schema
- Star Cluster Schema
- Snowflake Schema

Incrementa redundancia

Incrementa complejidad

Facultad de Ingeniería - In.Co. *Sistemas de Data Warehousing - 2003* 35



Resumen

- **Más casos de afinamiento:**
 - Constelación
 - Galaxia
 - Star Cluster

Facultad de Ingeniería - In.Co. *Sistemas de Data Warehousing - 2003* 36