

# Diseño Lógico de DW

Alejandro Vaisman  
Instituto Tecnológico de Buenos Aires, Argentina  
avaisman@itba.edu.ar

# Modelos Lógicos

- El modelo lógico de un DW representa los hechos y las dimensiones utilizando el modelo de datos de la BD sobre la cual se implementará el DW
- Tres alternativas básicas:
  - Relational OLAP (ROLAP): almacena los datos en bases de datos, se consulta en SQL
    - Utilizado para alta dimensionalidad y para almacenar el DW organizacional
  - Multidimensional OLAP (MOLAP): almacena los datos en estructuras propietarias MD, como arreglos multidimensionales
    - Usado cuando el número de dimensiones es  $< 10$
    - Se consulta en MDX
    - Se implementa en los OLAP servers, y se materializan todas las sumalizaciones posibles
  - Hybrid OLAP (HOLAP): Combina ambas tecnologías

# Diseño Relacional de DWs

- **La única estructura del modelo relacional es la tabla**
- Los hechos y las dimensiones se almacenan en tablas
  - Tablas de hechos y tablas de dimensión
  - Se pierde la semántica del modelo conceptual

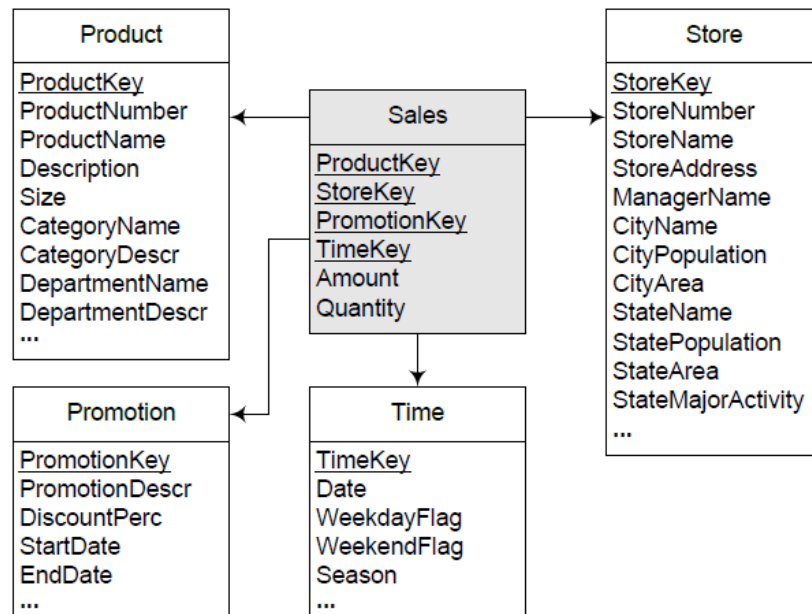
## CUATRO ALTERNATIVAS

- Esquema Estrella (Star schema)
  - Integridad referencial entre tablas de hechos y dimension
  - Tablas de dimension desnormalizadas, tablas de hechos normalizadas
- Esquema Snowflake (Copo de nieve)
  - Tablas de dimension normalizadas, tablas de hechos normalizadas
- Esquema Starflake
  - Una tabla de hechos normalizada, dimensiones normalizadas y desnormalizadas
- Esquema Constellation
  - Igual que Starflake, pero con más de una fact table

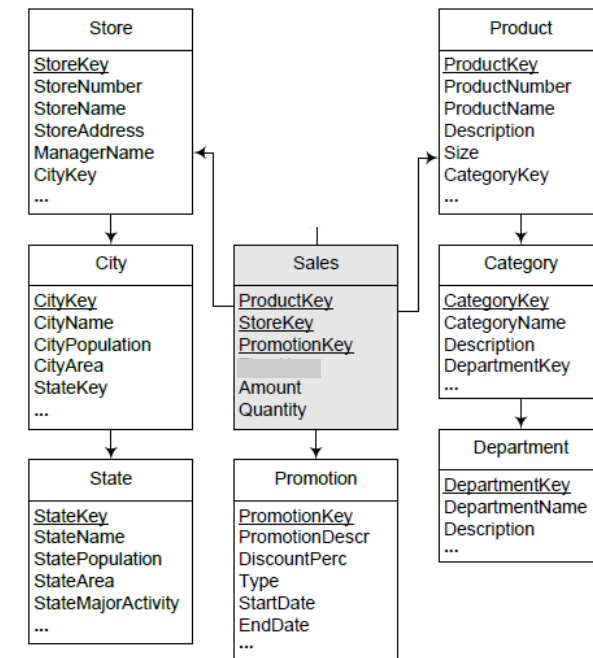
# Star vs Snowflake

- Fact table compuesta por las FKs de las tablas de dimension involucradas
- Notar la representación de la dimension tiempo
- **ACTIVIDAD: explicar los motivos por los cuales la dimension tiempo se representa en la forma descripta**

## Star schema



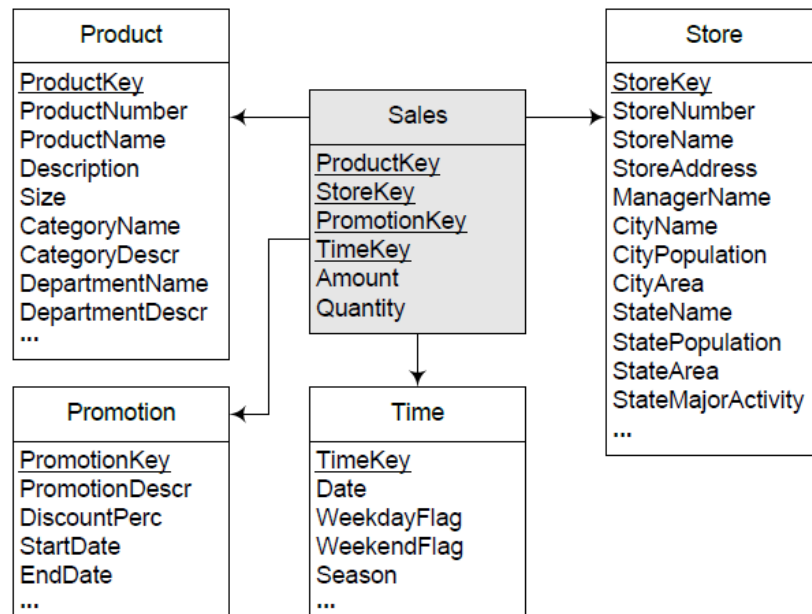
## Snowflake Schema



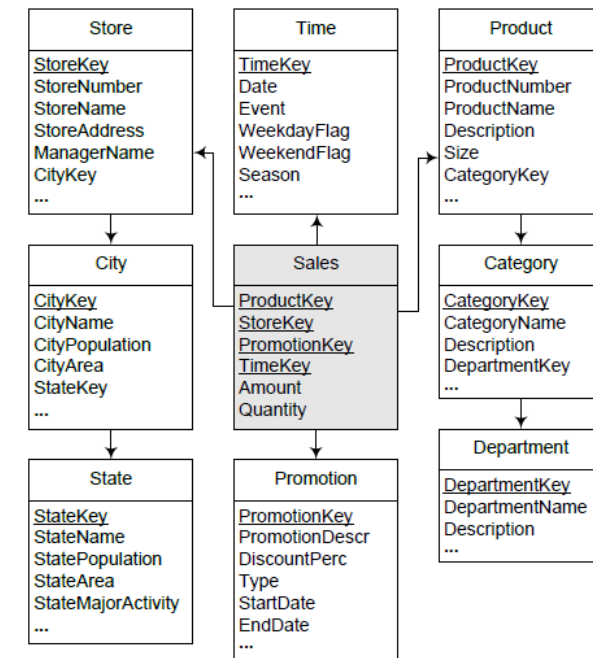
# Star vs Snowflake

- Fact table compuesta por las FKs de las tablas de dimension involucradas
- Notar la representación de la dimension tiempo
- **ACTIVIDAD: explicar los motivos por los cuales la dimension tiempo se representa en la forma descripta**

## Star schema

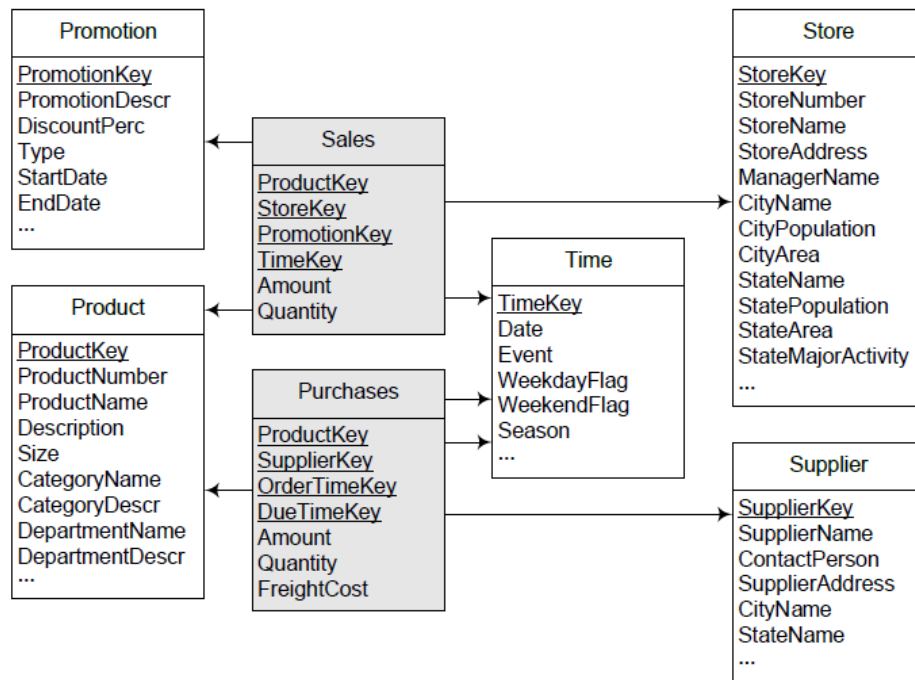


## Starflake Schema



# Constellation schema

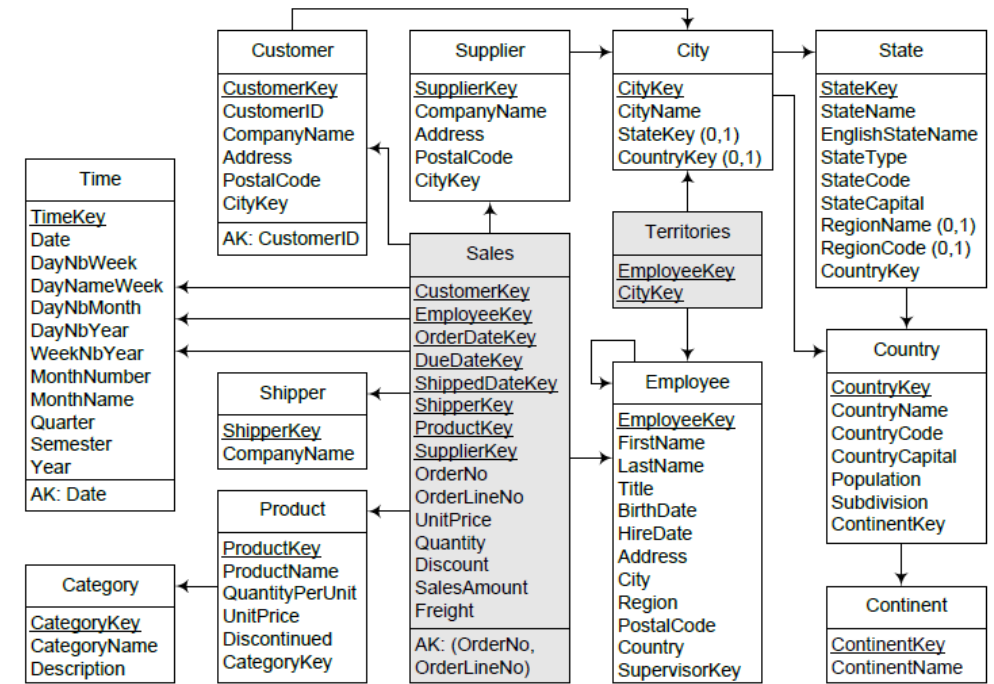
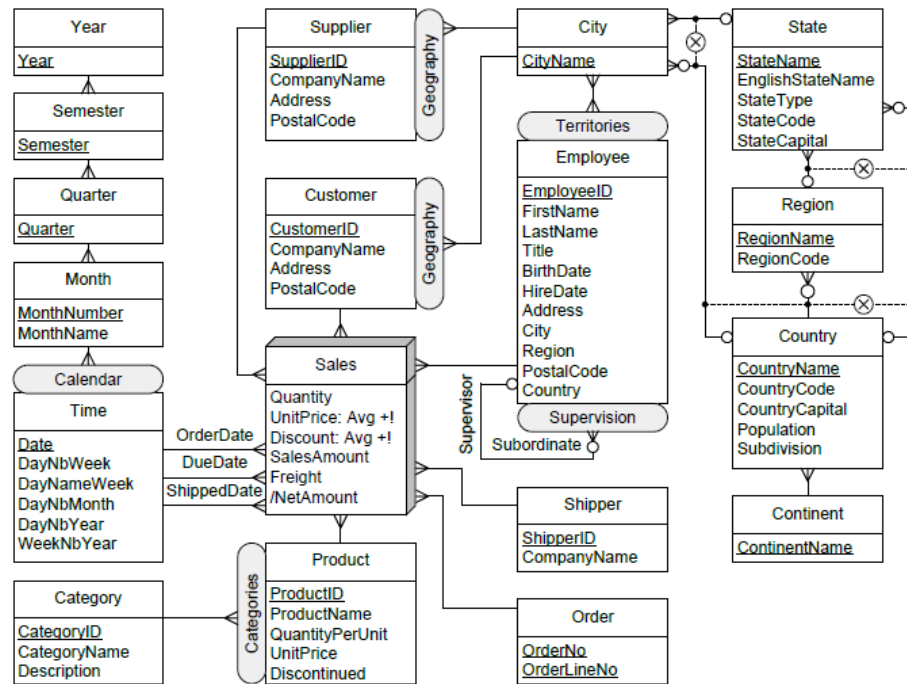
- Combinación de los dos anteriores



**ACTIVIDAD:** Analizar y discutir ventajas y desventajas de cada representación. En función de qué criterios se elije entre estas alternativas?

# Northwind conceptual vs lógico

- Notar las FKs de Sales y cómo se representan las role-playing dimensions
  - Jerarquías no estrictas: se transforman en una tabla de hechos sin métrica (eventualmente, con factor de distribución)
  - AK: Clave alternativa
  - (0,1) para representar las jerarquías ragged
  - Jerarquía recursiva: FK sobre la misma tabla
- **Actividad: analizar intuitivamente cómo se implementa el modelo conceptual en NW. Qué tipo de esquema es NW?**



# La Dimensión Tiempo

- Presente en prácticamente todos los DW
- Un DW es una base de datos histórica
- OLTP: una fecha, tipo DATE, y la información se obtiene de funciones especialmente definidas para ese tipo de dato
- En OLAP, la información se almacena en forma explícita, para mejorar la eficiencia. Por ejemplo: “Ventas totales en fines de semana”

```
SELECT SUM(SalesAmount)
FROM Time T JOIN Sales S ON T.timeKey=S.TimeKey
WHERE T.WeekendFlag=1
```

**Actividad: Escribir la consulta en SQL asumiendo que la WeekendFlag no está presente en el diseño.**

- Normalmente existen funciones predefinidas que permiten generar y poblar la dimension Tiempo automáticamente.



# Diseño Lógico de Jerarquías

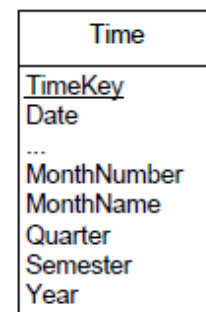
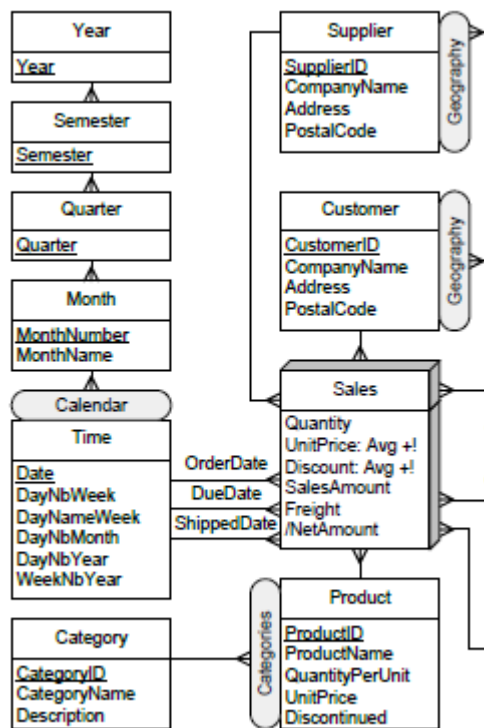
- Estudia cómo representar las jerarquías conceptuales en el modelo relacional

# Diseño Relacional de Jerarquías Estrictas

- Estudia cómo representar las jerarquías conceptuales en el modelo relacional

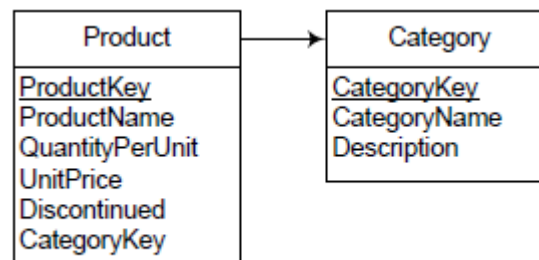
# Jerarquías Balanceadas

- El diseño puede ser Estrella (star) o Copo de nieve (snowflake)
- Ejemplo: Porción de NW DW



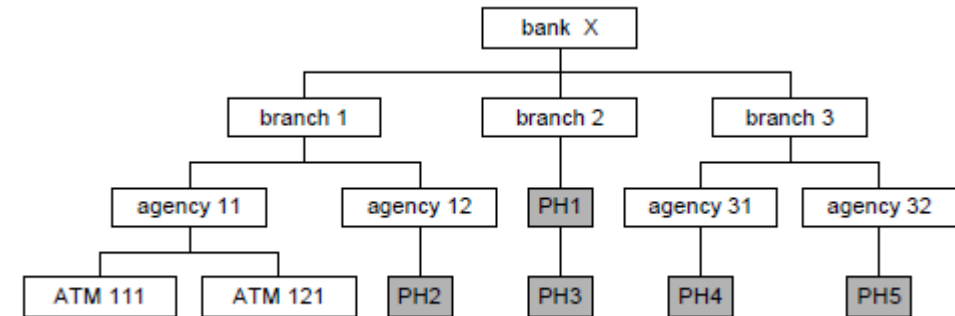
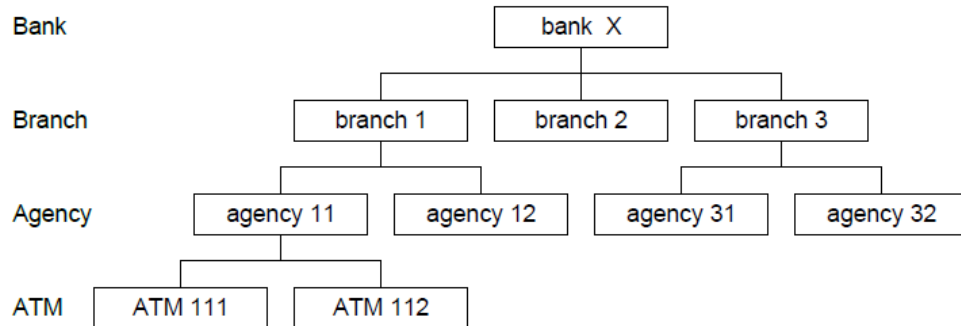
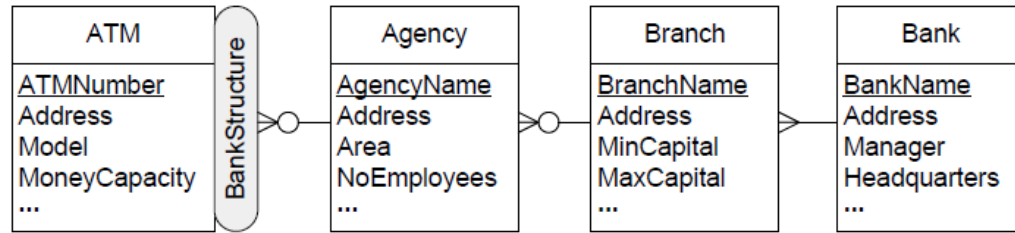
Estrella

Copo de nieve  
(snowflake)



# Jerarquías Desbalanceadas

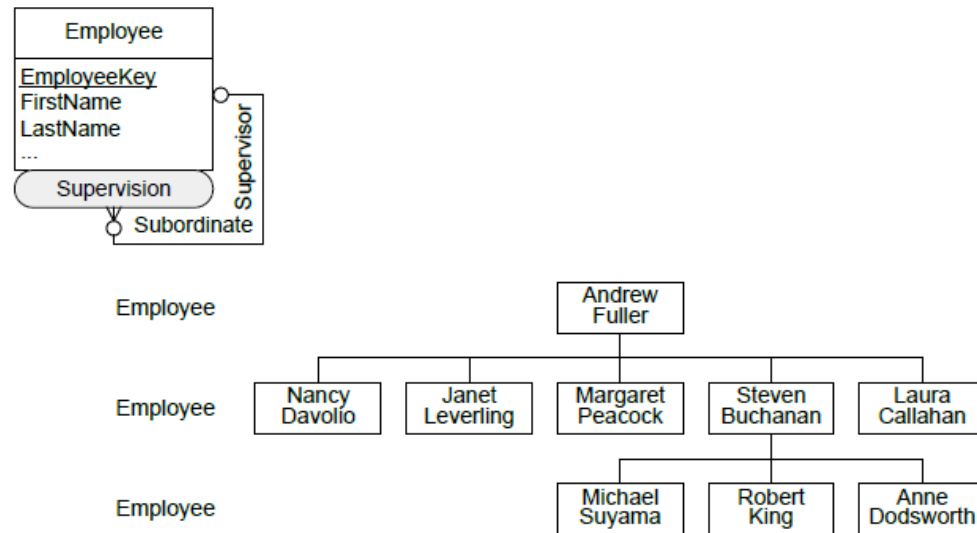
- Se transforman en estrictas mediante el uso de dummies, a nivel de datos



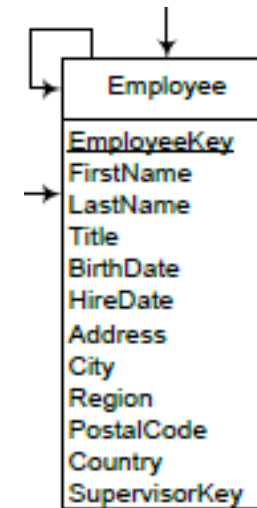
# Jerarquías Rekursivas

- Se agrega como clave foránea, a la clave primaria, referenciando a la propia tabla  
SUPERVISORKEY ... FOREIGN KEY REFERENCES Employee.EmployeeKey
- Generalmente requiere computar una consulta recursiva
- Se suele implementar como vista materializada

## Conceptual

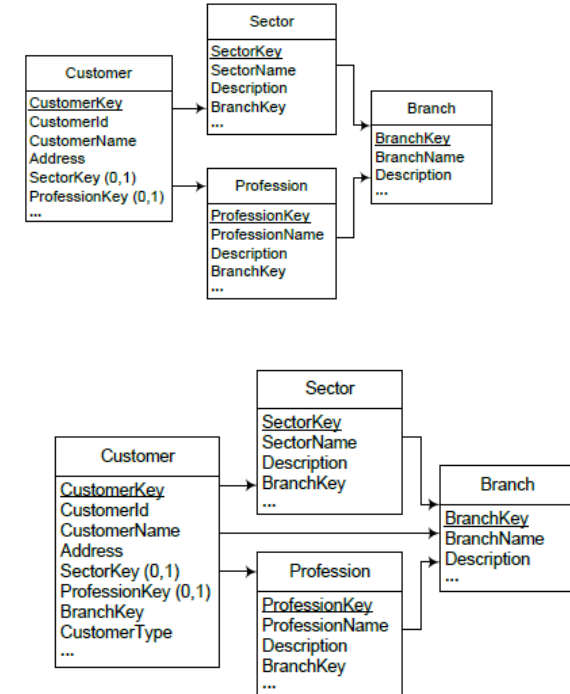
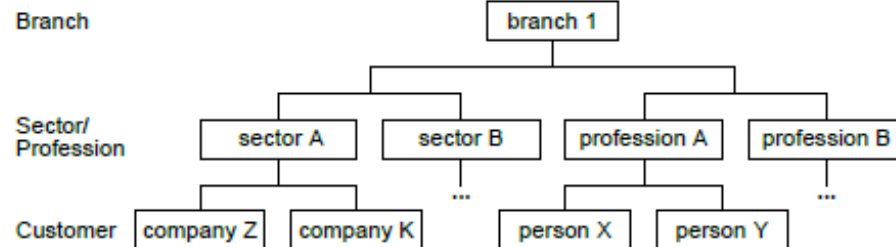
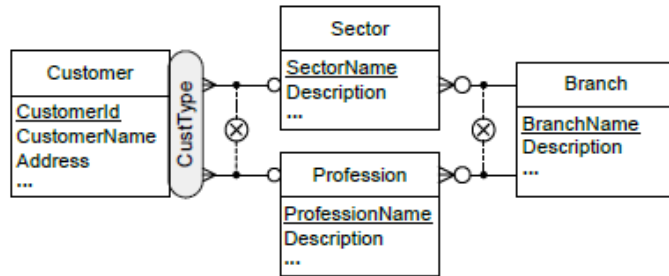


## Relacional



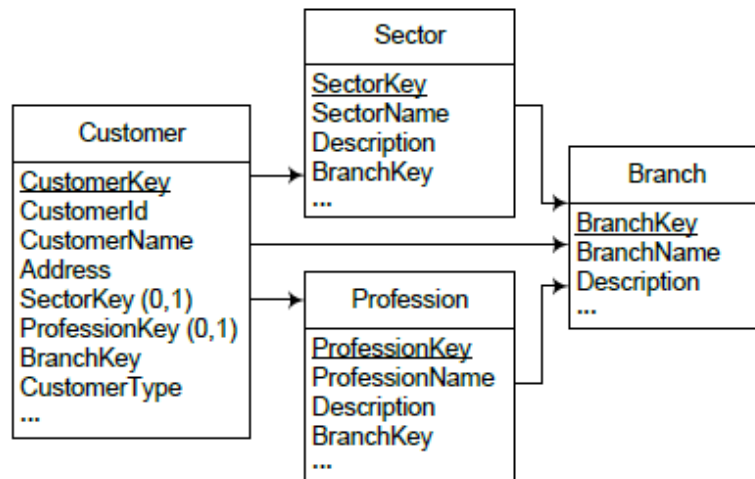
# Jerarquías Generalizadas

- Distintas formas de mapping:
  - Una tabla por cada nivel de la jerarquía => Snowflake
  - Una representación plana, con valores NULL
  - Una tabla para los niveles comunes, y otra para los específicos
  - **ACTIVIDAD: Mapear el modelo de la parte inferior izquierda, al modelo relacional según las tres alternativas, y dar un ejemplo de una instancia en cada caso.**



# Jerarquías Generalizadas

- Implementación de la generalización: SQL
- Notar que el atributo CustomerType es una restricción que debe ser verificada



```
ALTER TABLE Customer ADD CONSTRAINT CustomerTypeCK
CHECK ( CustomerType IN ('Person', 'Company') )
ALTER TABLE Customer ADD CONSTRAINT CustomerPersonFK
CHECK ( (CustomerType != 'Person') OR
( ProfessionKey IS NOT NULL AND SectorKey IS NULL ) )
ALTER TABLE Customer ADD CONSTRAINT CustomerCompanyFK
CHECK ( (CustomerType != 'Company') OR
( ProfessionKey IS NULL AND SectorKey IS NOT NULL ) )
```

# Diseño Relacional de Jerarquías No-Estrictas

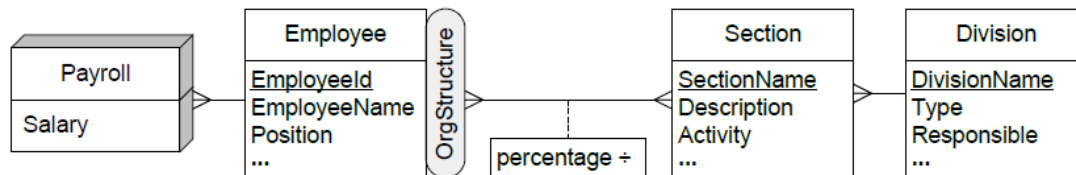
- Incluyen al menos una relación M:N
- Problemas de sumarizabilidad!!!



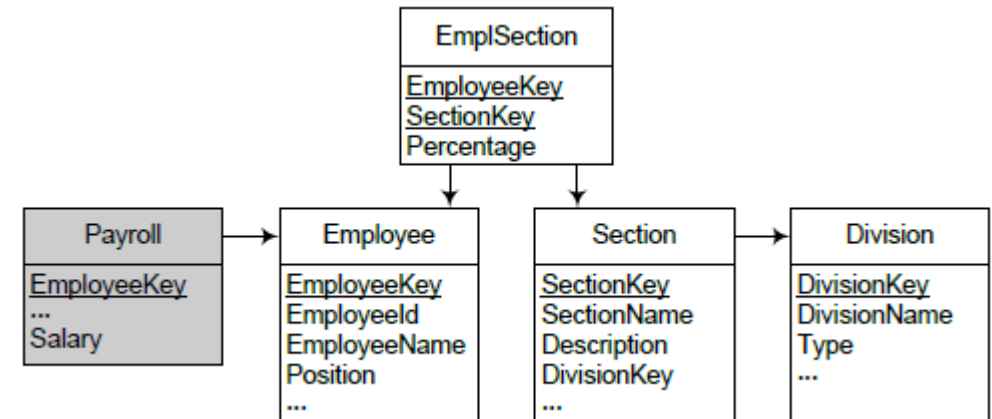
# Jerarquías No-Estrictas

- Esquema: Al menos una relación M:N
- Mapping: Crea tablas, llamadas **BRIDGE TABLES**
- En el ejemplo, la tabla EmplSection. “Percentage” es el factor de distribución

## Conceptual



## Relacional

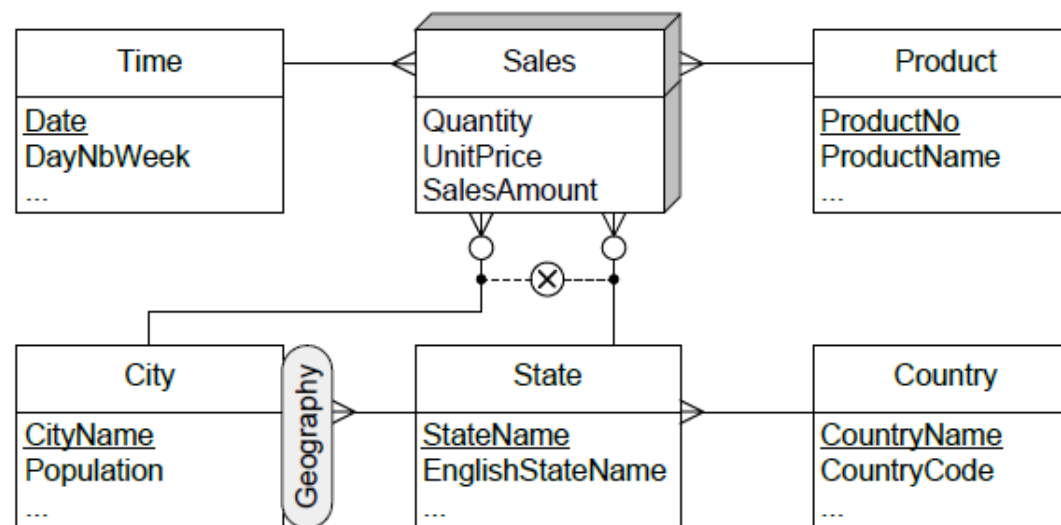


# Casos especiales

- Facts con multiples granularidades
- Dimensiones M:N
- Relaciones entre facts

# Hechos con más de una granularidad

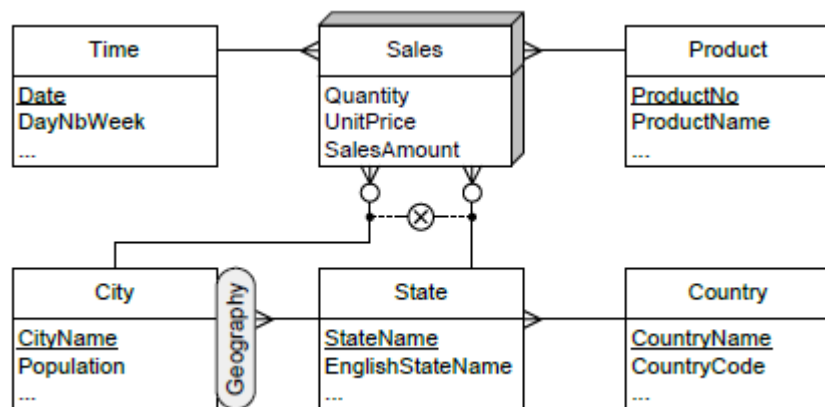
- Situación muy común
- Ej: una empresa recibe datos de venta por ciudad, en ciudades grandes, o por provincial, para aquellas ciudades pequeñas
- Otro ejemplo: para algunos pacientes en un DW médico, se conoce la condición específica, para otros, una familia de condiciones (e.j, enfermedades respiratorias, y gripe, respectivamente)



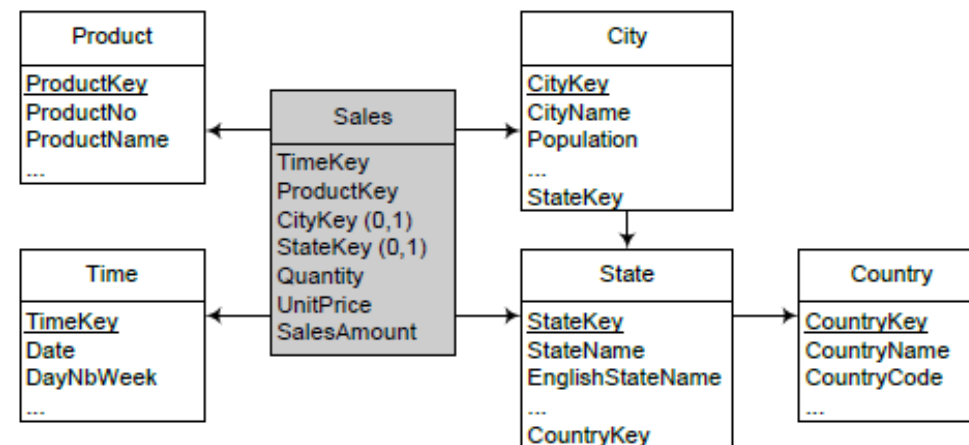
# Hechos con más de una Granularidad

- Primer solución: Usar varias FKs, como en el caso de las jerarquías ragged

## Conceptual



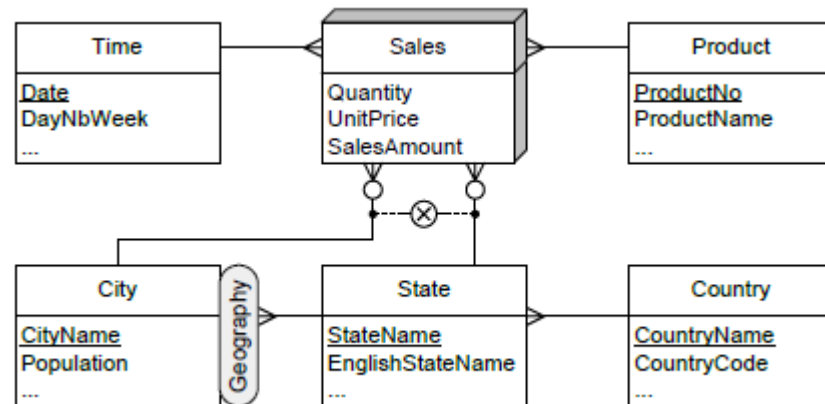
## Relacional



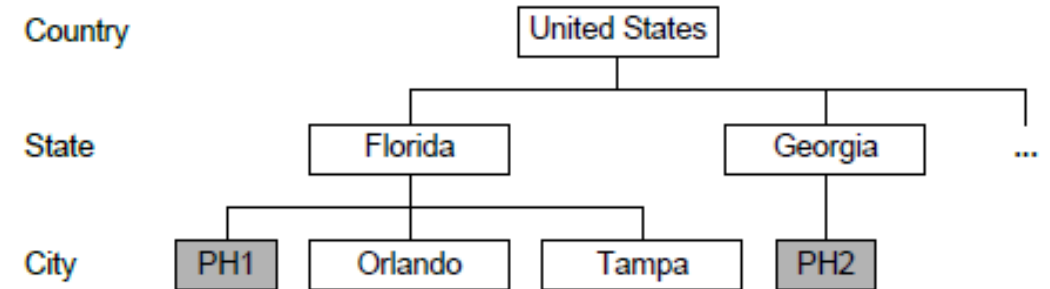
# Hechos con más de una Granularidad

- Segunda solución: Eliminar las distintas granularidades utilizando placeholders a nivel de instancia
- **Actividad:** Generar las tablas relacionales para ambos casos, y discutir las soluciones.

## Conceptual

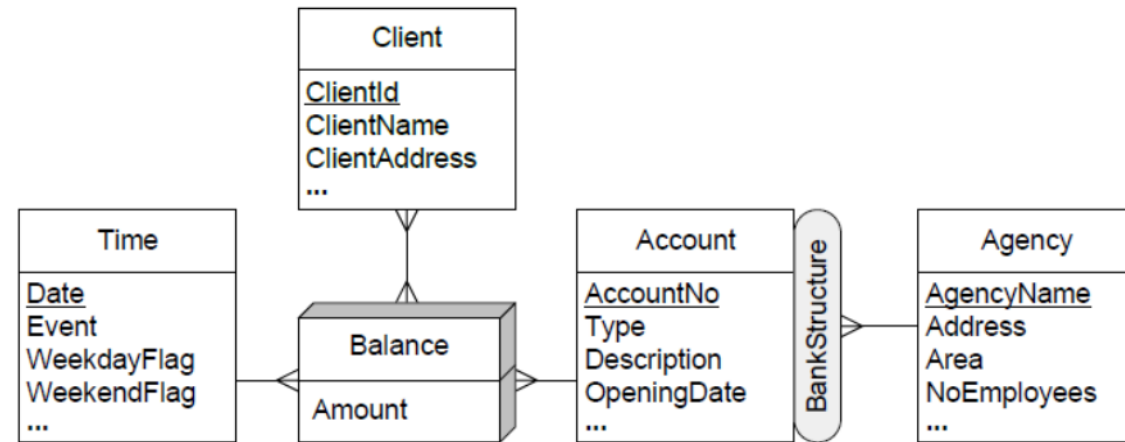


## Instancia Relacional



# Dimensiones M:N

- Relación M:N entre fact y nivel hoja de una dimension (distinto de jerarquías no estrictas)
- Problema de doble conteo: el saldo de una cuenta se suma tantas veces como cotitulares tenga la cuenta (ver ejemplo parte inferior)

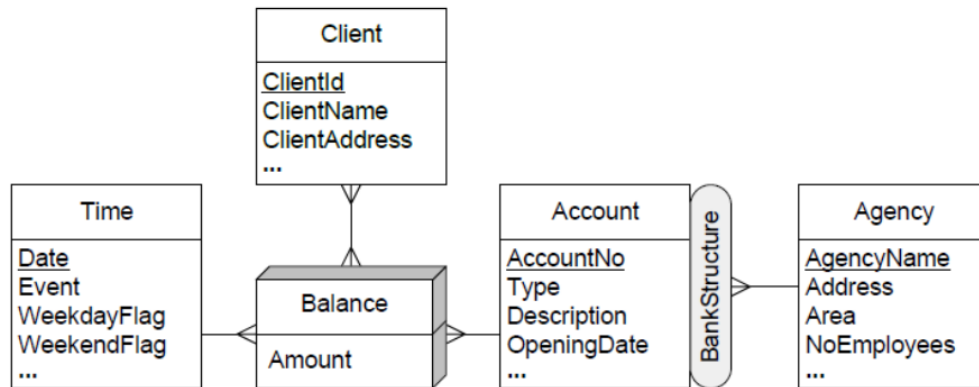


Time	Account	Client	Balance
T1	A1	C1	100
T1	A1	C2	100
T1	A1	C3	100
T1	A2	C1	500
T1	A2	C2	500

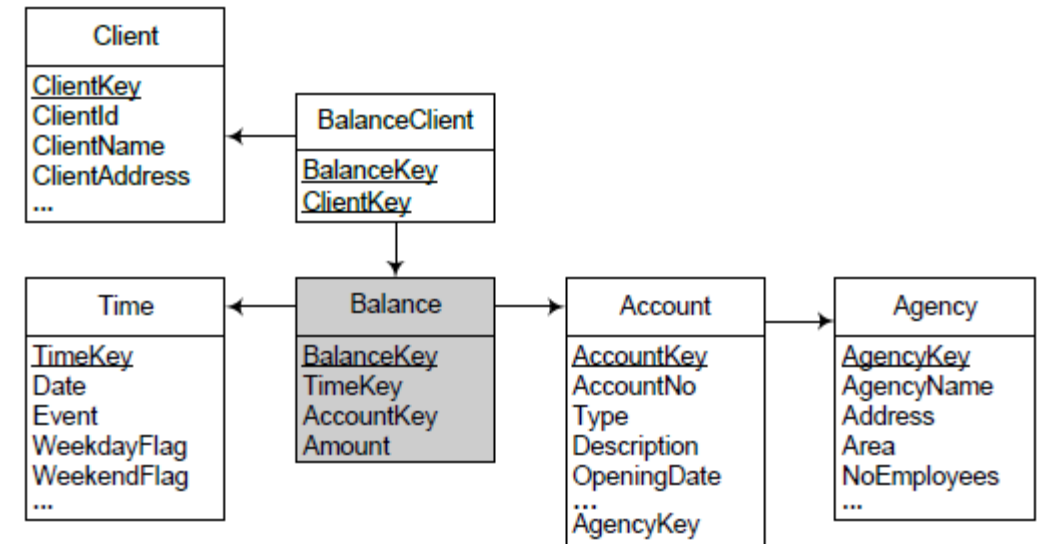
# Jerarquías No-Estrictas

- **Solución 1: crear una jerarquía no-estricta**
- Solución 2: crear dos facts

## Conceptual



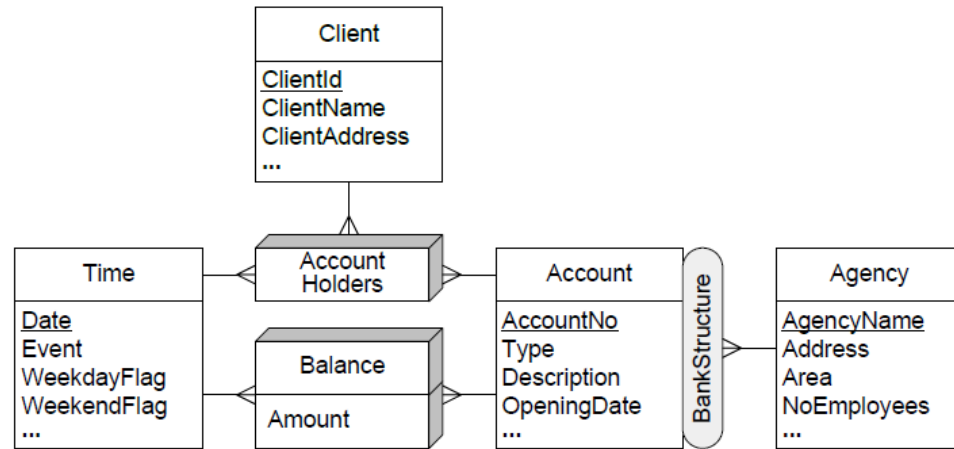
## Relacional



# Jerarquías No-Estrictas

- Solución 1: crear una jerarquía no-estricta
- **Solución 2: crear dos facts**

## Conceptual



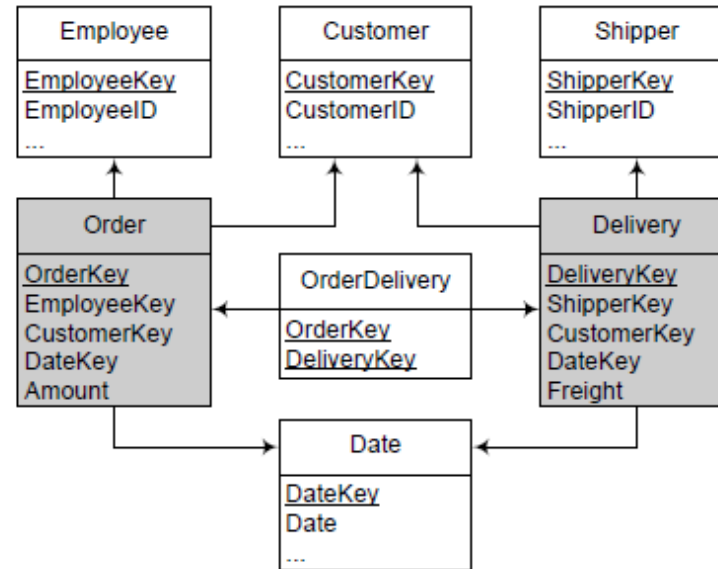
## Relacional ?

- **Actividad:** Dibujar el modelo relacional para esta solución
- Discutir los pros y contras de ambas soluciones.



# Relaciones entre Facts

- La traducción depende de las cardinalidades
- En el caso m:n se agrega una bridge table
- Vinculan cubos a través de joins entre fact tables



Order	Date	Customer	Employee	Amount
O1	1/5/2014	C1	E1	250.00
O2	1/5/2014	C1	E1	147.00
O3	3/5/2014	C1	E2	133.00
O4	5/5/2014	C2	E2	190.00

Order	Delivery
O1	D1
O3	D1
O2	D2
O2	D3
O4	D4

Delivery	Date	Customer	Shipper	Freight
D1	6/5/2014	C1	S3	25.00
D2	7/5/2014	C1	S2	30.00
D3	8/5/2014	C1	S2	27.00
D4	9/5/2014	C2	S1	50.00

# Actividades

- **Actividad 1.** Mapear los modelos coneptuales obtenidos en los ejercicios de Diseño Conceptual, a los modelos Estrella y Snowflake.
- **Actividad 2.** Mapear el siguiente diagrama MultiDim al modelo Estrella / o Snowflake

