

# Introducción

## Evolución Histórica:

- ▶ Años 50's: Preparación de resúmenes en departamentos de informática.
- ▶ En los 60's nacen los sistemas gestores de bases de datos.
- ▶ Después aparecen los motores relacionales.
- ▶ A finales de los 80's aparece el data warehouse.

# Conceptos Básicos

- ▶ **Bases de Datos.**

Es un conjunto de datos persistentes que es utilizados por un sistema de alguna empresa.

- ▶ **Sistema de Bases de Datos.**

Sistema computarizado para llevar los registros de alguna compañía en particular.

- ▶ **Sistema Manejador de Bases de Datos.**

Parte de un sistema de bases de datos encargado de gestionar todas transacciones que se llevan a cabo en la base de datos.

# Conceptos Básicos

- ▶ Los sistemas de bases de datos se diseñan para la gestión de grandes cantidades de información.
- ▶ Esto implica:
  - ▶ La definición de estructuras para almacenar la información.
  - ▶ La provisión de mecanismos para la manipulación de información.
- ▶ Además, los sistemas de bases de datos deben proporcionar fiabilidad.

# Conceptos Básicos

- ▶ Un sistema de base de datos comprende cuatro componentes principales:
  - ▶ Datos: Integrados y Compartidos.
  - ▶ Hardware: Necesario para el Sistema.
  - ▶ Software: DBMS, Utilerías, herramientas.
  - ▶ Usuarios: Programadores de Aplicaciones, Usuarios Finales y el Administrador de la Base de Datos.

# Ventajas de un DBS

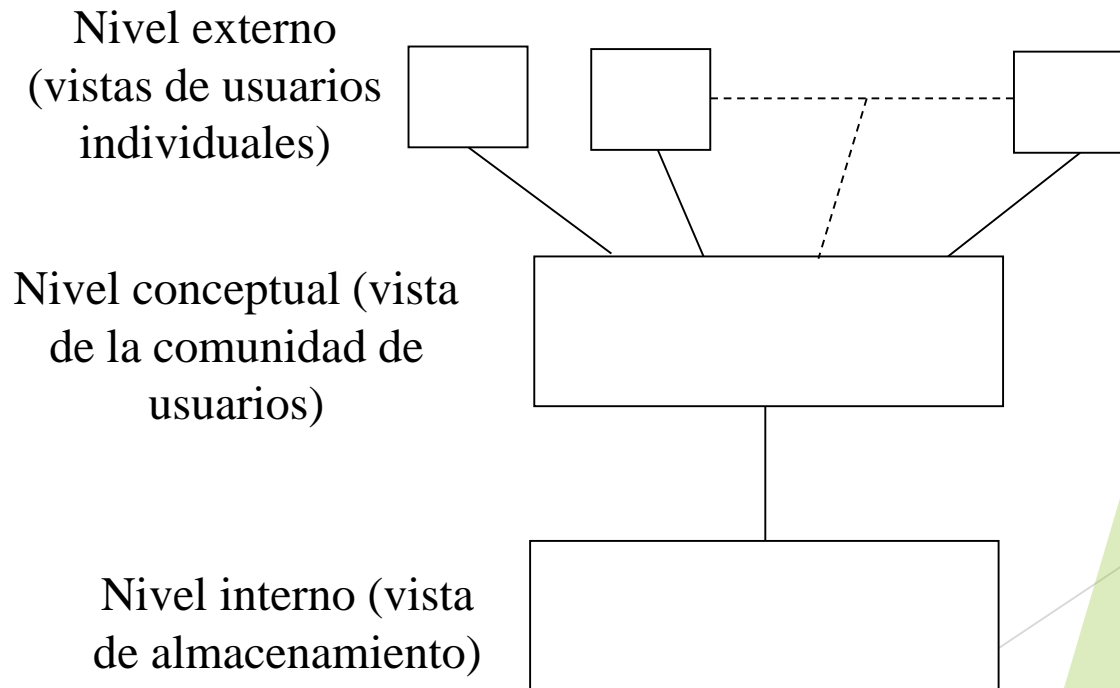
- ▶ Compactación.
- ▶ Velocidad.
- ▶ Menos trabajo laborioso.
- ▶ Actualidad.
- ▶ Ofrecimiento de un control centralizado.
- ▶ Los datos pueden compartirse.
- ▶ Se puede reducir la redundancia y la inconsistencia.
- ▶ Se puede mantener la integridad.
- ▶ Es posible mantener la seguridad.

# Lenguajes de Bases de Datos

- ▶ Un Sistema de Bases de Datos proporciona:
  - ▶ Un Lenguaje de Definición de Datos (DDL) para especificar el esquema de la Base de Datos, que permite la definición de objetos en la Base de Datos.
  - ▶ Un Lenguaje de Manipulación de Datos (DML) para expresar las consultas a la Base de Datos y las modificaciones, que permite la manipulación o procesamiento de dichos objetos.
- ▶ En la práctica ambos lenguajes no son dos lenguajes separados, tal como lo es SQL.

# Arquitectura ANSI/Sparc

## ► Niveles en la arquitectura:



# Arquitectura ANSI/Sparc

## ► El Nivel Interno:

- Nivel Físico, es el que está más cerca del almacenamiento físico, es decir, es el que tiene que ver con la forma en que los datos están almacenados físicamente.

## ► El Nivel Externo:

- Nivel Lógico de Usuario es el más próximo a los usuarios, es decir, el que tiene que ver con la forma en que los usuarios ven los datos.

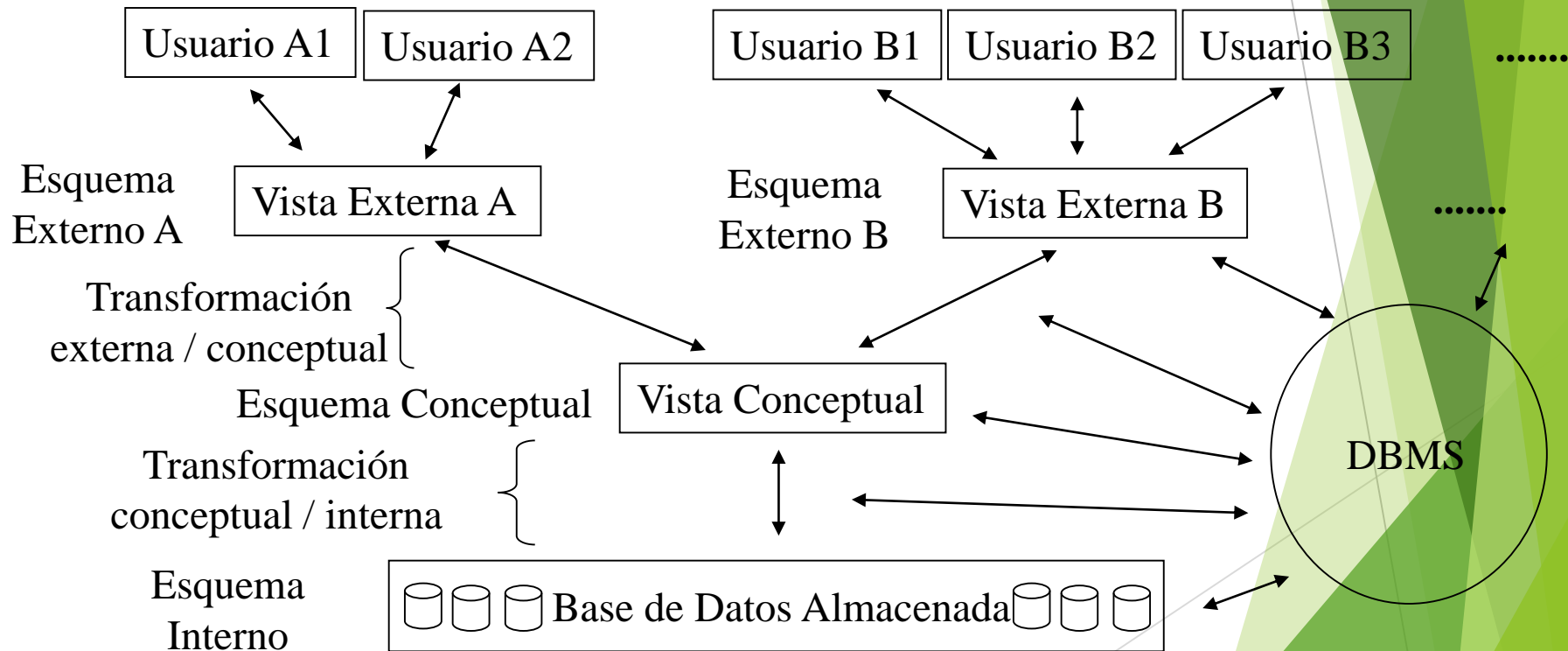
## ► El Nivel Conceptual:

- Nivel Lógico de la Comunidad, o en ocasiones sólo como el nivel lógico, sin calificar; es un nivel de indirección entre los otros dos.



# Arquitectura ANSI/Sparc

Arquitectura de un Sistema de Base de Datos:



# Independencia de Datos

- ▶ La **independencia de datos**, intenta poder modificar un esquema sin tener que modificar el esquema inferior.
- ▶ Cuando se tiene una **dependencia de datos** es cuando resulta imposible modificar un esquema sin tener que afectar le esquema inferior inmediato.

# Transformaciones

- ▶ La transformación interna/conceptual, define la correspondencia entre la vista conceptual y la base de datos almacenada, y especifica cómo están representados los registros y campos conceptuales en el nivel interno.
- ▶ La transformación externa/conceptual define la correspondencia entre una vista externa en particular y la vista conceptual. En general, las diferencias que puedan existir entre éstos dos niveles son análogas a aquellas.

# Modelos

- ▶ Bajo la estructura de la Base de Datos se encuentra el modelo de datos:
  - ▶ Colección de herramientas conceptuales para describir los datos, las relaciones, la semántica y las restricciones de consistencia.
- ▶ Entre los modelos más conocidos tenemos:
  - ▶ Modelos lógicos basados en registros.
  - ▶ Modelos basados en objetos.
  - ▶ Modelos físicos de datos.

# Modelos

- ▶ Modelos lógicos basados en registros.
  - ▶ Se usan para describir el nivel conceptual y físico.
  - ▶ Modelo de Red.
  - ▶ Modelo Jerárquico.
  - ▶ Modelo Relacional.
- ▶ Modelos basados en objetos.
  - ▶ Se usan para describir datos en el nivel conceptual y de visión.
  - ▶ Modelo Entidad - Relación.
  - ▶ Modelo Orientado a Objetos.

# Modelos

- ▶ Modelos físicos de datos.
  - ▶ Se usan para describir datos en el nivel más bajo.
  - ▶ Hay muy pocos modelos físico de datos en uso.
  - ▶ Modelo unificador.
  - ▶ Memoria de elementos.

# Administradores

- ▶ El administrador de datos (DA) es la persona que toma las decisiones de estrategia y política con respecto a los datos de la empresa.
- ▶ El administrador de la base de datos (DBA) es la persona que proporciona el apoyo técnico necesario para implementar dichas decisiones.
- ▶ El DBA es el responsable del control general del sistema al nivel técnico.

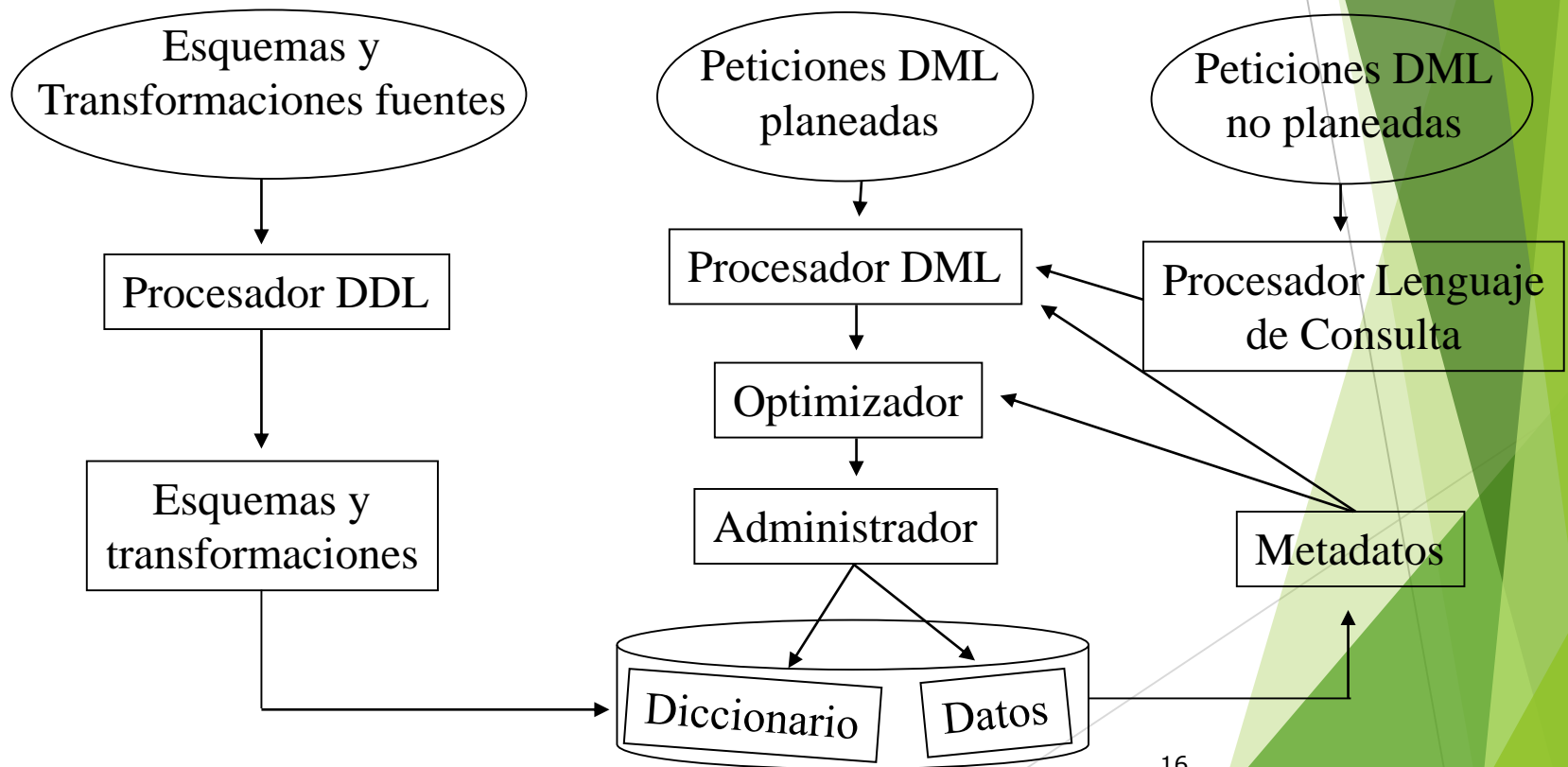
# Administradores

- ▶ Algunas de las tareas del DBA en general comprende:
  - ▶ Definir el esquema conceptual.
  - ▶ Definir el esquema interno.
  - ▶ Establecer un enlace con los usuarios.
  - ▶ Definir las restricciones de seguridad e integridad.
  - ▶ Definir las políticas de vaciado y recarga.
  - ▶ Supervisar el rendimiento y responder a los requerimientos cambiantes.



# DBMS

## Arquitectura de un Sistema Manejador de Base de Datos:



# Desarrollo de un DBS

Facultad de Ciencias de la Computación

# Desarrollo de Bases de Datos

- ▶ El término **Desarrollo de Bases de Datos**, se utiliza para describir el proceso de diseño y ejecución de bases de datos.
- ▶ El objetivo principal en el diseño de bases de datos es crear modelos de bases de datos completos normalizados, no redundantes, conceptuales, lógicos y físicos totalmente integrados.
- ▶ La fase de ejecución se incluye estructuras de almacenamiento, carga de datos, entre otros.

# Fases en el Desarrollo de BDS

- ▶ Recolección y Análisis de Requerimientos.
- ▶ Diseño Conceptual de la Base de Datos.
  - ▶ Diseño del esquema conceptual.
  - ▶ Diseño de transacciones.
- ▶ Elección de un DBMS.
- ▶ Transformación al Modelo de Base de Datos (Diseño Lógico).
- ▶ Diseño Físico de la Base de Datos.
- ▶ Implementación del Sistema de la Base De Datos.

# Modelo Conceptual

Facultad de Ciencias de la Computación

# Modelo Conceptual

- ▶ El modelo de datos **entidad - relación** (E-R) está basado en una percepción del mundo real consistente en objetos básicos:
  - ▶ Entidades
  - ▶ Relaciones
- ▶ Se desarrolló para facilitar el diseño de bases de datos permitiendo la especificación de un esquema de una empresa que representa la estructura completa.

# Modelo Conceptual

- ▶ Los conjuntos de entidades no son necesariamente disjuntos.
- ▶ Una entidad se representa mediante un conjunto de **atributos**.
- ▶ Los atributos describen propiedades que posee cada miembro de un conjunto de entidades.
- ▶ La designación de un atributo para un conjunto de identidades expresa que en la base de datos se va a guardar información similar.

# Modelo Conceptual

- ▶ Los conjuntos de entidades no son necesariamente disjuntos.
- ▶ Una entidad se representa mediante un conjunto de **atributos**.
- ▶ Los atributos describen propiedades que posee cada miembro de un conjunto de entidades.
- ▶ La designación de un atributo para un conjunto de identidades expresa que en la base de datos se va a guardar información similar.

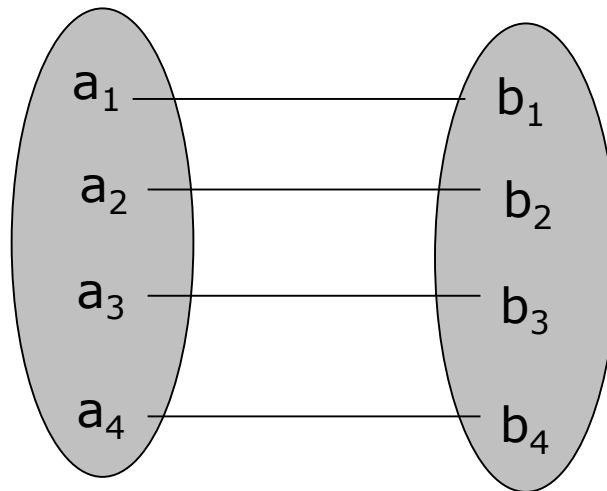


# Modelo Conceptual

- ▶ **Correspondencia de cardinalidad**, expresa el número de entidades a las que otra entidad puede estar asociada vía un conjunto de relaciones.
- ▶ La correspondencia de cardinalidades es más útil describiendo conjunto de relaciones binarias.
- ▶ Para un conjunto de relaciones binarias  $R$  entre los conjuntos de entidades  $A$  y  $B$ , la correspondencia de cardinalidades debe ser:

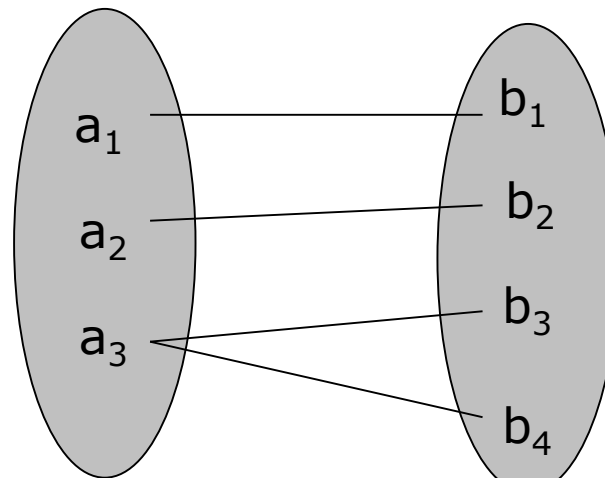
# Cardinalidad

- **Uno a uno:** Una entidad en  $A$  se asocia con a lo sumo una entidad de  $B$ , y una entidad en  $B$  se asocia con a lo sumo una entidad en  $A$ .



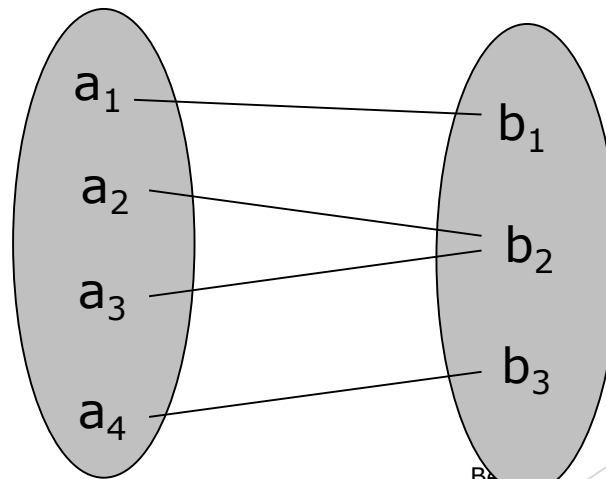
# Cardinalidad

- **Uno a varios:** Una entidad  $A$  se asocia con cualquier número de entidades en  $B$  (ninguna o varias). Una entidad  $B$ , en cambio, se puede asociar con a lo sumo una entidad en  $A$ .



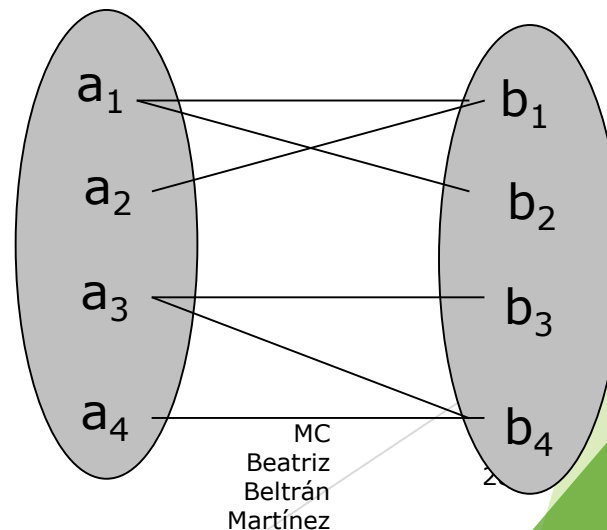
# Cardinalidad

- **Varios a uno:** Una entidad  $A$  se asocia con a los sumo asocia con una entidad en  $B$ . Una entidad  $B$ , en cambio, se puede asociar con cualquier número de entidades (ninguna o varias) en  $A$ .



# Cardinalidad

- **Varios a varios:** una entidad  $A$  se asocia con cualquier número de entidades (ninguna o varias) en  $B$ , y una entidad  $A$  se asocia con cualquier número de entidades (ninguna o varias) en  $B$ .



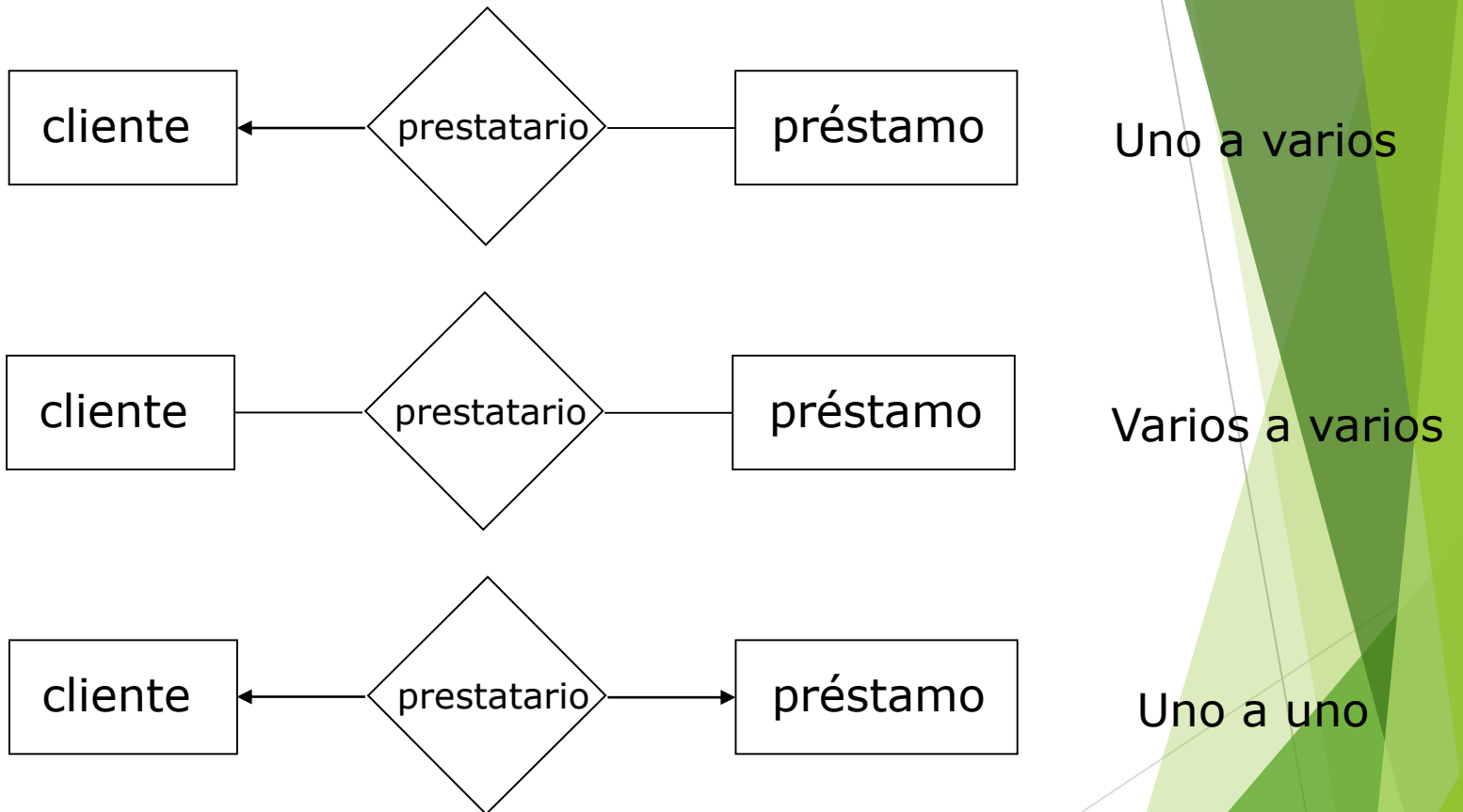
# Diagrama E - R

- ▶ La estructura lógica de una base de datos se puede expresar gráficamente mediante un diagrama E - R.
- ▶ Los diagramas son simples y claros.
- ▶ Los componentes son:
  - ▶ Rectángulos: conjuntos de identidades.
  - ▶ Elipses: atributos.
  - ▶ Rombo: relaciones.
  - ▶ Líneas: que unen conjuntos.

# Diagrama E - R

- ▶ Elipses dobles: atributos multivalorados.
- ▶ Elipses discontinuas: atributos derivados.
- ▶ Líneas dobles: participación total de una entidad en un conjunto de relaciones.
- ▶ Rectángulos dobles: conjunto de entidades débiles.
- ▶ La clave primaria se subraya.
- ▶ Para distinguir los tipo se tiene:
  - ▶ Línea dirigida: Denota uno.
  - ▶ Línea no dirigida: Denota varios.

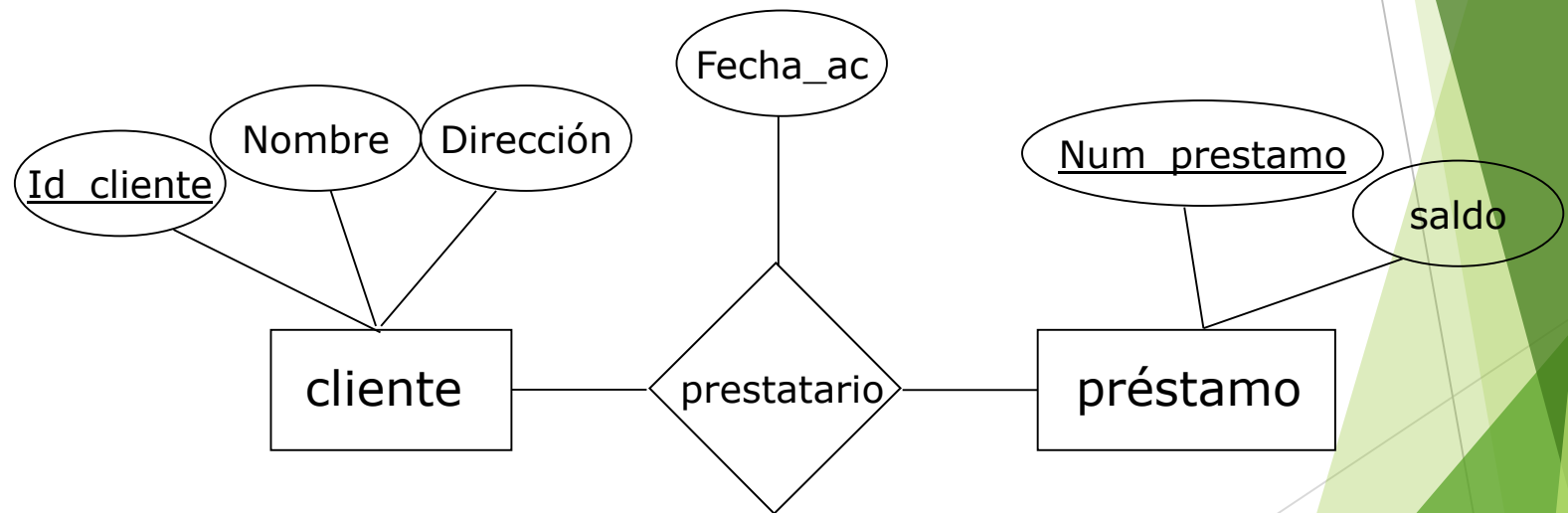
# Diagrama E - R





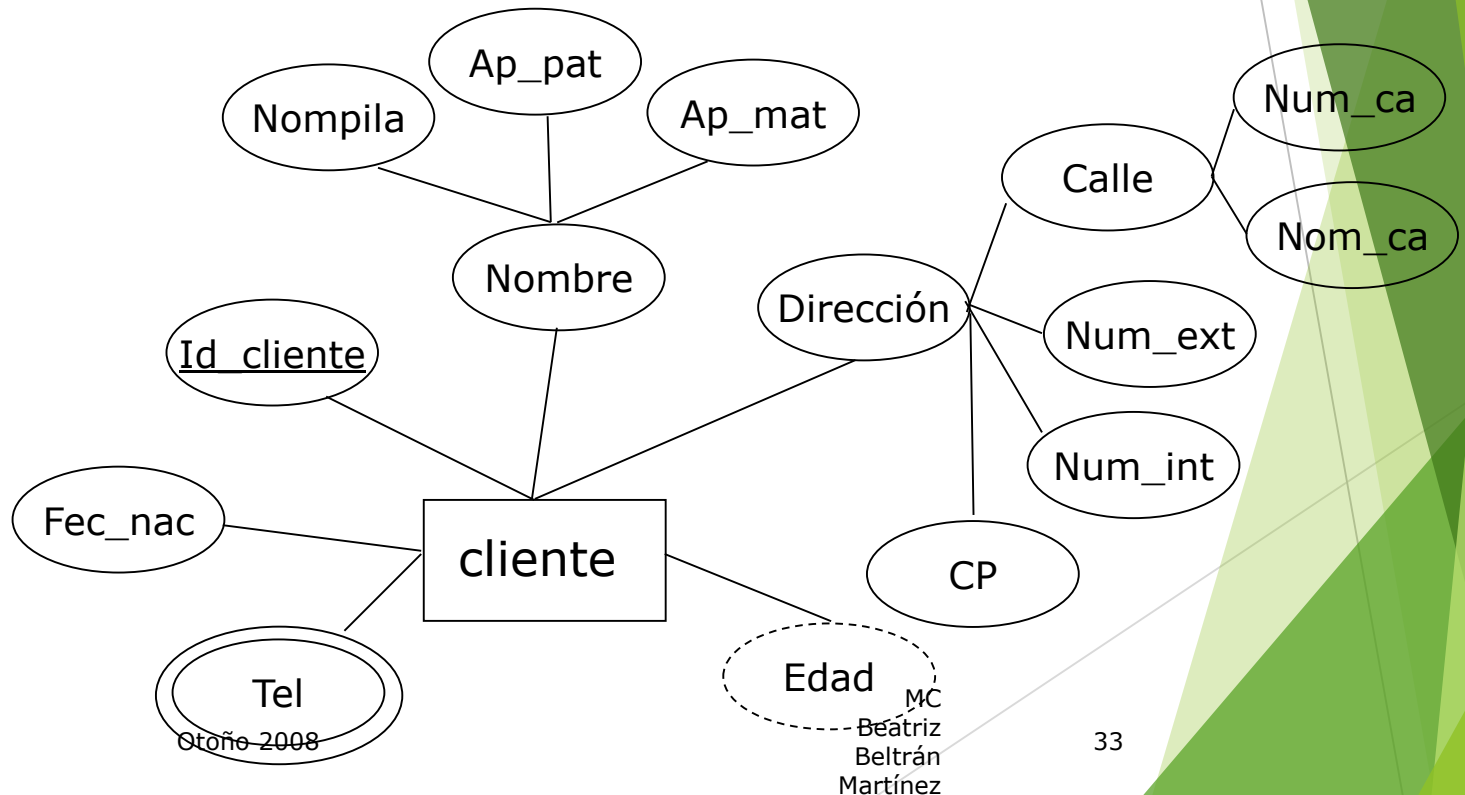
# Diagrama E - R

- Se pueden tener atributos unidos a un conjunto de relaciones.



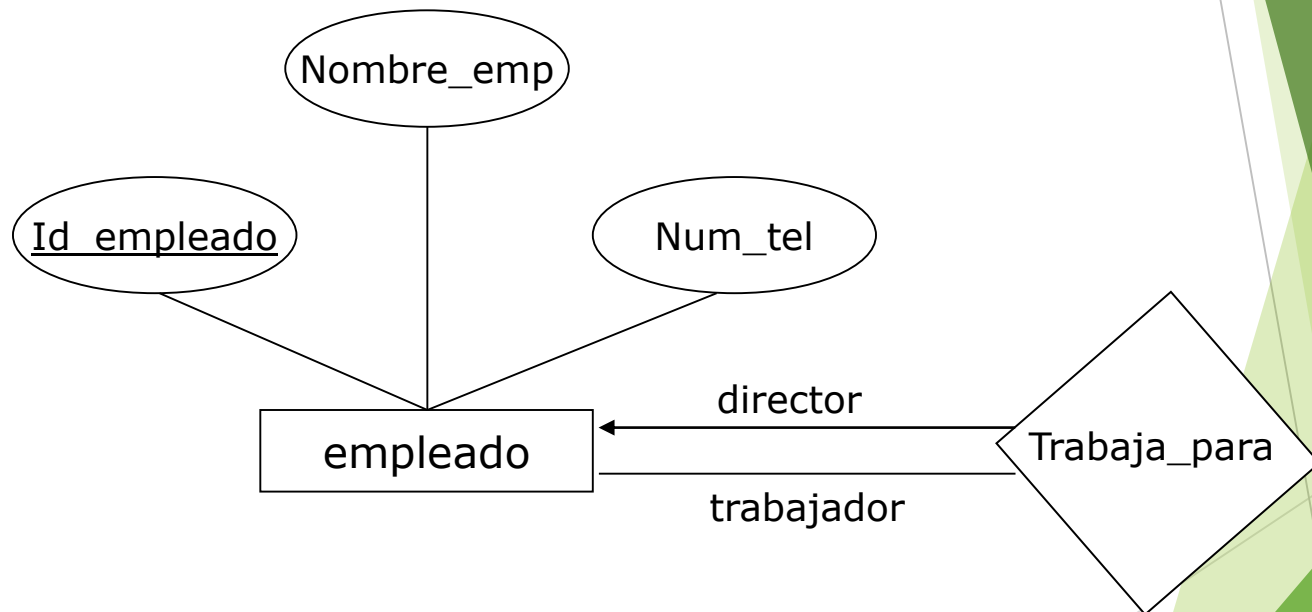
# Diagrama E - R

- Atributos compuestos, multivalorados y derivados.



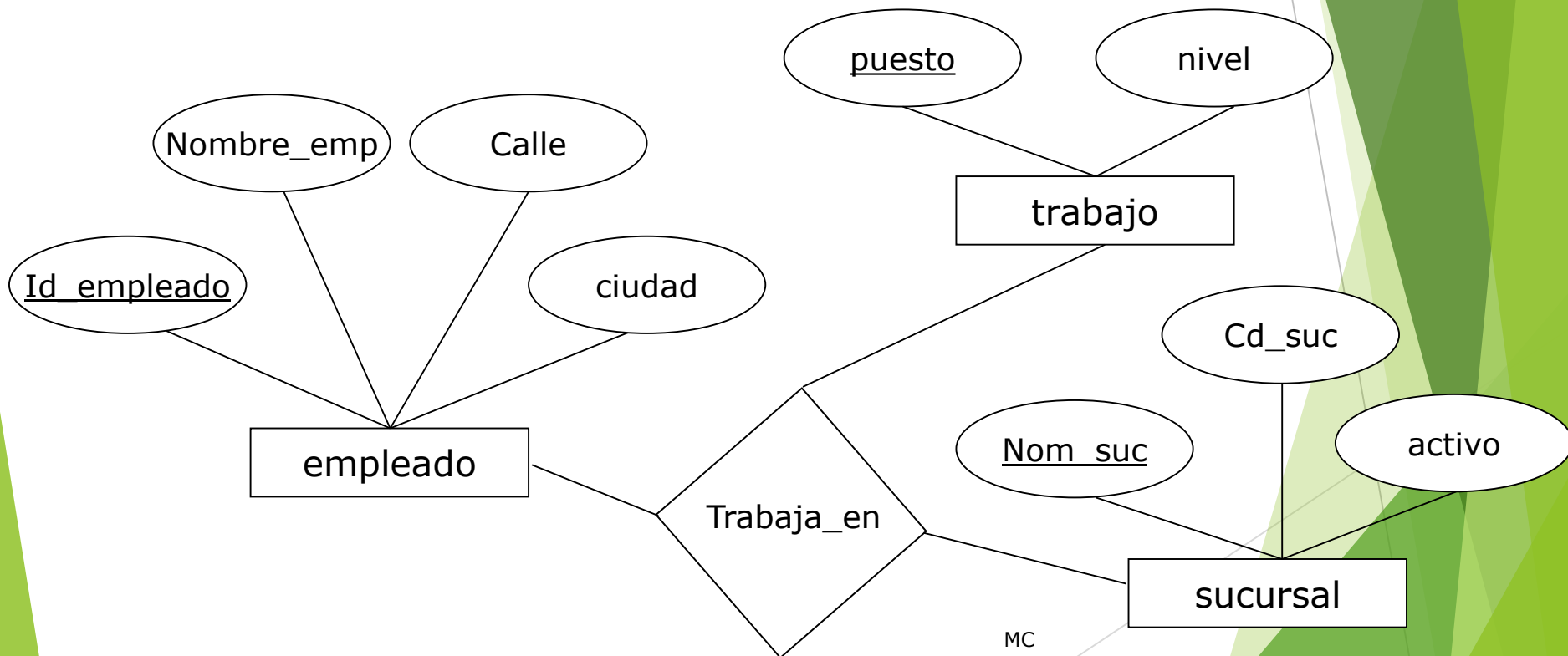
# Diagrama E - R

- Para indicadores de papeles



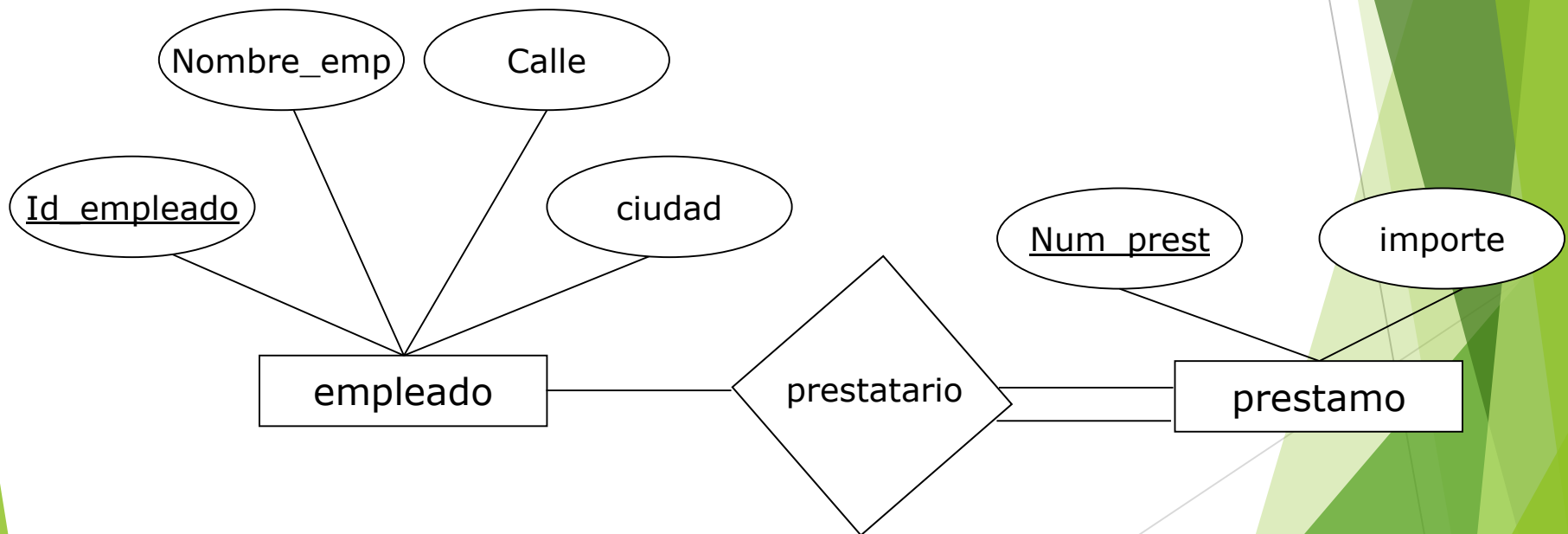
# Diagrama E - R

## ► Ejemplo de relación ternaria.



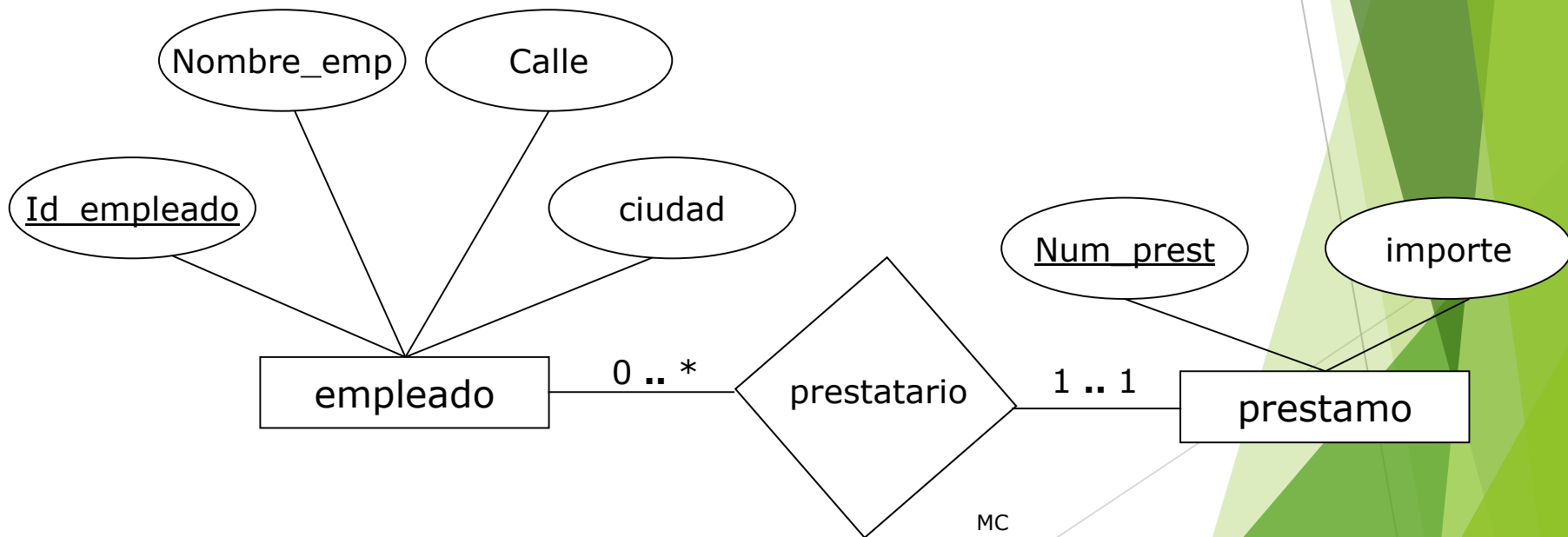
# Diagrama E - R

- Participación total de un conjunto de entidades en un conjunto de relaciones.



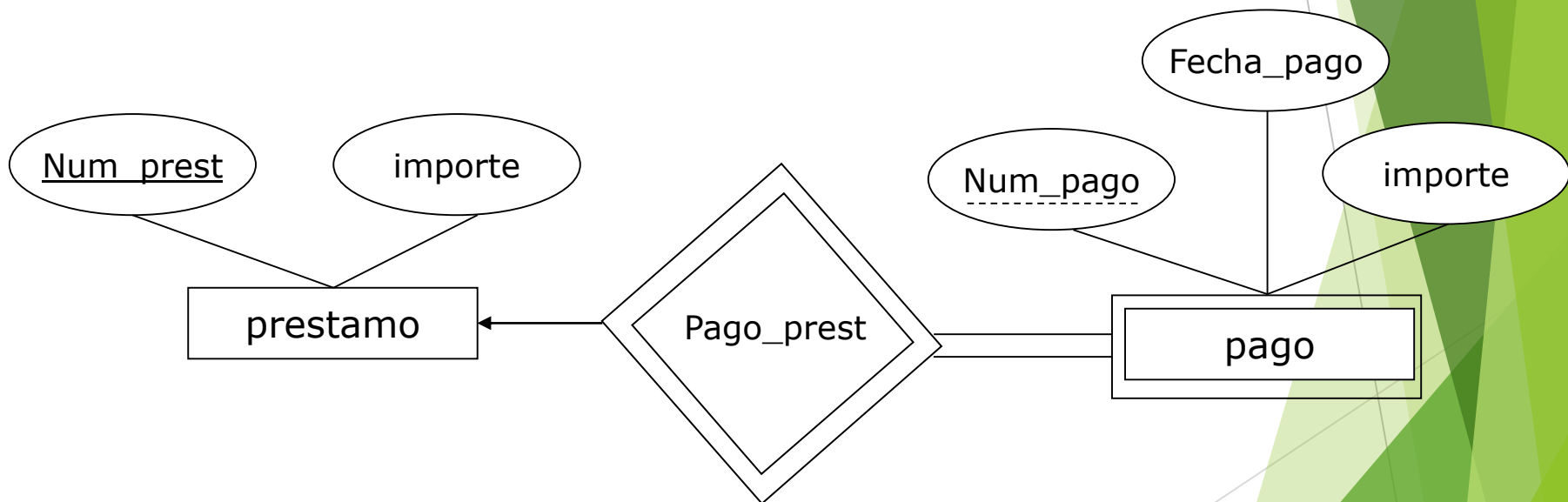
# Diagrama E - R

- Límites de cardinalidad en conjunto de relaciones.



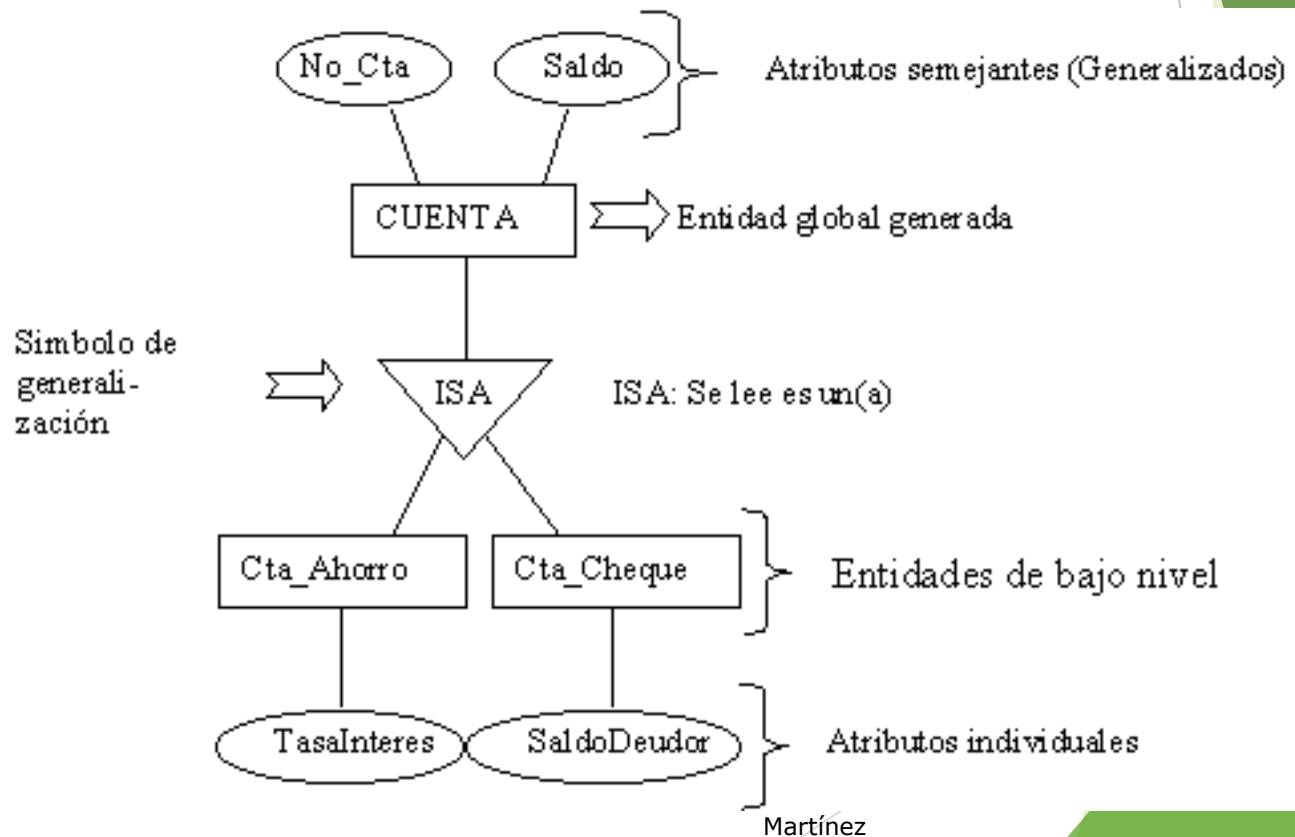
# Diagrama E - R

- Conjunto de entidades débiles.



# Diagrama E - R

## ► Generalización.





# Modelo Lógico

Facultad de Ciencias de la Computación

# Diseño de un esquema

- ▶ Un modelo de datos que se ajusta a un esquema de bases de datos E - R se puede representar por medio de tablas.
- ▶ Se forma una tabla única por cada conjunto de:
  - ▶ Entidades de la base de datos.
  - ▶ Relaciones de la base de datos.
- ▶ Cada tabla tendrá sus columnas con un nombre único.

# Diseño de un esquema

- ▶ Entidades fuertes.
  - ▶ Sea  $E$  un conjunto de entidades fuertes con los atributos descriptivos  $a_1, a_2, \dots, a_n$ .
    - ▶ Esta entidad se representa mediante una tabla llamada  $E$  con  $n$  columnas distintas.
    - ▶ Cada fila de la tabla corresponde a una entidad del conjunto de entidades  $E$ .

# Diseño de un esquema

- ▶ Entidades Débiles.

- ▶ Se representa el conjunto de entidades  $A$  mediante una tabla llamada  $A$  por cada uno de los atributos del conjunto:

- ▶  $\{a_1, a_2, \dots, a_m\} \cup \{b_1, b_2, \dots, b_n\}$

- ▶ El conjunto *pago* quedaría:

Num_pago	Fecha_pago	Importe	Num_prest

Otoño 2006

MC  
Beatriz  
Beltran  
Martínez

43

# Diseño de un esquema

- ▶ Atributos compuestos.
  - ▶ Se manejan creando un atributo separado por cada uno de los atributos componentes.
  - ▶ No se crea una columna separada para el propio atributo compuesto.
  - ▶ Si *dirección* esta compuesto por *calle* y *ciudad*; se crea en la tabla las columnas *calle\_dir*, *ciudad\_dir*, *dirección* no tiene una columna.

# Diseño de un esquema

## ► Atributos multivalorados.

- Para estos atributos se crea, como excepción; una tabla nueva.
- Para el atributo multivalorado  $M$  se crea la tabla  $T$  con una columna  $C$  que corresponde a la clave primaria del conjunto de entidades o conjunto de relaciones del que  $M$  es atributo.
- La columna  $C$  aparece en el conjunto de entidad donde esta el atributo multivalorado.

# Diseño de un esquema

- ▶ Generalización.
  - ▶ Hay dos métodos para transformar a forma tabular un diagrama de E - R de una generalización:
    1. Crear una tabla para el conjunto de entidades de nivel más alto. Para cada conjunto de entidades de nivel más bajo, crear una tabla que incluya una columna para cada uno de los atributos de ese conjunto de entidades más una columna por cada atributo de la clave primaria del conjunto de entidades del nivel más alto.

# Diseño de un esquema

2. Se tiene cuando la generalización es disjunta y completa. No se crea una tabla para el conjunto de entidades de más alto nivel. En su lugar, para cada conjunto de entidades de mas bajo nivel se crea una tabla que incluya una columna por cada atributo del conjunto de entidades de nivel más bajo más una columna por cada atributo del conjunto de entidades de nivel más alto.



# Modelo Relacional

- ▶ Consiste de un conjunto de **tablas** a cada una de las cuales se les asigna un nombre exclusivo.
- ▶ Cada tabla tiene la estructura, donde cada fila de la tabla representa una **relación** entre un conjunto de valores.
- ▶ Las columnas se les conoce como **atributos**.
- ▶ Para cada atributo hay un conjunto de valores permitidos, llamado **dominio** de ese atributo.

# Modelo Relacional

- ▶ Se utiliza los términos matemáticos **tupla** en lugar de *fila*; y **relación** en lugar de *tabla*.
- ▶ Se exige que para toda relación  $r$ , los dominios de todos los atributos de  $r$  sean atómicos.
- ▶ Un dominio es **atómico** si los elementos del dominio se consideran indivisibles.

# Modelo Relacional

- ▶ Se define una relación como subconjuntos de productos cartesianos de los dominios.
- ▶ Una variable tupla es una variable que representa a una **tupla**, esto es, una tupla que representa al conjunto de todas las tuplas.
- ▶ El orden en que aparecen las tuplas es irrelevante, dado que una relación es un conjunto.

# Modelo Relacional

- ▶ No se admiten valores múltiples, por lo tanto la intersección de un renglón con una columna tiene un sólo valor, nunca un conjunto de valores
- ▶ Es posible que varios atributos tengan el mismo dominio.
- ▶ Un valor de dominio que es miembro de todos los dominios posibles es el valor nulo, que indica que el valor es desconocido o no existe.

# Modelo Relacional

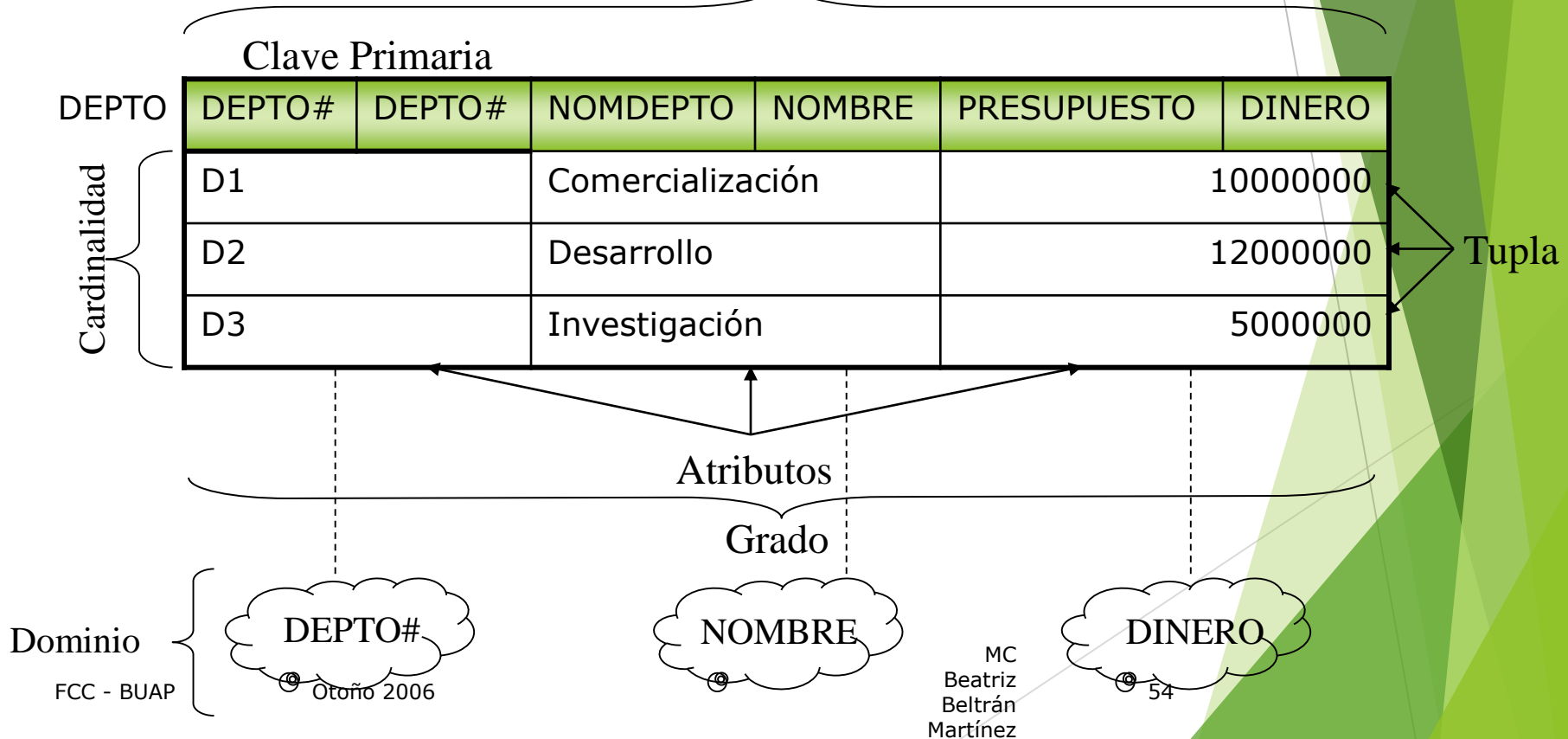
- ▶ Como cada atributo tiene un **dominio**, el cual es una descripción física y lógica de valores permitidos:
  - ▶ No existen 2 tuplas en la tabla que sean idénticas.
  - ▶ No hay un orden entre tuplas o atributos.
  - ▶ La información en las bases de datos son representados como datos explícitos, no existen apuntadores o ligas entre las tablas.

# Modelo Relacional

- ▶ Se consideran los conceptos de:
  - ▶ **Superclave:** es un conjunto de uno o más atributos que tomados colectivamente, permiten identificar de forma única una entidad en el conjunto de identidades.
  - ▶ **Claves candidatas:** superclaves minimales.
  - ▶ **Clave primaria:** denota una clave candidata que es elegida como elemento principal para identificar la entidad.

# Modelo Relacional

## Relación



# Álgebra Relacional

Facultad de Ciencias de la Computación



# Álgebra Relacional

- ▶ El modelo relacional se basa en el **modelo relacional de datos**.
- ▶ De manera intuitiva, significa que en dichos sistemas hay:
  1. **Aspecto estructural.**
  2. **Aspecto de integridad.**
  3. **Aspecto de manipulación.**
    - ▶ Operación restringir.
    - ▶ Operación proyectar.
    - ▶ Operación juntar.

# Álgebra Relacional

Sea:

DEPTO

DEPTO#	NOMDEPTO	PRESUPUESTO
D1	Comercialización	10000000
D2	Desarrollo	12000000
D3	Investigación	5000000

EMP

EMP#	NOMEMP	DEPTO#	SALARIO
E1	López	D1	40000
E2	Cheng	D1	42000
E3	Pérez	D2	30000
E4	Hernández	D3	35000

# Álgebra Relacional

## ► Restringir:

- DEPTO where PRESUPUESTO > 8000000.

DEPTO#	NOMDEPTO	PRESUPUESTO
D1	Comercialización	10000000
D2	Desarrollo	12000000

# Álgebra Relacional

## ► Proyectar:

► DEPTO {DEPTO#, PRESUPUESTO}

DEPTO#	PRESUPUESTO
D1	10000000
D2	12000000
D3	5000000

# Álgebra Relacional

► Juntar:

► DEPTO join EMP

DEPTO#	NOMDEPTO	PRESUPUESTO	EMP#	NOMEMP	SALARIO
D1	Comercialización	10000000	E1	López	40000
D1	Comercialización	10000000	E2	Cheng	42000
D2	Desarrollo	12000000	E3	Pérez	30000
D3	Investigación	5000000	E4	Hernández	35000

# Álgebra Relacional

## ► Operación Selección.

- Selecciona tuplas que satisfacen un predicado dado.
- En general se permiten las comparaciones que utilizan:  $=$ ,  $\neq$ ,  $\leq$  o  $\geq$  en el predicado de selección.
- Además se pueden combinar varios predicados en una mayor utilizando conectivas y ( $\wedge$ ) y o ( $\vee$ ).
- El predicado puede incluir comparaciones con dos atributos.
- Cualquier evaluación con nulo da **falso**.

# Álgebra Relacional

## ► Operación Proyección.

- Es una operación unaria que devuelve su relación de argumento, excluyendo algunos argumentos.
- Dado que las relaciones son conjuntos, se eliminan todas las filas repetidas.
- Se mencionan los atributos que se desea que aparezcan en la relación resultante,

# Álgebra Relacional

## ► Operación Unión.

- Operación binaria, que toma dos relaciones del mismo tipo y toma las tuplas que aparecen en una o bien en la otra relación.
- Dado que las relaciones son conjuntos se eliminan las tuplas repetidas.
- Se debe asegurar que las uniones se realicen entre relaciones compatibles.
  - Misma aridad.
  - Dominio  $i$ -ésimo iguales en ambas relaciones.



# Álgebra Relacional

- ▶ Operación Diferencia.
  - ▶ Operación binaria, que toma dos relaciones del mismo tipo y toma las tuplas que aparecen en la primera relación que no aparecen en la segunda relación.
  - ▶ Al igual que en la unión se debe considerar que las relaciones sean compatibles.

# Álgebra Relacional

- ▶ Operación Producto Cartesiano.
  - ▶ Permite combinar información de cualesquiera dos relaciones.
  - ▶ Es equivalente al producto cartesiano definido en matemáticas.
  - ▶ Se pide que los nombres de las relaciones no sean iguales.
  - ▶ Se crea una tupla a partir de cada par de tuplas posibles.

# Álgebra Relacional

- ▶ Operación Renombramiento.
  - ▶ Devuelve una relación con nombre.
  - ▶ Esta operación permite también cambiar el nombre de los atributos.

# Álgebra Relacional

## ► Operación Reunión Natural.

- Es una operación binaria, que permite combinar ciertas selecciones y un producto cartesiano en una sola operación.
- Forma un producto cartesiano de sus dos argumentos, realiza una selección forzando la igualdad de los atributos que aparecen en ambos esquemas de relación y finalmente, elimina los atributos repetidos.

# Álgebra Relacional

- ▶ Operación Intersección.
  - ▶ Operación binaria, que toma dos relaciones del mismo tipo y toma las tuplas que aparecen en la primera relación pero que también aparecen en la segunda relación.
  - ▶ Al igual que en la unión se debe considerar que las relaciones sean compatibles.

# Álgebra Relacional

## ► Operación Asignación.

- Actúa similar a la asignación de los lenguajes de programación convencionales.
- La evaluación de la asignación no hace que se muestre ninguna relación al usuario.
- La variable a la que se asignó puede utilizarse después en otras operaciones relacionales.

# Normalización

Facultad de Ciencias de la Computación

# Normalización

- ▶ Forma parte del diseño de Base de Datos.
- ▶ El objetivo es: Guardar la información con un mínimo de redundancia o sin redundancia innecesaria.
- ▶ Además la recuperación debe ser simple.
- ▶ Las forma normales se definen en términos de las dependencias funcionales.



# Normalización

## 1ª Forma Normal

- ▶ Se tiene un requisito elemental, y no exige información adicional.
- ▶ Un dominio es **atómico** si se considera que los elementos del dominio son unidades indivisibles.
- ▶ Se dice que el esquema  $R$  está en primera forma normal (1FN) si los dominios de todos los atributos de  $R$  son atómicos.

# Normalización

- ▶ Por ejemplo, argumentos que **no** se encuentran en 1FN:
  - ▶ Si la relación ***empleado***, contiene el atributo *hijos*, se tienen elementos cuyo dominio son conjuntos de nombres.
  - ▶ Si el número de identificación de los empleados de una compañía, se designa con las dos primeras letras que especifican el departamento, seguido de un número de 4 dígitos, formando un número único dentro de ese departamento.

# Normalización

## 2ª Forma Normal

NOMBRE_SUC	CIUDAD_SUC	ACTIVO	NOMBRE_CLIENTE	NUM_PREST	IMPORTE
Los Fuertes	Puebla	9000000	Santos	P-17	1000
Santa Ana	Tlaxcala	2100000	Gómez	P-23	2000
Narvarte	DF	1700000	López	P-15	1500
Los Fuertes	Puebla	9000000	Gutiérrez	P-14	1500
Tacuba	DF	400000	Santos	P-93	500
Taxqueña	DF	8000000	Abril	P-11	900
Centro	Colima	300000	Valdivieso	P-29	1200
Rio Blanco	Veracruz	3700000	López	P-16	1300
Los Fuertes	Puebla	9000000	González	P-18	2000
Narvarte	DF	1700000	Rodríguez	P-25	2500
Angelopolis	Puebla	7100000	Martínez	P-10	2200

# Normalización

NOMBRE_SUC	CIUDAD_SUC	ACTIVO	NOMBRE_CLIENTE	NUM_PREST	IMPORTE
Los Fuertes	Puebla	9000000	Santos	P-17	1000
Santa Ana	Tlaxcala	2100000	Gómez	P-23	2000
Narvarte	DF	1700000	López	P-15	1500
Los Fuertes	Puebla	9000000	Gutiérrez	P-14	1500
Tacuba	DF	400000	Santos	P-93	500
Taxqueña	DF	8000000	Abril	P-11	900
Centro	Colima	300000	Valdivieso	P-29	1200
Rio Blanco	Veracruz	3700000	López	P-16	1300
Los Fuertes	Puebla	9000000	González	P-18	2000
Narvarte	DF	1700000	Rodríguez	P-25	2500
Angelopolis	Puebla	7100000	Martínez	P-10	2200
Narvarte	DF	1700000	Fernández	P-31	1500

# Normalización

## Dependencias Funcionales

- ▶ Una **dependencia funcional** es un tipo de restricción que constituye una generalización de clave.
- ▶ La definición de **superclave** dice que ningún par de tuplas en una relación puede tener el mismo valor para el conjunto de atributos.
- ▶ El concepto de dependencia funcional generaliza la noción de superclave.

# Normalización

- Considérese el esquema de una relación  $R$  y sean  $\alpha \subseteq R$ ,  $\beta \subseteq R$ . La dependencia funcional

$$\alpha \rightarrow \beta$$

se cumple para el esquema  $R$  si, en cualquier relación para todo par de tuplas  $t_1$  y  $t_2$  tales que  $t_1[\alpha] = t_2[\alpha]$  también ocurre que  $t_1[\beta] = t_2[\beta]$ .

# Normalización

- ▶ Las dependencias funcionales que se tienen:
  - ▶  $\text{NUM\_PREST} \rightarrow \text{IMPORTE}$
  - ▶  $\text{NUM\_PREST} \rightarrow \text{NOMBRE\_SUC}$
- ▶ Pero no hay dependencia funcional en:
  - ▶  $\text{NUM\_PREST} \rightarrow \text{NOMBRE\_CLIENTE}$
- ▶ Por que cada préstamo se puede conceder a más de un cliente.

# Normalización

T1

V#	Status	Ciudad	P#	Cant
V1	20	Londres	P1	300
V1	20	Londres	P2	200
V1	20	Londres	P3	400
V1	20	Londres	P4	200
V1	20	Londres	P5	100
V1	20	Londres	P6	100
V2	10	París	P1	300
V2	10	París	P2	400
V3	10	París	P2	200
V4	20	Londres	P2	200
V4	20	Londres	P4	300
V4	20	Londres	P5	400



# Normalización

T2

V#	Status	Ciudad
V1	20	Londres
V2	10	París
V3	10	París
V4	20	Londres

T3

V#	P#	Cant
V1	P1	300
V1	P2	200
V1	P3	400
V1	P4	200
V1	P5	100
V1	P6	100
V2	P1	300
V2	P2	400
V3	P2	200
V4	P2	200
V4	P4	300
V4	P5	400

# Normalización

T2

V#	Status	Ciudad
V1	20	Londres
V2	10	París
V3	10	París
V4	20	Londres
V5	30	Atenas

T3

V#	P#	Cant
V1	P1	300
V1	P2	200
V1	P3	400
V1	P4	200
V1	P5	100
V1	P6	100
V2	P1	300
V2	P2	400
V3	P2	200
V4	P2	200
V4	P4	300
V4	P5	400