

# BASES DE DATOS DISTRIBUIDAS

Sistemas informáticos I

## ÍNDICE

- Introducción:
  - Gestores de bases de datos
  - Objetivos de las bases de datos
  - Modelo Entidad-Relación y Modelo Relacional
- SQL
  - DDL
  - DML
  - Subconsultas
  - Agrupaciones y funciones de agregación
- Procedimientos almacenados y Triggers
- SQL inmerso/embebido
- Optimización de consultas
- Temas avanzados:
  - Tecnologías de tratamiento de Big Data
  - Bases de datos NoSQL
  - Map-reduce y métodos similares



## ANTES DE METERNOS EN MATERIA...

¿Qué es una base de datos?



## ANTES DE METERNOS EN MATERIA...

- El concepto “base de datos distribuida”, o mejor aún, “base de datos como sistema distribuido” tiene varias acepciones:
  - Por una parte, es un ejemplo clásico de sistema C/S
    - Pudiendo los datos estar físicamente en un único o distintos servidores
  - Por otra parte, se puede entender que hace referencia a BBDD que mantienen distribuidos los propios datos
    - Aquí también sería un sistema C/S, pero ahora el servidor en sí mismo es un sistema distribuido, que puede tener una arquitectura C/S o P2P
- Aquí principalmente hablaremos de BBDD relacionales (datos centralizados), aunque más adelante explicaremos limitaciones de estos sistemas y las alternativas actuales (generalmente conocidas como soluciones NoSQL)



# 3.1 INTRODUCCIÓN

Bases de datos distribuidas



## GESTORES DE BASES DE DATOS

- Sistema que se encarga de la organización, almacenamiento, gestión y recuperación (eficiente) de la información
- Incluye:
  - Un lenguaje para modelar la información de acuerdo a un determinado modelo (DDL, *Data Definition Language*) → Curiosidad: uno de los aspectos que ha cambiado en la mayoría de las BBDD NoSQL
  - Estructuras de almacenamiento de la información optimizadas para trabajar con un **gran volumen de datos** (aunque no es lo que ahora se entiende por BigData)
  - Un lenguaje para recuperar/manipular la información almacenada mediante búsquedas dirigidas (DML, *Data Manipulation Language*)
  - Los mecanismos adecuados que le permitan integrarse en un sistema de acceso con **control transaccional**
- Los dos últimos puntos siguen siendo las ventajas frente a propuestas más modernas



## GESTORES DE BASES DE DATOS

- El modelo más habitual de gestores de bases de datos es el que sigue el **modelo relacional**
- El lenguaje más utilizado actualmente para modelar y recuperar la información es **SQL** (*Structured Query Language*)
  - Aunque hay una definición ISO para SQL el lenguaje soportado por cada SGBD suele ser ligeramente diferente
- Constituyen el núcleo en torno al cual se ha desarrollado "gran parte" de las aplicaciones actualmente en producción (distribuidas y no distribuidas)
- Algunos ejemplos de gestores de bases de datos:
  - IBM DB/2
  - Oracle
  - Sybase
  - Computer Associates Ingress
  - IBM Informix
  - Microsoft Access
  - Microsoft SQL Server
  - PostgreSQL
  - MySQL (MaríaDB)



## SOBRE EL ACCESO A LOS DATOS

- Originalmente las aplicaciones relacionadas con el manejo de datos se construían sobre un conjunto de ficheros
- Problemas originados por esta aproximación:
  - Volumetría
    - Si manejamos muchos datos el tamaño del fichero se hace inmanejable
  - Redundancia e Inconsistencia en los datos
  - Acceso a los datos ineficiente:
    - Formatos variados e información duplicada en diferentes ficheros
    - Hay que escribir un programa nuevo para cada nueva funcionalidad que se desee añadir
  - Los datos no están aislados (hay que modificar todos los programas si se varía la estructura de los datos)
  - Problemas de Integridad
    - Las restricciones tienen que ser reforzadas en cada programa y no por la base de datos (edad >0)
    - Es difícil añadir nuevas restricciones o variar las que se establecieron inicialmente



## SOBRE EL ACCESO A LOS DATOS

- Atomicidad de las modificaciones difícil de asegurar
  - La base puede quedarse en un estado inconsistente
    - por ejemplo: transferencia de dinero de una cuenta a otra
- Acceso simultáneo por varios usuarios
  - Se debe permitir el acceso simultáneo para ganar velocidad de proceso
  - Pero, ¿cómo asegurar que dos actualizaciones no son conflictivas?
    - por ejemplo dos personas sacando dinero de la misma cuenta simultáneamente
- Seguridad
  - ¿Cómo restringir parcialmente el acceso a los datos?
- El proceso de abstracción de los datos es difícil

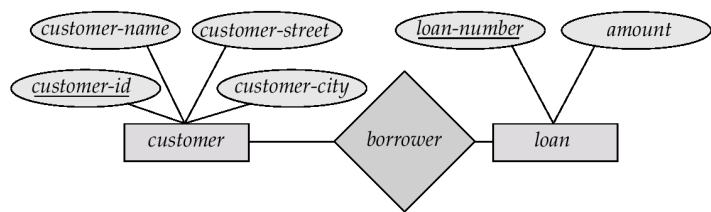


## MODELO ENTIDAD-RELACIÓN

- Modelo de datos conceptual → modelo abstracto independiente del SGBD
  - No confundir con un modelo relacional
- Percepción del mundo real
- Una base de datos puede modelarse como una colección de entidades y relaciones entre entidades
- Elementos básicos en el modelo:
  - Entidad: "cosa" u "objeto" distinguible de otros objetos
  - Atributo: propiedad de una entidad
  - Relación: asociación entre entidades
- Ejemplo:
  - Un banco desea tener almacenada la información sobre sus clientes, los préstamos que tienen éstos con el banco y los datos de sus empleados.

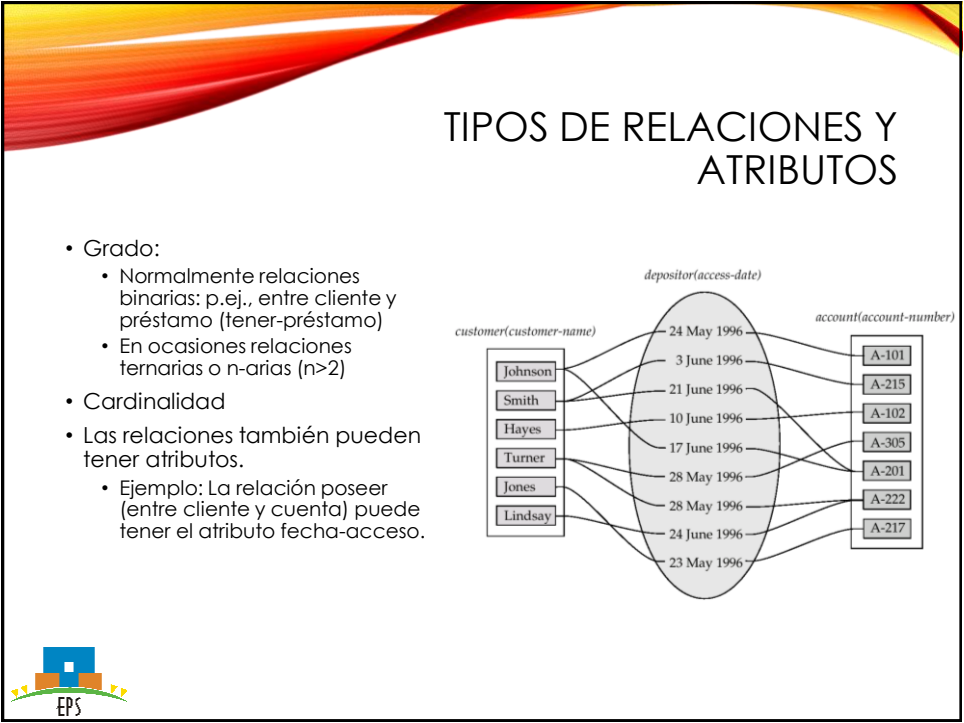
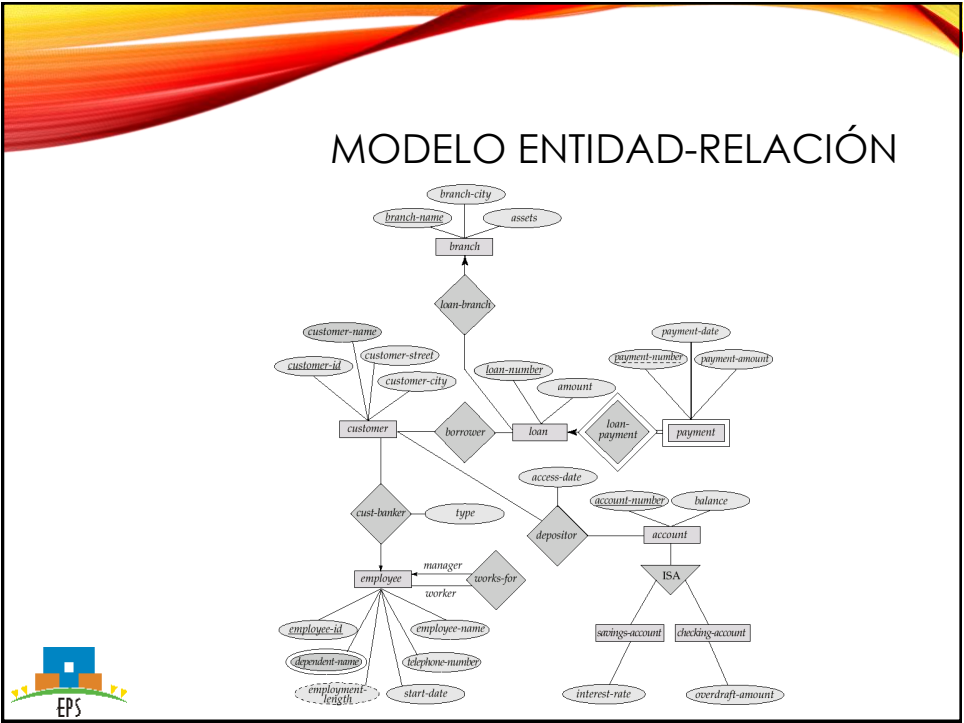


## MODELO ENTIDAD-RELACIÓN



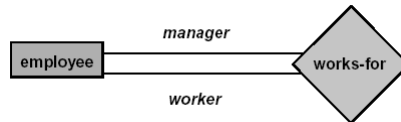
- **Rectángulos:** entidades
- **Rombos:** relaciones
- **Líneas:** enlaces entre entidad-atributo y entre entidad-relación
- **Elipses:** atributos
  - **Elipses dobles:** atributos multivalorados
  - **Elipses punteadas:** atributos derivados
  - **Subrayado:** clave primaria





## ROLES (EN RELACIONES)

- La función que una entidad juega en una relación se llama **rol**
- Normalmente los roles son obvios:
  - Ejemplo: cliente y préstamo en tener-préstamo
- Otras veces no son obvios los roles
  - Ejemplo: trabaja-para en un modelo sobre la organización de una empresa
  - Trabaja-para      empleado X empleado
  - ¿Quién es el empleado jefe y quien el trabajador?



## CLAVES

- Superclave: uno o más atributos que permiten identificar de forma única a una entidad en el conjunto de entidades
  - La combinación de *nombre-cliente* e *id-cliente* es una superclave del conjunto de entidades *cliente*
- Clave candidata: superclaves mínimas
  - *id-cliente* es una clave candidata de cliente.
- Clave primaria: clave candidata elegida para identificar de forma unívoca a una entidad en el conjunto de entidades
  - No puede tener valor nulo (NULL).
  - No se puede repetir
  - Sus valores no deberían cambiar





## CARDINALIDAD DE ASIGNACIÓN

- Restricción que indica el número de entidades con las que puede asociarse otra entidad mediante una relación
  - **Una a una (1-1)**: una entidad en A está asociada a lo sumo con una entidad en B, y una entidad en B está asociada a lo sumo con una entidad en A
  - **Muchas a una ( $\infty$ -1)**: una entidad en A puede estar asociada a lo sumo con una entidad en B, y una entidad en B está asociada con un número cualquiera de entidades en A
  - **Muchas a muchas ( $\infty$ - $\infty$ )**: una entidad en A puede estar asociada con un número cualquiera de entidades en B, y una entidad en B puede estar asociada con un número cualquiera de entidades en A

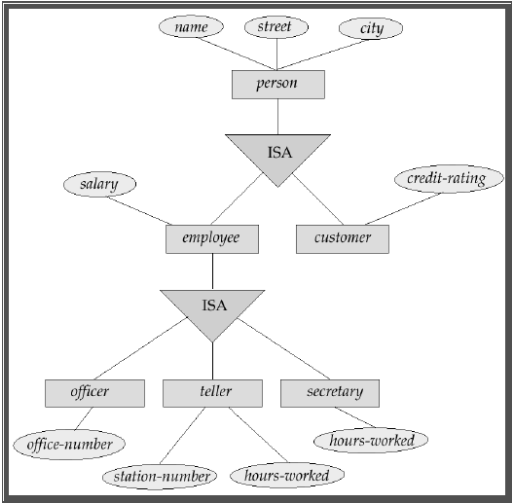


## RELACIÓN ISA

- Relación entre dos entidades para indicar que una de ellas es un subconjunto de la otra
  - Los conjuntos de entidades de nivel más alto: **superclase/superentidad**
  - Los conjuntos de entidades de nivel más bajo: **subclase/subentidad**
- **Herencia de atributos**: un conjunto de entidades de más bajo nivel hereda todos los atributos y la participación en las relaciones del conjunto de entidades de más alto nivel con la que está enlazada
- Herencia de atributos, superclase y subclase son conceptos importantes en el modelo entidad-relación que no tienen traducción directa al modelo relacional
- Proceso de diseño de arriba a abajo (top-down): un conjunto de entidades puede incluir subgrupos de entidades → **especialización**
- Proceso de diseño de abajo a arriba (down-top): varios conjuntos de entidades se sintetizan en un conjunto de entidades de más alto nivel basándose en características comunes → **generalización**

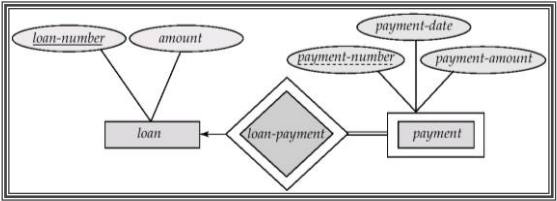


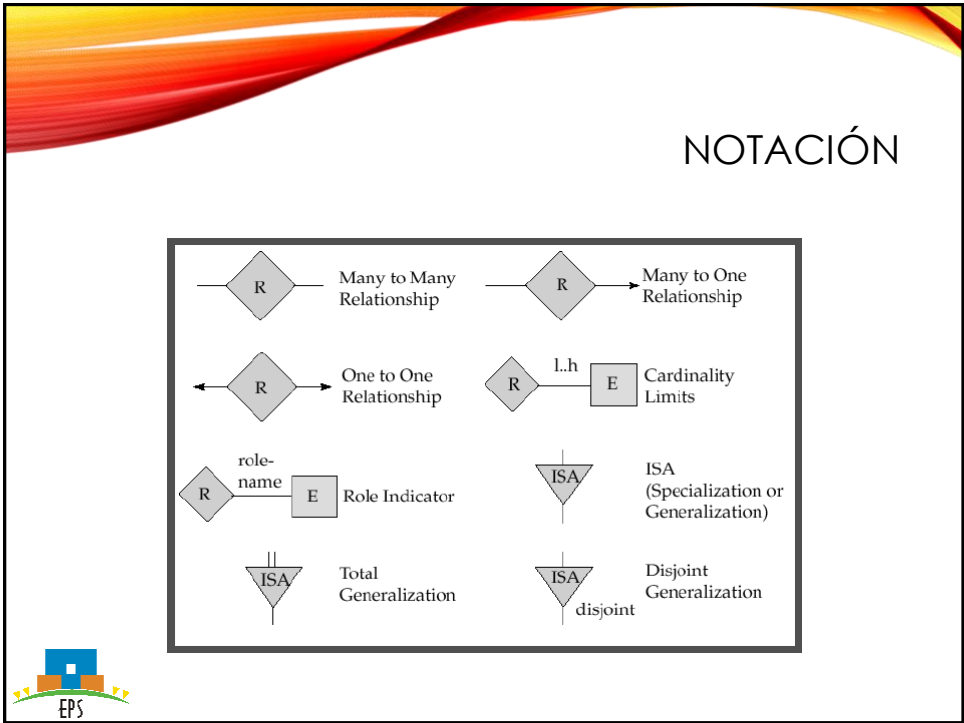
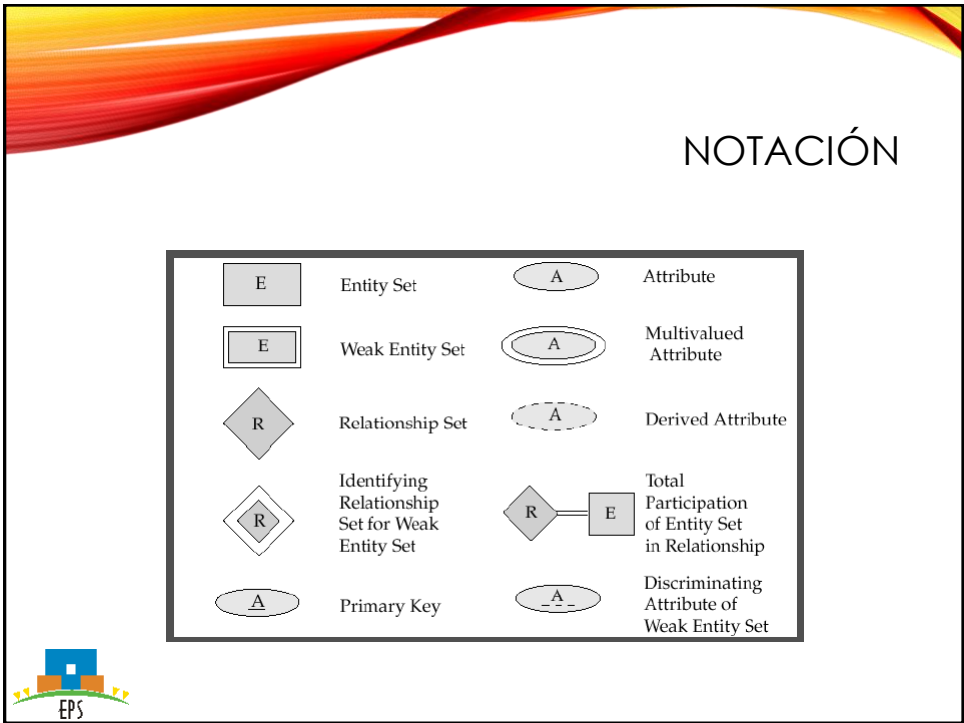
## EJEMPLOS DE RELACIONES ISA



## ENTIDADES DÉBILES


- Entidad que no tiene clave primaria
- No están bien definidas sino es con relación a otra entidad (entidad fuerte)
  - la relación tiene que ser muchos (lado de la entidad débil) a uno
- Existen unos atributos discriminantes o clave parcial que diferencia todas las entidades débiles relacionadas a la misma entidad fuerte
- La clave primaria de la entidad débil se forma por la unión de la clave primaria de la entidad fuerte asociada y los atributos discriminantes
- Se representan mediante líneas dobles
  - La clave parcial se identifica con una línea de puntos





## MODELO RELACIONAL

- Modelo lógico propuesto por E.F. Codd en 1970
- Todas las bases de datos que se pueden modelar siguiendo el modelo entidad-relación pueden implementarse siguiendo el modelo relacional
- Organiza y representa la información como una **colección de relaciones**
- Definición formal del concepto matemático de relación:  
Una relación sobre los conjuntos  $D_1, D_2, \dots, D_n$  es un subconjunto del producto cartesiano de los mismos, entendiéndose que el producto cartesiano  $(D_1 \times D_2 \times \dots \times D_n)$  es un nuevo conjunto formado por todas las tuplas ordenadas posibles  $(d_1, d_2, \dots, d_n)$  tales que  $d_i \in D_i$   
  
Ejemplo: dados los dominios:  
 $\text{nombre-cliente} = \{\text{Jones, Smith, Curry, Lindsay}\}$   
 $\text{dirección-cliente} = \{\text{Main, North, Park}\}$   
 $\text{ciudad-cliente} = \{\text{Harrison, Rye, Pittsfield}\}$   
  
Entonces  $r = \{$      $(\text{Jones, Main, Harrison}),$   
                          $(\text{Smith, North, Rye}),$   
                          $(\text{Curry, North, Rye}),$   
                          $(\text{Lindsay, Park, Pittsfield})\}$   
  
es una relación sobre  $\text{nombre-cliente} \times \text{dirección-cliente} \times \text{ciudad-cliente}$



## MODELO RELACIONAL

- Informalmente, una relación puede considerarse una tabla (conceptualmente)
  - Una base de datos que se ajusta al modelo relacional puede representarse como un conjunto de tablas
  - Convertir un diagrama E-R a tablas es el primer paso para obtener una base de datos relacional
  - Normalmente cada entidad y cada relación muchos a muchos da lugar a una tabla
  - Cada tabla tienen un conjunto de columnas que suelen corresponderse con los atributos



## INSTANCIA DE UNA RELACIÓN

- Valores actuales de la relación  $\equiv$  contenido de la tabla
- Un elemento  $t$  de  $r$  es una *tupla*  $\equiv$  fila de la tabla

<i>nombre-cliente</i>	<i>Direccion-cliente</i>	<i>Ciudad-cliente</i>
Jones	Main	Harrison
Smith	North	Rye
Curry	North	Rye
Lindsay	Park	Pittsfield

*cliente*

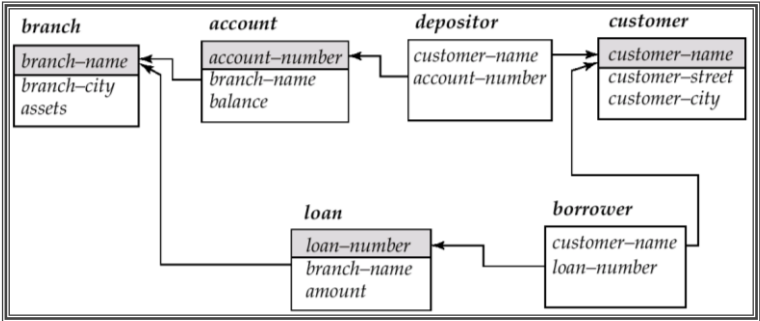


## ESQUEMA DE LA BASE DE DATOS

- Esquema BB.DD. = diseño lógico (descripción de la base de datos)
- Esquema BB.DD  $\neq$  datos
  - Esquema para la relación cliente:  
Esquema-cliente = (*nombre-cliente*, *direccion-cliente*, *ciudad-cliente*)
  - Cliente es una tabla/relación sobre el esquema Esquema-cliente:  
cliente(*esquema-cliente*)
- convenciones:
  - Nombres de los esquemas en mayúsculas
  - Nombres de las instancias/relaciones en minúsculas



# ESQUEMA DE LA BASE DE DATOS DEL EJEMPLO DEL BANCO



- Notación para los esquemas de las relaciones:
  - account(account-number, branch-name↑, balance)

