

Modelos de Base de Datos

Diseño y Administración de Base de Datos - UTN

Prof. Lic Pablo Ezequiel Fino

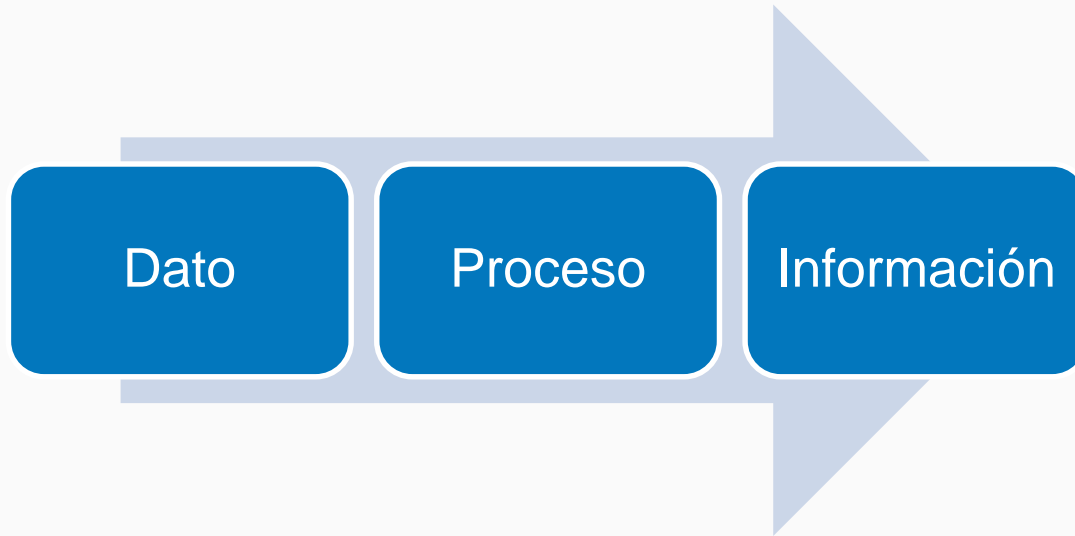
Prof. Lic Verónica Lourdes Tomich

Unidad 1: Modelos de Base de Datos

- Modelo Relacional
- Normalización de Base de Datos
- Modelo Cliente/Servidor

Introducción a las Bases de Datos

Las etapas del diseño de un programa consisten en lo siguiente:



Introducción a las Bases de Datos

DATOS

- son los elementos básicos o fundamentales con los que se cuenta al momento de inicio de la tarea a realizar

PROCESO

- es el conjunto de acciones que se realizan en un orden específico con un principio y fin bien definidos

INFORMACION

- es el resultado que se obtiene como consecuencia de procesar los datos

Organización en archivos

- Un archivo esta compuesto por registros y cada registro esta compuesto por un conjunto de campos.
- Los inconvenientes en el uso de archivos pueden ser:

Organización en archivos

- **Redundancia de datos:**

- Si los sistemas de una empresa están soportados sobre archivos es muy probable que el área de recursos humanos posea un archivo con los datos más relevantes de sus empleados (legajo, nombre, dni, dirección, etc.) a su vez el departamento de ventas también podría tener un archivo propio el cual contendría los datos de las ventas realizadas por cada empleado, con el siguiente diseño (legajo, nombre, nroventa, totven, comisión).

Organización en archivos

- **Redundancia de datos:**
 - Queda claro que cada departamento deberá cargar los datos de los empleados en sus archivos, por lo tanto esos datos estarán cargados dos veces, a esto se lo denomina **redundancia de datos**.

Organización en archivos

- **Consistencia e integridad de datos:**
 - Un empleado tiene mal grabado su apellido, si el departamento de RRHH realiza la actualización del dato y el de ventas no, la información sobre el empleado queda inconsistente y como consecuencia se pierde la integridad de los datos, ya que para un mismo legajo se tienen dos personas distintas.

Organización en archivos

- **Consistencia e integridad de datos:**

- Además, sucede algo peor, como RRHH realizó la actualización del apellido y ventas no, las ventas registradas no pertenecen a ningún empleado actual, y podría haber errores en la liquidación de comisiones.

Organización en archivos

- **Modificar la estructura lógica:**
 - Si se desea agregar un nuevo campo a la estructura de cualquiera de los archivos, implica modificar el código del programa para que acepte el nuevo diseño.

Definición de Base de Datos

Una Base de Datos es un sistema formado por un **conjunto de datos** almacenados en discos que permiten el acceso directo a ellos y un **conjunto de programas** que manipulan ese conjunto de datos.

En una Base de Datos, existe una visión lógica de los datos que no tiene porqué ser la misma que la visión física, es decir, existe una independencia de los datos con respecto a los datos que los utilizan.

Definición de Base de Datos

- Un conjunto de datos interrelacionados
- Almacenados sin redundancias
- Sirven a una o más aplicaciones
- Son independientes de los programas que los manejan
- Contiene la definición de los datos

Definición de Base de Datos

- Refleja las relaciones y restricciones de los datos
- Los datos son coherentes
- Los datos son versátiles
- Los datos son accedidos por más de un usuario simultáneamente
- Permiten el acceso a datos del pasado
- Son seguros

Componentes de una Base de Datos

Objetos del usuario:

- Los objetos que un usuario de BD puede definir son: Tablas, Vistas, Funciones, Desencadenadores, Procedimientos Almacenados, Índices, Relaciones, Restricciones, Tipos de Datos.

Objetos del Sistema

- El sistema mantiene información necesaria para la gestión de la BD.

Sistema de Gestión de Bases de Datos (DBMS)

Un Sistema de Gestión de Base de Datos es un conjunto de programas que permiten administrar y gestionar la información de una Base de Datos, y sirve de interfaz entre los **datos**, el **usuario** y las **aplicaciones** que la utilizan.

Sistema de Gestión de Bases de Datos (DBMS)

Proporcionan a los usuarios de las Bases de Datos herramientas para realizar las siguientes tareas:

- 1.-Definición de los datos en los distintos niveles de abstracción
- 2.-Manipulación de los datos es decir, inserción, modificación, borrado y consulta
- 3.-Mantenimiento de la integridad de la Base de Datos
- 4.-Control de la seguridad

Objetivos de un DBMS:

- **Abstracción de Datos**
- **Independencia de Datos**
- **Consistencia**
- **Seguridad**
- **Integridad**
- **Respaldo**
- **Control de la concurrencia**
- **Manejo de Transacciones**
- **Tiempo de respuesta**

Objetivos de un DBMS:



Objetivos de un DBMS:

Nivel Físico o Interno:

- es el nivel mas bajo de abstracción, describe cómo se almacenan realmente los datos, y a este nivel tendremos cosas como: los archivos que contienen la información, lugar donde se encuentra el disco, forma de acceso a esos archivos.

Nivel conceptual:

- este nivel describe qué datos son realmente almacenados en la Base de Datos y las relaciones que existen entre los datos. Este nivel lo usan los DBA.

Nivel de visión o externo:

- describe un número de vistas, cada vista describe una parte de la Base de Datos completa que le interesa a un grupo particular de usuarios.

Objetivos de un DBMS:

Independencia Física:

- Es la capacidad de modificar el esquema físico sin provocar que se vuelvan a escribir los programas de aplicación. En este punto se puede encontrar cambios al tamaño de los bloques, longitud de los registros, creación y eliminación de índices, etc.

Independencia lógica:

- Es la capacidad de modificar el esquema conceptual sin provocar que se vuelvan a escribir los programas de aplicación. Esto es necesario siempre que se altere la estructura lógica de la BD. Los cambios pueden realizarse en los campos (nombre, tipo, etc.) o en los registros (cambio de nombre, introducir nuevos campos, borrado de los mismos, etc.).

Objetivos de un DBMS:

Consistencia:

- En aquellos casos en los que no se ha logrado eliminar la redundancia, será necesario vigilar que aquella información que aparece repetida se actualice de forma coherente, es decir, que todos los datos repetidos se actualicen de forma simultánea. En los SGBD existen herramientas que facilitan la programación de este tipo de condiciones.

Objetivos de un DBMS:

Seguridad:

- La información almacenada en una base de datos puede llegar a tener un gran valor. Los SGBD deben garantizar que esta información se encuentra segura frente a usuarios malintencionados, que intenten leer información privilegiada; frente a ataques que deseen manipular o destruir la información; o simplemente ante las torpezas de algún usuario autorizado pero descuidado. Normalmente, los SGBD disponen de un complejo sistema de permisos a usuarios y grupos de usuarios, que permiten otorgar diversas categorías de permisos.

Objetivos de un DBMS:

Integridad

- Se trata de adoptar las medidas necesarias para garantizar la validez de los datos almacenados. Es decir, se trata de proteger los datos ante fallos de hardware, datos introducidos por usuarios descuidados, o cualquier otra circunstancia capaz de corromper la información almacenada. Los SGBD proveen mecanismos para garantizar la recuperación de la base de datos hasta un estado consistente conocido en forma automática.

Objetivos de un DBMS:

Respaldo

- Los SGBD deben proporcionar una forma eficiente de realizar copias de respaldo de la información almacenada en ellos, y de restaurar a partir de estas copias los datos que se hayan podido perder.

Objetivos de un DBMS:

Control de la concurrencia

- En la mayoría de entornos, lo más habitual es que sean muchas las personas que acceden a una base de datos, bien para recuperar información, bien para almacenarla. Y es también frecuente que dichos accesos se realicen de forma simultánea. Así pues, un SGBD debe controlar este acceso concurrente a la información, que podría derivar en inconsistencias.

Objetivos de un DBMS:

Manejo de Transacciones

- Una Transacción es un programa que se ejecuta como una sola operación. Esto quiere decir que el estado luego de una ejecución en la que se produce una falla es el mismo que se obtendría si el programa no se hubiera ejecutado. Los SGBD proveen mecanismos para programar las modificaciones de los datos de una forma mucho más simple que si no se dispusiera de ellos.

Objetivos de un DBMS:

Tiempo de respuesta

- Lógicamente, es deseable minimizar el tiempo que el SGBD tarda en darnos la información solicitada y en almacenar los cambios realizados.

Usuarios de Bases de Datos:

Un objetivo primordial de un sistema de BD es proporcionar un entorno para recuperar información de y almacenar nueva información en la BD.

- **Administrador de una BD (DBA).**
- **Programadores de aplicaciones.**
- **Usuarios Finales.**

Lenguajes:

Un objetivo primordial de un sistema de BD es proporcionar un entorno para recuperar información de y almacenar nueva información en la BD.

- **Administrador de una BD (DBA).**
- **Programadores de aplicaciones.**
- **Usuarios Finales.**

Lenguajes:

DDL

- lenguaje de definición de datos y se utiliza para definir el esquema conceptual de la Base de Datos.

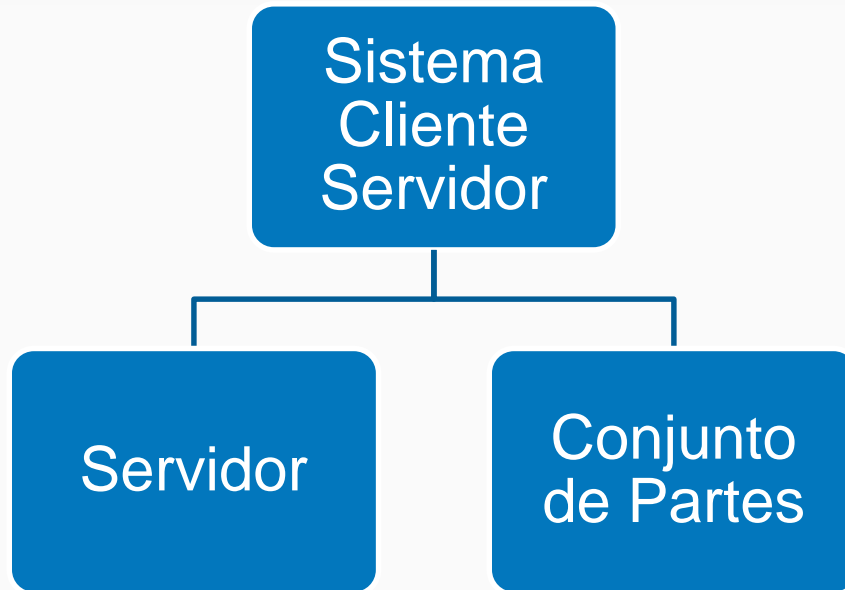
DML

- lenguaje de manipulación de datos, así podemos insertar datos, modificar los existentes, eliminar datos y hacer consultas.

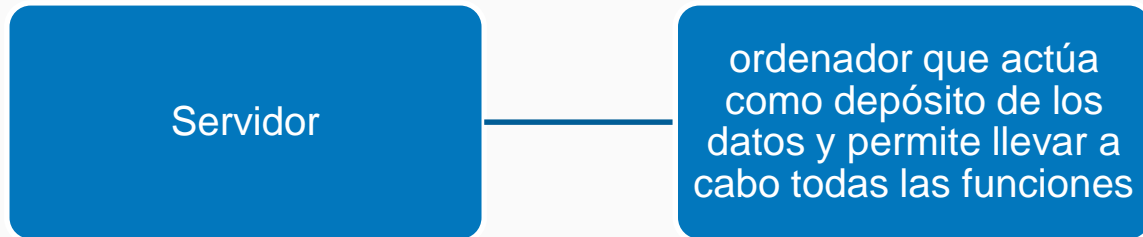
DCL

- lenguaje del control de datos, se utiliza para controlar el acceso a la información de la Base de Datos teniendo privilegios y tipo de acceso, así como para el control de la seguridad de los datos.

Arquitectura de acceso a Bases de Datos



Arquitectura de acceso a Bases de Datos:



Arquitectura de acceso a Bases de Datos:

Software Adecuado:

Software de gestión de datos que se encarga de la manipulación de datos y suele residir en el servidor.

Software de interacción con el usuario que se encarga de presentar los resultados que suelen estar en el cliente.

Software de desarrollo que se usa para realizar las aplicaciones, suele residir en puestos cliente pero de trabajadores del departamento de desarrollo.

Modelos de Base de Datos:

Un objetivo representar mediante abstracciones del mundo real toda la información necesaria para el cumplimiento de los fines.

Modelos de Base de Datos:

La estructura de una Base de Datos es necesario definir el concepto de **modelo de datos**, una colección de **herramientas conceptuales** para describir datos, **relaciones** entre ellos, **semántica** asociada a los datos y **restricciones** de consistencia.

Modelos de Base de Datos:

Modelos conceptuales, o **lógicos basados en objetos**, es la representación de la realidad no comprometida con ningún entorno informático:

- Modelo entidad-Relación
- Modelo orientado a objetos
- Modelo binario
- Modelo semántica de datos

Modelos de Base de Datos:

Modelos **lógicos basados en registros**, determinan unos criterios de almacenamiento y de operaciones de manipulación de los datos dentro de un entorno informático. Dentro de estos se encuentra:

- Modelo jerárquico (IMS)
- Modelo de red (Codasyl)
- Modelo relacional (Oracle, Microsoft SQL Server)

Modelos de Base de Datos:

Modelos Lógicos basados en Objetos:

Modelo Entidad - Relación

- El modelo de datos Entidad – Relación se basa en una percepción del mundo real, que consiste en un conjunto de objetos básicos llamados Entidades y de Relaciones entre ellos. Se emplea para interpretar, especificar y documentar los requerimientos para los sistemas de BD.

Modelos de Base de Datos:

Modelos Lógicos basados en Objetos:

Modelo Entidad - Relación

- Por lo tanto un modelo Entidad – Relación es un método de representación abstracta del mundo real centrado en las restricciones o propiedades lógicas de una BD.

Modelos de Base de Datos:

Modelos Lógicos basados en Objetos:

Modelo Entidad – Relación

ENTIDADES

- Una entidad es un objeto, real o abstracto, acerca del cual se recoge información de interés para la base de datos.

Modelos de Base de Datos:

Modelos Lógicos basados en Objetos:

Modelo Entidad – Relación

Entidades fuertes: tienen existencia por sí mismas. (Alumnos, Empleados, Dpto.)

Entidades débiles: dependen de otra entidad para su existencia (Hijos de Empleados)

Modelos de Base de Datos:

Modelos Lógicos basados en Objetos:

Modelo Entidad – Relación

ENTIDADES

- Cada entidad tiene propiedades particulares llamadas **atributos**.
- Los atributos describen las características de una entidad.

Modelos de Base de Datos:

Modelos Lógicos basados en Objetos:

Modelo Entidad – Relación

ENTIDADES – ATRIBUTOS

Entidad: Clientes

Atributos: legajo, nombre, domicilio, etc.

Modelos de Base de Datos:

Modelos Lógicos basados en Objetos:

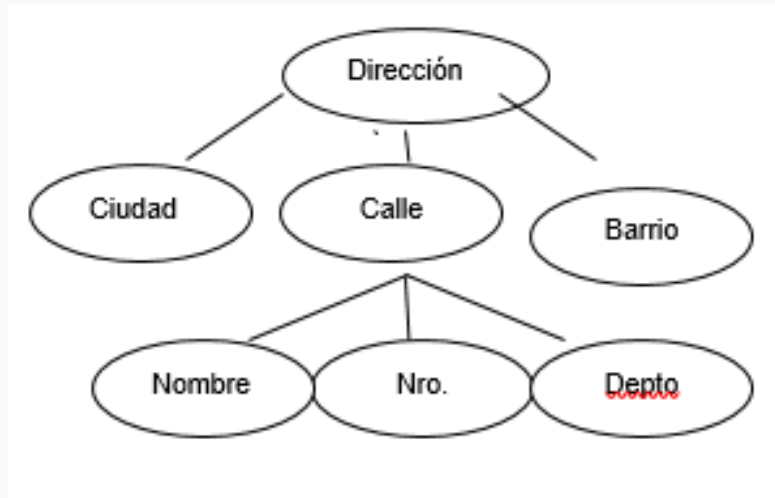
Modelo Entidad – Relación

Atómico: Son los atributos que no son divisibles. Ejemplo: Nombre, Ciudad.

Compuestos: Son los atributos que pueden dividirse en pequeñas sub partes. Los atributos compuestos pueden formar una jerarquía de atributos.

Modelos de Base de Datos:

Modelos Lógicos basados en Objetos:



Modelos de Base de Datos:

Modelos Lógicos basados en Objetos:

Atributo con simple valor: Cuando un atributo tiene un simple valor para una identidad particular.

Atributo Multivalor: Cuando un atributo tiene una serie de valores para identificarse.

Atributos Derivados: Cuando los valores de un atributo son afines y el valor para este tipo de atributo se puede derivar de los valores de otros atributos.

Atributo Clave: Una entidad usualmente tiene un atributo sobre el cual los valores son distintos para cada entidad individual, ésta nos permite identificar de forma única a una entidad en un conjunto de entidades.

Atributos Nulos: Se usa cuando una entidad no tiene valor para un atributo o que el valor es desconocido.

Modelos de Base de Datos:

Modelos Lógicos basados en Objetos (Ejemplos):

Atributo con simple valor: una persona que tiene un valor por su fecha de nacimiento y la fecha de nacimiento es un simple valor de la persona.

Atributo Multi valor: el atributo teléfonos de un cliente.

Atributos Derivados: la edad y fecha de nacimiento de una persona; si conocemos la fecha de nacimiento, podemos determinar su edad, en este caso se dice que la edad es derivada de la fecha de nacimiento.

Atributo Clave: en la entidad Clientes el atributo Documento puede ser un atributo clave. No necesariamente el atributo clave debe ser un solo atributo, hay casos en que varios atributos forman una llave. Por ejemplo: tipo más número de factura.

Atributos Nulos: un atributo que no es campo requerido y se inicializa como null

Modelos de Base de Datos:

Modelos Lógicos basados en Objetos:

Modelo Entidad – Relación

Dominio:

Un dominio es un conjunto posible de valores que puede tomar un atributo. Por ejemplo: el rango de edad de los empleados de una empresa es de 25 a 50 años, el estado civil, etc.

Modelos de Base de Datos:

Modelos Lógicos basados en Objetos:

Modelo Entidad – Relación

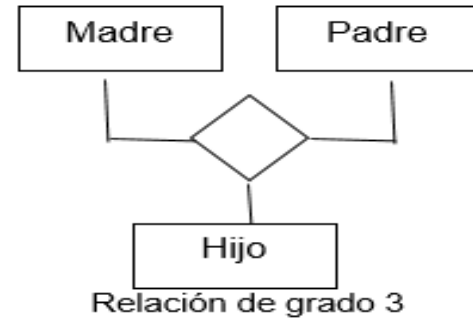
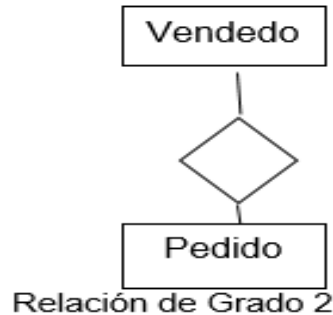
RELACIONES

Una relación es cualquier asociación R que pueda establecerse entre entidades.

Una relación puede incluir muchas entidades, la cantidad de entidades en una relación es el GRADO de la relación.

Modelos de Base de Datos:

Modelos Lógicos basados en Objetos:



Modelos de Base de Datos:

Modelos Lógicos basados en Objetos:

Relaciones Unitarias o Recursivas

- Se establece entre entidades de la misma clase.



Modelos de Base de Datos:

Modelos Lógicos basados en Objetos:

Relaciones N-arias

- La relación se establece entre N clases de entidades, siendo $N > 2$. Este tipo de relaciones tiene limitaciones, por lo que cada una de ellas suele ser sustituida por un conjunto de relaciones binarias.

Modelos de Base de Datos:

Modelos Lógicos basados en Objetos:

Propiedades de las relaciones

Roles de una relación: son las funciones que desempeñan cada una de las entidades asociadas. En toda relación existen dos roles diferentes correspondientes a las entidades relacionadas. Suponiendo que **A** representa la entidad formada por todos los profesores de un centro de estudios, y **B** la formada por todos los alumnos de dicho centro, entre las entidades podrían establecerse varias relaciones.

Modelos de Base de Datos:

Modelos Lógicos basados en Objetos:

Propiedades de las relaciones

Roles de una relación:

- Dar clase a / Recibir clase de
- Es tutor de / es dirigido por
- Es director de / es dirigido por

Modelos de Base de Datos:

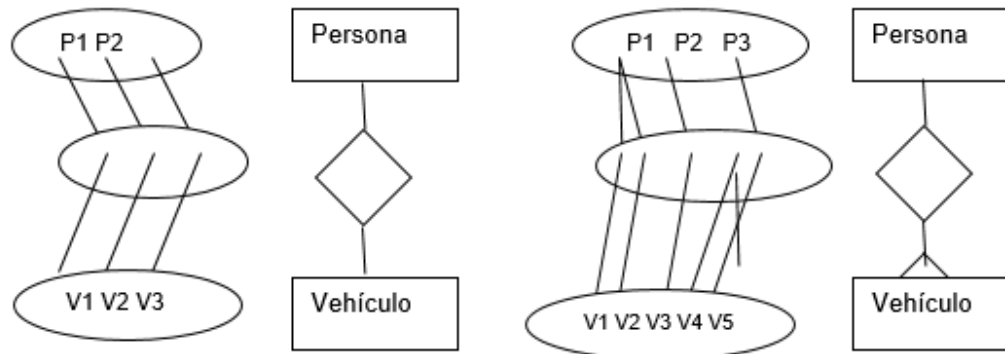
Modelos Lógicos basados en Objetos:

Relaciones N-arias

Cardinalidad de una Relación: Expresa el número de ocurrencias de una entidad que pueden asociarse a una ocurrencia de la otra entidad.

Modelos de Base de Datos:

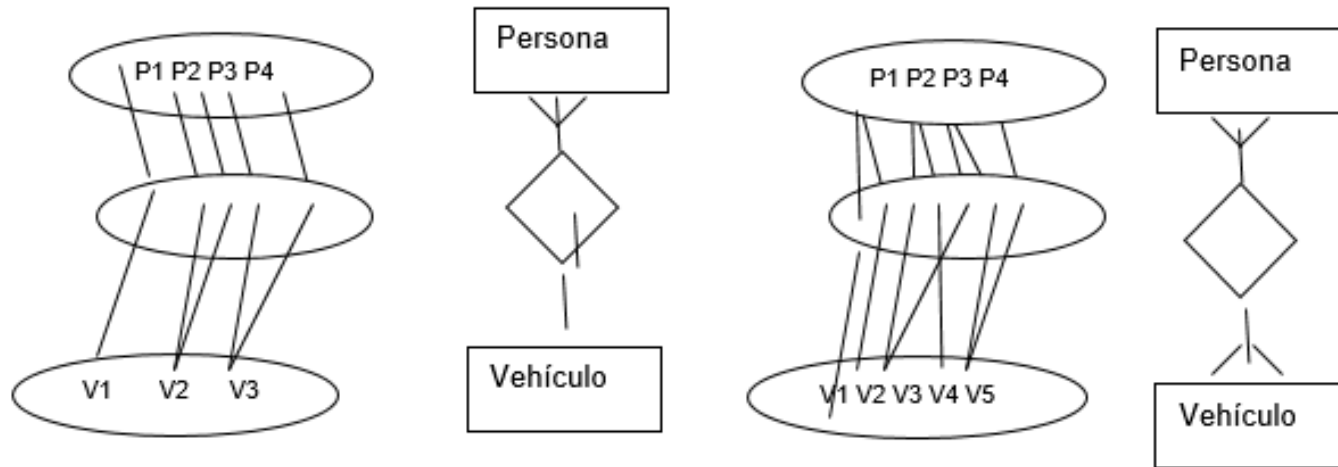
Modelos Lógicos basados en Objetos:



(A) Relación 1:1	(B) Relación 1:N
Cada persona maneja un vehículo, cada vehículo es manejado por una persona	Una persona maneja varios vehículos, un vehículo es manejado por una sola persona.

Modelos de Base de Datos:

Modelos Lógicos basados en Objetos:



(C) Relación N:1

Un vehículo puede ser manejado por algunas personas. Una persona maneja un vehículo.

(D) Relación N:M

Una persona puede manejar un número de vehículos, un vehículo puede ser manejado por algunas personas.

Modelos de Base de Datos:

Modelos Lógicos basados en Objetos:

Participación de un conjunto de entidades:

Especifica si todas las ocurrencias de una entidad participan o no en la relación establecida con ocurrencias de otras entidades, es decir, indica si todas las ocurrencias de una entidad se relacionan necesariamente con ocurrencias de otra entidad asociada.

Modelos de Base de Datos:

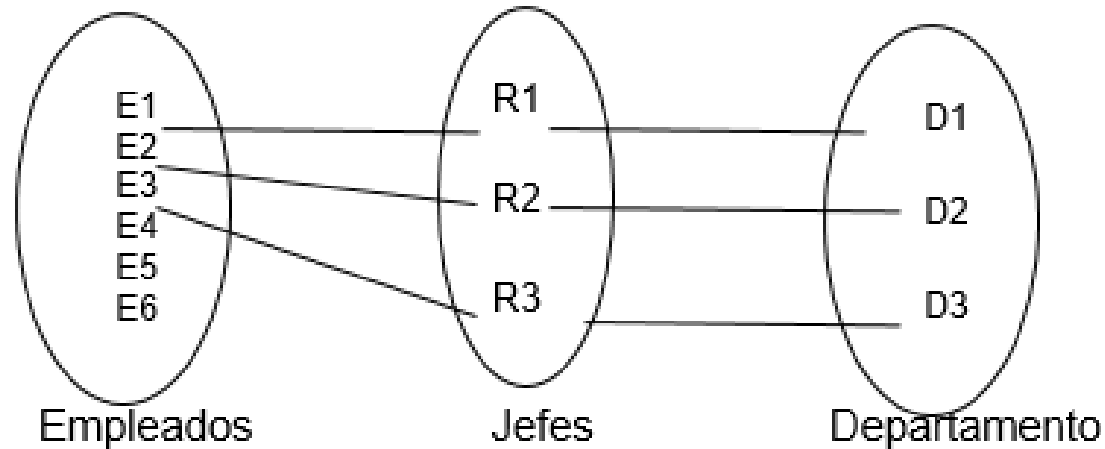
Modelos Lógicos basados en Objetos:

- Una participación es **total**, si cada ocurrencia de un conjunto participa en al menos una relación.
- Una participación es **parcial**, si solo alguna ocurrencia participa en las relaciones.

Modelos de Base de Datos:

Modelos Lógicos basados en Objetos:

Ejemplo:



Participación parcial del empleado y participación total del departamento.

Modelos de Base de Datos:

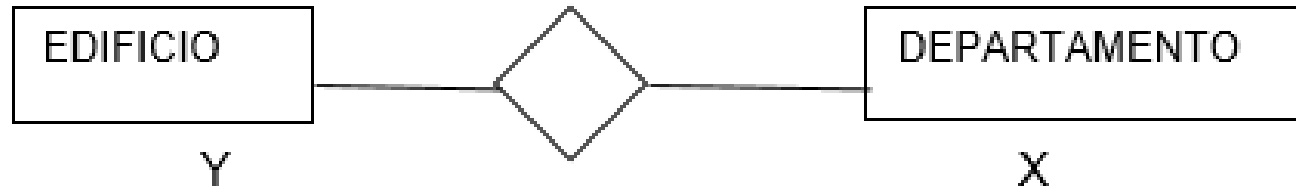
Modelos Lógicos basados en Objetos:

- **Dependencia de existencia:** si la existencia de una entidad X depende de la existencia de una entidad Y, entonces se dice que X es dependiente por existencia de Y. Operativamente eso significa que si se suprime Y, también se suprime X. La entidad Y se dice que es una entidad dominante y X una entidad subordinada.

Modelos de Base de Datos:

Modelos Lógicos basados en Objetos:

Ejemplo:



Modelos de Base de Datos:

Modelos Lógicos basados en Objetos:

Superclaves: conjunto de uno o más atributos cuyos valores identifican unívocamente cada ocurrencia de la entidad.

Clave Candidata: conjunto de uno más atributos cuyos valores identifican unívocamente cada ocurrencia de la entidad, sin que ningún subconjunto de ellos pueda realizar esta misma función. Una clave candidata es una superclave mínima. Puede haber varias claves candidatas.

Clave Primaria: es la clave candidata de una entidad o relación que se elige como identificador univoco de sus ocurrencias.

Modelos de Base de Datos:

DIAGRAMA ENTIDAD-RELACION

1.- Rectángulos, que representan conjuntos de entidades.

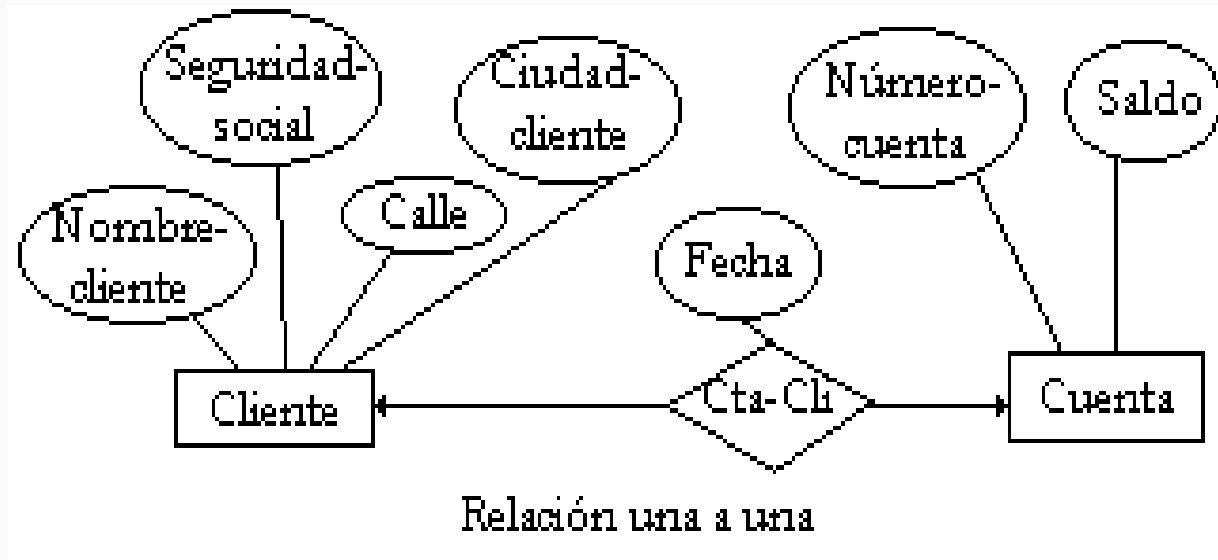
2.- Elipses, que representan atributos.

3.- Rombos, que representan conjuntos de relaciones.

4.- Líneas, que enlazan atributos a conjuntos de entidades y conjuntos de entidades a conjuntos de relaciones. Para distinguir entre las diferentes cardinalidades, dibujaremos líneas, con o sin dirección entre los conjuntos de relaciones y el conjunto de entidades en cuestión. Una línea con dirección (→) marcará la cardinalidad una, mientras que una línea sin flecha (--) indicará la cardinalidad muchas.

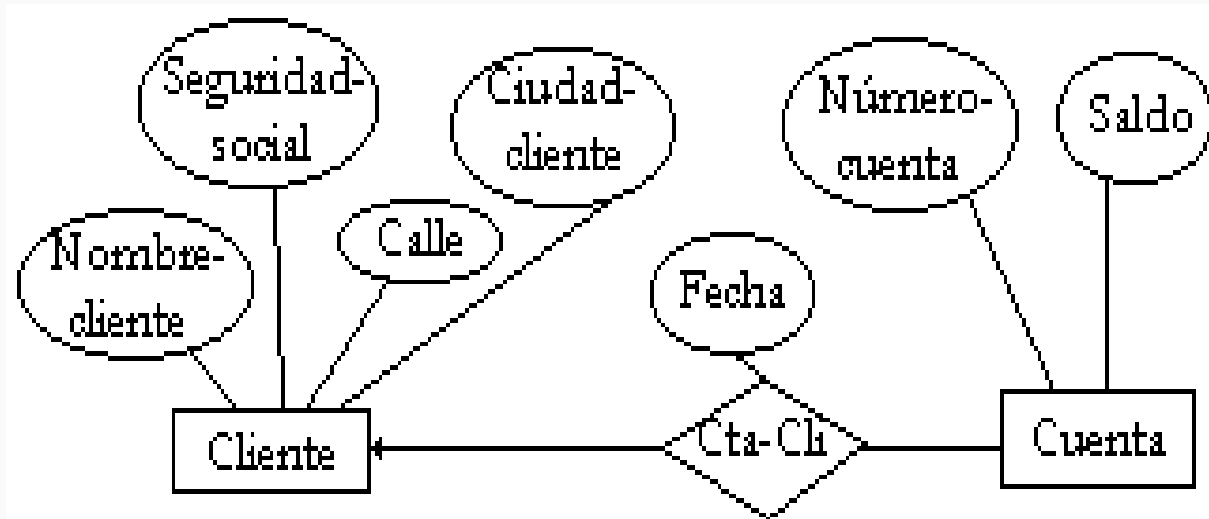
Modelos de Base de Datos:

DIAGRAMA ENTIDAD-RELACION



Modelos de Base de Datos:

DIAGRAMA ENTIDAD-RELACION



Relación una a muchas

Modelos de Base de Datos:

DIAGRAMA ENTIDAD-RELACION:

- En un hospital se tiene un registro de pacientes, un registro de personal y uno de salas con funcionarios que trabajan en esas salas y con pacientes internados en esas salas.
- Del personal nos interesa el número de empleado, el nombre, la dirección y el teléfono.
- Sabemos que dos empleados no tienen el mismo número.

Modelos de Base de Datos:

DIAGRAMA ENTIDAD-RELACION:

- De los pacientes nos interesa el número de registro (le es asignado cuando ingresa) y el nombre mientras que de las salas nos interesa el nombre y la cantidad de camas que tiene. También se sabe que un empleado trabaja en una única sala y que en una sala trabajan varios empleados. Lo mismo ocurre con los pacientes.

Modelos de Base de Datos:

DIAGRAMA ENTIDAD-RELACION:

1. identificar los objetos del problema.
2. identificar las relaciones entre los objetos.
3. representar las propiedades que interesan de los objetos.
4. determinar las restricciones que se desea imponer.

Modelos de Base de Datos:

DIAGRAMA ENTIDAD-RELACION: SOLUCION

- Conjunto de Objetos: Pacientes, Sala, Personal.
- Relaciones entre esos Conjuntos: Los pacientes están internados en las salas y el personal trabaja en las salas.
- Características que interesan de los objetos:
 - Personal: nro. de funcionario, nombre, dirección y teléfono.
 - Pacientes: nro. de registro, nombre.
 - Salas: nombre, cantidad de camas.

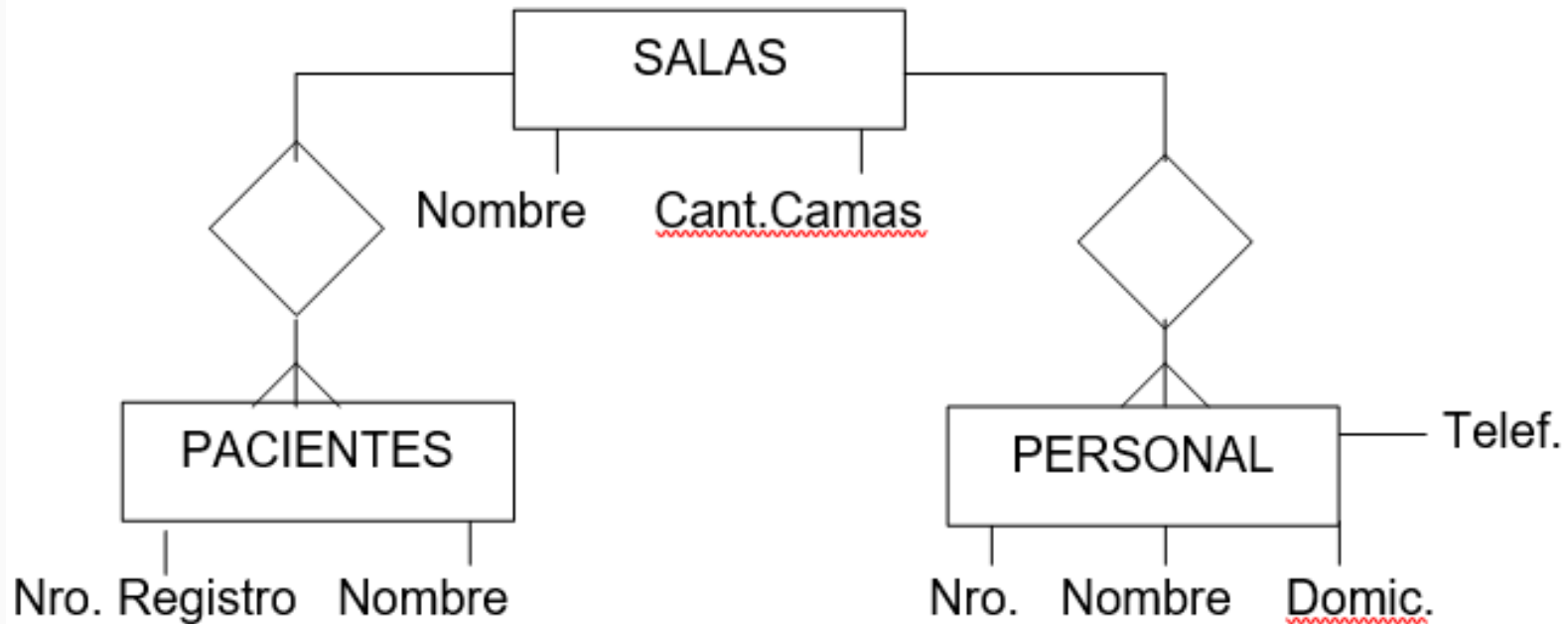
Modelos de Base de Datos:

DIAGRAMA ENTIDAD-RELACION: SOLUCION

- Restricciones: un empleado trabaja en una única sala y en una sala trabajan varios empleados. Un paciente está internado en una sola sala pero en una sala hay varios pacientes.

Modelos de Base de Datos:

DIAGRAMA ENTIDAD-RELACION



Modelos de Base de Datos:

Reducción de los DER a Tablas:

- Una base de datos que se ajusta a un DER puede representarse por medio de una colección de tablas.
- Para cada conjunto de entidades, y para cada conjunto de relaciones en la base de datos, existe una tabla única a la que se asigna el nombre de la entidad o de la relación según corresponda. Cada tabla tiene un número de columnas que a su vez tienen nombres únicos.
- En esta etapa transformaremos el esquema conceptual (modelo de entidad-relación) a un esquema relacional.

Modelos de Base de Datos:

Reducción de los DER a Tablas, PRINCIPIOS:

- Todo tipo de entidad se convierte en una tabla.
- Los atributos en columnas de la tabla
- El identificador único se convierte en clave primaria
- Todo tipo de relación N:N se transforma en una tabla.
- Todo tipo de relación 1: N se traduce en el fenómeno de propagación de la clave o bien se crea una nueva relación.

Modelos de Base de Datos:

Reducción de los DER a Tablas, PRINCIPIOS:

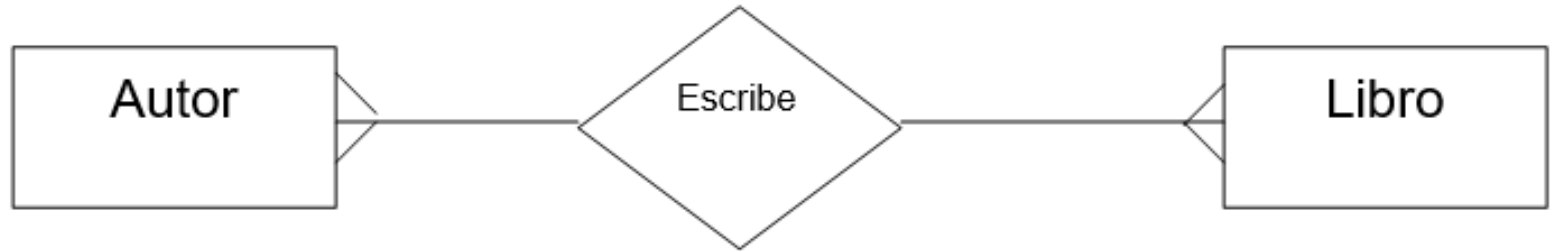
Relación N:N

- Un tipo de relación N:N se transforma en una relación que tendrá como clave primaria la concatenación de los identificadores de cada entidad que asocia. Cada uno de los atributos que forman la clave primaria de una tabla derivada de una relación N:N son claves foráneas respecto de cada una de las tablas de las entidades que se relacionan.

Modelos de Base de Datos:

Reducción de los DER a Tablas, PRINCIPIOS:

Relación N:N



AUTOR (cod_ autor, Nombre, Apellido)

LIBRO (cod_ libro, titulo)

ESCRIBE (cod_ autor, cod_ libro)

Modelos de Base de Datos:

Reducción de los DER a Tablas, PRINCIPIOS:

Relación 1:N

- Existen casos en que es necesario transformar la relación en una tabla, como si se tratara de una relación N:N, siendo conveniente en los siguientes casos:
 - el número de ocurrencias de la entidad que propaga la clave (empleado) es muy pequeño, evitando los valores nulos.

Modelos de Base de Datos:

Reducción de los DER a Tablas, PRINCIPIOS:

Relación 1:N

Empleados

Nombre	Apellido	ND
Juan	López	1
Martín	Jaime	1
Pablo	Solís	<u>Null</u>
Lucas	Estévez	2

Departamentos

Nro. Dep.	Nombre
1	Cómputos
2	Contable

Modelos de Base de Datos:

Reducción de los DER a Tablas, PRINCIPIOS:

Relación 1:N

- Para evitar los valores nulos de la tabla empleados se recomienda crear una nueva tabla de la relación.
- Cuando se prevé que en el futuro dicha relación se convierta en N:N.
- Cuando la relación tiene atributos propios.

Modelos de Base de Datos: Reducción de los DER a Tablas, PRINCIPIOS: Relación 1:1

- Son casos en que se puede crear una tabla o bien propagar la clave.
- Si ambas entidades tienen una participación parcial en la relación es recomendable crear una nueva tabla de la relación para evitar los valores nulos.

Modelos de Base de Datos:

Reducción de los DER a Tablas, PRINCIPIOS:

Relación 1:1



MUJER (dni_mujer, nombre,...)
HOMBRE (dni_hombre, nombre,...)
MATRIMONIO (dni_mujer, dni_hombre)