

Módulo 1



Sistemas de Información y el Modelo de Datos Relacional



Lic. Lidia Lorelí Zamora Nunfio
znlorel@yahoo.com.mx



Contenido



Sistemas de Información y el Modelo de Datos Relacional (15 hrs.)

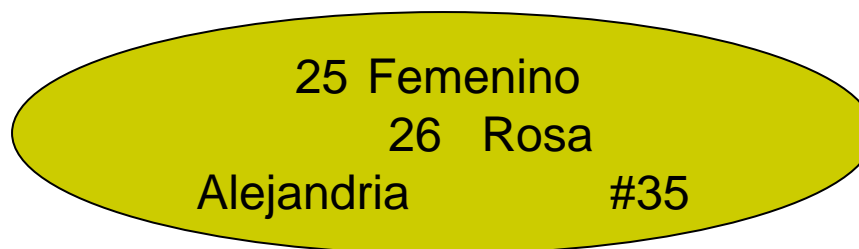
- ¿Qué es un dato?
- ¿Qué es la información?
- Análisis de la información
- Definición de una base de datos
- Características de las bases de datos
- Modelo de datos
- Modelo Relacional
- La independencia de datos y la integridad referencial
- Modelo Entidad-Relación
- Normalización
- Modelado de datos. Casos Prácticos
- Herramientas CASE



1.1. ¿Qué es un dato?



- Un dato es la unidad mínima de información, hechos sin evaluar ó un valor sin significado.

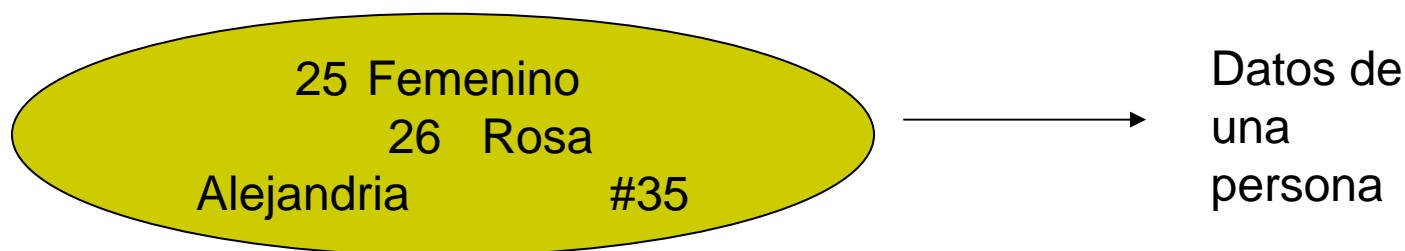




1.2. ¿ Qué es la información?



- La información se obtiene asociando los hechos en un contexto determinado, es decir, la adición o el procesamiento de los datos, proporcionan el conocimiento o entendimiento de ciertos factores.





1.2. ¿ Qué es la información?



- Conjunto de datos interrelacionados entre sí .
- En un sentido más amplio, un **sistema** es un conjunto de componentes que interactúan entre sí para lograr un objetivo común para la toma de decisiones



1.3. Análisis de la información



- El análisis de la información es un modelo de datos que consiste en la representación conceptual de la problemática que se desea resolver y cuya característica primordial es la claridad de su contenido.
- Algo importante de mencionar es que el costo de producción de la información es tangible y se puede medir gracias a los dispositivos y medios utilizados, pero la información es conceptual por naturaleza y no tiene características tangibles salvo representaciones simbólicas.



Características del valor de la información.



- **Accesible:** es la facilidad y rapidez con que se obtiene la información resultante.
- **Clara:** se refiere a la integridad y entendimiento de la información sin ambigüedades.
- **Precisa:** que sea lo más exacta posible.
- **Propia:** debe de haber relación entre el resultado y lo solicitado por el usuario.
- **Oportuna:** menor duración del ciclo (entrada, procesamiento y entrega al usuario).
- **Flexible:** adaptabilidad de la información a la toma de decisiones.
- **Verificable:** que se pueda examinar la información.
- **Imparcial:** no se puede alterar o modificar la información (sólo por el dueño).
- **Cuantificable:** todo dato procesado – produce información.



1.4. Definición de una Base de Datos



- Una base de datos es un conjunto de datos relacionados entre sí con un objetivo común almacenados en algún lugar, que contemplan persistencia de la información y que cuyos datos están organizados.

¿Ejemplos de bases de datos?



1.4. Definición de una Base de Datos Relacional



- Es una colección de relaciones. El contenido de una base de datos engloba a la información concerniente de una organización, de tal manera que los datos estén disponibles para los usuarios en tiempo real y son compatibles con usuarios concurrentes, una finalidad de la base de datos es eliminar la redundancia o al menos minimizarla.
- Los cuatro componentes principales de un sistema de base de datos son el hardware, el software, los usuarios y los datos a manejar.



1.5. Arquitectura para las bases de datos



Hace unos años atrás, las bases de datos eran el resultado de una compleja programación y de complicados mecanismos de almacenamiento. Con la popularización de la informática, la aparición de aplicaciones específicas también trajo con ella la disponibilidad de herramientas de gestión de datos que dieron lugar a los denominados Sistemas Manejadores de Bases de Datos, identificados por sus siglas SGBD por su acrónimo en inglés DBMS (Data Base Management Systems).



1.5. Arquitectura para las bases de datos



Para dotar a los usuarios de cierta seguridad en el intercambio de datos entre diferentes sistemas y en el diseño de archivos y bases de datos, fue necesario normalizar los esquemas que guiaban la creación de las bases de datos.

Un sistema de base de datos se encuentra dividido en módulos cada uno de los cuales controla una parte de la responsabilidad total de sistema. En la mayoría de los casos, el sistema operativo proporciona únicamente los servicios más básicos y el sistema de la base de datos debe partir de esa base y controlar además el manejo correcto de los datos.



1.5. Arquitectura para las bases de datos



Los componentes de un sistema de base de datos, son los siguientes:

- **Hardware.** En donde se encuentra toda la infraestructura física de la base de datos.
- **Software.** Todos los programas aplicativos que se involucran para generar un ambiente idóneo.
- **Datos.** La información en sí.
- **Usuarios.** Cualquier persona que interactúe con la base de datos.



1.6. Características de las bases de datos



Redundancia

- La redundancia de datos se refiere, a la existencia de información repetida o duplicada innecesaria para la base de datos.
- La redundancia conduce a muchos problemas que tienen que ver con la integridad y consistencia de los datos. La redundancia de los datos requiere múltiples procedimientos de entrada y actualización de los mismos.
- Dentro de una base de datos relacional la redundancia debe ser mínima y controlada. En ocasiones existirán motivos válidos de negocios o técnicos para mantener varias copias de los mismos datos almacenados.



1.6. Características de las bases de datos



Consistencia

- Es la evaluación de las reglas del negocio, esto es, verificar que los datos estén siguiendo dichas reglas.
- Es muy probable que surjan incongruencias al almacenar la misma información en más de un lugar; ya que al modificar, eliminar o agregar un dato, en esas condiciones, debe realizarse en cada una de las instancias del mismo con el riesgo de no realizarlo en su totalidad, generando en este caso datos inconsistentes.



1.6. Características de las bases de datos



Integridad

- La integridad es la facultad de poder implementar los mecanismos necesarios para que los datos guarden consistencia.
- Existiendo diferentes tipos de integridad como la de campo, la de entidad, la de clave, la referencial.



1.6. Características de las bases de datos



La integridad de la base de datos se puede lograr mediante :

- El mantenimiento una redundancia mínima y controlada.
- El establecimiento de llaves primarias o índices primarios.
- La validación de las dependencias entre tablas relacionadas.
- La creación de reglas de validación durante la inserción y edición de datos.



1.6. Características de las bases de datos



Seguridad

- Hoy en día se considera a la información de una empresa como uno de los activos más valiosos e importantes, por lo que la seguridad de la misma es muy importante.
- La seguridad implica asegurar que los usuarios están autorizados para llevar a cabo lo que tratan de hacer.



1.7. Modelo de datos



Modelo

Es una representación de la realidad que contiene las características generales de algo que se va a realizar. En base de datos, esta representación la elaboramos de forma gráfica.

Modelo de datos

Es una colección de herramientas conceptuales para describir los datos, las relaciones que existen entre ellos, semántica asociada a los datos y restricciones de consistencia.

Los modelos de datos se dividen en tres grupos: lógicos basados en objetos, lógicos basados en registros y físicos de datos.

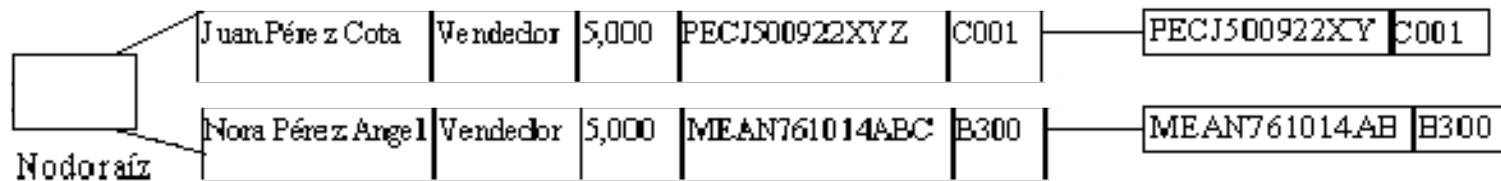


1.7. Modelo de datos



Modelo jerárquico

La forma de representar las relaciones y datos es por medio de registros y sus ligas. La diferencia radica en que están organizados por conjuntos de árboles en lugar de gráficas arbitrarias.



En este tipo de modelos la organización se establece en forma de árbol, donde la raíz es un nodo ficticio. Así tenemos que, una base de datos jerárquica es una colección de árboles de este tipo.

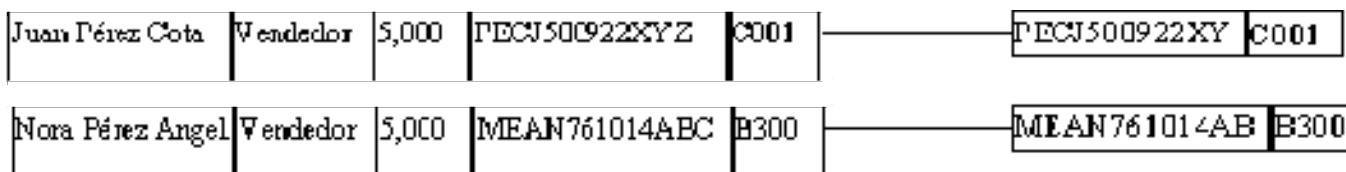


1.7. Modelo de datos



Modelo de red

Este modelo representa los datos mediante colecciones de registros, sus relaciones se representan por medio de ligas o enlaces, los cuales pueden verse como punteros.



Una base de datos de red como su nombre lo indica, esta formada por una colección de registros, los cuales están conectados entre sí por medio de enlaces. El registro es similar a una entidad como las empleadas en el modelo entidad-relación.



1.7. Modelo de datos



Modelo relacional

En este modelo se representan los datos y las relaciones entre estos, a través de una colección de relaciones, en las cuales los renglones (tuplas) equivalen a cada uno de los registros que contendrá la base de datos y las columnas corresponden a las características (atributos) de cada registro localizado en la tupla, por ejemplo, para las relaciones empleado y articulo, tendríamos:



1.7. Modelo de datos



* Cada una de las columnas representa a los atributos de la entidad empleado

Tabla del empleado

Nombre	Puesto	Salario	R.F.C
Juan Pérez Cota	Vendedor	5,000	PECT500922XYZ
Nora Méndez Angel	Vendedor	5,000	MEAN761014ABC

* Registros que contienen la información de la entidad empleado



1.7. Modelado de datos

Las formas de representar las relaciones en este modelo son:

- Haciendo una tabla que contenga cada una de las llaves primarias de las entidades involucradas en la relación, por ejemplo, tomando en cuenta que la llave primaria del empleado es su RFC, y la llave primaria del artículo es la clave.

La relación de nuestro modelo resulta:

RFC	Clave
PECT500922XYZ	C001
MEAN761014ABC	B300

- Incluyendo en alguna de las relaciones de las entidades involucradas, la llave de la otra tabla.

Incrustamos la llave primaria del artículo en la tabla del empleado

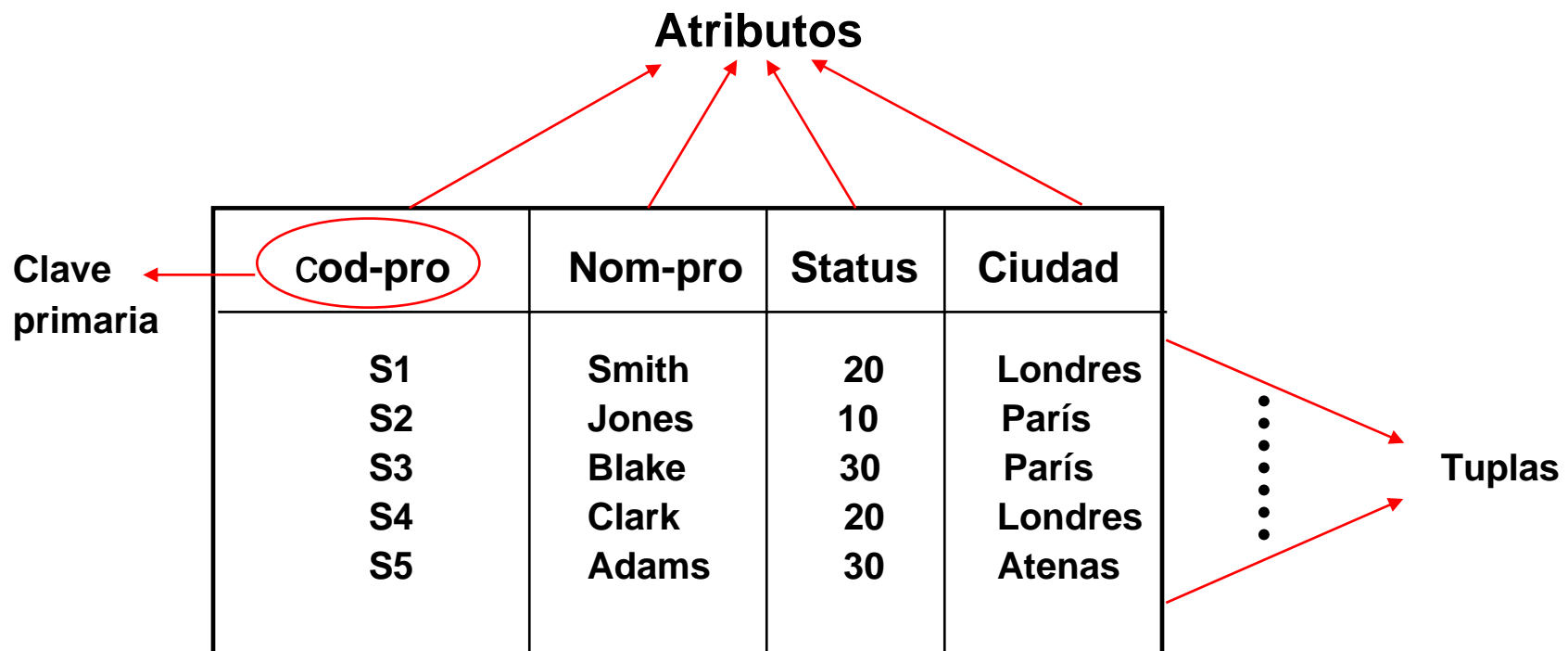
Nombre	Puesto	Salario	R.F.C	Clave
Juan Pérez Cota	Vendedor	5,000	PECT500922XYZ	C001
Nora Méndez Angel	Vendedor	5,000	MEAN761014ABC	B300



1.7. Modelado de datos



Ejemplo:





1.7. Modelado de datos



Modelo orientado a objetos

- El modelo de bases de datos orientado a objetos es una adaptación a los sistemas de bases de datos. Se basa en el concepto de encapsulamiento de datos y código que opera sobre estos en un objeto. Los objetos estructurados se agrupan en clases.
- El propósito de los sistemas de bases de datos es la gestión de grandes cantidades de información. Las primeras bases de datos surgieron del desarrollo de los sistemas de gestión de archivos. Estos sistemas primero evolucionaron en bases de datos de red o en bases de datos jerárquicas y, más tarde, en bases de datos relacionales.



Ventajas

- Facilita la comprensión y organización de un sistema de información, en términos de objetos.
- Facilita documentar las reglas de negocio, si las organizamos y agrupamos en torno a los objetos con los que se relacionan.

Desventajas:

- No hay variedad de software que permitan el modelado y después su alimentación a diversos DBMS.
- La desventaja anterior, obliga a usar herramientas no integradas; es decir, primero diseñar Objeto, por ejemplo, y después elaborar manualmente el diseño derivado de los objetos en otra herramienta, o a emular representación de Objetos semánticos bajo ER en diseñadores de tablas.



1.9. Modelo Relacional



La siguiente tabla hace una síntesis de la evolución del Modelo Relacional, desde su surgimiento a fines de la década de los sesenta hasta la actualidad.

Años	Sucesos
1968	Surge el Modelo Relacional (Codd).
1970	Desarrollo teóricos, álgebra relacional (Codd, 1972).
1973	Prototipos (Ingres, Sistema R, etc.)
1979	Oracle
1981	SQL
1982	Sybase, Informix
1984	SQL/ANS
1986	SQL ISO
1990	Modelo Relacional versión 2 (RM/V2) Codd
1992	SQL2 estándar
1994	SQL3 Aun no estandarizado BDOO



1.9. Modelo Relacional



El trabajo publicado por Codd en ACM (1970) presentaba un nuevo modelo de datos que perseguía una serie de objetivos, que se resumen en los siguientes líneas:

- **Independencia física.** El modo en el que se almacenan los datos no influye en su manipulación lógica y por tanto, los usuarios que acceden a esos datos no tienen que modificar sus programas por cambios en el almacenamiento físico.
- **Independencia lógica.** El añadir, eliminar o modificar objetos de la base de datos no repercute en los programas y/o usuarios que están accediendo a subconjuntos parciales de los mismos (vistas).
- **Flexibilidad.** En el sentido de poder presentar a cada usuario los datos de la forma en que éste prefiera.
- **Uniformidad.** Las estructuras lógicas de los datos presentan un aspecto uniforme, lo que facilita la concepción y manipulación de la base de datos por parte de los usuarios.
- **Sencillez.** Las características anteriores, así como unos lenguajes de usuario muy sencillos, producen como resultado que el modelo de datos relacional sea fácil de comprender y de utilizar por parte del usuario final.



1.9. Modelo Relacional



En el modelo relacional los datos se almacenan, al menos conceptualmente, de un modo en que los usuarios entienden con mayor facilidad.

Los datos se almacenan como tablas y las relaciones entre las filas y las tablas son visibles en los datos. Este enfoque permite a los usuarios obtener información de la base de datos sin asistencia de sistemas profesionales de administración de información.



1.9. Modelo Relacional



Es importante saber que las entradas en la tabla tienen un solo valor (son atómicos); no se admiten valores múltiples, por lo tanto la intersección de un renglón con una columna tiene un solo valor, nunca un conjunto de valores.

Todas las entradas de cualquier columna son de un solo tipo. Por ejemplo, una columna puede contener nombres de clientes, y en otra puede tener fechas de nacimiento. Cada columna posee un nombre único, el orden de las columnas no es de importancia para la tabla, las columnas de una tabla se conocen como atributos. Cada atributo tiene un dominio, que es una descripción física y lógica de valores permitidos.



1.9. Modelo Relacional



En el enfoque relacional es sustancialmente distinto de otros enfoques en términos de sus estructuras lógicas y del modo de las operaciones de entrada/salida. En el enfoque relacional, los datos se organizan en tablas llamadas relaciones, cada una de las cuales se implanta como un archivo. En terminología relacional una fila en una relación representa un registro o una entidad. Cada columna en una relación representa un campo o un atributo.

Así, una relación se compone de una colección de entidades (o registros) cuyos propietarios están descritos por cierto número de atributos predeterminados implantados como campos.



1.10. La independencia de los datos y la integridad referencial



La independencia de los datos

Una de las principales ventajas que provee una base de datos es la independencia entre los datos y los tratamientos que se hacen de ellos ya que en los sistemas orientados a procesos los datos eran sumamente dependientes de los programas.



1.10. La independencia de los datos y la integridad referencial



Como tal, la independencia de los datos se refiere a la protección contra los programas de aplicación que puedan originar modificaciones cuando se altera la organización física o lógica de la base de datos. Existen 2 niveles de independencia de datos:

- Independencia física de datos
- Independencia lógica de datos



1.10. La independencia de los datos y la integridad referencial



Independencia física de los datos

Es la capacidad de modificar el esquema físico sin provocar que se vuelvan a escribir los programas de aplicación. Dicho de otra forma, el esquema conceptual no es afectado por cambios al esquema físico de datos.

Es decir se refiere al ocultamiento de los detalles sobre las estructuras de almacenamiento a las aplicaciones de usuario, o sea la descripción física de datos puede cambiar sin afectar a las aplicaciones de usuario.

Si el DBMS modifica su organización interna de ficheros, no pasa nada con el esquema conceptual.



1.10. La independencia de los datos y la integridad referencial



Ejemplos:

- Los datos pueden ser movidos de un disco a otro.
- La organización de los datos puede cambiar
- Cuando hay un cambio de versión
- Al migrar a otro DBMS



1.10. La independencia de los datos y la integridad referencial



Independencia lógica de los datos

Capacidad de modificar el esquema conceptual sin provocar que se vuelvan a escribir los programas de aplicación.

Algunos elementos del esquema externo no son afectados por cambios al esquema conceptual.

Ejemplos:

- Modificar columnas sólo afecta las vistas que incluyen esas columnas
- Agregar más columnas no afecta las vistas
- La creación de una nueva relación.
- El reordenamiento lógico de algunos atributos



1.10. La independencia de los datos y la integridad referencial



La integridad referencial

El término de integridad referencial se enmarca en la segunda regla de integridad y se aplica a las claves foráneas:

“Si en una relación hay alguna clave foránea, sus valores deben coincidir con valores de la clave primaria a la que hace referencia, o bien, deben ser completamente nulos”.

Lo que en realidad trata de decir el texto anterior es que las claves foráneas no pueden dejar de tener correspondencia con la clave primaria de la tabla externa;



1.10. La independencia de los datos y la integridad referencial



Clave primaria: Es aquel atributo que identifica de manera única a un registro. Esto es, no debe haber dos tuplas que tengan el mismo valor, por lo tanto, con sólo conocer el valor de la clave primaria para una determinada tupla será suficiente para identificarlo de manera única.



1.10. La independencia de los datos y la integridad referencial



Clave candidata: Es el atributo o conjunto de atributos que podrían servir como llaves primarias.

Una llave candidata debe cumplir dos condiciones:

- **Unicidad:** no pueden existir dos tupías con el mismo valor en todos los atributos que forman la llave candidata.
- **Minimidad:** no existe ningún subconjunto de la llave que cumpla la regla de unicidad.



1.10. La independencia de los datos y la integridad referencial



Clave secundaria: Son aquellas claves candidatas que no se eligieron como llave primaria, es decir, tienen todas las características para ser claves primarias, pero que por alguna razón no fueron tomadas como tal debido quizás a que hubo otra que cumplía mejor con ese objetivo.



1.10. La independencia de los datos y la integridad referencial



Clave foránea: Es una clave primaria en otra relación, estas representan las asociaciones entre las diferentes entidades, es decir, son claves que están siendo compartidas por dos tablas para formar una relación entre ellas.



1.11. Modelo Entidad - Relación



El modelo Entidad - Relación, es una técnica de diseño de bases de datos gráfica, que incorpora información relativa a los datos y la relación existente entre ellos, para poder así plasmar una visión del mundo real sobre un soporte informático.



1.11. Modelo Entidad - Relación



Entre sus principales características tenemos se caracteriza fundamentalmente por:

- Sólo reflejar la existencia de los datos sin expresar lo que se hace con ellos.
- La independencia de la base de datos y de los sistemas operativos.
- La inclusión de todos los datos sin considerar las aplicaciones que se tendrán.



1.11. Modelo Entidad - Relación



Conceptos fundamentales



1.11. Modelo Entidad - Relación



Entidades

Se puede definir como entidad a cualquier objeto, real o abstracto, que existe en un contexto determinado o que puede llegar a existir y del cual deseamos guardar información, por ejemplo, un profesor, un alumno o bien una materia



1.11. Modelo Entidad - Relación



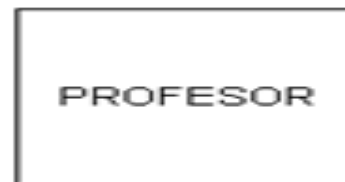
Las entidades las podemos clasificar en:

- Regulares
- Débiles



1.11. Modelo Entidad - Relación

Regulares: Son aquellas entidades que existen por sí mismas, es decir, la existencia de un ejemplar de la entidad no depende de la existencia de otros ejemplares en otra entidad, por ejemplo, la entidad "PROFESOR".





1.11. Modelo Entidad - Relación



Débiles: Son aquellas entidades en las que su existencia depende de la existencia de ejemplares en otras entidades, por ejemplo, la existencia de la entidad “*PROFESOR*” depende de la existencia de la entidad “*ESCUELA*”.





1.11. Modelo Entidad - Relación



Atributos

Las entidades se componen de atributos que son cada una de las propiedades o características que tienen las entidades. Cada ejemplar de una misma entidad posee los mismos atributos, tanto en nombre como en número, diferenciándose cada uno de los ejemplares por los valores que toman dichos atributos.



1.11. Modelo Entidad - Relación



Si consideramos la entidad "PROFESOR" y definimos los atributos Nombre, Cursos, Teléfonos y Edad, podríamos obtener los siguientes ejemplares:

PROFESOR			
NOMBRE	CURSOS	TELÉFONOS	EDAD
Luís García	Algebra	555-55-55	80.500
Juan Antonio Álvarez	Diseño	666-66-66	92.479
Marta López	Base de Datos	777-77-77	85.396



1.11. Modelo Entidad - Relación



Dominios

Se define dominio como un conjunto de valores que puede tomar un determinado atributo dentro de una entidad. Por ejemplo: 111

Atributo	Dominio
Fecha de Alta	Calendario Gregoriano
Teléfono	Conjunto de números de teléfonos
Cobro de Incentivos	SI / NO
Edad	16 - 65



1.11. Modelo Entidad - Relación



Claves

El modelo Entidad - Relación exige que cada entidad tenga un identificador, se trata de un atributo o conjunto de atributos que identifican de forma única a cada uno de los ejemplares de la entidad.



1.11. Modelo Entidad - Relación



Un ejemplo de identificador es el atributo "RFC", que en la entidad "PROFESOR", identifica de forma única a cada uno de los profesores. Estos identificadores reciben el nombre de Clave Primaria o Primary Key (PK). Como ya se había mencionado antes, puede ser que existan más identificadores, a estos atributos se les conoce como Identificadores Candidatos (IC).





1.11. Modelo Entidad - Relación



Interrelaciones

Se entiende por interrelación a la asociación, vinculación o correspondencia entre entidades. Por ejemplo, entre la entidad "*PROFESOR*" y la entidad "*CURSO*" podemos establecer la relación "*IMPARTE*" por que el profesor imparte cursos.



1.11. Modelo Entidad - Relación



Al igual que las entidades, las interrelaciones se pueden clasificar en regulares y débiles, esto de acuerdo al tipo de entidad que estén asociando, entidades regulares o entidades débiles, con otra de cualquier tipo. Las interrelaciones débiles se subdividen en dos grupos:

En existencia: Cuando los ejemplares de la entidad débil no pueden existir si desaparece el ejemplar de la entidad regular del cual dependen.

En identificación: Cuando además de ser una relación en existencia, los ejemplares de la entidad débil no se pueden identificar por sí mismos y exigen añadir el identificador principal de la entidad regular del cual dependen para ser identificados.



1.12. Normalización



El proceso de cristalización de las entidades y sus relaciones en formatos de tabla usando los conceptos relacionales se llama proceso de normalización y consiste en agrupar a los campos de datos en un conjunto de relaciones o tablas que representan a las entidades, sus características y sus relaciones de forma adecuada.

La razón de la normalización es asegurar que el modelo conceptual de la base de datos funcionará. Esto no significa que una estructura no normalizada no funcionará, sino que puede causar algunos problemas cuando los programadores de aplicación traten de modificar la base de datos para insertar, actualizar o eliminar datos.



1.12. Normalización

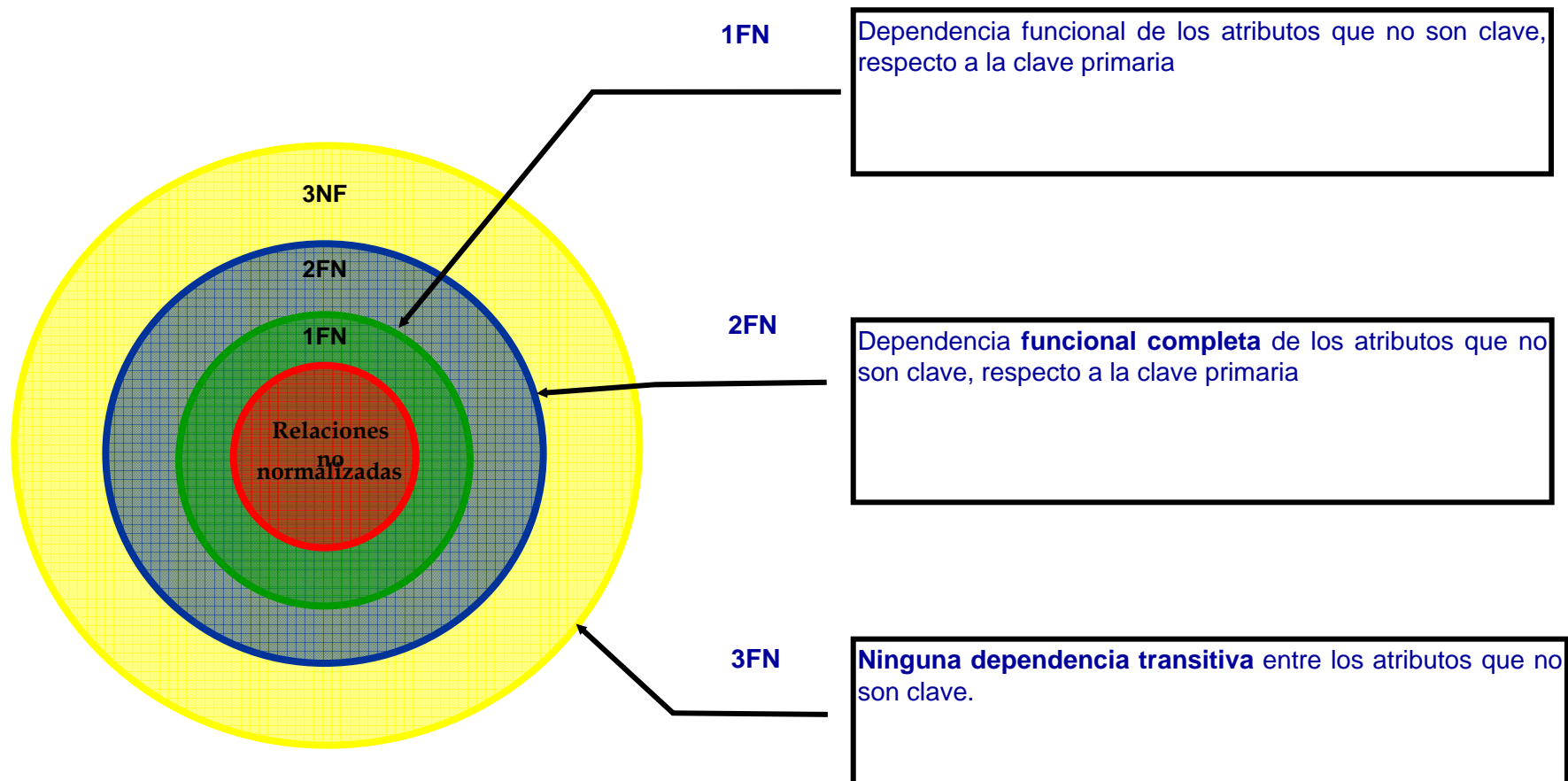


Las formas de normalización fueron propuestas originalmente por Codd, entre 1971 y 1972. Posteriormente varios investigadores continuaron trabajando en esta teoría y a lo largo del tiempo han surgido varias formas de normalización que complementan y refuerzan a las enunciadas por Codd.

Las *formas normales* son una serie de restricciones que se definen sobre las estructuras relacionales para evitar anomalías al efectuar adiciones, eliminaciones o actualizaciones de tuplas.



1.12. Normalización





1.12. Normalización



Las ventajas de la normalización son las siguientes:

- Evita anomalías en inserciones, modificaciones y borrados.
- Mejora la independencia de datos.
- No establece restricciones artificiales en la estructura de los datos.
- Están encaminadas a eliminar redundancias e inconsistencias de dependencia en el diseño de las tablas



1.12. Normalización



Primera Forma Normal (1FN)

Una relación está en primera forma normal si, y sólo si, todos los dominios de la misma contienen valores atómicos, es decir, no hay grupos repetitivos. Si se ve la relación gráficamente como una tabla, estará en 1FN si tiene un solo valor en la intersección de cada fila con cada columna.



1.12. Normalización



Segunda Forma Normal (2FN)

Una relación está en segunda forma normal si, y sólo si, está en 1FN y, además, cada atributo que no está en la clave primaria es completamente dependiente de la clave primaria.



1.12. Normalización



Tercera Forma Normal (3FN)

Una relación está en tercera forma normal si, y sólo si, está en 2FN y, además, cada atributo que no está en la clave primaria no depende transitivamente de la clave primaria. La dependencia es transitiva si existen las dependencias siendo atributos o conjuntos de atributos de una misma relación.



1.14. Herramientas CASE



Desde el inicio de la creación de software ha existido la necesidad de crear herramientas automatizadas que permitan incrementar la productividad de los diseñadores de software, en un inicio, los esfuerzos se direccionaron hacia programas traductores, compiladores, ensambladores, procesadores de macros, montadores y cargadores.

Al ver los beneficios de este conjunto de aplicaciones se generó una gran demanda por nuevo software con características similares. El significado de las siglas CASE viene de su acrónimo en inglés **C**omputer **A**ided **A**ssisted **A**utomated **S**oftware **S**ystems **E**ngineering.



1.14. Herramientas CASE



No existe una única clasificación de herramientas CASE y, en ocasiones, es difícil incluirlas en una clase en común. Podrían clasificarse así:

- Las plataformas que soportan.
- Las fases del ciclo de vida del desarrollo de sistemas que abarca.
- La arquitectura de las aplicaciones que produce.

Las herramientas CASE, en función de las fases del ciclo de vida que cubre, se pueden agrupar de la forma siguiente:



1.14. Herramientas CASE



- Herramientas integradas, I-CASE (Integrated CASE): Abarcan todas las fases del ciclo de vida del desarrollo de sistemas son llamadas CASE workbench.
- Herramientas de alto nivel, U-CASE (Upper CASE): Orientadas a la automatización y soporte de las actividades desarrolladas durante las primeras fases del desarrollo, análisis y diseño.
- Herramientas de bajo nivel, L-CASE (Lower CASE): Dirigidas a las últimas fases del desarrollo, construcción e implantación.
- Juegos de herramientas, (Tools CASE): Son el tipo más simple de Herramientas CASE, automatizan una fase dentro del ciclo de vida. Dentro de este grupo se encontrarían las herramientas de reingeniería, orientadas a la fase de mantenimiento.



1.14. Herramientas CASE



Componentes de una herramienta CASE

- *Repositorio:* Este se puede definir como la base de datos central de una herramienta CASE. El repositorio amplía el concepto de diccionario de datos para incluir toda la información que se va generando a lo largo del ciclo de vida del sistema, por ejemplo: componentes de análisis y diseño.
- *Módulos de diagramación y modelación:* Algunos de los diagramas y modelos utilizados con mayor frecuencia son: diagrama de flujo de datos, modelo E-R, y técnicas matriciales.



1.14. Herramientas CASE



- *Herramienta de prototipazo:* El objetivo principal de esta herramienta es poder mostrar al usuario, desde los momentos iniciales del diseño, el aspecto que tendrá la aplicación una vez desarrollada. Ello facilitará la aplicación de los cambios que se consideren necesarios, todavía en la fase de diseño.
- *Generador de código:* Normalmente se suele utilizar sobre ordenadores personales o estaciones de trabajo, por lo que el paso posterior del código al host puede traer problemas, al tener que compilar en ambos entornos.
- *Módulo generador de documentación:* El módulo generador de la documentación se alimenta del repositorio para transcribir las especificaciones allí contenidas.



1.14. Herramientas CASE



Herramientas Case más utilizadas

- **ERwin:** Es una herramienta para el diseño de base de datos, que Brinda productividad en su diseño, generación, y mantenimiento de aplicaciones. Desde un modelo lógico de los requerimientos de información, hasta el modelo físico perfeccionado para las características específicas de la base de datos diseñada, además ERwin permite visualizar la estructura, los elementos importantes, y optimizar el diseño de la base de datos. Genera automáticamente las tablas y miles de líneas de *stored procedure* y *triggers* para los principales tipos de base de datos.
- **EasyCASE:** Es un producto para la generación de esquemas de base de datos e ingeniería reversa, esta herramienta permite automatizar las fases de análisis y diseño dentro del desarrollo de una aplicación, para poder crear las aplicaciones eficazmente desde el procesamiento de transacciones a la aplicación de bases de datos de cliente/servidor, así como sistemas de tiempo real, es una herramienta multi-usuario.



1.14. Herramientas CASE



- **Oracle Designer:** Oracle Designer es un conjunto de herramientas para guardar las definiciones que necesita el usuario y automatizar la construcción rápida de aplicaciones cliente/servidor gráficas. Este está integrado con Oracle Developer, lo cual provee una solución para desarrollar sistemas empresariales de segunda generación.
- **System Architect:** Herramienta que posee un repositorio único que integra todas las herramientas, y metodologías usadas. En la elaboración de los diagramas, el System Architect conecta directamente al diccionario de datos, los elementos asociados, comentarios, reglas de validaciones, normalización.
- **DBDesigner:** Producto destacable por su sencillez, permite modelar sobre MySQL y dispone de la capacidad de generar documentación e incluso pantallas de administración sobre PHP.
- **TableDesigner:** Herramienta que permite la creación de bases de datos en Access y SQL Server de Microsoft.