

Nivelación

Introducción a Base de Datos

DISEÑO DE BASE DE DATOS

Conceptos generales

Dato

1950

Es una representación simbólica. Por si solo no tiene significado.

Información

1950



- Dato que tiene un significado, que fue procesado y se convirtió en información.

Dato Es una representación simbólica. Por si solo no tiene un significado. Diferencia entre Dato e Información. Ejemplo con un nro, que puede ser un año, código de materia, nro de puerta, etc.

Información: Dato que tiene un significado, el dato fue procesado y se convirtió en información.

Base de Datos

Conjunto de datos relacionados.



Debe cumplir:

- Representar algún aspecto de la realidad
- Tener un significado implícito.
- Servir para un propósito específico.

Sistema de gestión de Base de Datos (DBMS)



Programa que provee un conjunto de servicios para el mantenimiento y construcción de bases de datos.

Una **Base de Datos** es un conjunto de **datos relacionados entre sí**. NO son un DBMS. Esta mal decir que Oracle es una base de datos. No se necesita un DBMS para tener una base de datos, ejemplo, un fichero, planillas Excel, archivos secuenciales, separados por comas, etc.

Por **datos** entendemos hechos que pueden ser registrados. Esta definición puede ser demasiado amplia, por lo que para referirnos al término “base de datos” que vamos a estar utilizando, vamos a agregarle a esa definición, que los datos deben cumplir además las siguientes propiedades:

- Representar algún aspecto de la realidad.
- Tener un significado implícito.
- Sirvan para un propósito específico destinado a un grupo de usuarios concreto.

Ejemplo: Guía telefónica, búsqueda por persona o por teléfono

DISEÑO DEL MODELO CONCEPTUAL

Modelo Entidad Relación (MER)

El diseño de una **Base de Datos**, se compone de tres etapas:

- Diseño del Modelo Conceptual
- Diseño del Modelo Relacional
- Diseño del Modelo Físico

Modelo Entidad Relación (MER)

Es un modelo conceptual de alto nivel y se utiliza para la definición de datos. Se basa en representar en forma gráfica objetos (entidades) y relaciones entre esos objetos.

Entidad



Es un objeto distinguible de la realidad, del cual nos interesa almacenar datos.

Por ejemplo: Cliente, País, Producto, etc.

Un **modelo entidad-relación o diagrama entidad-relación** es una herramienta para el modelado de datos que permite representar las entidades relevantes de un sistema de información así como sus interrelaciones y propiedades

Permite:

- Descripción concreta de los requisitos de información (tipos de datos, relaciones, restricciones que los datos deben cumplir).
- Fácil de entender por usuarios no técnicos.

Entidad Es un objeto del mundo real que podemos distinguir de los demás objetos, y del que nos interesa almacenar datos

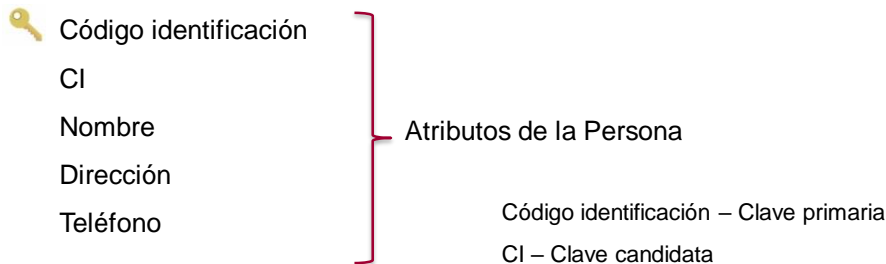
Atributo: Todas aquellas características de la entidad que nos interesa conocer.

Las entidades pueden estar relacionadas entre sí, y es posible que existan atributos que correspondan a esa relación.

Atributo

Es una características de la entidad que nos interesa conocer.

Ejemplo: Entidad Persona



Consideremos la entidad Persona.

Existen ciertos datos que son comunes a todas las personas:

- Código de identificación
- Cédula
- Nombre
- Dirección
- Teléfono

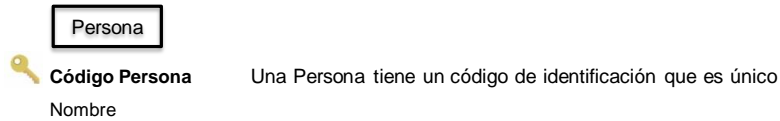
Dentro de este conjunto de atributos, existirá uno (o un conjunto) que representará y hará única a cada persona. En este ejemplo contamos con un código identificatorio de la persona y la cédula de identidad.

Si tomamos el código de la persona, podemos decir que será la **clave primaria** de la entidad, no pudiendo existir dos personas con el mismo código de identificación.

También se podría elegir la Cédula de Identidad como clave primaria ya que no existen dos personas con la misma cédula, pero en este ejemplo cumplirá el rol de **clave candidata**, ya que por su característica funcionales opera como un atributo único pero no es la clave primaria de la entidad.

Atributo: Clave Primaria

Es el atributo (o conjunto de atributos) que identifican en forma única a cada elemento de la Entidad.



Atributo: Clave Foránea

Atributo que es clave primaria en la entidad A y participa como atributo en la entidad B haciendo referencia a la entidad A.



RELACIONES ENTRE ENTIDADES

Cardinalidad



Es la forma en que se relacionan las entidades entre sí.

Relaciones entre Entidades



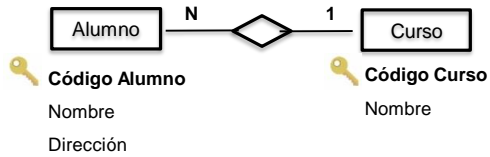
La Cardinalidad entre dos entidades, es la forma en que se relacionan esas entidades{

Existen 4 tipos de relaciones que pueden establecerse entre entidades, las cuales establecen con cuantas ocurrencias de entidad de tipo B se puede relacionar una ocurrencia de entidad de tipo A:

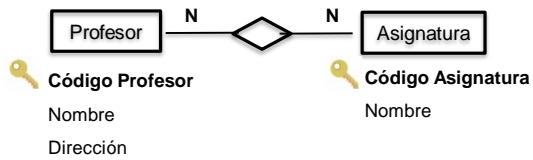
- **Relación uno a uno (1-1):** Un elemento de la entidad A se relaciona con un elemento de la entidad B, y un elemento de la entidad B se relaciona con un elemento de la entidad A.
- **Relación uno a varios (1-N):** Un elemento de la entidad A se relaciona con varios elemento de la entidad B, y un elemento de la entidad B se relaciona con 1 elemento de la entidad A.
- **Relación varios a varios (N-N):** Un elemento de la entidad A se relaciona con varios elementos de la entidad B, y un elemento de la entidad B se relaciona con varios elementos de la entidad A.

Ejemplos

1) Un Alumno se inscribe a un Curso, y en un Curso se inscriben muchos Alumnos.

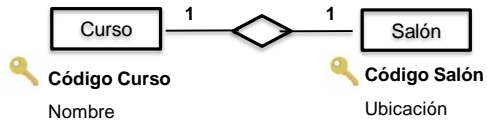


2) Un Profesor dicta muchas Asignaturas, y una Asignatura es dictada por muchos Profesores.



Ejemplos

3) Un Curso se dicta en un Salón, y en un Salón se dicta un solo Curso.



Entidades débiles

Son aquellas entidades que expresan dependencia con respecto a otra. Una Entidad débil no existe por sí sola sino que depende de la existencia de otra.

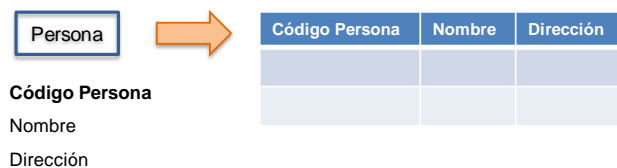


DISEÑO DEL MODELO LOGICO

Modelo Relacional (MR)

Modelo Lógico

Toda Entidad fuerte declarada en el MER será una tabla física en el Modelo Lógico.

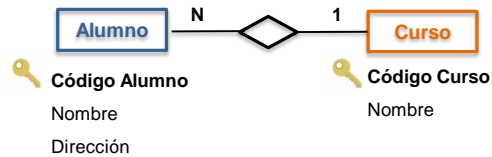


Cada entidad fuerte del MER se traducirá a una tabla en el MR. Recordemos que una entidad fuerte es un elemento de la realidad que necesitamos conocer y almacenar sus datos.

Hay que considerar siempre los aspectos de cardinalidad que existan entre las entidades definidas en el MER para la creación de las tablas.

Representación de la relación 1 - N

Un Alumno se inscribe a un Curso, y en un Curso se inscriben muchos Alumnos.



Se agrega la clave primaria del Curso como clave foránea en Alumno.



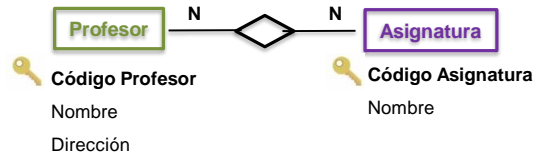
El atributo Código Curso es clave primaria en Curso y clave foránea en Alumno. Solamente se podrá indicar un Curso por Alumno.

De esta forma se representa que un Alumno está asociado a un Curso, pero que un Curso puede estar asociado a muchos Alumnos.

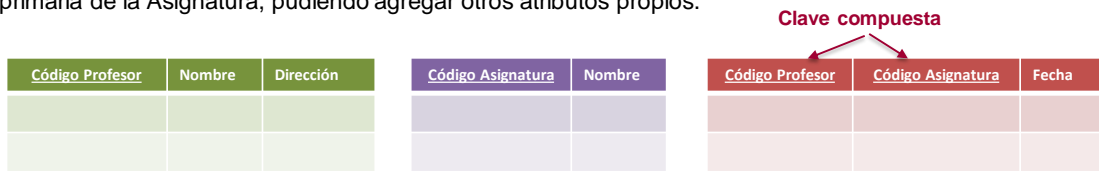
De esta forma podemos tener que:
 María Rodríguez cursa Diseño Web
 Luis Gómez cursa Técnico en Redes
 Andrés García cursa Diseño Web
 Etc.

Representación de la relación N - N

Un Profesor dicta muchas Asignaturas, y una Asignatura es dictada por muchos Profesores.



La relación “dicta” se traduce a una tercera tabla que combina la clave primaria del Profesor con la clave primaria de la Asignatura, pudiendo agregar otros atributos propios.



Primero debemos asegurarnos que Profesor y Asignatura existan con todos sus datos.

Como un Profesor puede dictar muchas Asignaturas, y una Asignatura puede ser dictada por muchos Profesores, la relación “dicta” se traduce como una tercera tabla donde al menos contiene la clave primaria del Profesor y la clave primaria de la Asignatura. Puede además contener otros atributos como puede ser en este caso la fecha de dictado.

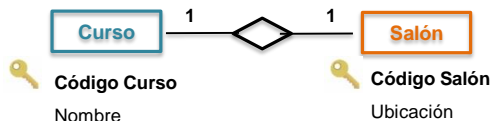
En este ejemplo, la clave primaria de la relación queda compuesta por:
Código Profesor, Código Asignatura

De esta forma podemos tener que:

Juan Pérez dicta Base de Datos, Programación y GeneXus.
María Gómez dicta Sistemas Operativos y Programación
Ricardo López dicta GeneXus y Redes
Etc.

Representación de la relación 1 - 1

Un Curso se dicta en un Salón, y en un Salón se dicta un solo Curso.



Se puede representar de dos formas:

- Para cada Curso referenciar el Código de Salón.
- Para cada Salón referenciar el Código del Curso.

Código Curso	Ubicación	Código Salón

VS

Código Salón	Ubicación	Código Curso

Esta relación se puede representar de dos formas:

- Agregando el Código de Salón como clave foránea en el Curso.
- Agregando el Código del Curso como clave foránea en el Salón.

En cualquiera de los dos casos de alguna forma se deberá controlar que dichas asignaciones no tengan repeticiones. O sea, que ningún Código de Salón se repita en el Curso, o que ningún Código de Curso se repita en el Salón.

La elección de la alternativa tendrá que ver con la forma en que se quiere disponer de los datos, y con el orden en que se quiere ingresar:

- Ingresar primero los Cursos para luego poder hacer la referencia desde el Salón.
- Ingresar primero los Salones para luego poder hacer la referencia desde el Curso.

CONSTRUCCION

Modelo Físico (MF)

Construcción

Se determina la forma en que se almacenan los datos.

- Tipo de dato
- Relaciones entre los datos (integridad)
- Restricciones
 - Restricciones de Dominio
 - Restricciones de Unicidad de clave
 - Restricciones de Integridad referencial

Para que las tablas reflejen fielmente la realidad, se define un conjunto de conceptos para determinar cómo se almacenan los datos.

- Tipo de Datos – Característica que debe cumplir el dato para ser aceptado (si debe ser un dato numérico, fecha, carácter, etc).
- Relaciones entre los datos (integridad): Claves primarias y foráneas. Esto es lo que define la integridad y relación entre los datos.
- Restricciones:
 - Dominio: Un atributo solamente admitirá el tipo de dato que se le indique, y no otro.
 - Unicidad de la clave: El valor de un atributo clave primaria no puede repetirse ni modificarse.
 - Integridad referencial: Se refiere a que debe haber consistencia entre los datos de las distintas tablas de una base de datos relacional. Las tablas de una base de datos relacional se encuentran relacionadas por atributos que tienen en común. Estas relaciones implican que los datos de las tablas no son independientes, sino que al insertar, modificar y eliminar registros en una tabla, se deben tener en cuenta los datos de las otras tablas para que siempre se conserve la consistencia de la información en la base de datos.

NORMALIZACION

Normalización

- Proceso durante el cual se descomponen las estructuras de tablas no satisfactorias.
- Garantiza que no se produzcan situaciones anormales al momento de ingresar, modificar o eliminar datos.
- Implica llevar las tablas a la primera, segunda y tercera forma normal.

1ra
Forma
Normal

El contenido de todas las columnas de la tabla es un valor único y no una lista de valores.

2da
Forma
Normal

La tabla está en 1ra Forma Normal y además todos los valores declarados en ella dependen de la clave primaria.

3ra
Forma
Normal

La tabla está en 2da Forma Normal y además no existen dependencias transitivas entre las comunas de una tabla. Es decir, no existen atributos no principales (que no sea clave primaria) que dependan funcionalmente de otros atributos no principales

Existen más niveles de normalización, pero una tabla se considera normalizada si se lleva al menos hasta la 3era forma normal.

ÍNDICES

- Los índices son una estructura de la base de datos, que ayuda a mejorar la velocidad de las operaciones. Un índice básicamente sirve para buscar datos de forma veloz, y no tener que recorrer toda la tabla secuencialmente en busca de alguna fila concreta.

TABLA CLIENTE:

Ordenado por clave primaria
(ordenación por defecto)

Ordenado por nombre

Índice por: ClientId

ClientId	ClientName
1	Leo
2	Jacob
3	Theo
4	Jonah
5	Alex
6	Henry
7	Nora

Índice por: ClientName

ClientId	ClientName
5	Alex
6	Henry
2	Jacob
4	Jonah
1	Leo
7	Nora
3	Theo

Los índices de una base de datos funcionan de forma similar al índice de un libro, donde se guardará el elemento que se desea indexar y su posición. De esta forma, para buscar un elemento que esté indexado, sólo habrá que buscar en el índice dicho elemento, evitando así recorrer toda la tabla que estemos navegando en búsqueda del dato que estamos necesitando.

Se pueden también crear índices únicos, en el que no se permite que dos filas tengan el mismo valor en la columna de clave del índice. Es decir que no permite valores duplicados.

OPERACIONES A LA BASE DE DATOS

Unidad Lógica de Trabajo (UTL)

- Una Unidad Lógica de Trabajo (UTL) es una secuencia de operaciones sobre la Base de Datos, realizadas como una sola unidad, en la cual intervienen sentencias que modifican datos en una o más tablas de la misma. Las UTL corresponden al concepto de “transacciones de base de datos”.
- Si una UTL tiene éxito, todas las modificaciones de los datos realizadas durante la transacción se confirman y convierten en modificaciones permanentes. (**Commit**)
- Si durante el transcurso de una UTL, alguna de las sentencias encuentra errores, se cancelan todas las modificaciones llevadas a cabo en la transacción (**Rollback**)

Sentencias SQL **Commit** y **Rollback**, para confirmar o deshacer la UTL:

- **Commit:** Confirma como permanentes las modificaciones realizadas en una UTL, marca el final correcto de la misma.
Garantiza que todas las modificaciones de la transacción se conviertan en una parte permanente de la base de datos.
- **Rollback:** Si se produce un error en una UTL el usuario decide cancelar la misma, hay que volver para atrás con los cambios realizados hasta el momento. Esto se hace con la instrucción ROLLBACK, que devuelve los datos al estado en que estaban al inicio de la transacción
Marca un final incorrecto de una transacción. Aborta todos los cambios efectuados desde el comienzo de ella.
Los Rollbacks son importantes para la integridad de la base de datos, a causa de que significan que la base de datos puede ser restaurada a un estado consistente incluso después de que se han realizado operaciones erróneas. Cuando una base de datos tiene esta capacidad, se dice que tiene “Integridad Transaccional”.

GeneXus™

Videos

training.genexus.com

Documentation

wiki.genexus.com

Certifications

training.genexus.com/certifications