Universidad Andrés Bello®
Conectar · Innovar · Liderar

FUNDAMENTOS DE BASES DE DATOS RELACIONALES

Modelo Relacional

El modelo relacional y sus diferencias con el modelo conceptual

En el modelo relacional se utiliza un grupo de tablas para representar los datos y las relaciones entre ellos. Cada tabla está compuesta por varias columnas, y cada columna tiene un nombre único.

Características del modelo relacional

- Los datos se organizan en relaciones compuestas por tuplas de atributos. Si se convierte esta definición a tablas, se tiene que los datos se organizan en tablas compuestas por filas (registros) y columnas (campos).
- A cada tabla se le asigna un nombre único.
- Una tabla tiene 0 o más filas, y cada fila contiene la información de un determinado 'sujeto' de la relación.
- Las filas en un principio están desordenadas.
- La lista de los atributos dispuestos en un orden específico de izquierda a derecha y que forman la definición de una tabla se denomina "esquema" de la tabla, mientras que los valores concretos de los datos que están almacenados en la tablase llaman "ocurrencias".
- Todos los valores de una columna determinada tienen el mismo tipo de datos, y éstos están extraídos de un conjunto de valores legales llamado "dominio" de la columna. Muchas veces el dominio se corresponderá con un tipo de datos estándar del sistema.
- Además de los valores del dominio, la columna puede contener un valor especial,el valor nulo. El valor nulo (NULL) es importante porque representa la ausencia de valor en el campo y no es lo mismo que el valor cero "0" o la cadena vacía o espacios en blanco. De hecho, es un valor tan especial que no funciona como los demás valores; por ejemplo, no se puede comparar (con el operador de comparación =) un campo con el valor nulo, y por lo mismo se tiene que utilizar un operador especial (IS NULL). Incluso se han tenido que redefinir los operadores lógicos para tener en cuenta el valor nulo.
- En una tabla cada columna tiene un único nombre, y éste no se puede utilizar para nombrar otra columna de la misma tabla, pero sí de otra tabla.
- En una tabla no se admiten dos filas con los valores coincidentes en todos sus campos. Esta restricción no se suele cumplir.
- Toda tabla debe tener una clave principal (clave primaria).
    - Una clave primaria es cualquier una columna (o combinación de columnas) que permite identificar de forma unívoca cada una de las filas de la tabla. Para que pueda cumplir su cometido, la clave primaria no puede contenervalores nulos ni valores duplicados (no podrá haber dos filas con el mismovalor en este campo).
    - Hay SGBD que incluyen el concepto de clave primaria pero no la hacen obligatoria, por lo que en estos sistemas se pueden definir tablas sin clave primaria.

    }

- Otro concepto muy importante, fundamental en las bases de datos relacionales, es la clave ajena (externa o foránea).
    - Una clave ajena es un campo (o combinación de campos) que contiene la referencia a una fila de otra tabla; también puede referirse a la misma tabla.
    - En otras palabras, es un campo que señala a un registro de otra tabla, y que contiene un valor que identifica un registro de la otra tabla.
    - Son los campos que se utilizan para relacionar las tablas entre sí.
- Una clave ajena puede contener valores duplicados y valores nulos.
- Una tabla puede tener 0, una o varias claves ajenas (externas, foráneas).
- El SGBD deberá velar por la integridad de los datos; para ello incluye varias reglas de integridad que se comprobarán de forma automática sin necesidad de la intervención externa de los usuarios o de los programas de aplicación.
- Existen distintos tipos de reglas de integridad:
    - La **integridad de entidades** (integridad de claves): "Toda tabla debe tener una clave primaria que permite identificar unívocamente los registros que contiene, por lo tanto, no puede contener el valor nulo ni valores duplicados".
    - La **integridad referencial**: "En una clave ajena no puede haber un valor no nulo que no exista en la tabla de referencia". Tal como se analizó en temas anteriores, para que no existan errores de integridad referencial en la base de datos, el sistema comprueba automáticamente que los valores introducidos en las claves ajenas existan en el campo de referencia en la otra tabla, si no existe, no dejará insertar el registro.

- A nivel de control sobre los datos, el SGBD debe de proporcionar herramientas para poder definir restricciones de dominio que se comprobarán de forma automática (se comprueba que el valor introducido en una columna pertenece al dominio de la columna, al tipo de datos), y reglas de negocio, que son reglas específicas sobre los datos.

Diferencias entre el modelo Entidad-Relación y Relacional

Tanto el Modelo E-R como el Modelo Relacional son tipos de Modelado de Datos.

El modelo de datos describe una manera de diseñar la base de datos a nivel físico, lógico y de vista.

La principal diferencia entre el Modelo E-R y el Modelo Relacional es que el Modelo E-R es específico para cada entidad, y el Modelo Relacional es específico para cada tabla.

A continuación, se mostrará una tabla comparativa de ambos tipos de modelo:

	Modelo E/R	Modelo Relacional
Labor	Representa la colección de objetos llamada entidades y la relación entre esas entidades.	Representa la colección de tablas y la relación entre esastablas.
Describir	El modelo de relación de entidad describe los datos como conjunto de entidades, conjunto de relaciones y atributos.	El Modelo Relacional describe los datos en una tabla como Dominio, Atributos, Tuplas.
Relación	En el modelo E/R es más fácil enteder la relación entre las entidades.	Comparativamente, es menos fácil derivar una relación entre tablas en el Modelo Relacional.
Cartografía	El modelo E/R describe la asignación de cardinalidades.	El Modelo Relacional no describe las cardinalidades de mapeo.

Las diferencias principales entre ambos modelos son las siguientes:

- La diferencia básica entre el Modelo E-R y el Modelo Relacional es que el modelo E-R trata específicamente con las entidades y sus relaciones. Por otro lado, el Modelo Relacional se ocupa de las Tablas y de la relación entre los datos de esas tablas.
- Un Modelo E-R describe los datos con conjuntos de entidades, conjuntos de relaciones y atributos. Sin embargo, el modelo relacional describe los datos con las tuplas, atributos y dominio del atributo.

- Se puede entender más fácilmente la relación entre los datos en el Modelo E-R en comparación con el Modelo Relacional.
- El Modelo E-R tiene la Cardinalidad del Mapeo como una restricción, mientras que el Modelo Relacional no tiene tal restricción.

Reglas de transformación

Un Diagrama E/R puede utilizarse de forma muy simplificada con el propósito de especificar un modelo de dominio (relaciones entre conceptos del problema). Más adelante en el proceso, dicho modelo de dominio podría refinarse hasta constituir el Modelo Conceptual de datos, es decir, el modelo de datos en el ámbito de análisis. Este modelo conceptual de datos debería tener trazabilidad hacia el modelo de datos en su nivel de abstracción correspondiente al diseño lógico relacional, es decir, una especificación del esquema relacional que pueda servir para generar una implementaciónposterior en un determinado gestor de bases de datos relacional. A continuación, se presentarán pautas muy simples (incluso ya están automatizadas en muchas herramientas) que permiten obtener un Modelo Lógico Relacional (o su representación gráfica como Diagramas de Tablas) a partir de un Diagrama E/R.

Cabe destacar que estas mismas pautas pueden emplearse para obtener un Modelo Lógico Relacional a partir de un Diagrama de Clases. Para hacerlo se debe obviar la información del Diagrama de Clases, ocultando los métodos de las clases, además de las clases y los atributos no persistentes (aquellas clases y atributos que no requieran almacenarse en la Base de Datos). Este es el simple truco para usar un Diagrama de Clasesde forma similar a un Diagrama E/R.

Transformación de Entidades y Atributos

- Transformar cada entidad en una tabla. El identificador de la entidad será la clave primaria de la tabla. Si la entidad no tiene definido un identificador, para la tabla se crea un atributo adicional que se utilizará como clave primaria.
- Analizar la conveniencia de crear tablas adicionales para cada atributo multivaluado. Las nuevas tablas tendrán como clave primaria el identificador de laentidad junto al atributo multivaluado.
- Los atributos compuestos se "aplanan" dejando como atributos de la tabla sólo los componentes de más bajo nivel.
- Los atributos derivados pueden tratarse como atributos normales indicando que se trata de atributos derivados. La decisión de almacenar o no físicamente el valor del atributo derivado podría retrasarse hasta la implementación.

- Los atributos que no pueden tener valor nulo deben indicarse (salvo si son parte de la clave primaria, caso en el cual se entiende que no pueden tomar valores nulos).

La figura siguiente ilustra la obtención de tablas a partir de una entidad y sus atributos.

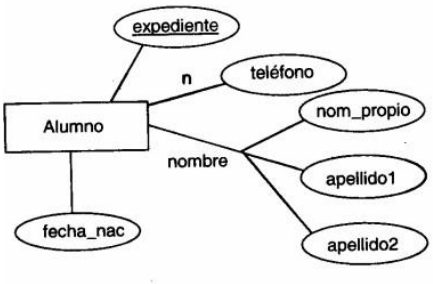

Ilustración 1: Ejemplo de transformación de entidad y atributos en tablas

```
ALUMNO(Nro_exp, Nombre, Apellido1, Apellido2, Fec_nac)

TELEFONOS_ALUMNO(Nro_exp, Telefono)
```

Transformación de Supertipos y Subtipos

Para esta parte de la transformación se plantean las siguientes alternativas:

I. Sólo una tabla incluyendo los atributos del supertipo más la unión de todos los atributos (y relaciones) de los subtipos.
   - La clave primaria es el identificador del supertipo.
   - Considerando que las jerarquías son disjuntas, debería establecerse una restricción de integridad que controle que los grupos de atributos tengan valores en forma excluyente unos de otros.
   - Puede incluirse un atributo derivado que indique el subtipo al cual pertenece una tupla.

II. Una tabla para el supertipo y una para cada subtipo. La tabla que representa a cada subtipo tiene como atributo adicional el identificador del supertipo, pudiendo ser ésta la clave primaria de la tabla del subtipo, o solamente una clave ajena que referencia a la tabla del supertipo.
   - Puede incluirse un atributo derivado que indique el subtipo al cual pertenece una tupla de la tabla del supertipo.

III. Sólo las tablas correspondientes a cada subtipo, incluyendo en cada una de ellas todos los atributos (y relaciones) del supertipo. Las claves primarias de las tablas de los subtipos se determinan como en el caso anterior.

Las ilustraciones 2 y 3 indican transformaciones de generalizaciones a tablas, proponiendo alternativas posibles en cada caso.

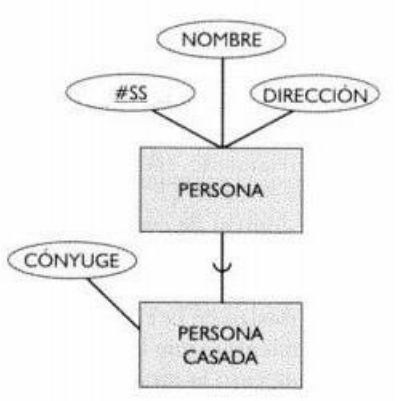

Ilustración 2: Ejemplo de transformación de generalización a tablas

```
ALTERNATIVAS:

   I.    PERSONA(SS#, Nombre, Dirección, Cónyuge)

II. PERSONA(SS#, Nombre, Dirección)
 PERSONA_CASADA(SS#, Cónyuge)
```

Ilustración 3: Otro ejemplo de transformación de generalización a tablas

```
ALTERNATIVAS:

I. CLIENTE(#Cliente, Fec_nac, Sexo, Nro_de_empleados,
Representante, Tipo_org)
RI: (Fec_nac, Sexo) no Nulo o (Nro_empleados,
Representante, Tipo_org)
No Nulo, en forma exclusiva.

II. CLIENTE(#Cliente)
```

```
CLIENTE_HUMANO(#Cliente, Fec_nac, Sexo)
CLIENTE_INSTITUCION(#Cliente, Nro_de_empleados,
Representante, Tipo_org)
```

Transformación de Relaciones

Cualquier tipo de relación podría ser tratada como un tipo de entidad (tipo de entidad asociativa) y transformarse como tal. Según esto, cada tipo de relación generaría una tabla adicional. Esto se aplicará a relaciones binarias N:M, ternarias o superiores); en el resto de relaciones binarias se recomienda dar un tratamiento especial atendiendo a las multiplicidades asociadas. La idea es reducir el número de tablas para evitar excesivo "join" en las consultas. Sin embargo, debe existir un compromiso entre dicho propósito y la evolución que pueda tener el modelo en el futuro.

Las relaciones entre entidades se representan en las tablas como claves ajenas.

Relación Binaria 1:1

Se propone analizar las siguientes alternativas:

I. Si una de las entidades no puede existir sin participar en la relación o es poco relevante (en el contexto de estudio), incluir toda la información del tipo de entidad menos relevante (y atributos del tipo de relación, si existen) en la tabla que representa al otro tipo de entidad.
II. Incluir la clave primaria de la tabla que representa a una entidad en la tabla de la otra (será una clave ajena para ésta). Hacer lo mismo con los atributos del tipo derelación, si existen.

La ilustración 4 muestra alternativas para obtener tablas a partir de una relación binaria 1:1. La ilustración siguiente muestra el caso para una relación binaria reflexiva 1:1.

Ilustración 4: Ejemplo de transformación de relación 1:1

ALTERNATIVAS:

I. EMPLEADO(Atributos Empleado)
PC(Atributos PC)

PC_ASIGNADO(CP_Empleado, CP_PC, Atributos Relación)

II. EMPLEADO(Atributos Empleado, Atributos PC, Atributos Relación)
PC(Atributos PC, Atributos Empleado, Atributos Relación)

III. EMPLEADO(Atributos Empleado, CP_PC, Atributos Relación)
PC(Atributos PC)

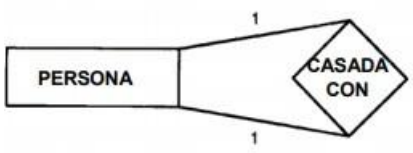

Ilustración 5: Ejemplo de transformación de relación unaria 1:1

ALTERNATIVAS:

I. PERSONA(Atributos Persona, CP_Otra_Persona, Atributos Relación)

II. PERSONA(Atributos Persona)
CASADA_CON(CP_Persona, CP_Otra_Persona, Atributos Relación)

Tipo de relación binaria 1:N (“uno a muchos”)

Los atributos del tipo de relación y la clave primaria de la tabla que representa al tipo deentidad de la parte “uno” se incorporan en la tabla que representa al tipo de entidad de la parte “muchos”. Así, La clave primaria de la parte “uno” se convierte en una clave ajenapara la tabla de la parte “muchos”.

```
DEPTO(Atributos Departamento)
```

Ilustración 6: Ejemplo de transformación de relación binaria 1:N

```
EMPLEADO(Atributos Empleado, CP_Departamento, Atributos Relación)
```

Relaciones Binarias N:M y Tenarias (o superiores)

En este caso se crea una tabla para cada entidad participante y además se añade unatabla para la relación.

Ilustración 7: Ejemplo de transformación de relación binaria M:N

```
ALUMNO(Atributos Alumno)
CURSO(Atributos Curso)
INSCRITO_EN(CP_Alumno, CP_Curso, Atributos Relación)
```

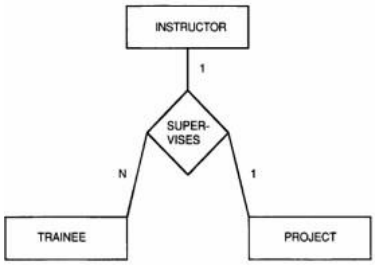

Ilustración 8: Ejemplo de transformación de relación ternaria

```
INSTRUCTOR(Atributos Instructor)
AYUDANTE(Atributos Ayudante)
PROYECTO(Atributos Proyecto)
SUPERVISA(CP_Instructor, CP_Ayudante, CP_Proyecto,
Atributos Relación)
```

Ejemplo 1: Fábrica de pelotas "Golazo"

Solicitan nuestros servicios para resolver el almacenamiento de datos de un sistema de gestión de la producción de una fábrica de pelotas. La fábrica se compone de una serie de plantas, cada una identificada por un color. De las plantas conocemos la superficie en metros cuadrados y la lista de procesos que se llevan a cabo dentro de ellas; de estos procesos sólo conocemos su nombre y un grado de complejidad asociado. Dentro de cada planta se encuentran las máquinas. Cada máquina es de una marca y un modelo, y se identifica por un número; este número es único a lo largo de todas las plantas.
Cada máquina es operada por técnicos, debemos conocer en qué rango de fechas los técnicos estuvieron asignados a esa máquina, y además en qué turno (mañana, tarde o noche).
De los técnicos conocemos su DNI, nombre, apellido y fecha de nacimiento, aparte de una serie de números telefónicos de contacto.
Existen situaciones normales en las que una máquina sale de servicio y debe ser reparada, lo único que nos interesa conocer aquí es cuál otra máquina está asignada para tomar el trabajo que ella no puede realizar.

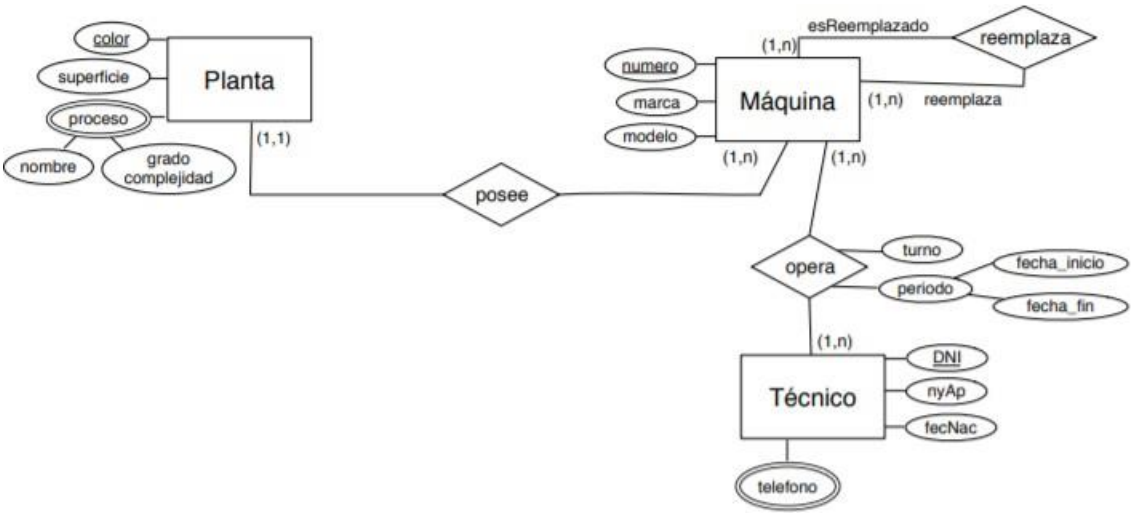

Ilustración 9: Modelo E/R ejemplo 1

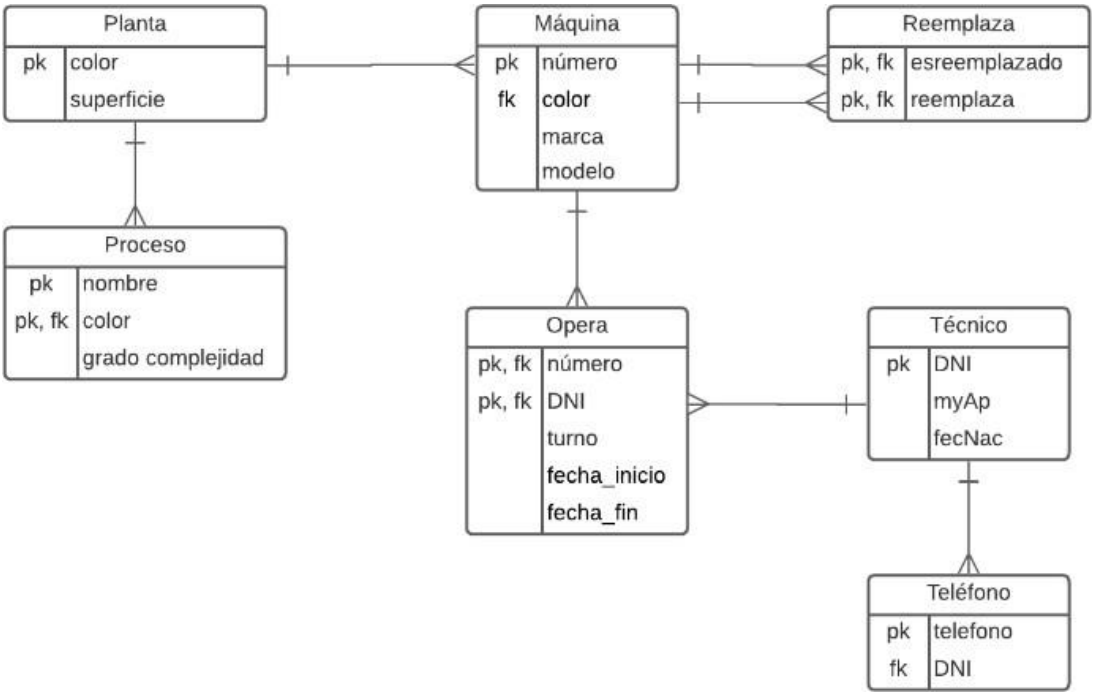

Ilustración 10: Modelo relacional ejemplo 1

Asignando tipos de datos y restricciones al modelo

Tal como se indicó antes y se reflejó en los ejemplos, cada atributo de las entidades del modelo E/R se convierte en un campo de una tabla. Al mismo tiempo, los atributos obtenidos desde tablas generadas desde relaciones entre entidades también se representan como campos.

Cada campo de las tablas debe tener un tipo específico de acuerdo a su naturaleza. Esta última característica indica si en un determinado campo se almacenará un número entero, un número decimal, un texto, un dato booleano, etc; además es importante establecer la precisión del campo, esto es, la cantidad máxima de caracteres o dígitos, según sea el caso, que tendrá en la tabla.

Es importante recordar que, si bien a nivel de modelo de datos los campos no tienen un tipo específico, cuando se pasa desde un modelo conceptual a un modelo lógico, la manera de crearlo va a depender directamente del RDBMS seleccionado.

Una vez definidos los tipos de datos, se debe proceder a crear las relaciones foráneas como elementos constitutivos de la base de datos. La forma de crearlas va a depender enlo específico del motor de base de datos, ya que la creación de restricciones a nivel de comandos difiere en aspectos muy específicos.

Finalmente, definido el script o código que creará el modelo en la base de datos, se debeejecutar y validar posteriormente que la acción no haya generado inconvenientes, y que todas las restricciones incluidas en el mismo hayan sido creadas.

Ejemplo 2: Complemente el modelo relacional del ejemplo anterior, asignando tipos de datos adecuados a cada campo.

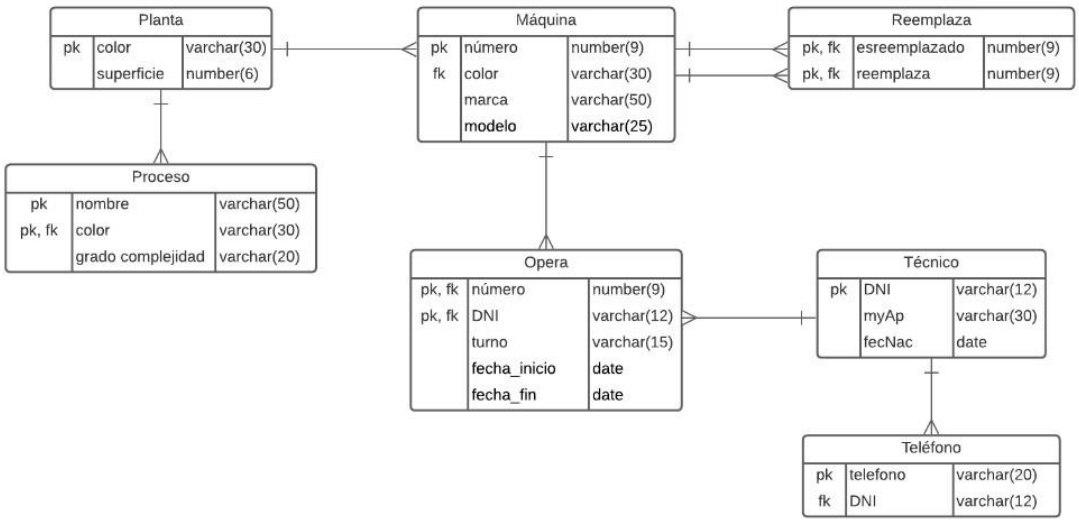

Ilustración 11: Modelo relacional ejemplo 2

Normalización (1FN, 2FN, 3FN)

La normalización es el proceso de organizar los datos de una base de datos, valga la redundancia. Se debe tener en cuenta la creación de tablas y las reglas que se usan para definir las relaciones; estas reglas son diseñadas para proteger los datos, y para que la base de datos sea flexible con el fin de eliminar redundancias y dependencias incoherentes.

Las bases de datos relacionales se normalizan para:

- Evitar la redundancia de los datos.
- Disminuir problemas de actualización de los datos en las tablas.
- Proteger la integridad de los datos.
- Facilitar el acceso e interpretación de los datos.
- Reducir el tiempo y complejidad de revisión de las bases de datos
- Optimizar el espacio de almacenamiento.
- Prevenir borrados indeseados de datos.

Para que las tablas de una base de datos estén normalizadas deben cumplir las siguientesreglas:

- Cada tabla debe tener su nombre único.
- No puede haber dos filas iguales.
- No se permiten los duplicados.
- Todos los datos en una columna deben ser del mismo tipo.

Para normalizar una base de datos existen principalmente 3 reglas, las cuales se deberíancumplir para evitar redundancias e incoherencias en las dependencias. A estas reglas se les conoce como "Forma normal", y van de la 1 a la 3, y si la base de datos cumple con cada regla se dice que está en la "primera o segunda o tercera forma normal".

Aunque son posibles otros niveles de normalización, la tercera forma normal se considera el máximo nivel necesario para la mayoría de las aplicaciones.

Primera forma normal

- Eliminar los grupos repetidos de las tablas individuales.
- Crear una tabla independiente para cada conjunto de datos relacionados.
- Identificar cada conjunto de datos relacionados con una clave principal.

Segunda forma normal

- Cree tablas independientes para conjuntos de valores que se apliquen a varios registros.
- Relacione estas tablas con una clave externa.

Tercera forma normal

- Elimine los campos que no dependan de la clave.

Cada nivel de normalización exige cierta complejidad, pero se debe analizar si en realidad se necesita llegar hasta la tercera forma normal; siempre se debe buscar lo que mejor seadapte al problema planteado.

El diccionario de datos

Es un listado organizado de todos los campos y tablas pertenecientes a una base de datos. La información contenida en él deberá incluir aquellas características que describan e identifiquen cada elemento. Las anotaciones, métodos y herramientas utilizadas para desarrollar este apartado deberán estar estandarizadas. Son desarrollados durante el modelamiento de datos, y ayuda a los analistas a tener una mejor interpretación en la determinación de los requerimientos del sistema.

En un diccionario de datos se encuentra la lista de todos los elementos que forman parte del modelo de la base de datos. El diccionario de datos guarda los detalles y descripción de todos estos elementos.

Ventajas del diccionario de datos

- Contiene la lista de todos los objetos que forman parte de la base de datos.
- Hace más fácil el manejo de los detalles en los sistemas de gran volumen, permitiendo una mayor visión de los objetos en la base de datos.
- Localizar errores y omisiones en el proceso de diseño es más sencillo cuando se toma como referencia un diccionario de datos.
- Todos los encargados de la base de datos tendrán un conocimiento universal estandarizado, facilitando la comunicación en el grupo de trabajo.
- Si los analistas desean conocer cuántos caracteres abarca un determinado dato o qué otro nombre recibe en distintas partes del sistema, o dónde se utiliza, encontrarán las respuestas en un diccionario de datos desarrollado en forma apropiada.

Elementos Del Diccionario De Datos

- **Datos Elementales:** Es la parte más pequeña de los datos que tiene significado en el sistema de información, pero al combinarlos con varios elementos de datos se puede obtener una estructura de datos que provea la información completa que se desea consultar. Por ejemplo, un nombre; por si solo no representa nada, pero cuando se relaciona con su número de cédula o algunos otros atributos se tendrá una estructura dedatos acerca de esa persona.

- **Nombre de los datos:** Se usan para distinguir un dato de otro; se les asignan nombres significativos a los objetos en las bases de datos para tener un mayor control de la información.

- **Descripción de los Datos:** Establece brevemente lo que representa el dato en el sistema; por ejemplo, la descripción para FECHA-DE-FACTURA indica que es la fecha en la cual se está preparando la misma para distinguirla de la fecha en la que se envió por correo o serecibió. Las descripciones de datos se deben escribir suponiendo que la gente que los leano conoce nada en relación al sistema.

- **Alias:** Son los distintos nombres que puede recibir un dato, dependiendo de quién y cuálsea el uso que se le va a dar a dicho dato.

- **Longitud de campo:** Es la cantidad de espacio que ocupa un dato.

Aplicabilidad

- Los diccionarios de datos se deben aplicar cuando se desarrolla una base de datoso una aplicación, ya que la información que ellos proveen son de gran ayuda en elproceso de desarrollo.
- Coordinan la actividad de la base de datos en los objetos de entrada de datos.
- Proveen a un programa los servicios de validación y actualización de la base de datos.

Ejemplo 3: Diccionario de datos

Nombre de Archivo: APEmpleado. **Fecha de Creación:** 23/07/2007
Descripción: Archivo Principal de Empleados, contendrá información de cada uno de ellos.

Campo	Tipo	Tamaño	Descripción
cveEmpleado	Numérico	12	Clave de Empleado
cNombre	Caractér	50	Nombre del empleado
nSueldo	Numérico	10:2	Sueldo del empleado con dos decimales para fraccionarios
cveDpto	Numérico	5	Clave del departamento a donde pertenece el empleado.
cveArea	Numérico	5	Clave del área donde se encuentra el empleado.
cdespuesto	Caractér	75	Descripción del puesto que ocupa.

Relaciones:
CatDepto con el campo cveDpto.
CatArea con el campo cveArea.

Campos Clave:
cveEmpleado, cveDpto, cveArea

Ilustración 12: Ejemplo 3 - Tabla APEmpleado

Nombre del Archivo: CatDepto. **Fecha de Creación:** 23/07/2007
Descripción: Archivo para el catálogo de Departamentos

Campo	Tipo	Tamaño	Descripción
cveDpto	Numérico	5	Clave de departamento.
cDescripcion	Caratér	75	Nombre del departamento.
cveArea	Numérico	5	Clave del área a donde pertenece el departamento.

Relaciones:
CatArea con el campo cveArea.

Campos Clave:
cveDpto, cveArea

Ilustración 13: Ejemplo 3 - Tabla CatDepto

Nombre de Archivo: CatArea **Fecha de Creación:** 23/07/2007
Descripción: Archivo con información de las áreas que componen la empresa.

Campo	Tipo	Tamaño	Descripción
cveArea	Numérica	5	Clave de área de la empresa
cDescripcion	Caractér	75	Nombre del área que de la empresa.

Relaciones:
Ninguna

Campos Clave:
cveArea

Ilustración 14: Ejemplo 3 - Tabla CatArea

Anexo: Referencias

1.- Modelo relacional
Referencia:
https://sites.google.com/site/basdededatosrelacionales/home/contenido/subtema-2/modelo-relacional

2.- Elementos de una base de datos relacional
Referencia: https://sites.google.com/site/basededatosiiucb2/tipos-de-bases-de-datos/elementos-de-una-base-de-datos-relacional

3.- Diferencias E/R y modelo relacional
Referencia: https://pc-solucion.es/2018/04/18/diferencias-entre-el-modelo-entidad-relacion-y-relacional/

4.- Normalización de bases de datos
Referencia: https://ed.team/blog/normalizacion-de-bases-de-datos

5.- Diccionarios De Datos En Las Bases De Datos
Referencia: https://dapayox.wordpress.com/2011/06/24/diccionario_base_datos/