

TEMA 35: La definición de datos. Niveles de descripción. Lenguajes. Diccionario de datos.

Índice de contenidos

- 1.- Introducción
- 2.- La definición de datos
 - 2.1.- Modelos lógicos basados en objetos
 - 2.1.1.- Modelo de Entidad-Relación
 - 2.1.2.- Modelo Orientado a Objetos
 - 2.2.- Modelos lógicos basados en registros
 - 2.2.1.- Modelo Relacional
 - 2.2.2.- Modelo de Red
 - 2.2.3.- Modelo Jerárquico
 - 2.3.- Modelos físicos de datos
- 3.- Niveles de descripción. El modelo Entidad-Relación
 - 3.1.- Modelo Entidad-Relación
 - 3.2.- Diagramas Entidad-Relación
- 4.- Lenguajes. SQL
 - 4.1.- Sentencias SQL para definición de datos
- 5.- Diccionario de datos
- 6.- Conclusiones
 - 6.1.- Otras consideraciones
 - 6.2.- Ámbitos de aplicación docente
- 7.- Referencias bibliográficas

1.- Introducción

El tema que se desarrolla pertenece al bloque de Bases de Datos, por lo que comenzaremos recordando los principales conceptos de este campo de la ciencia.

Una **base de datos (BD)** es un sistema formado por una colección de datos almacenados, relativos a diversas temáticas y categorizados de distinta manera, pero que comparten entre sí algún tipo de vínculo o contexto que

permiten el acceso directo a ellos, así como su relación. Estos datos se encuentran almacenados sobre un soporte físico.

Llanos Ferraris sitúa la aparición del concepto de bases de datos a comienzos de la década de 1960, en un simposio celebrado en Santa Mónica (EE.UU), y la define como “(...) un conjunto, colección o depósito de datos almacenados en un soporte informático de acceso directo. Los datos deben estar relacionados y estructurados de acuerdo con un modelo capaz de recoger el contenido semántico de los datos almacenados. Dada la importancia que tienen en el mundo real las relaciones entre los datos, es imprescindible que la base de datos sea capaz de almacenar estas interrelaciones. Esta es una de las principales diferencias respecto a los ficheros tradicionales, en los que no se almacenan dichas relaciones. Además, las bases de datos modernas también almacenan las restricciones semánticas que están presentes en los datos y a las que se les está concediendo una importancia creciente”.

Por su parte, el **sistema de gestión de la base de datos (SGBD)** es un conjunto de programas, de propósito general, que facilita la definición, construcción y manipulación de las BD.

Dentro de las funciones que debe cumplir un SGBD, podemos destacar las siguientes:

- **Definición de datos:** Los lenguajes de definición de datos (DDL) permiten especificar el esquema o diseño de una BD, mediante la definición de los objetos que la constituyen, la estructura y las interrelaciones.
- **Manipulación de datos:** consiste en realizar ciertas acciones sobre los datos existentes, o añadir nuevos. Estas acciones son:
 - Consultar
 - Insertar
 - Eliminar
 - Modificar o actualizar

Para poder realizar estas tareas, se utilizan lenguajes de manipulación de datos (DML).

- **Recuperación:** se debe prever que se produzcan fallos, debiendo existir mecanismos para recuperar los datos ante errores. Dichos fallos pueden ser:
 - En la transacción o del software
 - En el sistema o de hardware
 - Del medio de almacenamiento
- **Integridad:** unos datos que relacionan una tabla con otra deben ser coherentes.
- **Concurrencia:** una BD debe asegurar la consistencia de los datos, que se puede ver alterada ante un sistema multiusuario, que permite la multiprogramación.
- **Diccionario de datos:** el diccionario de datos es una BD del sistema que contiene metadatos, o lo que es lo mismo, incluye datos referentes a los datos que constituyen la BD original. El SGBD es el responsable de su definición y mantenimiento.

En el desarrollo de este tema, nos centraremos en la definición de datos, según los niveles de abstracción interno, conceptual y externo.

Asimismo, dentro del nivel lógico o conceptual, analizaremos los principales modelos lógicos de definición de la BD, haciendo un especial hincapié en el más extendido, esto es, el Modelo Entidad-Relación.

Seguidamente, se hablará de los lenguajes de definición de datos (DDL), haciendo una introducción al lenguaje más extendido en el mercado: SQL.

Y, por último, trataremos la definición del diccionario de datos, así como su contenido y sintaxis, entre otros.

Finalizará la redacción del tema con las preceptivas conclusiones, en las que, además de destacar lo más significativo del mismo, se llevará a cabo una relación de los principales contenidos del tema con los módulos y materias en los que un Profesor de Enseñanza Secundaria de la Especialidad de Informática las pone en práctica, así como un recordatorio de diversos aspectos que, por las obvias

limitaciones de tiempo, no han sido abordados (junto a la relación de otros temas del temario donde se tratan). Por último, se expondrán las referencias bibliográficas.

2.- La definición de datos

Entre las funciones con las que debe contar un SGBD, se encuentra la definición de datos. Esto consiste en describir los objetos que vamos a representar, su estructura y las relaciones existentes entre ellos, así como las reglas de integridad, controles de acceso, características de tipo físico y las vistas externas de los usuarios.

Para esto, contamos con un lenguaje específico: **DDL**, que es propio de cada SGBD y que debe suministrar los mecanismos necesarios para definir las tres estructuras de datos (externa, lógica global e interna), haciendo posible la especificación de las características de los datos en cada uno de estos niveles.

Podemos clasificar los modelos de datos más extendidos en la literatura en tres tipos:

- Modelos lógicos basados en objetos
- Modelos lógicos basados en registros
- Modelos físicos de datos

2.1.- Modelos lógicos basados en objetos

Estos modelos se utilizan para describir datos en los niveles externo y conceptual, es decir, con este modelo representamos los datos tal y como los captamos en el mundo real. Poseen una capacidad de definición bastante flexible y permiten especificar restricciones de integridad de los datos. Pese a que el modelo Entidad-Relación será el que más desarrollemos en este libro y, en definitiva, el más extendido, otros también importantes son:

2.1.1.- Modelo de Entidad-Relación

Se trata del modelo de datos más extendido en el mundo. Presenta un alto nivel de abstracción y es un modelo basado en percibir la realidad como una serie de entidades (objetos que existen en la realidad) y de relaciones entre esos objetos. Tanto las entidades como las relaciones contienen atributos (información que los

definen), que nos servirán para diferenciar cada entidad de otras similares.

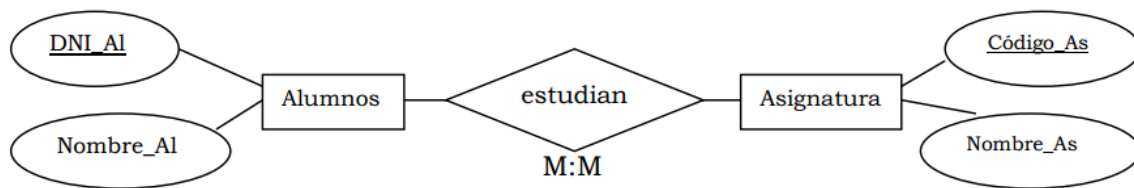


Figura 1. Ejemplo de modelo Entidad-Relación

2.1.2.- Modelo Orientado a Objetos

Este modelo también se basa en una colección de objetos, con valores almacenados (como variables de instancia) dentro de ellos. La principal diferencia radica en que estos objetos también almacenan fragmentos de código, con las operaciones fundamentales de estos, llamados métodos.

La forma en la que un objeto puede acceder a los datos de otro es mediante la invocación de los métodos del segundo, acción se conoce como paso de mensajes.

Al contrario que las entidades del modelo E-R, cada objeto dispone de identidad propia, aunque pueda poseer la misma información que otro objeto de los de su clase. Dicha distinción se establece a nivel físico (mediante la asignación de una posición en memoria diferente para almacenar el citado objeto)

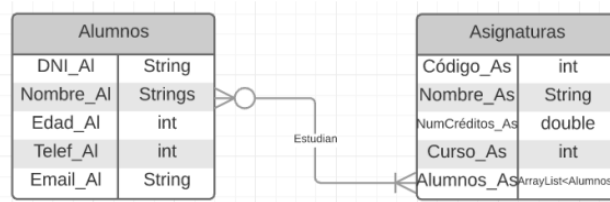


Figura 2. Ejemplo de un modelado OO

Del mismo modo, existen otros modelos, tales como:

- Modelo de datos semántico
- Modelo de datos funcional

2.2.- Modelos lógicos basados en registros

Es un modelo de menor abstracción, que se emplea para describir los datos en los niveles conceptual y físico. A diferencia de los modelos anteriores, la BD se estructura en registros de formato fijo, diferentes, en función del tipo de elemento

a almacenar. De esta forma, para cada tipo de registro se define un registro de longitud fija y de número de campos también fijo, lo que simplifica la implementación de la base de datos.

Al igual que ocurría en el anterior modelo, nos encontramos con diversos modelos, fundamentalmente diferenciados en las estructuras de datos para almacenar estos registros y para relacionarlos.

2.2.1.- Modelo relacional

En este modelo se emplea una colección de tablas para expresar tanto los datos, como sus relaciones. Cada tabla contiene varias columnas (con nombre único) y las relaciones se representan mediante columnas comunes en dichas tablas.

2.2.2.- Modelo de red

Los datos del modelo de red se representan mediante colecciones de registros, pero sus relaciones se representan mediante enlaces o punteros a otros registros. El esquema final del modelado se asemeja a un grafo dirigido.

2.2.3.- Modelo jerárquico

Este modelo es muy similar al modelo de red, con la diferencia de que los registros se organizan como colecciones de árboles, en lugar de como grafos dirigidos.

2.3.- Modelos físicos de datos

Se trata de estructuras de datos de bajo nivel implementadas dentro del propio manejador. A diferencia del modelo de datos lógico, hay pocos modelos de datos físicos que se usan en la práctica.

Algunos exponentes de estos modelos son el modelo unificador y el modelo de memoria de elementos. Suelen emplear, como estructuras de datos con las que se implementan estos modelos, los Árboles B+, o las Tablas Hash.

3.- Niveles de descripción. El modelo Entidad-Relación

Existen tres características propias de los SGBD: la separación entre los programas de aplicación y los datos, la necesidad de emplear un diccionario de datos para almacenar el esquema de la BD y el uso de múltiples vistas por parte de los usuarios.

El comité **ANSI SPARC** propuso, en 1975, una arquitectura de tres niveles para los SGBD, dividiéndola en los niveles de abstracción: interno, conceptual y externo.

- ❖ **Nivel interno o físico:** Su responsabilidad reside en el almacenamiento físico de los datos (el tamaño de los bloques de datos, los métodos de direccionamiento, los índices, etc.), así como los métodos de acceso. Es el considerado como más exhaustivo.
- ❖ **Nivel conceptual o lógico:** Se encarga de la descripción de la estructura de los datos y de sus relaciones (describe las entidades, atributos, relaciones, restricciones, etc.).
- ❖ **Nivel externo:** Define las vistas, es decir, las partes de la BD visibles para las distintas aplicaciones y usuarios que trabajan sobre ella.

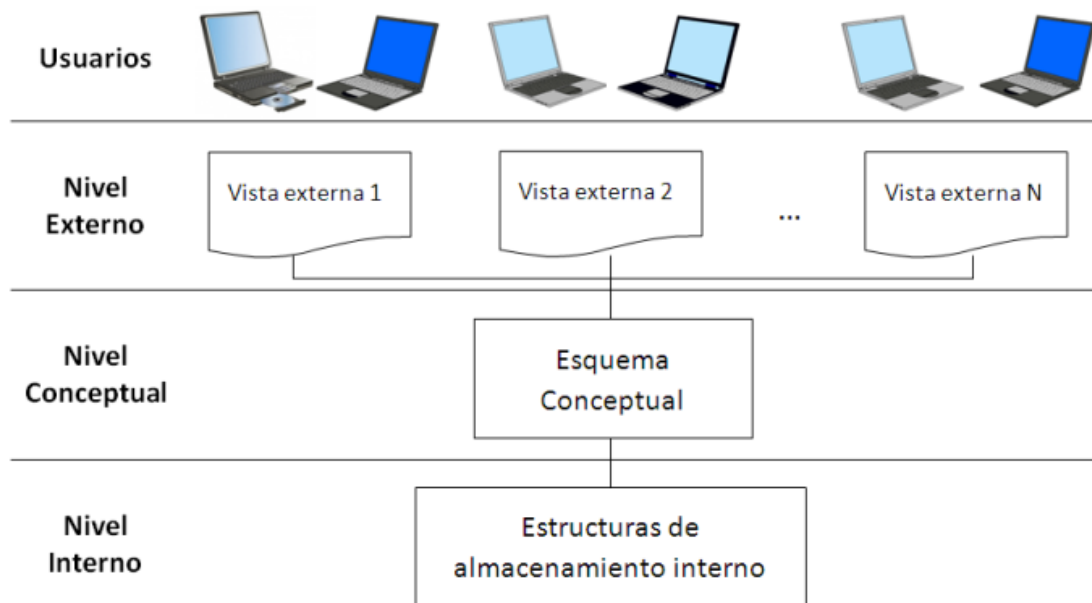


Figura 3. Esquema de la Arquitectura de un SGBD.

Dentro del nivel conceptual, podemos definir el modelo de datos como una colección de herramientas conceptuales, reglas y convenciones utilizadas para la definición de datos, es decir, el modelo de datos proporciona la abstracción necesaria a la BD para describir:

→ Datos.

- Relaciones entre datos.
- Significado de los datos.
- Restricciones de consistencia.

La correspondencia conceptual-interna es la que existe entre la vista conceptual y la base de datos almacenada; especifica cómo se representan los registros y campos conceptuales en el nivel interno. Si se modifica la estructura de la base de datos almacenada -es decir si se altera la definición de la estructura de almacenamiento- la correspondencia conceptual-interna deberá modificarse también de acuerdo con ello, para que no varíe el esquema conceptual. Dicho de otra manera, los efectos de las alteraciones deberán aislarse por debajo del nivel conceptual, a fin de conservar la independencia de los datos.

La correspondencia externa-conceptual es la que existe entre una determinada vista externa y la vista conceptual. Las diferencias que pueden existir entre esos dos niveles son similares a las que pueden existir entre la vista conceptual y la base de datos almacenada. Puede existir cualquier cantidad de vistas externas; cualquier número de usuarios puede compartir una determinada vista externa.

Estas correspondencias están relacionadas con otro objetivo de los SGBD, la **independencia**, que es la capacidad para modificar un esquema de definición sin afectar a los programas de aplicación. Nos encontramos con dos niveles:

- **Independencia física:** se puede cambiar el soporte físico de la base de datos sin que esto afecte a la base de datos ni a los programas que la utilizan.
- **Independencia lógica:** se pueden modificar los datos contenidos en la base de datos, las relaciones o incluso incluir nuevos datos, sin que se vean afectados los programas que las usan.

La existencia de estos niveles de descripción de datos, hace posible el hecho de que no todos los usuarios necesitan disponer de la misma percepción de la base de datos. Supone además un mecanismo de seguridad para evitar que los usuarios accedan a datos para los que no deberían estar autorizados.

3.1.- El modelo Entidad-Relación

Propuesto por Peter Chen en 1976, el modelo Entidad-Relación (E-R) es el modelo conceptual de alto nivel más extendido en la actualidad. Presenta un alto nivel de abstracción y se fundamenta en percibir la realidad como una serie de entidades (objetos que existen en la realidad) y de relaciones entre esos objetos. Tanto las entidades como las relaciones contienen atributos.

Entidades

Una entidad es un objeto que existe y se puede distinguir de otros. El conjunto de las entidades del mismo tipo se denomina Conjunto de Entidades (por economía, Entidades).

Entidades fuertes y entidades débiles

Se puede hacer distinción entre dos tipos de entidades:

- **Entidades Fuertes:** Su existencia no depende de ninguna otra entidad (rectángulo simple).
- **Entidades débiles:** Su existencia está condicionada a la existencia de otra entidad (doble rectángulo).

Por ejemplo, si quisiéramos almacenar una entidad llamada *Cualidades_Alumnos* dependiente de *Alumnos*, esto se representaría de la siguiente forma:

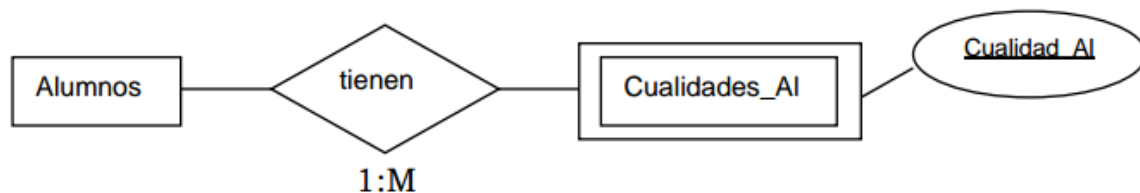


Figura 4. Representación de dos entidades (fuerte-débil) en el modelo E-R

Interrelaciones: grado, cardinalidad y rol

Las interrelaciones o relaciones representan asociaciones entre entidades. El conjunto de las relaciones del mismo tipo se denomina *Conjunto de Relaciones*.

Siguiendo el mismo ejemplo anterior, en la BD de la academia de enseñanza donde almacenamos Alumnos y Asignaturas, existirá una relación que asocie ambas entidades, que hemos denominado Estudian la cual informa sobre qué alumnos están estudiando qué asignaturas.

Grado

El grado hace referencia al número de entidades relacionadas. Los tipos de grado pueden ser:

- Relaciones de grado 1 (reflexivas): En la que se relaciona una entidad consigo misma.
- Relaciones de grado 2 (binarias): Cuando se relacionan dos entidades.
- Relaciones de grado superior a 2 (ternarias, cuaternarias, etc.): Aquellas que asocian más de 2 entidades en la relación.

Cardinalidad

Los conjuntos de relaciones pueden tener restricciones. La más frecuente es la cardinalidad de asignación, que limita el número de entidades relacionadas con una entidad del otro conjunto de entidades: uno a uno (1:1); uno a muchos (1:n); muchos a uno (n:1) o muchos a muchos (n:m).

Rol

Referente a las relaciones de grado 1, opcionalmente se puede especificar el rol que desempeña cada parte de la entidad en la relación.

Dominios y valores

Podemos definir el dominio de un atributo (cuya definición la veremos en el siguiente apartado) como la especificación de todos los valores que pueden estar contenidos en dicho atributo. Se trata pues de una restricción de cuáles son los valores aceptables para ese atributo concreto.

Algunos dominios usuales para una base de datos suelen ser cadenas de texto, carácter, enteros, reales, fechas, etc. Los dominios pueden ser también compuestos, a partir de otros (año, mes y día → fecha).

Atributos

Los atributos almacenan las propiedades que nos interesa conservar de las entidades y se representan mediante óvalos.

Claves primarias

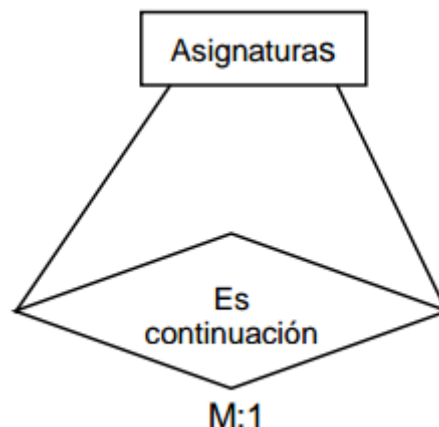
Las claves primarias son aquellas claves candidatas (las que identifican un atributo de manera inequívoca) que el diseñador de la BD ha escogido (porque considera más óptimas) para identificar las entidades de un conjunto de entidades. Por supuesto, en caso de que una relación posee una única clave candidata, esta será la clave primaria; si, por el contrario, la entidad ofrece varias claves candidatas, el diseñador de la BD tendrá que tomar tal decisión.

3.2.- Diagramas Entidad-Relación.

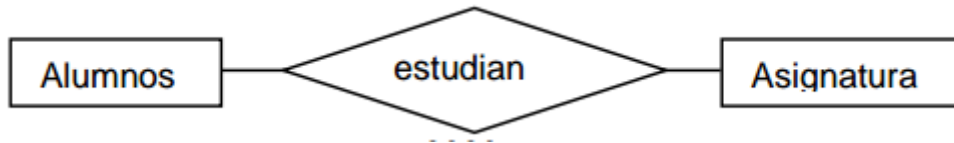
Las **entidades** se representan mediante un rectángulo, en caso de entidades fuertes y de doble rectángulo, si esta es débil.

A su vez, las **relaciones** se representan mediante un rombo, que une la entidad o las entidades involucradas en dicha relación.

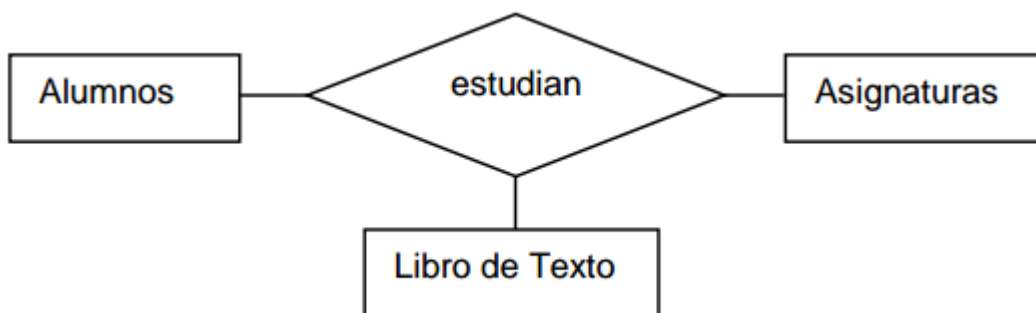
Relaciones de grado 1 (reflexivas): Una relación Es continuación que especifique si una/s Asignatura/s son continuación de otra.



Relaciones de grado 2 (binarias): El ejemplo de la relación Estudian, entre las entidades Alumnos y Asignaturas.



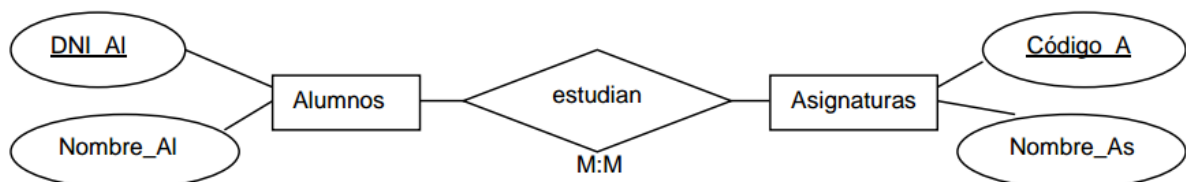
Relaciones de grado superior a 2 (ternarias, cuaternarias, etc.): Ejemplo en el que involucramos otra entidad Libro de Texto, y una relación que asocia qué Alumno estudia qué Asignatura y, además, con qué Libro de Texto estudia.



Por su parte, los atributos también se dividían en tres tipos:

- Atributos de una entidad.
- Atributos que sirven para identificar tuplas, conocidos como clave primaria.
- Atributos de una relación, conocidos como atributos descriptivos.

Dichos atributos se representan mediante una elipse, subrayando el nombre del atributo, en caso de ser clave primaria.



Posteriormente, otros autores han ido incorporando mejoras al modelo Entidad-Relación original, dando lugar al modelo Entidad-Relación Extendido, el

cual incluye mecanismos como: generalización, especialización, categorización y agregación; con el objetivo de alcanzar una mejor representación del mundo real mediante bases de datos.

4.- Lenguajes. SQL

Cuando se ha terminado el diseño de la BD y se ha elegido el SGBD concreto en el que implementar la BD, hay que especificar los esquemas conceptual e interno de la base de datos, así como las correspondencias entre ellos.

En muchos SGBD en los que no existe una separación estricta entre niveles, el DBA (Administrador) y los diseñadores usan un mismo lenguaje, llamado lenguaje de definición de datos (DDL: data definition language) para especificar los dos esquemas.

El SGBD dispone de un compilador del DDL encargado de procesar sentencias escritas en DDL para identificar las descripciones de los elementos de los esquemas y almacenar dicha descripción del esquema en el diccionario de datos del SGBD.

LENGUAJE ANFITRIÓN: es el lenguaje que se encarga de todos los aspectos de la base de datos, tanto los relacionados directamente con la misma, como los no relacionados (por ejemplo, variables locales -temporales-, operaciones de cálculo, lógica condicional, etc).

DDL Externo

Toda la vista externa se define mediante un *esquema externo*, para lo que se emplean las sentencias específicas de DDL para este fin.

DDL Conceptual

Mediante este tipo de declaraciones, se consigue implementar el *esquema conceptual* de la base de datos, logrando la independencia de los datos, las cuales no deben implicar consideraciones sobre las estructuras de almacenamiento de los datos.

DDL Interno

En la definición del *esquema interno*, el cual no solo define los distintos tipos de datos almacenados, sino también especifica índices, la representación de los campos almacenados, la secuencia física en que se almacenan los datos, etc.

4.1.- Sentencias SQL para definición de datos

En lenguaje SQL¹ (Structured Query Language) encontramos diversas sentencias correspondientes al DDL:

- CREATE TABLE
- RENAME TABLE
- DROP TABLE
- ALTER TABLE
- CREATE VIEW
- DROP VIEW
- ALTER VIEW
- CREATE INDEX
- DROP INDEX

Además, mediante los modificadores adecuados podemos especificar las restricciones necesarias sobre cada elemento, como podría ser las **restricciones sobre columnas**:

- NOT NULL: indica que la columna no puede tener valores nulos.
- UNIQUE: indica que la columna no puede tener valores repetidos. Es una clave alternativa.
- PRIMARY KEY: indica que la columna no puede tener valores repetidos ni nulos. Es la clave primaria.
- REFERENCES <tabla> [<columna>]: en este caso, la columna es la clave foránea de la columna de la tabla especificada.
- CHECK (condiciones): la columna debe cumplir las condiciones..

También existen **restricciones sobre tablas**:

- UNIQUE <nombre_campo/s>: indica que el conjunto de las columnas especificadas no puede tener valores repetidos. Es una clave alternativa.
- PRIMARY KEY <nombre_campo/s>: El conjunto de columnas o campos especificados no pueden tener valores nulos ni repetidos. Es una clave primaria.
- FOREIGN KEY <nombre_campo/s> REFERENCES <nombre_tabla> (<nombre_campo/s2>): Indica que el conjunto de los campos especificados

¹ Manual de Referencia MySQL 8.0 en <https://dev.mysql.com/doc/refman/8.0/en/>

es una clave foránea que referencia la clave primaria formada por el conjunto de las columnas2 de la tabla dada. Si las columnas y las columnas2 se denominan exactamente igual, entonces no sería estrictamente necesario poner columnas2 (nosotros lo haremos siempre).

- CHECK (condiciones): La columna debe cumplir las condiciones
- especificadas.

5.- Diccionario de datos

El diccionario de datos es una referencia de datos sobre los datos (metadatos). El SGBD debe ser capaz de proporcionar, de forma automática, toda la información sobre la organización de las bases de datos y su funcionamiento. En el diccionario de datos se almacenarán:

- Las descripciones interna, conceptual y externa de la base de datos, así como las restricciones definidas sobre ella.
- Las descripciones de los campos, registros y referencias cruzadas entre registros. Información sobre la seguridad de los datos.
- Los esquemas externos que son empleados por cada aplicación, los usuarios autorizados y los permisos asignados.

En los SGBD relacionales recibe el nombre de **catálogo**. Además, este catálogo puede ser accedido a través del mismo lenguaje con el que se gestiona el SGBD.

Las anotaciones contenidas en el DD pueden ser:

- ★ Información de los elementos: nombre, tipo de dato, formato, tamaño, alias, dónde se usa, descripción detallada, información adicional (restricciones, limitaciones, condiciones, usuarios permitidos, etc...)
- ★ Información sobre los esquemas correspondientes a los tres niveles de descripción de datos, además de estadísticas de uso y espacio ocupado por los datos.

6.- Conclusiones

Podemos remontar los orígenes de las bases de datos a la antigüedad, donde ya existían bibliotecas con ciertos sistemas de catalogación de obras y relacionadas con otros ámbitos prácticos como la recogida y archivo de información sobre cosechas o censos. Sin embargo, el gran problema de esos sistemas de almacenamiento de datos primitivos residía en que las búsquedas eran lentas y

poco eficaces, ya que todos los procesos de búsqueda, consulta, actualización, etc. eran manuales.

De este modo, el término de bases de datos siempre ha estado relacionado con el de informática, a partir de la necesidad de almacenar grandes cantidades de datos, para su posterior consulta. Así, diferentes autores sitúan la aparición del término de bases de datos, como hoy lo conocemos, a partir de las necesidades producidas por las nuevas industrias de la Revolución Industrial. Concretamente, tal informatización de las bases de datos se inició, en la década de los 50s, con la invención de las cintas magnéticas.

En el desarrollo de este tema hemos estado estudiando que la evolución del manejo de información está relacionada con el desarrollo de los Sistemas de Gestión de Bases de Datos (SGBD), pudiendo concluir las principales ventajas e inconvenientes del uso de bases de datos.

Comenzando por las **ventajas**, podemos destacar las siguientes:

- ❖ **Independencia física:** La forma de almacenar los datos (servidores locales, servidores “en la nube”, ordenador local, etc.) no influye en su manipulación lógica.
- ❖ **Independencia lógica:** Las aplicaciones que utilizan la base de datos no deben ser modificadas porque se modifiquen elementos de la base de datos.
- ❖ **Flexibilidad y Seguridad:** Las bases de datos ofrecen distintas vistas, en función de los usuarios y aplicaciones que trabajen sobre ellas, por lo que, a su vez, también restringen el acceso a los elementos que no se deban mostrar.
- ❖ Del mismo modo, las bases de datos permiten el acceso simultáneo a los datos, facilitando el control de **acceso concurrente**.
- ❖ **Uniformidad:** Las estructuras lógicas siempre tienen una única forma conceptual (las tablas).
- ❖ **Menor redundancia y mayor integridad** de datos: Las bases de datos reducen la repetición de datos y generan mayor dificultad de perder información. Todo ello reduce la probabilidad de incoherencias con los datos.
- ❖ Otras ventajas relacionadas con esto es la **reducción del espacio de almacenamiento** y el **acceso más eficiente** a la información.
- ❖ **Sencillez**.

Del mismo modo, entre las **desventajas** podemos destacar las siguientes:

- **Necesidad de personal cualificado:** Para aprovechar las ventajas

anteriormente comentadas y en función de las necesidades de la base de datos, del software que funciona con ella y, en definitiva, de las necesidades de la empresa que la explota (número de perfiles de usuarios distintos, necesidades de mantenimiento, backups, velocidad de respuesta, etc.), podría ser necesaria la especialización de personal o la formación de programadores y analistas sobre las posibilidades y las

- limitaciones de las bases de datos.
- Requerimientos adicionales hardware:** Generalmente la implantación de un SGBD recomienda la adquisición de equipos hardware adicionales, tales como servidores locales, ampliación de memoria, discos duros, servidores en la nube, etc.
- Por este motivo, antes de implantar un SGBD es preciso analizar si realmente es necesario. En ocasiones, si se tienen pocos datos, son usados por un único usuario simultáneamente y no hay que realizar consultas complejas sobre dichos datos, quizás sea mejor solución usar otras alternativas.
- Falta de rentabilidad a corto plazo:** Debido a los altos costes software, hardware y de personal en el momento de la implantación de la base de datos.
- Ausencia de estándares reales, que se traduce en excesiva dependencia de los sistemas comerciales del mercado (este problema cada vez es menor).

6.1.- Otras consideraciones

Debido a las restricciones de tiempo en este tipo de pruebas, se ha tenido que llevar a cabo una selección de los contenidos del presente tema, por lo que, a modo de comentario, voy a exponer algunos contenidos directamente relacionados con los expuestos:

- Modelo Entidad-Relación Extendido
- Modelo Relacional
- Lenguajes de manipulación de datos (DML)

6.2.- Ámbitos de aplicación

Técnico Superior en Administración de Sistemas Informáticos en Red (ASIR):

- Gestión de bases de datos.
- Administración de sistemas de gestores de bases de datos.

Técnico Superior en Desarrollo de Aplicaciones Multiplataforma (DAM):

- Bases de datos.

- Acceso a datos.

Técnico Superior en Desarrollo de Aplicaciones Web (DAW):

- Bases de datos.

Y, por último, también se trabajan dichos contenidos en las materias de

- Tecnologías de la Información y la Comunicación (en 4º de ESO) y
Tecnologías de la Información y la Comunicación I (en 1º de Bachillerato).

7.- Referencias bibliográficas

Martínez López, F.J. y Gallegos Ruiz, A. Programación de bases de datos relacionales. Ed. Ra-Ma.

Date, C.J. Introducción a los sistemas de bases de datos. Ed. Pearson. Prentice-Hall.

Korth, H.F., Silberschatz, A. Fundamentos de bases de datos. Ed. Mac GrawHill.

De la Peña, S. SGBD e instalación. Certificado de Profesionalidad. Ed. Paraninfo.

Hueso, L. Administración de Sistemas Gestores de Bases de Datos. Ed. Ra-Ma.