

CERTIFICADO

DESARROLLO DE APLICACIONES CON TECNOLOGÍA WEB

MODULO PROGRAMACION WEB EN ENTORNO SERVIDOR

- **Desarrollo de aplicaciones web en entorno servidor**
- **Acceso a datos en aplicaciones web del entorno servidor**
- **Desarrollo de aplicaciones web distribuidas**



TEMA 5: MODELO DE DATOS

CONCEPTO DE DATO. CICLO DE VIDA DE LOS DATOS

Un **modelo de datos** es un lenguaje orientado a hablar de una Base de datos
Típicamente un modelo de datos permite describir:

- estructuras de datos de la base: El tipo de los datos que hay en la base y la forma en que se relacionan.
- restricciones de integridad: Un conjunto de condiciones que deben cumplir los datos para reflejar la realidad deseada.

Operaciones de manipulación de los datos: típicamente, operaciones de agregado, borrado, modificación y recuperación de los datos de la base.

Otro enfoque es pensar que un **modelo de datos** permite describir los elementos de la realidad que intervienen en un problema dado y la forma en que se relacionan esos elementos entre sí.

TEMA 5: MODELO DE DATOS

CONCEPTO DE DATO. CICLO DE VIDA DE LOS DATOS

Un **modelo de datos** es un lenguaje que, típicamente, tiene dos sub lenguajes:

- Un *Lenguaje de Definición de Datos* o DDL (Data Definition Language), orientado a describir de una forma abstracta las estructuras de datos y las restricciones de integridad.
- Un *Lenguaje de Manipulación de Datos* o DML (Data Manipulation Language), orientado a describir las operaciones de manipulación de los datos.
- A la parte del DML orientada a la recuperación de datos, usualmente se le llama Lenguaje de Consulta o QL (Query Language).

Un **dato** es una representación simbólica (numérica, alfabética, algorítmica, espacial, etc.) de un atributo o variable cuantitativa o cualitativa.

TEMA 5: MODELO DE DATOS

CONCEPTO DE DATO. CICLO DE VIDA DE LOS DATOS

Los datos describen hechos empíricos, sucesos y entidades. Es un valor o referente que recibe el computador por diferentes medios, los datos representan la información que el programador manipula en la construcción de una solución o en el desarrollo de un algoritmo.

EN programación, un dato es la expresión general que describe las características de las entidades sobre las cuales opera un algoritmo.

estructura de datos, es la parte mínima de la información.



Un dato por sí mismo no constituye información, es el procesamiento de los datos lo que nos proporciona información.

TEMA 5: MODELO DE DATOS

CONCEPTO DE DATO. CICLO DE VIDA DE LOS DATOS

El ciclo de vida de los datos, se representa de esta manera



TEMA 5: MODELO DE DATOS

CONCEPTO DE DATO. CICLO DE VIDA DE LOS DATOS

CICLO DE VIDA DE LOS DATOS (explicación de imagen anterior):

- **Generación y Captura de Datos:** Introducción de datos en el sistema
- **Evaluación:** los datos son evaluados para comprobar si requieren de algún tipo de procesamiento
- **Destrucción:** Los datos que no tienen valor deben ser eliminados
- **Almacenamiento:** los datos con valor son almacenados
- **Transporte:** Los datos son transportados entre procesos
- **Recuperación:** el proceso de recuperación de datos es habitual
- **Clasificación:** los datos deben ser ordenados para que puedan ser procesados
- **Análisis:** los datos pueden ser analizados para su clasificación
- **Operación:** las operaciones que se pueden realizar sobre los datos
- **Síntesis:** Agrupación de los datos para darles un sentido dentro de un dominio
- **Utilización:** los datos procesados se convierten en información

TEMA 5: MODELO DE DATOS

Tipos de datos SQL: BASICOS

Tipo de Datos	Longitud	Descripción
BINARY	1 byte	Para consultas sobre tabla adjunta de productos de bases de datos que definen un tipo de datos Binario.
BIT	1 byte	Valores Si/No ó True/False
BYTE	1 byte	Un valor entero entre 0 y 255.
COUNTER	4 bytes	Un número incrementado automáticamente (de tipo Long)
CURRENCY	8 bytes	Un entero escalable entre 922.337.203.685.477,5808 y 922.337.203.685.477,5807.
DATETIME	8 bytes	Un valor de fecha u hora entre los años 100 y 9999.
SINGLE	4 bytes	Un valor en punto flotante de precisión simple con un rango de -3.402823×10^{38} a $-1.401298 \times 10^{-45}$ para valores negativos, 1.401298×10^{-45} a 3.402823×10^{38} para valores positivos, y 0.
DOUBLE	8 bytes	Un valor en punto flotante de doble precisión con un rango de $-1.79769313486232 \times 10^{308}$ a $-4.94065645841247 \times 10^{-324}$ para valores negativos, $4.94065645841247 \times 10^{-324}$ a $1.79769313486232 \times 10^{308}$ para valores positivos, y 0.
SHORT	2 bytes	Un entero corto entre -32,768 y 32,767.
LONG	4 bytes	Un entero largo entre -2,147,483,648 y 2,147,483,647.
LONGTEXT	1 byte por carácter	De cero a un máximo de 1.2 gigabytes.
LONGBINARY	Según se necesite	De cero 1 gigabyte. Utilizado para objetos OLE.
TEXT	1 byte por carácter	De cero a 255 caracteres.

TEMA 5: MODELO DE DATOS

Tipos de datos SQL: REGISTROS

- Los **registros** son un conjunto de datos que conforman una estructura.
Cada elemento de esta se denomina **campo** de registro.
- La unión de estos campos y sus distintos tipos determina el **tipo** de formato de registro.
- Por ejemplo un registro de un alumno incluirá los campos DNI, nombre, apellido y teléfono.
- Desde el punto de vista de una base de datos, **un registro es una fila** en una tabla de la base de datos.
- **Los tipos de datos dinámicos** son iguales Que los registros, salvo que estos pueden variar los campos que contienen para ajustarse a las necesidades temporales. Los campos de los registros , por el contrario, son siempre los mismos

TEMA 5: MODELO CONCEPTUAL

Un **modelo de datos** :lenguaje orientado a hablar de una Base de Datos.

Permite describir:

- Las **estructuras de datos** de la base:
El tipo de los datos que hay en la base y la forma en que se relacionan.
- Las **restricciones de integridad**:
Un conjunto de condiciones que deben cumplir los datos para reflejar algo
- Operaciones de **manipulación de los datos**: típicamente, operaciones de agregado, borrado, modificación y recuperación de los datos de la base.

Otro enfoque es pensar que un **modelo de datos** permite describir los elementos de la realidad que intervienen en un problema dado y la forma en que se relacionan esos elementos entre sí.

No hay que perder de vista que una **Base de Datos** siempre está orientada a resolver un problema determinado, por lo que los dos enfoques propuestos son necesarios en cualquier desarrollo de software.

TEMA 5: MODELO CONCEPTUAL

Un **modelo de datos** tiene dos sublenguajes:

- ◆ Un *Lenguaje de Definición de Datos* o **DDL** (Data Definition Language), orientado a describir de una forma abstracta las estructuras de datos y las restricciones de integridad.
- ◆ Un *Lenguaje de Manipulación de Datos* o **DML** (Data Manipulation Language), orientado a describir las operaciones de manipulación de los datos.
- ◆ A la parte del **DML** orientada a la recuperación de datos, usualmente se le llama Lenguaje de Consulta o QL (Query Language).

TEMA 5: MODELO CONCEPTUAL

Patrones de diseño: Una solución a un problema concreto. En el desarrollo de aplicaciones informáticas el ingeniero de software se encuentra a menudo con problemas comunes de diseño. Estos son documentados y sobre ellos se generan posibles soluciones, las cuales van evolucionando con nuevas soluciones o mejoras de las existentes.

Este proceso permite un nivel de reutilización a niveles de abstracción muy elevados con la consiguiente mejora en el desarrollo y mantenimiento del software

Por ejemplo, un patrón de diseño muy utilizado en el acceso a base de datos es el Data Access. Este patrón define un componente de software que suministra una interface común entre la aplicación y uno o más distintivos de almacenamiento de datos, tales como una base de datos o un archivo.

TEMA 5: MODELO CONCEPTUAL

MODELO GENÉRICO

Los modelos genérico son soluciones estándar que personalizamos para resolver un problema concreto

A diferencia de los patrones, estos son soluciones completas y se les llama frameworks.

Un ejemplo podría ser la realización de una tienda en internet

Una primera opción sería hacerla desde cero y otra utilizar un modelo genérico como prestashop y adaptarlo a nuestra tienda en internet

El grado de reutilización de estas soluciones permite una reducción drástica tanto en costes como en tiempo

TEMA 5: MODELO RELACIONAL

◆ DESCRIPCION

En este modelo todos los datos son almacenados en relaciones, y como cada relación es un conjunto de datos, el orden en el que estos se almacenen no tiene relevancia (a diferencia de otros modelos como el **jerárquico** y el de red). Esto tiene la considerable ventaja de que es más fácil de entender y de utilizar por un usuario no experto. La información puede ser recuperada o almacenada por medio de consultas que ofrecen una amplia flexibilidad y poder para administrar la información.

- ◆ **Este modelo considera la base de datos** como una colección de relaciones. De manera simple, una relación representa una tabla que no es más que un conjunto de filas, cada fila es un conjunto de campos y cada campo representa un valor que interpretado describe el mundo real. Cada fila también se denomina registro y a cada columna también se le puede llamar campo o atributo.

TEMA 5: MODELO RELACIONAL

ENTIDADES Y TIPOS DE ENTIDADES

- Una entidad es cualquier objeto (real o abstracto) o bien una persona, lugar cosa concepto o suceso.
- distinguible o inconfundible con otra (p.ej. el alumno Juan es distinto del alumno Jorge)
- Una entidad queda caracterizada o descrita por un conjunto de propiedades o atributos.
- Las entidades se representan en el Diagrama entidad-relación por un rectángulo con su nombre (sustantivo común en singular)

La decisión sobre si un determinado objeto o concepto debe ser una entidad dentro de Modelo Entidad-Relación uno es algo sencillo inicialmente. Por ejemplo el color es habitualmente una característica de una entidad (por ejemplo un producto, pero en una fábrica de pinturas probablemente sería una entidad)

TEMA 5: MODELO RELACIONAL

💧 Para ayudar a precisar el concepto de entidad, Tardieu propone tres reglas generales que debe cumplir una entidad:

1. Tiene que tener existencia en el universo
2. Cada instancia de un tipo de entidad debe poder distinguirse de las demás.
3. Todos los ejemplares de un tipo de entidad deben tener las mismas propiedades características.

Un atributo es una unidad de información que no se puede dividir y que almacena una propiedad de un entidad. Por lo tanto, una entidad esta descrita y se representa por sus propiedades o atributos. Por ejemplo, la entidad cliente tiene como propiedades: CIF, Nombre, Apellido, Fecha de nacimiento

TEMA 5: MODELO RELACIONAL

RELACIONES, CARDINALIDAD

RELACIÓN

- Se puede definir como la unión o enlace entre entidades dentro de un sistemas de información . Las entidades, por si solas, no caracterizan el sistema de información , es necesario establecer las relaciones entre ellas. Por ejemplo podemos establecer la correspondencia entre un cliente y su cuenta corriente asociada.
- Las interrelaciones se representan en el diagrama Entidad-Relación con un rombo y un nombre(verbo en singular) de la misma.

CARDINALIDAD

- Define el tipo de correspondencia o número de ocurrencias de una entidad que se interrelacionan con otra entidad. Puede ser: 1:1, 1:N, N:M, 0:M , 0:1, etc

TEMA 5: MODELO RELACIONAL

CLAVES, TIPOS DE CLAVES

- Una clave candidata es una agrupación de atributos que identifican sin ambigüedad y de forma unívoca a todas las posibles “tuplas” (registros) de una tabla. El número mínimo de atributos que la conforman es 1 y el máximo, el conjunto de atributos que tenga la tabla.
- Una relación puede tener más de una clave candidata, entre las cuales se debe distinguir:
 - Clave primaria o principal: se escoge de entre las claves candidatas para identificar las “tuplas” de la relación de forma unívoca.
 - Claves alternas o alternativas: el resto de claves candidatas no escogidas como clave primaria.

Se denomina clave ajena o foránea de una relación R , a un conjunto no vacío de atributos cuyo valores han de coincidir con los valores de la clave candidata de una relación

TEMA 5: MODELO RELACIONAL

NORMALIZACION. FORMAS NORMALES

- La normalización tiene por objetivo la eliminación de dependencias entre atributos. Además evita la redundancia y protege la integridad de los mismos. Se dice que un esquema de relación está en una determinada forma normal si satisface un conjunto determinado de restricciones sobre los atributos. Cuantas más restricciones menos será el número de relaciones que las satisfagan y viceversa

TEMA 5: CONSTRUCCION DEL MODELO LÓGICO DE DATOS

Vamos a ver ahora cómo se transforma el modelo conceptual, dado por el Diagrama Entidad-Relación, en modelo relacional.

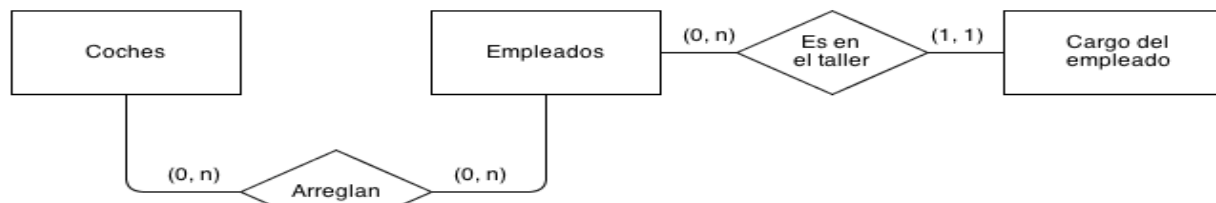
Es un proceso que se lleva a cabo siguiendo una pautas precisas.

Objetivo: crear un esquema relacional con todas las tablas y sus atributos correspondientes para que sean finalmente implementadas en un SGBD

💧 ESPECIFICACION DE TABLAS

Lo primero es obtener la tablas que compondrán el esquema relacional.

Sintaxis nombre_tabla (clave-primaria, campo1, campo2,.....campo n)
el nombre de la tabla no debe contener caracteres especiales ni espacios en blanco. Además se puede observar que el campo que ejerce como clave primaria aparecerá subrayado.



**Ejemplo de
diagrama
entidad-
relación**

TEMA 5: CONSTRUCCION DEL MODELO LÓGICO DE DATOS

Las relaciones dependen del tipo de cardinalidad. La transformación se realiza de la siguiente forma en función de esta.

- **Relaciones con cardinalidad. N:M** (varios a varios): Una entidad en A se puede relacionar con 0 o con muchas entidades en B y viceversa (ejemplo asociaciones-ciudadanos, donde muchos ciudadanos pueden pertenecer a una misma asociación, y cada ciudadano puede pertenecer a muchas asociaciones distintas).
- **Relación 1:N** (uno a varios): Un registro en una entidad en A se relaciona con cero o muchos registros en una entidad B. Pero los registros de B solamente se relacionan con un registro en A. (ejemplo: dos entidades, vendedor y ventas, con llaves primarias, codigo_vendedor y venta, respectivamente, un vendedor puede tener muchas ventas pero una venta solo puede tener un vendedor).
- **Relación 1:1** Un registro de una entidad A se relaciona con solo un registro en una entidad B. (ejemplo dos entidades, profesor y departamento, con llaves primarias, codigo_profesor y jefe_depto respectivamente, un profesor solo puede ser jefe de un departamento y un departamento solo puede tener un jefe).

TEMA 5: CONSTRUCCION DEL MODELO LÓGICO DE DATOS

◆ DEFINICION DE COLUMNAS

Los atributos , representados por elipses en el Diagrama Entidad-relación, pasan a ser campos de las tablas de las entidades a las que se caracterizan. Debe especificarse si la columna admitirá valores nulos o no y qué columnas formarán la clave primaria subrayando su nombre. NO se permitirán nombre de columnas repetidas.

- ◆ Los nombres tampoco deben tener espacios en blanco , ni caracteres especiales. Además sería recomendable que guardaran relación con el dato que almacena, para mayor claridad a la hora de realizar las consultas.
- ◆ Para crear las relaciones entre tablas , será necesario crear columnas adicionales para poder asociarlas con las claves principales de las tablas que relacionan, conforme se ha visto en el apartado anterior.

TEMA 5: CONSTRUCCION DEL MODELO LÓGICO DE DATOS

ESPECIFICACION DE CLAVES

- Como se ha visto anteriormente, cada tabla debe especificar su clave primaria y nombrar todas las claves foráneas de las tablas que relacionan. En apartados anteriores se especificaron las formas normales, en este se va a poner en valor esos conocimientos, a través de un conjunto de ejemplos para la creación de tablas.
- Primera forma Normal (1FN): una relación se encuentra en primera forma normal cuando no hay grupos repetitivos en sus atributos. Es decir no puede haber más de un valor por atributo

Cliente

ID Cliente	Nombre	Apellido	Teléfono
123	Rachel	Ingram	555-861-2025
456	James	Wright	555-403-1659 555-776-4100
789	Cesar	Dure	555-808-9633

En este caso se debe generar dos tablas diferentes (clientes y teléfono de clientes) con para cada teléfono de James, cada uno debe llevar su clave

TEMA 5: CONSTRUCCION DEL MODELO LÓGICO DE DATOS

Segunda forma Normal (2FN):

una relación se encuentra en segunda forma normal, si además de estar en 1FN, todos los atributos que no forman parte de ninguna clave candidata tienen dependencia funcional completa respecto de cada una de las claves.

En este caso se debe tener claro que todos los atributos dependan de la clave principal de forma completa

1° Se muestra una tabla con los datos de una biblioteca

BIBLIOTECA				
C_LIBRO	N_TITULO	N_AUTOR	C_LECTOR	N_LECTOR
10002	FISICA 1	ALEXANDER VASQUEZ	2596	CARLOS
10003	CALCULO 2	J. MILLER	4853	JUANA
10004	CONTABILIDAD	JOAQUIN LINO	2253	PABLO
10005	ECONOMIA	MARIA CASAS	5420	ANGELA

Como se puede ver en la tabla mostrada, el nombre del lector no está en dependencia funcional con el código del libro, por lo que debemos separar la tabla para que este en 2FN.

BIBLIOTECA		
C_LIBRO	N_TITULO	N_AUTOR
10002	FISICA 1	ALEXANDER VASQUEZ
10003	CALCULO 2	J. MILLER
10004	CONTABILIDAD	JOAQUIN LINO
10005	ECONOMIA	MARIA CASAS

LECTOR	
C_LECTOR	N_LECTOR
2596	CARLOS
4853	JUANA
2253	PABLO
5420	ANGELA

TEMA 5: CONSTRUCCION DEL MODELO LÓGICO DE DATOS

Tercera forma Normal (3FN):

una relación se encuentra en segunda forma normal, si además de estar en 2FN, todos los atributos que no forman parte de ninguna clave candidata facilitar información solo acerca de las claves y no acerca de otros atributos.

La idea es eliminar las dependencias transitivas entre atributos que no son claves candidatas.

2ª Se muestra una tabla con el código de algunos cursos, el profesor que dicta tal curso y el aula donde lo hace

C_CURSO	N_PROFESOR	#_AULA
CI32	CARLOS LOPEZ	C-22
SW41	MARIO SOTO	A-15
NI85	LUIS RAMIREZ	D-58

Los atributos no clave como nombre de profesor y número de aula presentan dependencia transitiva, el nombre del profesor determina el número de aula que ocupará, es por ello que se aplica la 3FN.

C_CURSO	N_PROFESOR	N_PROFESOR	#_AULA
CI32	CARLOS LOPEZ	CARLOS LOPEZ	C-22
SW41	MARIO SOTO	MARIO SOTO	A-15
NI85	LUIS RAMIREZ	LUIS RAMIREZ	D-58

TEMA 5: CONSTRUCCION DEL MODELO LÓGICO DE DATOS

🟢 Forma Normal Boyce-Codd(FNBC):
una relación se encuentra en Boyce-Codd, si además de estar en 3FN, el conocimiento de las claves permite averiguar todas las relaciones existentes entre los datos de la relación.

1º Se presenta la siguiente tabla de restaurantes de pizza, el tipo y el lugar de envío:

<u>N- Restaurante</u>	<u>N-Tipo de pizza</u>	<u>N-Lugar de envío</u>
Pizza Hut	Americana	Surco
Pizza Hut	Hawaiana	Miraflores
Papa Johns	Americana	San Borja
Papa Johns	Hawaiana	San Luis

Se puede observar que hay dependencia multivalor entre el nombre del restaurante y el número de pizza, por ello se aplica la cuarta forma normal que daría paso a dos tablas:

<u>N- Restaurante</u>	<u>N-Tipo de pizza 1</u>	<u>N-Tipo de pizza 2</u>
Pizza Hut	Americana	Hawaiana
Papa Johns	Americana	Hawaiana

<u>N- Restaurante</u>	<u>N-Lugar de envío</u>
Pizza Hut	Surco
Pizza Hut	Miraflores
Papa Johns	San Borja
Papa Johns	San Luis

TEMA 5: CONSTRUCCION DEL MODELO LÓGICO DE DATOS

- Cuarta Forma Normal Boyce-Codd(4FN):

- Una relación está en 4FN si lo está en FNBC, y se asegura de que las dependencias multi-valuadas” independientes estén correctas y eficientemente representadas en un diseño de base de datos.

3º Se muestra una tabla de restaurantes de pizza, el tipo y lugar de envío

N_RESTAURANTE	T_TIPOPIZZA	L_LUGARENVIO
Dominos pizza	Hawaiiana	Surco
Pizza Hut	Americana	Miraflores
Pizza Hut	Americana	San Miguel
Dominos pizza	Hawaiiana	Ate

Se observa que hay una dependencia multivalor entre el nombre del restaurante y el número de pizza, por eso separamos la tabla para convertir la tabla a 4FN.

N_RESTAURANTE	L_LUGARENVIO
Dominos pizza	Surco
Pizza Hut	Miraflores
Pizza Hut	San Miguel
Dominos pizza	Ate

N_RESTAURANTE	T_TIPOPIZZA1	T_TIPOPIZZA2
Dominos pizza	Hawaiiana	Hawaiiana
Pizza Hut	Americana	Americana

Normalmente se produce en las relaciones con cardinalidad varios a varios y consiste en que entidades independiente no pueden ser almacenadas en la misma tabla.

TEMA 5: CONSTRUCCION DEL MODELO LÓGICO DE DATOS

Quinta Forma Normal Boyce-Codd(5FN):

- Una relación está en 4FN si y solo si está en 4FN, y cada dependencia de unión en ella está implicada por la claves candidatas.
- Ejemplo: profesores que solo pueden impartir unas asignaturas determinadas

EL MODELO FISICO DE DATOS.

FICHEROS DE DATOS

- ◆ los ficheros son la forma más primitiva de almacenar datos, aunque presentan algunos problemas como son: la gestión de la integridad de los datos, concurrencia...
- ◆ Un fichero es un conjunto de información relacionada, grabada en el sistema de **almacenamiento secundario y a la que se hace referencia mediante un nombre**

EL MODELO FISICO DE DATOS.

FICHEROS DE DATOS

◆ **TIPOS DE FICHEROS** Pueden organizarse en dos grandes categorías:

- ◆ **ficheros de texto (o ASCII)** : Con líneas de texto representadas en código ASCII

Ejemplo ficheros de programas fuente escritos en cualquier lenguaje de programación

Ventaja: son estándar para todo tipo de sistema operativo. Estos pueden ser editados por cualquier editor de textos básico

- ◆ **binarios** Son aquellos ficheros que tienen información codificada en lenguaje binario. Estos ficheros utilizan un formato que es específico de la aplicación que los genera.

Ejemplo típico de fichero binario son los ficheros ejecutables. También se utilizan mucho como ficheros multimedia, es decir para almacenar imágenes sonido y video

EL MODELO FISICO DE DATOS.

FICHEROS DE DATOS

MODOS DE ACCESO:

Los modos de acceso son las distintas formas que existen a la hora de leer la información que contiene un fichero. Los mas habituales son los siguientes:

- 💧 **ACCESO SECUENCIAL:** Como su propio nombre indica, se produce una lectura del fichero desde el principio hasta que se encuentra el dato buscado. Este método es muy poco eficiente, ya que en el peor de los casos tendríamos que leer todo el fichero para buscar un único registro
- 💧 **ACCESO DIRECTO:** La información se puede encontrar de una forma directa, sin necesidad de recorrer todo el fichero. Esto hace que este tipo de acceso sea optimo en la obtención de registros
- 💧 **ACCESO INDEXADO:** El acceso al fichero se produce a través de un índice, el cual tiene un enlace a la parte del fichero donde se encuentra el registro buscado

EL MODELO FISICO DE DATOS. FICHEROS DE DATOS

ORGANIZACIÓN DE FICHEROS

La elección de una determinada organización para un fichero es una decisión de diseño, que depende de ciertos parámetros como :

- 💧 Tipo de procesos por los que se va a ver afectado el fichero
- 💧 Factores que afectan al fichero que se quiere optimizar
- 💧 Parámetros del fichero
- 💧 Características del soporte donde va a ser almacenado el fichero

EL MODELO FISICO DE DATOS. FICHEROS DE DATOS

ORGANIZACIÓN DE FICHEROS

- La organización de un fichero puede ser de tres tipos:
 - Organización Serial proporciona registros físicos en serie, esto es que se registran uno detrás de otro y se acceden en ese orden. El ejemplo típico de soporte serial es la cinta magnética.
 - Organización Secuencial: surge a partir de la secuencia serial, introduciendo un orden en los registros. Esta se debe a la existencia de una clave de ordenación física.
 - Organización Direcccionada: los registros físicos pueden ser localizados , ya que poseen una dirección física en el soporte. Un ejemplo es el disco duro

TRANSFORMACION DE UN MODELO LÓGICO EN UN MODELO FISICO DE DATOS

- ◆ Decisiones que debe tomar el ingeniero que realiza esta labor:
 - ◆ Como convertir relaciones en tablas físicas, determinando el formato del fichero
 - ◆ Qué organización deben tener los ficheros
 - ◆ Cómo organizar las columnas de las tablas, sobre todo, las que contengan campos claves
 - ◆ Qué índices deben definirse en las tablas
 - ◆ Que vistas deben definirse en las tablas
 - ◆ Cómo des normalizar las tablas
 - ◆ Como resolver relaciones de varios con varios
 - ◆ Que diseño pueden beneficiarse de acceso hash

El diseño físico está bajo continua supervisión para mejorar las características de rendimiento e integridad de los datos.

TEMA 5 HERRAMIENTAS PARA LA REALIZACIÓN DE MODELOS DE DATOS

- ◆ Herramientas que Permitan trabajar con los distintos modelos expuestos. La mayor parte de ellas trabajan por igual en los distintos sistemas operativos:
 - ◆ DIA
 - ◆ VISIO
 - ◆ MYSQL WORKBENCH
 - ◆ ORACLE SQL DEVELOPED

TEMA 6:DEFINICION DE SGBD

- ◆ Un sistema gestor de base de datos (SGBD) es un conjunto de programas que permiten el almacenamiento, modificación y extracción de la información en una BASE DE DATOS
- ◆ Las base de datos, además de proporcionar herramientas para añadir, borrar, modificar y analizar los datos. Los usuarios pueden acceder a la información usando herramientas específicas de interrogación y de generación de informes, o bien mediante aplicaciones al efecto.
- ◆ Estos sistemas también proporcionan métodos para mantener la integridad de los datos, para administrar el acceso de usuarios a los datos y para recuperar la información si el sistema se corrompe. Permiten presentar la información de la base de datos en variados formatos. La mayoría incluyen un generador de informes. También pueden incluir un módulo gráfico que permita presentar la información con gráficos y tablas

TEMA 6:DEFINICION DE SGBD

- ◆ Generalmente se accede a los datos mediante lenguajes de interrogación, lenguajes de alto nivel que simplifican la tarea de construir las aplicaciones. También simplifican la interrogación y la presentación de la información.
- ◆ Un SGBD permite controlar el acceso a los datos, asegurar su integridad, gestionar el acceso concurrente a ellos, recuperar los datos tras un fallo del sistema y hacer copias de seguridad. Las bases de datos y los sistemas para su gestión son esenciales para cualquier área de negocio, y deben ser gestionados con esmero.

TEMA 6: COMPONENTES DE UN SGDB. ESTRUCTURA

- Un SGBD se divide en módulos que tratan cada una de las responsabilidades del sistema general.
- Los componentes de un SGDB son:
 - Procesador de consultas.** Traduce las sentencias de un lenguaje de consultas a instrucciones de bajo nivel que puedan ser interpretadas por el gestor de la base de datos
 - Gestor de la base de datos.** Proporciona la interfaces entre los datos de bajo nivel almacenados en la propia base de datos y los programas de aplicación y las consultas que se hacen en el sistema
 - Gestor de archivos** Gestiona la asignación de espacio en la memoria del disco duro y de las estructuras de datos usadas para representar la información almacenada en disco
- Precompilador del lenguaje de Manipulación de Datos (DML) te permite consultar o manipular los datos, organizados por el modelo adecuado. El más popular es el SQL.

TEMA 6: COMPONENTES DE UN SGDB. ESTRUCTURA

- ◆ **Compilación del lenguaje de Definición de Datos** te permite llevar a cabo las tareas de definición de las estructuras que almacenarán los datos, así como los métodos o funciones que permitan consultarlos.
- ◆ **DML:** te permite consultar o manipular los datos, organizados por el modelo adecuado. El más popular es el SQL.
- ◆ **Gestor del diccionario de Datos**
El diccionario de datos es el lugar donde se guarda toda la información de todos los datos que forman la base de datos. Dentro de él se encuentra la lista de todos los elementos que forman parte del flujo de datos de todo el sistema. En su contenido están las descripciones de todos los demás objetos (archivos, programas,...) que existen en el sistema, almacena el conjunto numeroso de esquemas y especifica cada archivo y su ubicación, también incluye información acerca de qué programas utilizan qué datos, y a qué usuarios les interesa un informe u otro.
Está integrado dentro de la misma base de datos, puede tolerar descripciones de los modelos conceptual, lógico, interno y externo. Está guardado en un medio con acceso directo por si llegásemos a perder información poder recuperarla con facilidad.

TEMA 6: COMPONENTES DE UN SGDB. ESTRUCTURA

- ◆ **Control de autorización** . Comprueba los permisos de usuario antes de llevar a cabo la operación que solicita
- ◆ **Procesador de comandos** El sistema comprueba los permisos de usuario y pasa el control al procesador de comandos
- ◆ **Control de integridad**. Cuando se produce una modificación en los datos este módulo comprueba que esta operación satisface todas las restricciones de integridad
- ◆ **Optimizador de consultas** Este módulo optimiza las distintas estrategias para la ejecución de las consultas
- ◆ **Gestor de transacciones**. Este módulo realiza el procesamiento de las transacciones
- ◆ **Planificador** Gestiona los conflictos de las operaciones concurrentes sobre la base de datos

TEMA 6: COMPONENTES DE UN SGDB. ESTRUCTURA

- ◆ **Gestor de recuperación.** Gestiona la consistencia en el caso de que se genere algún error
- ◆ **Gestor de buffers** Gestiona la transferencia de datos entre la memoria principal y los dispositivos de almacenamiento secundario.

GESTION DE ALMACENAMIENTO

Los datos que maneja un SGDB son almacenados físicamente en algún tipo de soporte físico.

Las principales funciones de un gestor de almacenamiento son las siguientes:

- ◆ Comprobar que se satisfacen las restricciones de integridad y la autorización de los usuarios para acceder a los datos
- ◆ Asegurar que la base de datos se mantenga consistente a pesar de los fallos del sistema y que las ejecuciones de transacciones concurrentes se realicen sin conflictos. Controlar los sistemas de copia de seguridad y recuperación
- ◆ Gestionar la reserva de espacio de almacenamiento de disco y las estructuras de datos usadas para representar la información almacenada
- ◆ Responsable de traer los datos que se encuentran en memoria permanente a la memoria principal y de la gestión de esta última

TEMA 6: COMPONENTES DE UN SGDB. ESTRUCTURA

◆ **GESTION DE CONSULTAS**

LAS CONSULTAS SON EL MEDIO A TRAVÉS DEL CUAL LOS USUARIOS EXTRAEN LA INFORMACIÓN DE LAS BASES DE DATOS. LA GeSTION DE CONSULAS SIGUE LOS SIGUIENTES PASOS:

- ◆ El usuario envía una consulta, que es recibida por el servidor
- ◆ El servidor la valida sistemáticamente y posteriormente comprueba que los tipos de datos son correctos . A continuación se pasa al optimizador de consultas, el cual la optimiza antes de ejecutarla y finalmente esta es generada. Si en cualquier paso se produce un error, este es enviado al usuario

TEMA 6: COMPONENTES DE UN SGDB. ESTRUCTURA

MOTOR DE REGLAS

- ◆ Las reglas de Negocio o Conjunto de Regla de Negocio describe la políticas, normas, operaciones, definiciones y restricciones presentes en una organización y que son de vital importancia para alcanzar los objetivos. Las reglas de negocio definen los procesos de un sistema de información.
- ◆ En su forma más simple un montón de reglas de negocio está compuesto de tres elementos :
 - ◆ un conjunto de reglas,
 - ◆ la base de conocimientos (conocida como área de trabajo)
 - ◆ y el procesador de reglas. Este utilizará la base de conocimientos para decidir qué reglas deben activarse

TEMA 6: ADMINISTRACION DE UN SGDB

EL PAPEL DEL DBA

Un administrador de bases de datos (también conocido como DBA, en inglés database administrator) es aquel profesional que administra las tecnologías de la información y la comunicación, siendo responsable de los aspectos técnicos, tecnológicos, científicos, inteligencia de negocios y legales de bases de datos.

Sus tareas incluyen las siguientes:

- ◆ Implementar, dar soporte y gestionar bases de datos corporativas.
- ◆ Crear y configurar bases de datos relacionales.
- ◆ Ser responsables de la integridad de los datos y la disponibilidad.
- ◆ Diseñar, desplegar y monitorizar servidores de bases de datos.
- ◆ la distribución de los datos y las soluciones de almacenamiento.
- ◆ Garantizar la seguridad de las bases de datos, realizar copias de seguridad y llevar a cabo la recuperación de desastres.
- ◆ Planificar e implementar el aprovisionamiento de los datos y aplicaciones.
- ◆ Diseñar planes de contingencia.

TEMA 6: ADMINISTRACION DE UN SGDB

- ◆ Diseñar y crear las bases de datos corporativas de soluciones avanzadas.
- ◆ Analizar y reportar datos corporativos que ayuden a la toma de decisiones en la inteligencia de negocios.
- ◆ Producir diagramas de entidades relacionales y diagramas de flujos de datos, normalización esquemática, localización lógica y física de bases de datos y parámetros de tablas.

TEMA 6: ADMINISTRACION DE UN SGDB

GESTION DE INDICES

- ◆ Un índice es una estructura de datos que mejora la velocidad de las operaciones, permitiendo un rápido acceso a los registros de una tabla. Normalmente se utilizan sobre aquellos campos sobre los cuales se hacen las búsquedas mas frecuentes.
- ◆ Un índice funciona de forma similar al índice de un libro guardando parejas de elementos : el elemento que se desea indexar y su posición en el fichero de la base de datos. De tal forma que para buscar un elemento que esté indexado, solo hay que buscar en el índice dicho elemento, y una vez encontrado, obtener el registro correspondiente.
- ◆ El índice no deja de ser un fichero que debe estar actualizado y por tanto requiere de un mantenimiento.

TEMA 6: ADMINISTRACION DE UN SGDB

GESTION DE INDICES

- Podemos destacar dos tipos de índices generales
 - Hash** se refiere a una función o método para generar claves que representen de manera casi unívoca a un documento, registro, archivo, etc, y que resuma o identifique un dato a través de la probabilidad , utilizando una función hash o algoritmo hash
 - Arboles-Estructuras** de datos de árbol que se encuentran comúnmente en las implementaciones de bases de datos y un sistemas de archivos

TEMA 6: ADMINISTRACION DE UN SGDB

SEGURIDAD

- UN SGDB debe ser capaz de ofrecer unas herramientas de seguridad mínima: Gestionar los accesos de los usuarios, los permisos con respecto a los recursos de SGBS, las conexiones, las incidencias ante fallos del sistema, la seguridad y la integridad de los datos, sistemas de encriptación de datos

RESPALDOS Y REPLICACION DE BASES DE DATOS

- Los **respaldos** se refieren al mantenimiento de copias de seguridad de los datos. Dependiendo del sistema de información estas se realizan en periodos semanales, diarios o de horas. Los tipos de copias de seguridad que existen son : completos, incrementales, y diferenciales
- Replicación** consiste en duplicar el propio sistema en otro u otros equipos. Esto permite que exista redundancia de datos y se mejora la velocidad del propio sistema en general.

TEMA 6: ADMINISTRACION DE UN SGDB

- ◆ Replicación consiste en duplicar el propio sistema en otro u otros equipos. Esto permite que exista redundancia de datos y se mejora la velocidad del propio sistema en general. En la actualidad, con el paradigma de funcionamiento. Las ventajas que prestan son las siguientes:
 - ◆ Distribución de datos
 - ◆ Balanceo de carga
 - ◆ Copias de seguridad
 - ◆ Alta disponibilidad y recuperación de fallos

TEMA 6: GESTION DE TRANSACCIONES EN UN SGBD

- Una transacción en un Sistema de Gestión de Bases de Datos (SGBD), es un conjunto de órdenes que se ejecutan formando una unidad de trabajo.
- Un SGBD se considera transaccional, si es capaz de mantener la integridad de los datos, haciendo que estas transacciones no puedan finalizar en un estado intermedio. Cuando por alguna causa el sistema debe cancelar la transacción, empieza a deshacer las órdenes ejecutadas hasta dejar la base de datos en su estado inicial (llamado punto de integridad), como si la orden de la transacción nunca se hubiese realizado.
- Una transacción debe contar con ACID (un acrónimo inglés) que quiere decir: Atomicidad, Consistencia, Aislamiento y Durabilidad. Para esto, el lenguaje de consulta de datos

TEMA 6: GESTION DE TRANSACCIONES EN UN SGBD

COMONENTES E UN SISTEMAS DE TRANSACCIONES

SQL (Structured Query Language), provee los mecanismos para especificar que un conjunto de acciones deben constituir una transacción:

- BEGIN TRAN: Especifica que va a empezar una transacción.
- COMMIT TRAN: Le indica al motor que puede considerar la transacción completada con éxito.
- ROLLBACK TRAN: Indica que se ha alcanzado un fallo y que debe restablecer la base al punto de integridad.

TEMA 6: GESTION DE TRANSACCIONES EN UN SGBD

COMONENTES E UN SISTEMAS DE TRANSACCIONES

- Un **gestor transaccional** es un componente que procesa información descomponiendo la de forma unitaria en operaciones indivisibles, llamadas transacciones. Estas se diseñan para mantener a las bases de datos en un estado consistente, asegurando que todas las operaciones interdependientes se han completado con éxito o cancelado

TIPOS DE PROTOCOLOS DE CONTROL DE LA CONCURRENCIA

- El nivel e concurrencia viene dado por el número de transacciones activas en un instante dato. Es el parámetro más importante en el sistema de bases de datos distribuido.
- Los mecanismos de control de concurrencia buscan encontrar un balance entre el mantenimiento de la consistencia de la base de datos y el mantenimiento de un alto nivel de concurrencia

TEMA 6: GESTION DE TRANSACCIONES EN UN SGBD

RECUPERACION DE TRANSACCIONES

- Los métodos más habituales para recuperar la información y mantener constante la base de datos son :
 - Rollback. Consiste en registrar todos los estados intermedarios mientras la base de datos se modifica. En caso de error , estos sirven para devolver a la base de datos a un estado consistente
 - Rollforward consiste en mantener una copia de todas aquellas modificaciones realizadas sobre una base de datos
 - Bloqueos mutuos
cuando dos transacciones compiten en el mismo instante de tiempo por un recurso se produce un bloqueo muto, los sistemas de transacciones están diseñados para detectar este tipo de bloqueos cuando ocurren y actúan en concordancia. Generalmente, se cancelan las transferencias, se realiza un rollback de todos
 - Los cambios y luego se vuelve a ejecutar en un orden predeterminado

TEMA 6: SOLUCIONES DE SGBD

DISTRIBUIDAS

Una base de datos distribuida, es un conjunto de múltiples bases de datos lógicamente relacionadas, la cuales se encuentran distribuidas en diferentes espacios lógicos y geográficos e interconectados por una red de comunicaciones.

Dichas BDD tienen la capacidad de realizar procesamiento autónomo , lo que permite realizar operaciones locales o distribuidas.

Un sistemas de Bases de datos distribuido es un sistema en el que múltiples sitios de Bases de datos están ligados por un sistema de comunicaciones, de tal forma que un usuario en cualquier sitio puede acceder a los datos en cualquier parte de la red, exactamente igual que si se accediera a ellos de forma local

TEMA 6: SOLUCIONES DE SGBD

Los principales factores que distinguen un SBDD de un sistema centralizado son los siguientes:

Hay múltiples ordenadores, los cuales se denominan sitios o nodos

Estos nodos se comunican a través de una red de comunicaciones para transmitir datos y ordenes entre ellos

Sistemas comerciales altamente utilizados son los siguientes: Apache Cassandra, Oracle.....

ORIENTADAS A OBJETOS

- En este tipo de bases de datos la información se representa mediante objetos como en el paradigma orientado a objetos. Cuando se integran las características de una base de datos con las de un lenguaje de programación orientado a objetos, el resultado es un sistema gestor de base de datos orientada a objetos (ODBMS)
- Las bases de datos y los lenguaje orientados a objetos usan exactamente el mismo modelo .
- Ejemplo de sistemas de bases de datos orientado a objetos de código abierto : ONTOS, Objetivity....

6: SOLUCIONES DE SGBD

ORIENTADAS A DATOS ESTRUCTURADOS (XML)

las **bases de datos no relacionales** (NoSQL, este tipo de sistemas no requieren de esquemas fijos y está especialmente diseñadas para escalar horizontalmente. La mayor parte de ellas se basan en una estructura de clave-valor o bases de datos orientadas a documentos

ALMACENES DE DATOS (DATAWAREHOUSES):

- **ORIENTADO A TEMAS:** los datos en la base de datos están organizados de manera que todos los elementos de datos relativos al mismo evento u objeto del mundo real queden unidos entre sí
- **Variante en el tiempo** Los cambios producidos en los datos a lo largo del tiempo quedan registrados para que los informes que se pueden generar reflejen esas variaciones
- **No volátil** La información no se modifica, ni se elimina. Una vez almacenado un dato se convierte en información solo de lectura y se mantiene para futuras consultas
- **Integrado.** La base de datos contiene los datos de todos los sistemas operacionales de la organización y dichos datos deben ser consistentes.
- Los almacenes de datos contienen normalmente una gran cantidad de información. Por eso se subdividen en unidades lógicas más pequeñas. En este apartado IBM cuenta con un gran número de soluciones comerciales

TEMA 6: CRITERIOS PARA LA SELECCIÓN DE SGBD COMERCIALES

- Lo que mas influye son el tipo de datos, las aplicaciones utilizadas y el rendimiento requerido. También se incluye lógicamente , los costes asociados tanto del punto de vista del hardware como del software.
- Una lista de SGBD comerciales clasificados por tipo de licencia, podría ser la siguiente:
 - ✓ SGBD libres: postgresQL, MySQL, Firebird, SQLite, DB2Express-C, Apache-Derby
 - ✓ SGBD gratuitos. Microsoft SQL, Server Compact Edition, Sybase ASE Express Edition para linux
 - ✓ SGBD comerciales: dBase, Fox Pro, IBM DB2 Universal Database, IBMInformix, Microsoft Access, Microsoft SQL Server, Oracle, Pradox, Sybase

TEMA 7:

LENGUAJES DE GESTION DE BASES DE DATOS. EL LENGUAJE SQL

1. SQL:

lenguaje estándar ANSI/ISO

- Se encarga de la definición, manipulación y control de base de datos relacionales
- Muy similar al lenguaje natural
- Estándar que utilizan los SQBD relacionales
- Operaciones de SQL: reciben el nombre de sentencias
- Las sentencias están formadas por cláusulas

TEMA 7:

LENGUAJES DE GESTION DE BASES DE DATOS. EL LENGUAJE SQL

2.CREACION DE BASES DE DATOS

- **CREATE SCHEMA**: para la creación de bases de datos.

Sintaxis:

CREATE SCHEMA {[nombre_esquema]} | [AUTHORIZATION usuario]][lista_de_elementos_del_esquema];

Hay que tener en cuenta que:

Las palabras en negrita son palabras reservadas del lenguaje SQL

La notación entre corchetes [..] quiere decir que es un elemento opcional

{A|...|B} quiere decir que tenemos que elegir una de las opciones de forma obligatoria de entre todas las posibles

La lista de elementos que se puede añadir al esquema no son únicamente tablas, también se pueden añadir dominios, vistas, privilegios....

TEMA 7:

LENGUAJES DE GESTION DE BASES DE DATOS. EL LENGUAJE SQL

- DROP SCHEMA: Sentencia para el borrado de BBDD
Sintaxis:
DROP SCHEMA nombre_esquema {RESTRICT|CASCADA};
OPCIONES DE BORRADO
 - Restrict: lo borra solo si no contiene elementos.
 - Cascade: lo borra en cualquier caso..

TEMA 7:

LENGUAJES DE GESTION DE BASES DE DATOS. EL LENGUAJE SQL

CREATE TABLE: Sentencia para la creación de tablas
la tabla, es la estructura de almacenamiento de los datos del modelo relacional
Sintaxis:

```
CREATE TABLE nombre_tabla (nombre_columna {tipo_datos | dominio} [def_defecto]  
[restrict_col]
```

```
[,nombre_columna {tipo_datos | dominio} [def_defecto] [restrict_col]...
```

```
[,restricciones_tabla]);
```

Esta sentencia tiene los siguientes elementos: el nombre de la tabla(nombre_tabla). Y los atributos que conforman las columnas de la tabla (nombre_columna). Cada columna tiene un tipo de datos o un dominio asignado. También se le puede asignar definiciones por defecto y restricciones de columna.

TEMA 7:

LENGUAJES DE GESTION DE BASES DE DATOS. EL LENGUAJE SQL

SQL permite definir dominios específicos. Para crear un dominio se utiliza la sentencia `CREATE DOMAIN nombre dominio [AS] tipos_datos [def_defecto] [CONSTRAINT nombre_restricción] CHECK (condiciones) ;`

Un ejemplo podría ser la creación de los distintos departamentos que tiene una empresa (ej libro)

De esta forma cuando se define en la tabla departamentos el campo `tipo_departamento`, lo único que tendremos que indicar es el tipo `dom_departamentos`. Para borrar un dominio definido por el usuario se utiliza la sentencia `drop domain` que tiene este formato

`DROP DOMAIN nombre_dominio {RESTRICT|CASCADA};`

`restricti`: lo borra sólo si no contiene elementos

`cascade`: lo borra en cualquier caso

TEMA 7:

LENGUAJES DE GESTION DE BASES DE DATOS. EL LENGUAJE SQL

Para modificar un dominio semántico es necesario utilizar la sentencia alter domain. Veamos su formato

```
ALTER DOMAIN nombre_dominio {SET defec_defecto | DROP DEFAULT} | {ADD restricciones_dominio | DROP CONSTRAINT nombre_restricción}};
```

TEMA 7:

LENGUAJES DE GESTION DE BASES DE DATOS. EL LENGUAJE SQL

DEFINICIONES POR DEFECTO

La opción `def_defecto` permite especificar valores por omisión. Por ejemplo si aún descocemos la ubicación de un determinado departamento se puede incluir una por defecto (`default "edificio1"`), o bien definir la opción de valor nulo (`null`) de la siguiente forma (`default null`). En el caso de elegir esta última opción, la columna de ubicación tiene que poder admitir valores nulos.

La opción `default` tiene el siguiente formato

`DEFAULT (literal | función | NULL)`

TEMA 7:

LENGUAJES DE GESTION DE BASES DE DATOS. EL LENGUAJE SQL

DEFINICIONES Y CREACION DE INDICES. CLAVES PRIMARIAS Y EXTERNAS

Los índices:

permiten obtener los datos de una tabla de forma mas rápida, puesto que si se quiere localizar una parte concreta, no hará falta recorrerlo entero.

Funcionan de forma parecida al índice de un libro. lo que se hace es que se consulta el índice para ubicar la página donde está la información buscada y luego se va a ésta. Por tanto, la utilización de índices hace las búsquedas más eficientes.

Un Sistema Generador de Base de Datos, utiliza los mismo principios para obtener la información almacenada en una tabla de la base de datos. Primero usa el índice para ubicar la fila o filas que contienen los datos y posteriormente se dirige a ellas para devolverlas.

Para una tabla, lo que hace el SGBD es una consulta. Por lo tanto es recomendable crear índices en las tablas.

TEMA 7:

LENGUAJES DE GESTION DE BASES DE DATOS. EL LENGUAJE SQL

El índice puede cubrir una o más columnas. La sintaxis general es la siguiente:

```
CREATE INDEX "nombre_indice" ON "nombre_tabla" (NOMBRE_COLUMNNA);
```

CLAVE PRIMARIA:

Se utiliza para identificar de forma única cada fila de una tabla. El campo seleccionado como clave primaria puede ser un dato del registro, como por ejemplo el DNI, o puede ser un campo creado de forma artificial, por ejemplo, cod_departamento. Una clave primaria puede contener más de un campo de la tabla. Cuando se utilizan múltiples campos como clave primaria, se los denomina claves compuestas. Las clave primarias pueden especificarse cuando se crea la tabla (utilizando create table o cambiando la estructura existente de la tabla (utilizando alter table). A continuación se presentan ejemplos para la especificación de una clave primaria cuando se crea una tabla:

```
CREATE TABLE departamento (cod_departamento integer, nombre_departamento  
varchar (30), dirección_departamento varchar (30), PRIMARY KEY (cod_departamento);
```

TEMA 7:

LENGUAJES DE GESTION DE BASES DE DATOS. EL LENGUAJE SQL

CLAVE EXTERNA:

Una clave externa es un campo o campos que señala la clave primaria de otra tabla.

Propósito: asegurara la integridad referencial de los datos . Es decir solo podemos encontrar valores en la clave ajena existentes en la comuna que tiene la clave principal en la otra tabla

Ej: si tenemos dos tablas : departamento que incluye todos los datos de los departamentos existentes y otra de profesor, que incluye el departamento en el que trabaja como clave externa. En este caso se deben asociar los profesores con el departamento en el que realizan su labor. Para ello se colocará una clave externa en la tabla profesor y se relacionará con la clave primaria de la tabla departamento

De esta forma, se asegura que todos los departamentos que puedas ser asociados a un docente existen en la tabla departamento

TEMA 7:

LENGUAJES DE GESTION DE BASES DE DATOS. EL LENGUAJE SQL

ENLACE ENTRE BASES DE DATOS :

En ocasiones, se producen consultas en las que intervienen dos tablas de bases de datos distintas

```
SELECT * FROM base1.dbo.tabla t1
```

```
INNER JOIN base2.dbo.otratabla t2
```

```
ON t1.clave = t2.clave
```

El usuario debe tener permisos sobre los objetos de ambas bases de datos

TEMA 7:

LENGUAJES DE GESTION DE BASES DE DATOS. EL LENGUAJE SQL

3.GESTION DE REGISTROS DE TABLAS

INSERT INTO VALUES para insertar,

UPDATE SET WHERE para modificar,

DELETE FROM WHERE para borrar

¿QUÉ ES UNA CONSULTA?

Se trata de una solicitud al SGBD para que nos devuelva los datos almacenados en una o varias tablas:

Una consulta se puede realizar de distintas formas que veremos en los siguientes apartados:

SELECT permite seleccionar campos de una tabla, Y SELECT FROM WHERE permite seleccionar una fila concreta de la tabla

TEMA 7:

LENGUAJES DE GESTION DE BASES DE DATOS. EL LENGUAJE SQL

4. CONSULTAS

Una consulta es una solicitud al SGBD para que nos devuelva los datos almacenado en una o varias tablas. Una consulta se puede realizar de distintas formas que veremos en los siguientes apartados.

Estructura de una consulta

- Consultas sobre una tablas se realizarán con la sentencia select from
- La sentencia select from punto en la clausula where permiten seleccionar una fila concreta de la tabla

Selección de columnas. Obtención de valores únicos

- Para evitar filas resultantes repetidas se utiliza la palabra clave disitnct después de select. La opción por defecto es all, donde se muestran todas las filas con y sin repetición.

El formato es el siguiente:
SELECT

DISTINCT

FROM

WHERE

TEMA 7:

LENGUAJES DE GESTION DE BASES DE DATOS. EL LENGUAJE SQL

SELECCIÓN DE TABLAS. ENLACES ENTRE TABLAS

La mayor parte de las consultas que se realizan sobre el SGBD incluyen mas de una tabla. EN SQL esto se realiza especificando las tablas en la clausula from

```
SELECT profesor.cod_profesor.nombre, departamento.nombre_departamento
```

```
FROM profeor, departamento
```

```
WHERE profesor.cod_departamento=departamento.cod_departamento AND  
profesor.cod_profesor=100;
```


TEMA 7:

LENGUAJES DE GESTION DE BASES DE DATOS. EL LENGUAJE SQL

CONDICIONES, FUNCIONES ÚTILES EN LA DEFINICIÓN DE CONDICIONES

A continuación, se va a estudiar un conjunto de predicados muy utilizados en la definición de condiciones

BETWEEN. Para expresar una condición que quiere encontrar un valor entre dos limites concretos. Su formato es el siguiente:

```
SELECT  
FROM  
WHERE columna BETWEEN limite1 AND limite2;
```

IN
se utiliza para comprobar si un valor coincide con los elementos de una lista. En caso contrario se utilizara not in

```
SELECT  
FROM  
WHERE columna not in (valor1 .....,valorN);
```

TEMA 7:

LENGUAJES DE GESTION DE BASES DE DATOS. EL LENGUAJE SQL

CONDICIONES, FUNCIONES ÚTILES EN LA DEFINICIÓN DE CONDICIONES

LIKE

se utiliza para comprobar si una columna de tipo carácter cumple alguna propiedad determinada

SELECT

FROM

WHERE column LIKE características

Los patrones de SQL para expresar características son los siguientes

Se utiliza un símbolo para cada símbolo individual a considerar

Pondremos un símbolo % para expresar una cadena de símbolos, que no necesariamente tiene que ser un grupo

TEMA 7:

LENGUAJES DE GESTION DE BASES DE DATOS. EL LENGUAJE SQL

IS

NULL

Para comprobar si un valor es nulo, utilizaremos is null, y para averiguar si no lo es se utiliza is not null

ANY/SOME Y ALL

se utiliza para ver si una columna cumple en todas sus filas (all) o en (any/some) una condición

EXISTS

Se utiliza exists para comprobar si una sub-consulta produce alguna fila de resultados. En caso contrario se utilizara not exists

SIGNIFICADO Y USO DEL VALOR NULL

es un marcador especial utilizado en SQL para indicar que no existe un valor dentro de una base de datos.

La función es resolver que los SGDB puedan representar información desconocida o no aplicable. Es también una palabra reservada en el lenguaje SQL para identificar el marcador especial null

TEMA 7:

LENGUAJES DE GESTION DE BASES DE DATOS. EL LENGUAJE SQL

CONDICIONES, FUNCIONES ÚTILES EN LA DEFINICIÓN DE CONDICIONES

ORDENACION DEL RESULTADO DE UNA CONSULTA
para que los datos aparezcan ordenados , es preciso utilizar la clausula order by en la sentencia select.

SELECT

FROM

WHERE

ORDER BY

TEMA 7:

LENGUAJES DE GESTION DE BASES DE DATOS. EL LENGUAJE SQL

5. CONVERSION, GENERACION Y MANIPULACION DE DATOS

En los siguientes apartados se verán las funciones más habituales de manipulación y consulta de datos para distintos SGBD

FUNCIONES PARA LA MANIPULACION DE CADENAS DE CARACTERES

El procesamiento de cadenas de caracteres es una parte fundamental en el procesamiento de la mayoría de bases de datos. En los siguientes apartados se muestran las funciones utilizadas en SQL-99

- Delimitación de cadenas. La delimitación de una cadena se realiza con comillas, simples o dobles. Por ejemplo en Oracle son simples
- Concatenación de cadenas La concatenación d cadenas de caracteres se realiza en SQL y en Oracle con el operador ||
- Longitud de una cadena El cálculo de la longitud de una cadena de caracteres sigue el siguiente formato: Char_length(cadena)

TEMA 7:

LENGUAJES DE GESTION DE BASES DE DATOS. EL LENGUAJE SQL

- Extracción de una parte de una cadena Selecciona todos los caracteres de cadena comenzando con posición “pos”.

Substr (cadena, pos)

Conversión a mayúsculas y minúsculas

- Las funciones para convertir a mayúsculas y minúsculas son las siguientes:

upper(cadena)

lower(cadena)

TEMA 7:

LENGUAJES DE GESTION DE BASES DE DATOS. EL LENGUAJE SQL

- FUNCIONES PARA LA MANIPULACION DE NÚMEROS
 - Round(columna|expresión, n). Redondea a n posiciones decimales
 - Trunc (columna|expresión,n). trunca en la enésima posición decimal
 - Mod(m,n). Devuelve el resto de la division de m por n
- FUNCIONES DE FECHA Y HORA
 - Current_date() Para conseguir la fecha actual
 - Extract year from fecha date_part(year,fecha). Para extraer una parte de una fecha
 - To_char()Conversion de una fecha al formato deseado
- FUNCIONES DE CONVERSIÓN DE DATOS

TEMA 7:

LENGUAJES DE GESTION DE BASES DE DATOS. EL LENGUAJE SQL

- FUNCIONES DE CONVERSIÓN DE DATOS
 - La función cast de SQL permite convertir un tipo de dato a otro. Para ello debe existir ciertas congruencias en la naturaleza de los tipos. Por ejemplo convertir el varchar 'juan' a date no sería posible.

A continuación se muestran un conjunto de conversiones habituales

- De date a varchar y viceversa
- Conversión entre tipos numéricos
- Conversión entre intervalos temporales
- Cambiar un tipo numérico a un intervalo de tiempo o fracción
- De DATE a TIMESTAMP y viceversa

TEMA 7:

LENGUAJES DE GESTION DE BASES DE DATOS. EL LENGUAJE SQL

CONSULTAS MULTIPLES. UNIONES (JOINS)

- La operación join o combinacional permite mostrar columnas de varias tablas como si se tratase de una sola tabla, combinando entre si los registros relacionados usando par ello claves externa. Las tablas relacionadas se especifican en la clausula from y además hay que hacer coincidir los valores que relacionan las columnas de las tablas.
- Para evitar que se produzca como resultado el producto cartesiano de las dos tablas se utiliza la clausual where con el identificador del departamento, que es la clave existente en ambas
- DEFINICIÓN DE PRODUCTO CARTESIANO APLICADO A TABLAS
El producto cartesiano de dos tablas son todas las combinaciones de todas las filas de las dos tablas. Usando una sentencia select se encuentran todas las combinaciones de atributos posibles para ambas tablas. Los nombres de las tablas se indican en la clausula from separados por comas

TEMA 7:

LENGUAJES DE GESTION DE BASES DE DATOS. EL LENGUAJE SQL

- UNIONES DE TABLAS JOINS.
TIPOS: INNER, OUTER, SELF, EQUI, ETC...

En la versión SQL-92 se incorpora una nueva forma de realizar combinaciones utilizando la clausula JOIN en la clausula FROM. Permite que se relaciones de forma explicita ambas tablas sin involucrar la clave externa en las condiciones del select (o sea en el where) Es una manera más clara y limpia de llevar a cabo la relacion. Esto se puede ir aplicando a cuantas tablas se necesiten combinar en la consulta. En ocasiones una tabla puede unirse a si misma, produciéndose una auto-combinacion, self-join

- La combinación interna inner join esta solo se queda con las filas que tienen valores idénticos en la columnas de la tabla que compara . Las operaciones inner Join puede ser clasificada como de equivalencia, naturales y cruzadas

TEMA 7:

LENGUAJES DE GESTION DE BASES DE DATOS. EL LENGUAJE SQL

- Combinación interna natural (NATURAL JOIN) de dos tablas consiste básicamente , al igual que el álgebra relacional, en hacer una “equicombinacion” entre columnas del mismo nombre y eliminar las columnas repetidas
- De equivalencia
utiliza comparaciones de igualdad en el predicado join.
- Cruzada
presenta el producto cartesiano de todos los registros de las dos tablas

La combinación externa(outer join)
permite obtener todos los valores de la tabla que está a la derecha, los de la tabla que está a la izquierda o todos los valores de las dos tablas

Subconsultas

una sub-consulta es una consulta incluida dentro de una clausula WHERE o HAVING de otra consulta.

Si quisiéramos saber los códigos y los nombres de los proyectos de precio más elevado, en primer lugar tendríamos que encontrar los proyectos que tienen el precio más elevado

TEMA 7:

LENGUAJES DE GESTION DE BASES DE DATOS. EL LENGUAJE SQL

7.

AGRUPACIONES

Las siguientes cláusulas añadidas a select from permiten que las filas se puedan organizar en grupos:

Grupo by se denomina consulta agrupada ya que reúne los datos de la tabla origen y produce una única fila resumen por cada grupo formado

having. Nos permite seleccionar filas de la tabla resultante de una consulta de resumen

CONCEPTOS DE AGRUPACION DE DATOS

Las funciones de agrupación son utilizadas por los SGBD para operar sobre conjuntos de filas para dar un resultado por grupos. Existen dos tipos de funciones: funciones de fila única y funciones de agrupación

TEMA 7:

LENGUAJES DE GESTION DE BASES DE DATOS. EL LENGUAJE SQL

FUNCIONES

DE

AGRUPACION

Estas funciones se aplican a una sola columna, que especificaremos entre paréntesis, excepto la función count, que se puede aplicar a una columna, o indicar un “*”. La diferencia es que el count no cuenta los valores nulos para dicha columna

AGRUPACION

MULTICOLUMNA

Las composiciones multi_columna son menos habituales y se encuentran generalmente en consultas que afectan a claves foráneas compuestas

AGRUPACION

VIA

EXPRESIONES

Las funciones anteriores se utilizan en las consultas generando expresiones válidas para la manipulación de datos.

Usos correctos de estas expresiones podrías ser :

- Mezclar expresiones agregadas y no agregadas para agrupar columnas
- Utilizar mas de una expresión agregada en la clausula select

TEMA 7:

LENGUAJES DE GESTION DE BASES DE DATOS. EL LENGUAJE SQL

En muchas ocasiones estas expresiones se utilizan en forma incorrecta.
EJ de mal uso

- Una expresión agregada no puede aparecer en una clausula where
- Mezclar expresiones agregadas y no agregadas en un select

CONDICIONES DE FILTRADO DE GRUPOS

En ocasiones se quiere limitar el resultado según alguna de las funciones de agregación vistas anteriormente.

En vez de utilizar la clausula where en la instrucción SQL, se utilizará la clausula having con el siguiente formato:

La clausula having puede o no incluir la clausula group by

TEMA 7:

LENGUAJES DE GESTION DE BASES DE DATOS. EL LENGUAJE SQL

8.VISTAS

Concepto: Una vista es una consulta que se presenta como una tabla (virtual) a partir de un conjunto de tablas en una base de datos relacional.

Las vistas tienen la misma estructura que una tabla, filas y columnas. A diferencia de esta solo se almacena su definición y no los datos. Las operaciones que se pueden realizar son las de insertar , actualizar, borrar y seleccionar datos, igual que en una tabla normal.

CREACION, MODIFICADO Y BORRADO DE VISITAS

CREATE VIEW para crear una vista

with Check point para asegurarnos de que se cumpla la condición de la clausula where

TEMA 7:

LENGUAJES DE GESTION DE BASES DE DATOS. EL LENGUAJE SQL

9.FUNCIONES AVANZADAS

En los siguientes apartados se van a ver un conjunto de funciones avanzadas de las que disponen la mayoría de los SGBD

RESTRICCIONES, INTEGRIDAD DE BASES DE DATOS

Por integridad de datos nos referimos a la corrección y complementación de los datos en una base de datos . La modificación de los datos puede llevar consigo una pérdida de integridad de los mismos.

Ej: asociar un profe a un departamento que no existe

También pueden perderse datos debido a errores intrínsecos del propio sistema o simplemente a un fallo en el suministro eléctrico. Por tanto una de las funciones más importantes de un SGBD es presentar la integridad de los datos almacenados.

TEMA 7:

LENGUAJES DE GESTION DE BASES DE DATOS. EL LENGUAJE SQL

- Tipos de restricciones de integridad
 - datos requeridos: Establece que una columna tenga un valor distinto de null. Para ello se define que una columna sea not null en la creación de la tabla
 - Chequeo de validez. Cuando se crea una tabla cada columna tiene un tipo de datos concreto y SGBD deba asegurar que los datos que se introducen, sean solo de ese tipo
 - .integridad de entidad. La clave primaria de una tabla debe tener un valor único par cada fila de la tabla , si no, la base de datos perderá su integridad.
 - integridad referencial. Asegura la integridad entre las claves foráneas y primarias (relaciones padre/hijo).

TEMA 7:

LENGUAJES DE GESTION DE BASES DE DATOS. EL LENGUAJE SQL

- Existen cuatro operaciones que pueden corromper la integridad referencial.
 1. La inserción de una fila hijo se realiza sin que coincida la clave foránea y primarias con la clave principal del padre.
 2. La actualización en la clave foránea de la fila hijo donde se produce una actualización en la clave ajena de la fila hijo y la misma no coincide con ninguna clave principal del padre
 3. La supresión de una fila padre que tiene hijos asociados. Esto hace que los hijos queden huérfanos.
 4. La actualización de la clave primaria de una fila padre, lo que conlleva a que los hijos no la actualizan y queden huérfanos

TEMA 7:

LENGUAJES DE GESTION DE BASES DE DATOS. EL LENGUAJE SQL

- DISPARADORES

Un disparador o trigger es un programa almacenado, creado para ejecutarse automáticamente cuando ocurra un evento generado por las cláusulas insert, update y delete.

Se utilizan para mantener la integridad, calcular atributos derivados....

Para crear disparadores en el SGBD de MySQL se necesitan los privilegios super y trigger (formato en pagina 195)

Significado por orden:

- Indica al gestor de bases de datos qué usuario tiene privilegios en su cuenta, para la invocación de los triggers cuando surjan los eventos DML. Por defecto esta característica tiene el valor CURRENT_USER que hace referencia al usuario actual que está creando el trigger
- BEFORE/AFTER indica si el trigger se ejecuta antes o después del evento DML

TEMA 7:

LENGUAJES DE GESTION DE BASES DE DATOS. EL LENGUAJE SQL

- Aquí se elige que sentencia se utilizará para que se ejecute el trigger
- ON nombre de la tabla: El nombre de la tabla asociada
- FOR EACH ROW: establece que el trigger se ejecute por cada fila en la tabla asociada
- Bloque_de_instrucciones>define el bloque de sentencias que trigger ejecutará al ser invocado
- Identificadores new y old en triggers. Para relacionar el disparador con columnas específicas de una tabla deben usar los identificadores old y new.
OLD: indica el valor antiguo de la columna
NEW el valor nuevo.
- Para eliminar un trigger en MySQL, DROP TRIGGER

TEMA 7:

LENGUAJES DE GESTION DE BASES DE DATOS. EL LENGUAJE SQL

- En este apartado se van a ver las operaciones más básicas de gestión de usuario para el SGBD . Esto se va a realizar desde la linea de comandos shell.
- Crear un usuario. Comando create user dde MySQL, que nos va a permitir crear usuario y asignarles una contraseña con el parámetro identified by
- Eliminar un usuario con comando DROP USER
- Asignar privilegios al usuario con GRANT. Si se quiere crear un usuario y asignarle los privilegios a la vez se hará con el comando grant
- Visualizar privilegios de un usuario con SHOW GRANTS. El comando show grant s sirve para visualizar los privilegios asignados a un usuario
- Revocar privilegios de un usuario con revoke. Para eliminar privilegios al usuario
- Flush privileges este sirve para recargar la tabla de privilegios pero solo es necesario cuando se manipulan directamente las tablas de privilegios ejecutando insert, delete,en lugar de usar los comandos grant y revoke

TEMA 7:

LENGUAJES DE GESTION DE BASES DE DATOS. EL LENGUAJE SQL

OPTIMIZACIÓN

DE

CONSULTAS

depende de la arquitectura e la base de datos además de su propio desarrollo.

- Por ejemplo que los campos de las **tablas** tengan **tamaños** y **tipos de datos** adecuados permite que los volúmenes de las tablas sean inferiores, lo cual hace que la búsqueda sea más eficiente.
- Otro elemento fundamenta es **separar entre propiedades y características** , para hacer búsquedas independientes
- El **uso de índices** sobre tablas que se utilicen con mucha frecuencia es importante. OJO!!Sobrecargar el sistema con índices que no son necesarios hará que la base de datos sea mucho menos eficiente
- en cuanto a las consultas en si es importante **limitar las consultas a las necesidades específicas** de la información que hay que obtener. Realizar consultas del tipo selecto no es muy recomendable.
- también utilizar todas las funciones que los distintos SGBD ponen a nuestro alcance ya que normalmente suele ser un código que está previamente optimizado , y redundará en la velocidad de nuestras consultas