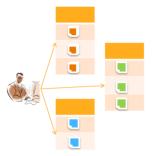
Definiciones de Base de Datos

Javier, CALLI OLVEA

DIFERENCIAS

- En cada una de nuestras actividades usamos y generamos información.
- La información puede ser usada ya sea en forma organizada o desorganizada, pero el hacerlo de esta última manera, no garantiza su





Organizada

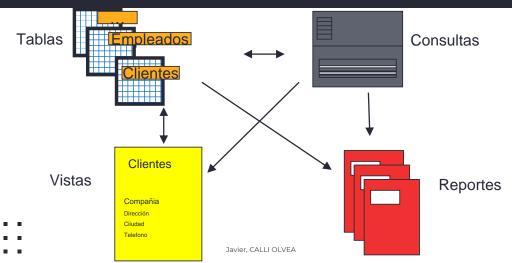
,

- Es un conjunto de datos relacionados entre sí.
- Una base de datos es un conjunto de información relacionada, almacenada sistemáticamente en forma organizada, en un medio electrónico.
- Los datos son hechos conocidos, que pueden registrarse y que tienen un significado implícito.

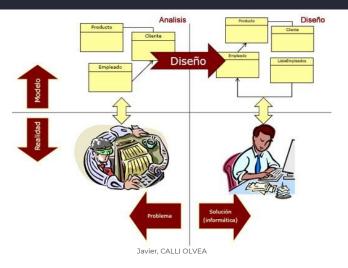
Javier, CALLI OLVEA

3

UNA BASE DE DATOS



DISEÑO DE BASES DE DATOS



Ę

PREGUNTAS QUE HACER

- ¿Qué tengo?
- -(Entradas)
- ¿Qué quiero?
- -(Resultados)
- ¿Qué tengo que hacer para llegar allí?
- -(Proceso)
- ¿Cómo voy a construirlo?
- -(Aplicación / Programa)



Javier, CALLI OLVEA

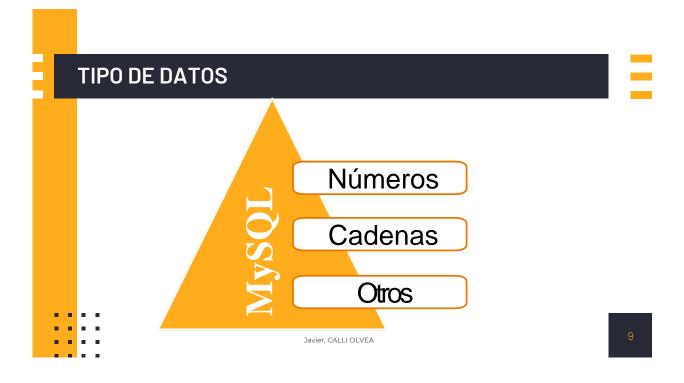


TIPO DE DATOS

Al diseñar nuestras tablas tenemos que especificar para cada campo, el tipo de datos y tamaño adecuado.

Una correcta Elección debe procurar que las tablas no se queden corta en su capacidad, que destine un tamaño apropiado a la longitud de los datos y la máxima velocidad de ejecución.

Javier, CALLI OLVEA



TIPO DE DATOS

Datos numéricos

- TINYINT
- SMALLINT
- MEDIUMINT
- INT o INTEGER
- BIGINT

Javier, CALLI OLVEA

TIPO DE DATOS

<u>Datos para guardar cadenas de caracteres (alfanuméricos)</u>

- CHAR
- VARCHAR
- BINARY
- VARBINARY
- TINYBLOB
- TINYTEXT

- BLOB
- TEXT
- MEDIUMBLOB
- MEDIUMTEXT
- LONGBLOB
- LONGTEX
- ENUM
- SET

Javier, CALLI OLVEA

1

TIPO DE DATOS

<u>Datos para almacenar fechas y</u> horas

- DATE
- DATETIME
- TIME
- TIMESTAMP
- YEAR

Atributos de los campos

- NULL
- DEFAULT
- BINARY
- INDEX
- PRIMARY KEY
- AUTO_INCREMENT
- UNIQUE
- FULLTEXT

Javier, CALLI OLVEA

En este tipo de campos solo pueden almacenarse números, positivos o negativos, enteros o decimales, en notación hexadecimal, científica o decimal.

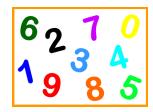
Los tipos numéricos tipo integer admiten los atributos **SIGNED** y **UNSIGNED** indicando en el primer caso que pueden tener valor negativo, y solo positivo en el segundo.

Javier, CALLI OLVEA

13

DATOS NUMÉRICOS

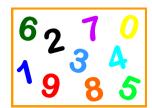
Los tipos numéricos pueden además usar el atributo **ZEROFILL** en cuyo caso los números se completaran hasta la máxima anchura disponible con ceros.



Ejemplo: CP INT(5) zerofill => valor 23 se almacenará como 00023

BIT o **BOOL**, para un número entero que puede ser 0 ó 1

TINYINT es un número entero con rango de valores válidos desde -128 a 127. Si se configura como unsigned (sin signo), el rango de valores es de 0 a 255



Javier, CALLI OLVEA

15

DATOS NUMÉRICOS

SMALLINT, para números enteros, con rango desde -32768 a 32767. Si se configura como unsigned, 0 a 65535.

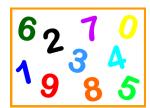
6270 1985

MEDIUMINT para números enteros; el rango de valores va desde -8.388608 a 8388607. Si se configura como unsigned, 0 a 16777215

Javier, CALLI OLVEA

INT para almacenar números enteros, en un rango de -2147463846 a 2147483647. Si configuramos este dato como unsigned, el rango es 0 a 4294967295

BIGINT número entero con rango de valores desde -9223372036854775808 a 9223372036854775807. Unsigned, desde 0 a 18446744073709551615.

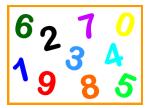


Javier, CALLI OLVEA

17

DATOS NUMÉRICOS

FLOAT (m,d) representa números decimales. Podemos especificar cuantos dígitos (m) pueden utilizarse (término también conocido como ancho de pantalla), y cuantos en la parte decimal (d). Mysql redondeará el decimal para ajustarse a la capacidad.



10

DOUBLE Número de coma flotante de precisión doble. Es un tipo de datos igual al anterior cuya única diferencia es el rango numérico que abarca

6270 1985

DECIMAL almacena los números como cadenas.

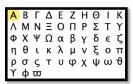
Javier, CALLI OLVEA

19

CARACTERES O CADENAS

CHAR Este tipo se utiliza para almacenar cadenas de longitud fija. Su longitud abarca desde 1 a 255 caracteres.

VARCHAR Al igual que el anterior se utiliza para almacenar cadenas, en el mismo rango de 1-255 caracteres, pero en este caso, de longitud variable



20

CARACTERES O CADENAS

campo Nota: un CHAR ocupará siempre el máximo de longitud que le hallamos asignado, aunque el tamaño del dato menor (añadiendo sea espacios adicionales que sean precisos). Mientras que VARCHAR solo almacena la longitud del dato. permitiendo que el tamaño de la base de datos sea menor. Eso si, el acceso a los datos CHAR es mas rápido que VARCHAR. Javier, CALLI OLVEA

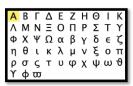


21

CARACTERES O CADENAS

Text es para cadenas de texto plano (sin formato) y case-insensitive (sin distinguir mayúsculas o minúsculas).

Blob se usa para objetos binarios, cualquier tipo de datos o información, desde un archivo de texto con todo su formato (se diferencía en esto de el tipo Text) hasta imágenes, archivos de sonido o video.



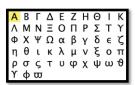
22

CARACTERES O CADENAS

TINYTEXT, **TINYBLOB** para un máximo de 255 caracteres.

TEXT y **BLOB** para cadenas con un rango de 255 - 65535 caracteres.

MEDIUMTEXT, MEDIUMBLOB textos de hasta 16777215 caracteres. LONGTEXT, LONGBLOB, hasta máximo de 4.294.967.295 caracteres



Javier, CALLI OLVEA

23

OTROS

DATE para almacenar fechas. El formato por defecto es YYYY MM DD desde 0000 00 00 a 9999 12 31.

TIME almacena una hora. El rango de horas va desde -838 horas, 59 minutos y 59 segundos a 838, 59 minutos y 59 segundos. El formato de almacenamiento es de 'HH:MM:SS'



2/

OTROS

YEAR almacena un año. El rango de valores permitidos va desde 1901 al año 2155. El campo puede tener tamaño 2 o 4 dependiendo de si queremos almacenar el año con 2 o 4 dígitos.



TIMESTAMP Combinación de fecha y hora. El rango va desde el 1 de enero de 1970 al año 2037. El formato de almacenamiento depende del tamaño del campo

25

OTROS

DATETIME Combinación de fecha y hora. El rango de valores va desde el 1 de enero del 1001 a las 0 horas, 0 minutos y 0 segundos al 31 de diciembre del 9999 a las 23 horas, 59 minutos y 59 segundos.



El formato de almacenamiento es de año-mes-dia horas:minutos:segundos

26

OTROS

SET un campo que puede contener ninguno, uno ó varios valores de una lista. La lista puede tener un máximo de 64 valores.

ENUM es igual que SET, pero solo se puede almacenar uno de los valores de la lista.

Javier, CALLI OLVEA

27

2. CICLO DE VIDA DE UNA BD

BD - BASE DE DATOS

CICLO DE VIDA DE UNA BD

- Las etapas del ciclo de vida de una aplicación de bases de datos son las siguientes:
 - 1. Planificación del proyecto.
 - 2. Definición del sistema.
 - 3. Recolección y análisis de los requisitos.
 - 4. Diseño de la base de datos.
 - 5. Selección del SGBD.
 - 6. Diseño de la aplicación.

- 7. Prototipado.
- 8. Implementación.
- 9. Conversión y carga de datos.
- 10. Prueba.
- 11. Mantenimiento.

29

Javier, CALLI OLVEA

1. PLANIFICACIÓN DEL PROYECTO

Esta etapa conlleva la planificación de cómo se pueden llevar a cabo las eta del ciclo de vida de la manera más eficiente.

Hay tres componentes principales:

- el trabajo que se ha de realizar.
- los recursos para llevarlo a cabo
- el dinero para pagar por todo ello.

Normalmente, este modelo de datos se representa mediante un diagrama ENTIDAD - RELACIÓN.

La planificación de la base de datos también incluye el desarrollo de estándares que especifiquen cómo realizar la recolección de datos, cómo especificar su formato, qué documentación será necesaria y cómo se va a llevar a cabo el diseño y la implementación.

30

2. DEFINICIÓN DEL SISTEMA

En esta etapa se especifica el ámbito y los límites de la aplicación de bases de datos, así como con qué otros sistemas interactúa. También hay que determinar quienes son los usuarios y las áreas de aplicación.



31

3. RECOLECCIÓN Y ANÁLISIS DE LOS REQUISITOS

En esta etapa se recogen y analizan los requerimientos de los usuarios y de las áreas de aplicación.

Esta información se puede recoger de varias formas:

- Entrevistando al personal de la empresa.
- Observando el funcionamiento de la empresa.
- Examinando documentos, sobre todo aquellos que se utilizan para recoger o visualizar información.
- Utilizando cuestionarios para recoger información de grandes grupos de usuarios.

La información recogida debe incluir las principales áreas de aplicación y de usuarios, la documentación utilizada o generada por estas áreas, las transacciones requeridas.

* 4

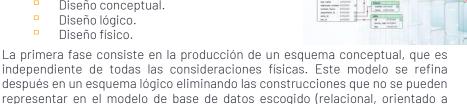
32

4. DISEÑO DE LA BASE DE DATOS

Esta etapa consta de tres fases:

- Diseño conceptual.
- Diseño lógico.
- Diseño físico.

objetos, etc.).



En la tercera fase, el esquema lógico se traduce en un esquema físico para el SGBD escogido. La fase de diseño físico considera las estructuras de almacenamiento y los métodos de acceso necesarios para proporcionar un acceso eficiente a la base de datos en memoria secundaria.

Javier, CALLLOLVEA

5. SELECCIÓN DEL SGBD

Si no se dispone de un SGBD, o el que hay se encuentra obsoleto, se debe escoger un SGBD que sea adecuado para el sistema de información. Esta elección se debe hacer en cualquier momento antes del diseño lógico.(Ejemplo)

- Apache Derby
- FoxPro
- Access
- SOL Server
- Firebird, Otros.

6. DISEÑO DE LA APLICACIÓN

En esta etapa se diseñan los programas de aplicación que usarán y procesarán la base de datos.

Esta etapa y el diseño de la base de datos, son paralelas. En la mayor parte de los casos no se puede finalizar el diseño de las aplicaciones hasta que se ha terminado con el diseño de la base de datos.

En esta etapa hay que asegurarse de que toda la funcionalidad especificada en los requisitos de usuario se encuentra en el diseño de la aplicación.

Además, habrá que diseñar las interfaces de usuario, aspecto muy importante que se suele ignorar. El sistema debe ser fácil de aprender, fácil de usar, ser directo y estar "dispuesto a perdonar". Si la interface no tiene estas características, el sistema dará problemas, sin lugar a dudas.

Javier, CALLI OLVEA

7 E

7. PROTOTIPADO

Un prototipo es un modelo de trabajo de las aplicaciones del sistema.

El prototipo no tiene toda la funcionalidad del sistema final, pero es suficiente para que los usuarios puedan utilizar el sistema e identificar qué aspectos están bien y cuáles no son adecuados, además de poder sugerir mejoras o la inclusión de nuevos elementos.

Este proceso permite que quienes diseñan e implementan el sistema sepan si han interpretado correctamente los requisitos de los usuarios.

Esta etapa es imprescindible cuando el sistema que se va a implementar tiene un gran coste, alto riesgo o utiliza nuevas tecnologías.

8. IMPLEMENTACIÓN

La implementación de la base de datos se realiza mediante las sentencias del lenguaje de definición de datos (LDD) del SGBD escogido.

Estas sentencias se encargan de crear el esquema de la base de datos, los ficheros en donde se almacenarán los datos y las vistas de los usuarios.

Partes de estas aplicaciones son transacciones sobre la base de datos, que se implementan mediante el lenguaje de manejo de datos (LMD) del SGBD.

También se implementan los menús, los formularios para la introducción de datos y los informes de visualización de datos mediante lenguajes de consultas no procedurales, generadores de informes, generadores de formularios, generadores de aplicaciones. También se implementan todos los controles de seguridad e integridad.

Javier, CALLI OLVEA

37

9. CONVERSIÓN Y CARGA DE DATOS

Esta etapa es necesaria cuando se está reemplazando un sistema antiguo por uno nuevo.

Los datos se cargan desde el sistema viejo al nuevo directamente o, si es necesario, se convierten al formato que requiera el nuevo SGBD y luego se cargan.

Si es posible, los programas de aplicación del sistema antiguo también se convierten para que se puedan utilizar en el sistema nuevo.

10. PRUEBAS

En esta etapa se prueba y valida el sistema con los requisitos especificados por los usuarios.

Para ello, se debe diseñar una batería de tests con datos reales, que se deben llevar a cabo de manera metódica y rigurosa.

Es importante darse cuenta de que la fase de prueba no sirve para demostrar que no hay fallos, sirve para encontrarlos.

Si la fase de prueba se lleva a cabo correctamente, descubrirá los errores en los programas de aplicación y en la estructura de la base de datos. Por último, en las pruebas se podrá hacer una medida de la fiabilidad y la calidad del software desarrollado.

Javier, CALLI OLVEA

70

11. MANTENIMIENTO

Una vez que el sistema está completamente implementado y probado, se pone en marcha. El sistema está ahora en la fase de mantenimiento en la que se llevan a cabo las siguientes tareas:

Monitorización de las prestaciones del sistema.

Si las prestaciones caen por debajo de un determinado nivel, puede ser necesario reorganizar la base de datos.

Mantenimiento y actualización del sistema.

Cuando sea necesario, los nuevos requisitos que vayan surgiendo se incorporarán al sistema, siguiendo de nuevo las etapas del ciclo de vida que se acaban de presentar.

40



Javier, CALLI OLVEA

PRINCIPALES - SGBD.

 Un Sistema Gestor de Base de Datos (SGBD) o DGBA (Data Base Management System)

Es un conjunto de programas no visibles que administran y gestionan la información que contiene una base de datos Los gestores de base de datos o gestores de datos hacen posible administrar todo acceso a la base de datos ya que tienen el objetivo de servir de interfaz entre ésta, el usuario y las aplicaciones.

Javier, CALLI OLVEA



















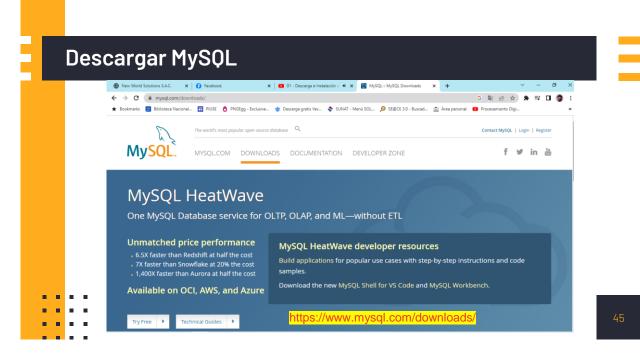
Javier, CALLI OLVEA

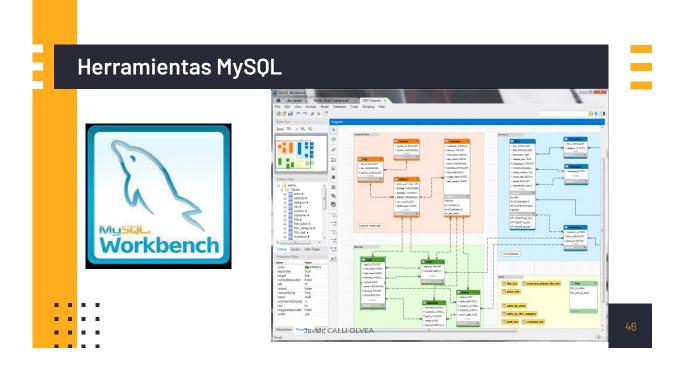
1.7

OTROS SGBD NoSQL

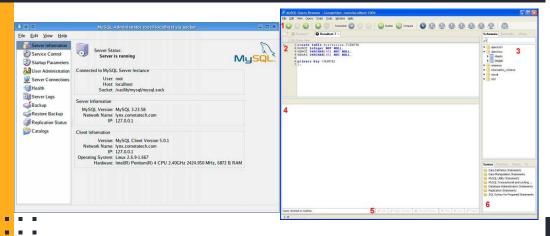
- Otros Sistemas Gestores de bases de datos no relacionales muy utilizados son:
 - Azure Cosmos DB
 - RavenDB
 - ObjectDB
 - Apache CouchDB
 - Neo4j
 - Google BigTable
 - Apache Hbase
 - Amazon DynamoDB

1.1.





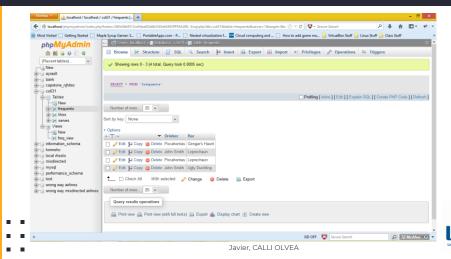




Javier, CALLI OLVEA

47

Herramientas MySQL











48

¿QUÉ ES UN SGBD?

Para **Definir** una BD hay que especificar:

- Las estructuras.
- Los tipos de datos.
- Las restricciones de los datos que se almacenarán en ella.

Javier, CALLI OLVEA

/.O

¿QUÉ ES UN SGBD?

Construir una BD.

Es el proceso de guardar los datos mismos en algún medio de almacenamiento controlado por el SGBD.

¿QUÉ ES UN SGBD?

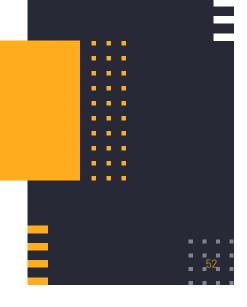
- En la manipulación de una BD intervienen funciones como:
 - <u>Consultar</u> la BD para obtener datos específicos.
 - <u>Actualizar</u> la BD para reflejar cambios en el minimundo.
 - Generar <u>informes</u> a partir de los datos.

Javier, CALLI OLVEA

51

4. LAS DECLARACIONES.

DCL, DDL y DML



Comandos DDL (Lenguaje de definición de datos)

Permite crear y modificar la estructura de una base de datos.

- CREATE: Utilizado para crear nuevas tablas, campos e índices.
- ALTER: Utilizado para modificar las tablas agregando campos o cambiando la definición de los campos.
- DROP: Empleado para eliminar tablas e índices.
- TRUNCATE: Empleado para eliminar todos los registros de una tabla.
- COMMENT: Utilizado para agregar comentarios al diccionario de datos.
- RENAME: Tal como su nombre lo indica es utilizado para renombrar objetos.

Javier, CALLI OLVEA

Comandos DML (Lenguaje de manipulación de datos)

- **SELECT:** Utilizado para consultar registros de la BBDD que satisfagan un determinado criterio.
- INSERT: Utilizado para cargar lotes de datos en la BB en una única operación.
- UPDATE: Utilizado para modificar los valores de los campos y registros especificados.
- DELETE: Utilizado para eliminar registros de una tabla de una BBDD.

DCL (Data Control Language)

Permite crear roles, permisos e integridad referencial, así como el control al acceso a la base de datos.

- GRANT: Usado para otorgar privilegios de acceso de usuario a la base de datos.
- REVOKE: Utilizado para retirar privilegios de acceso otorgados con el comando GRANT.

Javier, CALLI OLVEA

55

TCL (Transactional Control Language):

Permite administrar diferentes transacciones que ocurren dentro de una base de datos.

- COMMIT: Empleado para guardar el trabajo hecho.
- ROLLBACK: Utilizado para deshacer la modificación que hice desde el último COMMIT.







Javier, CALLI OLVEA

59

Herramientas para administración

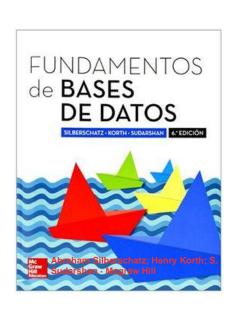


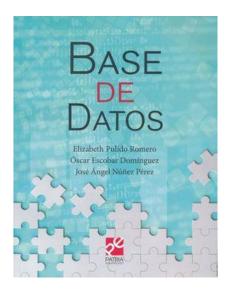




Javier, CALLI OLVEA

60





61

