Laboratorio de Bases de Datos (EBB)

Unidad I - Introducción

Departamento de Electricidad, Electrónica y Computación
Facultad de Ciencias Exactas y Tecnología
Universidad Nacional de Tucumán

Primer Cuatrimestre 2024

Contenido

Bases de Datos

- Tipos de Bases de Datos
- Objetos de Bases de Datos
- Organización de los objetos
- Sobre los nombres de los objetos

Tipos de Bases de Datos

Sistema Gestor de Base de Datos (SGBD)

- Conjunto de aplicaciones que se encargan de la creación y acceso a las BDs, permitiendo guardar y posteriormente acceder a los datos en forma rápida
- Ejemplos:
  - SQL Server
  - MySQL
  - MongoDB
  - CouchDB

Tipos de Bases de Datos

Un SGBD se ejecuta como servicio:

- Continúa su ejecución después que se desconecta el usuario
- Comienza a ejecutarse antes que se conecte el usuario
- Se ejecuta bajo una cuenta de sistema o una creada para este fin
- Se puede administrar remotamente

Servicio MySQL

Tipos de Bases de Datos

Según el SGBD, se deben instalar o no herramientas gráficas para su administración:

- Para MySQL una de estas herramientas puede ser “Workbench”

Conexión a MySQL con Workbench

Tipos de Bases de Datos

Base de Datos

- Conjunto de datos que pertenecen a un mismo contexto que se pueden guardar para ser usados posteriormente
- 2 grandes tipos:
  - **De sistema**: guardan información del SGBD (no hay que borrarlas)
  - **De usuario**: son las BDs de producción (el SGBD puede manejar muchas de éstas)

BDs de sistema y usuario

Objetos de Bases de Datos

Tablas: colección de filas con la misma cantidad de columnas

Tipos de datos: valores permitidos para cada columna

Restricciones (constraints): reglas que rigen los valores permitidos para las columnas (proporcionan un mecanismo de integridad)

Objetos de Bases de Datos

Índices: estructura de almacenamiento que brinda un acceso rápido a los datos y fuerza la integridad de los mismos

Valores por defecto: valores que toman las columnas cuando no se proporciona alguno

Reglas: información que define los valores para las columnas

Objetos de Bases de Datos

Vistas: forma de ver los datos de una o más tablas

Procedimientos almacenados: colección de sentencias SQL, con un nombre, que se ejecutan como un todo. Admiten parámetros de entrada y/o salida

Desencadenadores (triggers): procedimientos almacenados que se ejecutan automáticamente después de producida una acción

Objetos de Bases de Datos

Funciones: rutinas formadas por una o más sentencias SQL

Usuarios: entidad que puede solicitar un recurso del servidor de BD

Roles (grupos): conjunto de usuarios que comparten los mismos permisos

Organización de los objetos

Los objetos se organizan en una determinada jerarquía:

- Servidor
- BD
- Objeto

Hay 2 formas de referenciar los objetos:

- Relativa
- Absoluta

Organización de los objetos

Absoluta:

- select * from TrabajosGraduacion.Alumnos

Relativa:

- select * from Alumnos

Sobre los nombres de los objetos

- Pueden ir entre comillas simples invertidas (`)
- En los nombres de BDs y tablas no se puede emplear '.', '\' ni '/', ya que las BDs se implementan como directorios y las tablas como archivos
- Las palabras reservadas y los nombres de funciones son insensibles a mayúsculas/minúsculas, mientras que los nombres de BDs y tablas sí lo son

SGBDR

- Diseño de aplicaciones
- Implementación de BDs
- Administración de BDs

Diseño de aplicaciones

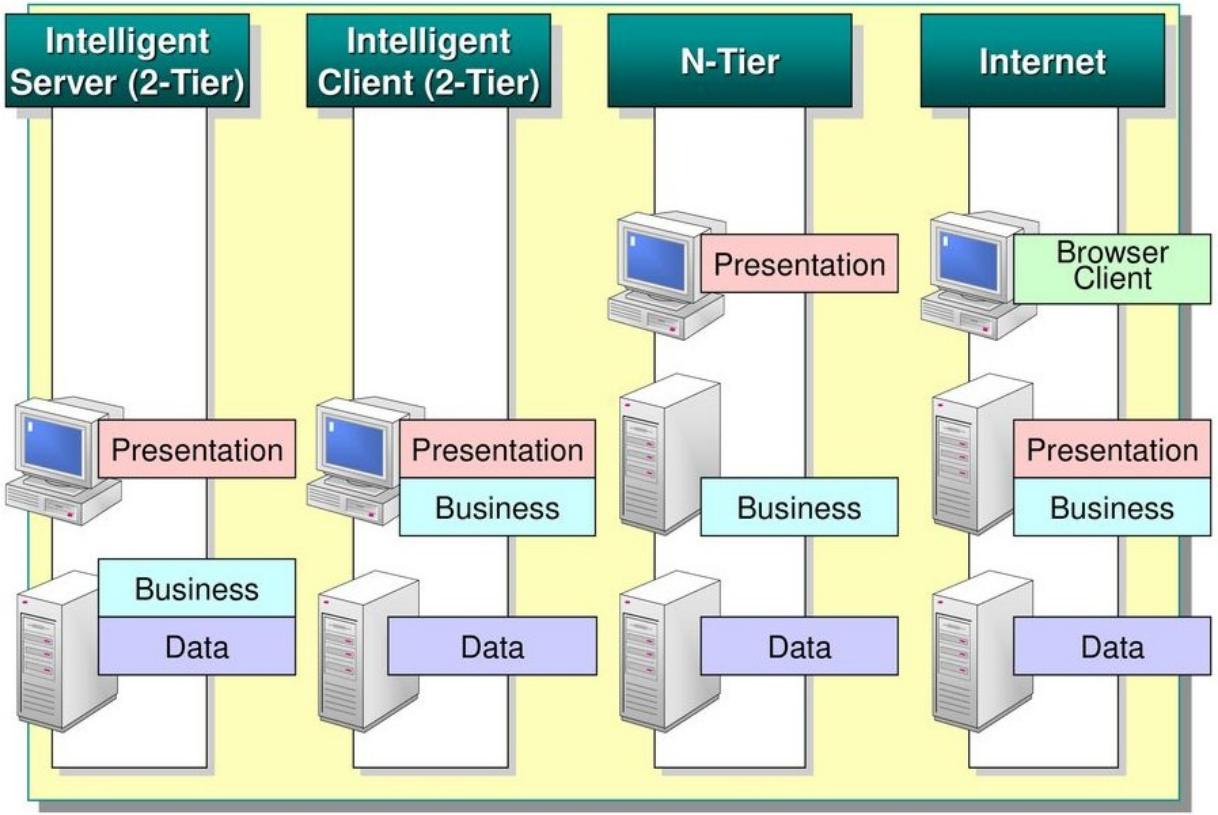
Capas para implementar un modelo C/S:

- **Presentación**: lógica para presentar los datos a los usuarios. Casi siempre la implementa el cliente

- **Negocio**: lógica de la aplicación y reglas de negocio. El SGBDR puede estar involucrado en esta capa

- **Datos**: definición de la BD, integridad, procedimientos almacenados, etc. El SGBDR se involucra estrechamente con esta capa

Diseño de aplicaciones



Diseño de aplicaciones

Arquitecturas típicas:

- Intelligent Server (2 capas)
- Intelligent Client (2 capas)
- N capas
- Internet

Diseño de aplicaciones

- **Intelligent Server (2 capas)**: la mayoría de los procesos ocurre en el servidor con la lógica de interfaz en el cliente. La lógica de negocios es implementada casi por completo en la BD

- **Intelligent Client (2 capas)**: la mayoría de los procesos ocurre en el cliente con los datos residiendo en el servidor. El rendimiento es malo por el tráfico en la red

Diseño de aplicaciones

- **N capas**: el procesamiento es dividido en un servidor de BD, uno o varios servidores de aplicación y los clientes. Es complejo pero muy escalable

- **Internet**: el proceso es dividido en 3 capas con los servicios de negocio y de presentación residiendo en un servidor web, y los clientes usando simples navegadores

Implementación de BDs

Implementar una BD implica:

- Diseñar la BD
- Crear la BD y sus objetos
- Probar y poner a punto la aplicación y la BD
- Planear el desarrollo
- Administrar la aplicación

Implementación de BDs

Diseñar la BD

- Uso eficiente del HW que permita crecimiento futuro, identificando y modelando los objetos de la BD y la lógica de la aplicación, la información de cada objeto y sus relaciones

Crear la BD y sus objetos

- Tablas, integridad y entrada de datos, procedimientos almacenados, vistas, índices, seguridad, etc

Implementación de BDs

Probar y poner a punto la aplicación y la BD

- Las tareas se deben ejecutar de forma rápida y correcta. Junto a un buen diseño, correcto uso de índices y RAID se consigue un buen rendimiento

Planear el desarrollo

- Analizar la carga de trabajo con el sistema en producción para recomendar una óptima configuración de índices

Implementación de BDs

Administrar la aplicación

- Configurar los servidores y clientes, monitorizar el rendimiento en todo momento, gestionar los trabajos, alertas y operadores, manejar la seguridad y las copias de respaldo, etc

Administración de BDs

Administrar una BD involucra:

- Instalar y configurar el SGBDR
- Establecer la seguridad de la red
- Construir las BDs
- Manejar las actividades del día a día

Administración de BDs

Construir las BDs:

- Reservar espacio en disco
- Crear trabajos automatizados

Manejar las actividades del día a día:

- Importación y exportación de datos
- Copias de respaldo y restauraciones
- Monitorizar y poner a punto la BD
- Automatizar todo lo anterior

Creación de BD

- Almacenamiento y Transacciones
- Creación y modificación de BDs
- Tipos de datos y creación de tablas
- Planificación de la capacidad

Almacenamiento y Transacciones

Transacción:

- Una o más unidades de trabajo donde todas tienen éxito o ninguna

- Puede ser un único cambio o un conjunto de cambios relacionados que deben completarse todos o ninguno

- Ejemplo: extracción de una cuenta y depósito en otra

Almacenamiento y Transacciones

Propiedades de una transacción (ACID):

- **Atomicidad**: se completan todas las operaciones de la transacción, o ninguna. Esto evita que las modificaciones se realicen parcialmente

- **Consistencia**: debe modificar los datos según las reglas definidas

- **Aislamiento**: cualquier otra actividad concurrente con la transacción no tiene efecto sobre esta última

Almacenamiento y Transacciones

Propiedades de una transacción (ACID):

- **Durabilidad**: cuando se completa una transacción, sus resultados se guardan

Almacenamiento y Transacciones

Las transacciones pueden ser:

- **Explícitas**: comienzo y final explícitamente definidos:

```
START TRANSACTION
    sentencias
COMMIT | ROLLBACK
```

- **Implícitas**: algunos comandos SQL terminan implícitamente una transacción (INSERT, ALTER, SELECT, UPDATE, etc)

Almacenamiento y Transacciones

En MySQL una BD está formada por 2 tipos de archivos:

- **Archivos de datos**: datos e índices de la tabla (*.ibd para InnoDB)

- **Registro de transacciones**: estructura de datos en disco que permite corregir fallos debidos a transacciones incompletas

Archivos de datos y del registro de transacciones

Almacenamiento y Transacciones

Funcionamiento del registro de transacciones:

- Una aplicación envía una modificación (transacción) de datos
- Cuando la transacción termina (COMMIT), MySQL la anota en el registro (se completaron con éxito todos los elementos de la transacción)
- Luego MySQL lee el registro y aplica todas las transacciones terminadas en la BD
- Si se produjera una falla (luego que la transacción esté anotada), cuando se vuelva a inicializar el servicio la misma se aplicará en la BD

Almacenamiento y Transacciones

Sobre el registro de transacciones:

- Se implementa mediante 2 archivos: ib_logfile0 e ib_logfile1

- MySQL se encarga de escribir en estos archivos de forma circular (escribe desde el comienzo y cuando llega al final vuelve al comienzo)

- Por defecto, el tamaño de cada uno de estos archivos es de 5 MB

Almacenamiento y Transacciones

Tipos de tablas:

- MySQL permite trabajar con distintos tipos de tablas
- En una misma BD se pueden usar distintos tipos de tablas
- Los 2 tipos más conocidos son:
  - InnoDB
  - MyISAM

Almacenamiento y Transacciones

InnoDB:

- Tipo transaccional (conforme a ACID)
- Puede aplicar y retrotraer transacciones
- Permite la recuperación en caso de problemas con la BD
- Guarda los datos usando índices agrupados
- Soporta integridad referencial mediante FK

Almacenamiento y Transacciones

MyISAM:

- Usa bloqueos a nivel tabla, limitando las operaciones de lectura/escritura
- Usado generalmente en escenarios de sólo lectura o data warehousing

Almacenamiento y Transacciones

Sobre los tipos de tablas:

- Se especifican en la sentencia CREATE TABLE

- Para cambiar el tipo de tabla:

ALTER TABLE <tabla> ENGINE=<tipo de tabla>;

Diferencias entre InnoDB y MyISAM

InnoDB	MyISAM
Bloqueos a nivel fila	Bloqueos a nivel tabla
Soporta FK	No soporta FK
Soporta transacciones (commit/roll back)	No soporta transacciones (no se puede commit/roll back)

Creación y modificación de BDs

Creación de BDs:

- Se puede especificar una sentencia SQL o usar alguna herramienta gráfica

- Se debe especificar el nombre

CREATE DATABASE | SCHEMA [IF NOT EXISTS] LBD2024;

Creación y modificación de BDs

Borrado de BDs:

- Se puede especificar una sentencia SQL o usar alguna herramienta gráfica:

```
DROP DATABASE | SCHEMA [IF EXISTS] LBD2024;
```

No se puede borrar una BD cuando:

- Está siendo restaurada
- Un usuario está conectado a la misma
- Se está publicando como parte de una replicación

Tipos datos y creación de tablas

Creación de tablas:

- Se puede especificar una sentencia SQL o usar alguna herramienta gráfica
- Se debe especificar un nombre (único dentro de la BD)
- Se deben especificar los nombres y tipos de datos de las columnas (únicos dentro de la tabla)
- Se debe especificar si cada columna admite o no valores nulos (NULL o NOT NULL)

Tipos datos y creación de tablas

Creación de tablas:

```
CREATE TABLE [IF NOT EXISTS] Socios (
    <col1> <tipo_dato> NOT NULL,
    <col2> <tipo_dato> NOT NULL,
    <col3> <tipo_dato> NOT NULL,
    <col4> <tipo_dato> NULL,
    ...
) ENGINE = <tipo de tabla>;
```

Tipos datos y creación de tablas

MySQL soporta diferentes tipos de datos:

- Numéricos: enteros, con punto decimal fijo, con punto decimal flotante, bits
- Fechas y horas
- Cadenas
- JSON
- Espaciales

Tipos datos y creación de tablas

Tipos de datos numéricos:

- **Enteros**: TINYINT, SMALLINT, MEDIUMINT, INT[EGER], BIGINT
- **Con punto decimal fijo**: DECIMAL, NUMERIC
- **Con punto decimal flotante**: FLOAT, DOUBLE
- **Bit**: BIT

Tipos datos y creación de tablas

Numéricos con punto decimal fijo:

- Guardan valores numéricos exactos (valores monetarios por ejemplo)
- Tipos: DECIMAL, NUMERIC
- Se especifica la precisión (cantidad total de dígitos) y la escala (cantidad de dígitos para la parte decimal)
- Ej: salario DECIMAL(5, 2)

Tipos datos y creación de tablas

Numéricos con punto decimal flotante:

- Guardan valores numéricos aproximados
- Tipos: FLOAT, DOUBLE
- También se especifica la precisión y la escala
- MySQL redondea al guardar los valores
  - Ej: salario FLOAT(7, 4)
  - Valor que se guarda: 999.00009
  - Resultado que se obtiene: 999.0001

Tipos datos y creación de tablas

Numéricos para bits:

- Guardan bits
- Tipo: BIT
- Para especificar los valores, se puede usar la notación b'valor':
  - Ej: b'1' y b'0'

Tabla con tipos de datos numéricos

Tipos datos y creación de tablas

Atributos para los tipos numéricos:

- Ancho de visualización
- ZEROFILL
- UNSIGNED
- AUTO_INCREMENT

Tipos datos y creación de tablas

Ancho de visualización:

- MySQL soporta una extensión para especificar, opcionalmente, el ancho con el cual se muestran los tipos enteros
- Por ejemplo, INT(4) especifica que un entero se muestre con 4 dígitos
- El ancho de visualización no restringe el rango de valores que se pueden guardar ni impide que valores más grandes que el mismo se muestren correctamente

Tipos datos y creación de tablas

ZEROFILL:

- Cuando se especifica un ancho de visualización junto con el atributo opcional (no estándar) ZEROFILL, el relleno por defecto de espacios se reemplaza con ceros

- Por ejemplo, una columna INT(4) ZEROFILL, en la cual se guarda el número 5, se muestra como 0005

Tipos datos y creación de tablas

UNSIGNED:

- Todos los tipos enteros pueden llevar este atributo opcional (no estándar), el cual permite sólo números positivos en una columna
- Se puede usar cuando se necesite un rango numérico más grande para la columna: una columna INT UNSIGNED, sigue teniendo el mismo rango de valores, pero los límites se mueven de -2147483648 y 2147483647 a 0 y 4294967295

Tipos datos y creación de tablas

AUTO_INCREMENT:

- Válido para los enteros y punto decimal flotante
- La primera fila que se inserta toma el valor 1 en esta columna
- La segunda fila toma el valor 2 y así sucesivamente
- Cada tabla admite una sola columna con esta propiedad, y su valor no se especifica
- Si se especifica un valor para esta columna, se inserta, y la siguiente fila toma el valor siguiente

Tipos datos y creación de tablas

AUTO_INCREMENT:

- MySQL exige que la columna con esta propiedad sea PK
- No permite valores negativos
- Requiere que la columna sea NOT NULL
- No soporta restricciones CHECK

Tipos datos y creación de tablas

Tipos de datos para fechas y horas:

- DATE
- DATETIME
- TIMESTAMP
- TIME
- YEAR

Tipos datos y creación de tablas

DATE:

- Es para valores con sólo la fecha
- Se recupera y visualiza en el formato ‘AAAA-MM-DD’ (por ejemplo: ‘2024-03-06’)
- Rango: '1000-01-01' a '9999-12-31'

Tipos datos y creación de tablas

DATETIME:

- Es para valores con fecha y hora

- Se recupera y visualiza en el formato ‘AAAA-MM-DD HH:MM:SS’ (por ejemplo: ‘2024-03-06 16:54:23’)

- Rango: '1000-01-01 00:00:00' a '9999-12-31 23:59:59'

Tipos datos y creación de tablas

TIMESTAMP:

- Es para valores con fecha y hora

- Se recupera y visualiza en el formato ‘AAAA-MM-DD HH:MM:SS’ (por ejemplo: ‘2024-03-03 16:54:23’)

- Rango: '1970-01-01 00:00:01' UTC a '2038-01-19 03:14:07' UTC

Tipos datos y creación de tablas

Sobre los tipos DATE, DATETIME y TIMESTAMP:

- Las fechas siempre se expresan en el orden año-mes-día (por ejemplo: ‘2023-03-06’)

- Se recomienda emplear 4 dígitos para los años para evitar problemas

- Los tipos DATETIME y TIMESTAMP pueden llevar hasta 6 dígitos, los cuales representan microsegundos

Tipos datos y creación de tablas

Sobre los tipos DATE, DATETIME y TIMESTAMP:

- Al guardar valores TIMESTAMP, MySQL los convierte de la zona horaria actual a UTC, y al recuperarlos de UTC a la zona horaria actual

Tipos datos y creación de tablas

Sobre los tipos DATE, DATETIME y TIMESTAMP:

- Cualquier columna DATETIME y TIMESTAMP puede:
  - Tomar el valor por defecto de la fecha y hora actual (DEFAULT CURRENT_TIMESTAMP)
  - Tomar el valor de auto actualización (ON UPDATE CURRENT_TIMESTAMP)
  - Ambos

Tipos datos y creación de tablas

Sobre los tipos DATE, DATETIME y TIMESTAMP:

- Además de CURRENT_TIMESTAMP se puede usar:
  - CURRENT_TIMESTAMP()
  - NOW()
  - LOCALTIME
  - LOCALTIME()
  - LOCALTIMESTAMP
  - LOCALTIMESTAMP()

Tipos datos y creación de tablas

Sobre los tipos DATE, DATETIME y TIMESTAMP:

- También se puede usar la cláusula DEFAULT para especificar un valor predeterminado constante (no automático):
  - Ejemplo: DEFAULT '2024-03-06 00:00:00'

Tipos datos y creación de tablas

TIME:

- Es para valores con sólo la hora
- Se recupera y visualiza en el formato ‘HH:MM:SS’ (o ‘HHH:MM:SS’ para valores con horas grandes)
- Rango: '-838:59:59' a '838:59:59'
- Puede incluir una parte fraccional para representar microsegundos

Tipos datos y creación de tablas

YEAR:

- Es para representar años
- Se recupera y visualiza en el formato ‘AAAA’
- Rango: '1901' a '2155' (o ‘0000’)

Tabla con tipos de datos para fechas

Tipos datos y creación de tablas

Tipos de datos para cadenas:

- **De largo fijo y variable**: CHAR, VARCHAR, BINARY, VARBINARY
- **Para grandes objetos**: TEXT, BLOB
- **Para selección de un elemento**: ENUM
- **Para selección de varios elementos**: SET

Tipos datos y creación de tablas

Cadenas de largo fijo y variable:

- El largo de una columna CHAR o BINARY, entre 0 y 255, se mantiene fijo al valor que se especifica en la creación (se completa con espacios en blanco a la derecha)

- El largo de una columna VARCHAR o VARBINARY, entre 0 y 65535, varía según el valor actualmente guardado

Tipos datos y creación de tablas

Cadenas de largo fijo y variable:

- CHAR, VARCHAR y TEXT comparan cadenas usando una cierta secuencia de ordenamiento (collation)
- BINARY, VARBINARY y BLOB comparan bytes

Tabla con tipos de datos CHAR y BINARY

Tipos datos y creación de tablas

Cadenas para objetos grandes:

- **TEXT**: cadenas desde 1 byte a 4 GB
  - Tiene 4 tipos:
    - TINYTEXT: 255 caracteres
    - TEXT: 65535 caracteres (64 kB)
    - MEDIUMTEXT: 16777215 caracteres (16 MB)
    - LONGTEXT: 4294967295 caracteres (4 GB)

Tipos datos y creación de tablas

Cadenas para objetos grandes:

- **BLOB**: similar a TEXT, pero las comparaciones se basan en los valores numéricos que contienen, no en una collation
  - Tiene 4 tipos:
    - TINYBLOB
    - BLOB
    - MEDIUMBLOB
    - LONGBLOB

Tipos datos y creación de tablas

Cadenas para selección de un elemento:

- El tipo ENUM es una cadena con un valor elegido a partir de una lista de valores permitidos

- Las cadenas que se especifican como valores de entrada se codifican como números

- En las consultas, los números se traducen a sus cadenas correspondientes

Tipos datos y creación de tablas

Cadenas para selección de un elemento:

- Los valores listados tienen una posición: 1, 2, 3, ...
- Si una columna ENUM vale NULL, la posición es NULL
- Si una columna ENUM vale ‘’, la posición es 0
- Un ENUM puede tener hasta 65535 elementos

Tabla con tipo de dato Enum

Tipos datos y creación de tablas

Cadenas para selección de varios elementos:

- El tipo SET es un objeto cadena que puede tener cero o más valores, elegidos a partir de una lista
- Una columna definida como SET('uno', 'dos') NOT NULL puede tener como valores:
  - ''
  - 'uno'
  - 'dos'
  - 'uno,dos'

Tabla con tipo de dato Set

Tipos datos y creación de tablas

JSON (JavaScript Object Notation):

- Formato texto para representar datos estructurados
- Muy usado para el intercambio de datos
- Es un subconjunto de JavaScript, totalmente independiente del mismo

Tipos datos y creación de tablas

JSON (JavaScript Object Notation):

- Un objeto JSON puede tomar 2 formas:
  - Una colección desordenada de pares clave:valor entre { }
  - Una lista ordenada de valores (vector) entre [ ]
  - Estas estructuras se pueden anidar

Tipos datos y creación de tablas

JSON (JavaScript Object Notation):

- Ejemplos:

  { }

  {"user":"Sammy", "online":true, "followers":987}

  [ ]

  [1, 2, true, false]

Tipos datos y creación de tablas

JSON (JavaScript Object Notation):

- Un valor puede ser:
  - Una cadena entre comillas dobles
  - Un número
  - true/false
  - null
  - Otro objeto JSON
  - Un vector

Tipos datos y creación de tablas

Borrado de tablas:

- Se puede especificar una sentencia SQL o usar alguna herramienta gráfica

```
DROP TABLE [IF EXISTS] Socios;
```

Borrado de tabla

Tipos datos y creación de tablas

Agregado / borrado de columnas:

- Se puede especificar una sentencia SQL o usar alguna herramienta gráfica

```
ALTER TABLE Socios
ADD domicilio VARCHAR(30) NULL;

ALTER TABLE Socios
DROP COLUMN foto;
```

Planificación de la capacidad

Al planificar una BD se debe tener en cuenta el espacio que ocupará en disco en forma de archivos:

- Tablas (de usuario y sistema)
- Para las tablas de usuario, se puede estimar para cada tabla la cantidad de filas y el tamaño de cada una
- Tamaño del registro de transacciones
- Índices (número y tamaño)

Resumen

- Tipos de Bases de Datos
- Objetos de Bases de Datos
- Arquitecturas de diseño de aplicaciones
- Actividades para implementar una BD
- Almacenamiento y Transacciones
- Creación y modificación de BDs
- Creación de tipos datos y tablas
- Planificación de la capacidad

Otros recursos

Registro de transacciones en MySQL:

- https://dev.mysql.com/doc/refman/8.0/en/innodb-redo-log.html
- https://dba.stackexchange.com/questions/72904/difference-between-transaction-log-and-redo-log-in-mysql

Tipos de datos

- https://dev.mysql.com/doc/refman/8.0/en/data-types.html

Otros recursos

JSON

- https://www.digitalocean.com/community/tutorials/an-introduction-to-json
- https://ed.team/blog/como-trabajar-con-json-en-mysql
- https://scotch.io/tutorials/working-with-json-in-mysql
- http://www.mysqltutorial.org/mysql-json/