

Unidad 4

Diseño físico

Base de datos

Diseño físico

Introducción

Modelo físico

Lenguaje SQL

MySQL

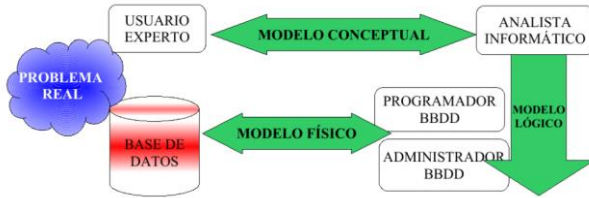
Data Definition Language (DDL)

Base de datos

Diseño físico

Fases

Introducción



Especificación de requisitos

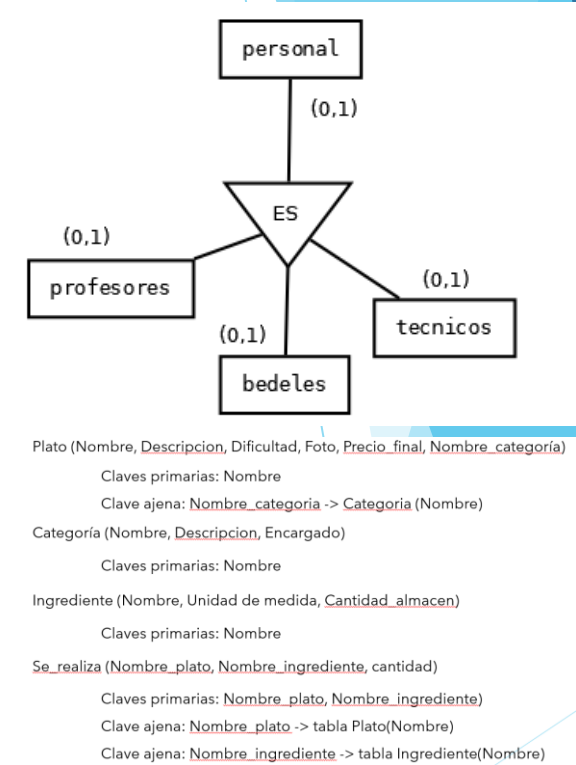
DISEÑO CONCEPTUAL

Esquema conceptual

DISEÑO LÓGICO

Esquema lógico

DISEÑO FÍSICO



```
CREATE TABLE plato  
( Nombre VARCHAR(9) PRIMARY KEY,  
  descripcion VARCHAR(100)
```

Base de datos

Diseño físico - Introducción

Representa la estructura de la base de datos

Tipos de datos del Sistema Gestor de Base de Datos.

Compuesto de tablas y relaciones

Uso del lenguaje SQL

[Tipos de datos \(Transact-SQL\) - SQL Server | Microsoft Docs](#)

[Tipos de Datos de Mysql \(desarrolloweb.com\)](#)

[Tipos de datos soportados \(oracle.com\)](#)

[PostgreSQL data types, tipos de datos más utilizados - TodoPostgreSQL](#)

Base de datos

Diseño físico

Introducción

Modelo físico

Lenguaje SQL

MySQL

Data Definition Language (DDL)

Base de datos

Diseño físico - **Lenguaje SQL**

SQL es un lenguaje de consulta

Interfaz para comunicarse con bases de datos

Realizar operaciones de acceso y manipulación de la información almacenada.

ORACLE, Microsoft SQL Server, MySQL o PostgreSQL.



SQL

Base de datos

Diseño físico - **Lenguaje SQL**

Principales características

- **Integridad de los datos.**

- Asegurar que los datos sean válidos, correctos y completos.

- **Lenguaje estandarizado.**

- Se pueden desplegar implementaciones del mismo lenguaje SQL en diferentes sistemas

- **Sencillez y claridad.**

- Lenguaje unificado, claro y simple, de fácil comprensión.

- **Flexibilidad.**

- Permite definir diferentes formas de ver los datos para satisfacer las especificaciones requeridas por parte del usuario.

Base de datos

Diseño físico - **Lenguaje SQL**

Versiones

- SQL-86: contiene la funcionalidad mínima para que un lenguaje se considere SQL.
- SQL-89: añade instrucciones para gestionar las claves ajenas (reglas de integridad referencial).
- SQL-92: contiene una gran cantidad de variaciones sobre le original.
- SQL:1999: se añaden extensiones hacia la programación orientada objetos.
- SQL:2003. Se han introducido características de SQL/ XML.
- A partir de aquí ha habido nuevas normativas en 2006, 2008, etc.; hasta SQL:2016.

Base de datos

Diseño físico - **Lenguaje SQL**

Grupos de sentencias

Dos categorías principales:

Lenguaje de Definición de Datos, DDL

Create Drop y Alter

Lenguaje de Manipulación de Datos, DML.

Select, Insert, Delete y Update

DCL lenguaje de consulta de datos y lenguaje de control de datos.

Grant

Declare, Open, Fetch y Close

Base de datos

Diseño físico - ***Tipo de datos SQL***

Datos cadena

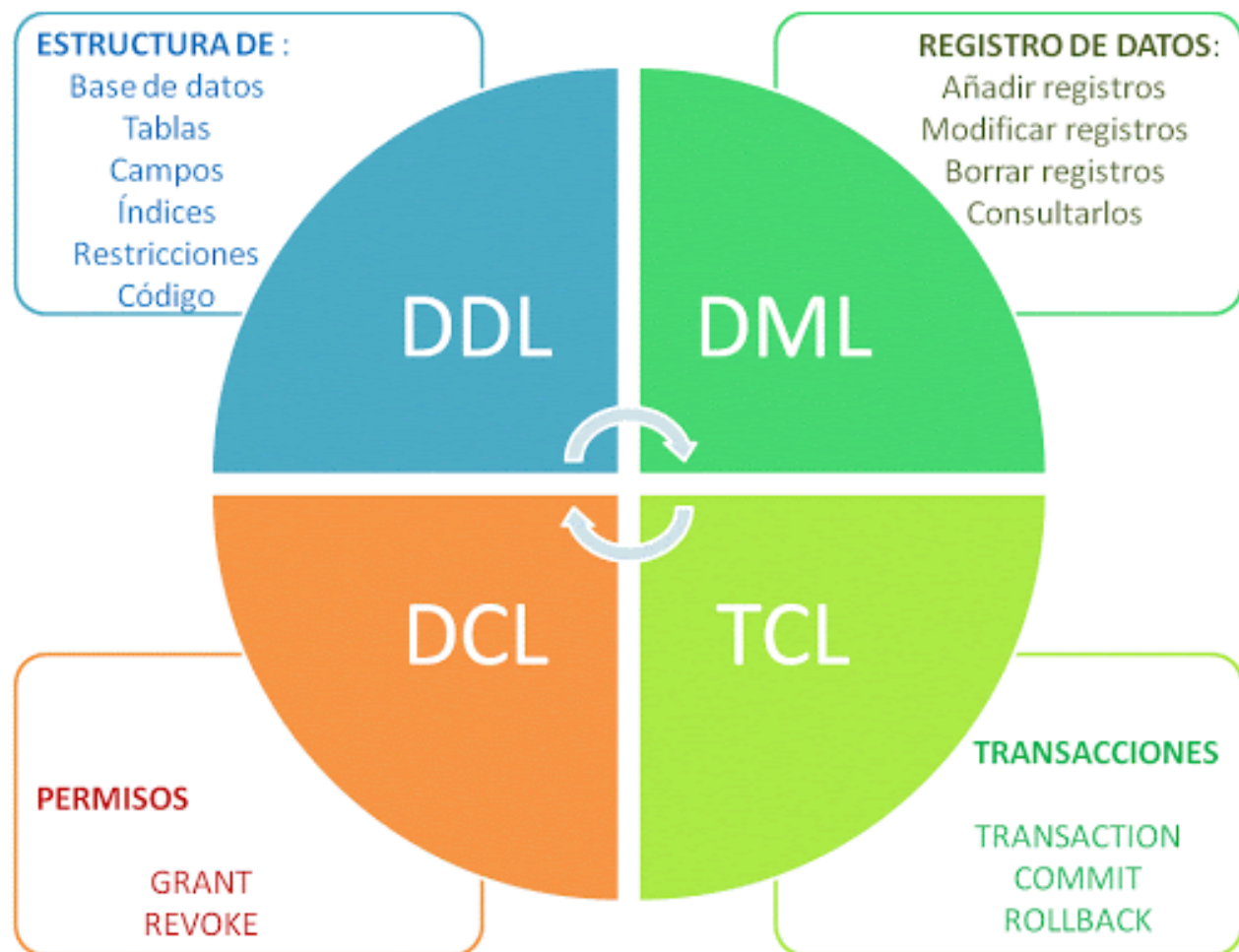
tipo	tamaño	uso
CHAR	0-255	cadena de longitud fija
VARCHAR	0-65535	cadena de longitud variable (a partir de 5.0.3)
TINYBLOB	0-255	No más de 255 caracteres en una cadena binaria
TINYTEXT	0-255	cadena de texto cortos
BLOB	0-65535	los datos de texto largo en forma binaria
TEXT	0-65535	los datos de texto largo
MEDIUMBLOB	0-16777215	forma binaria de datos de texto longitud media
MEDIUMTEXT	0-16777215	datos de texto longitud media
LOB	0-4294967295	los datos de texto grandes en forma binaria
LONGTEXT	0-4294967295	Gran datos de texto

Base de datos

Diseño físico - ***Tipo de datos SQL***

Datos fecha y hora

tipo	tamaño (Byte)	alcance	formato	uso
FECHA	3	01.01.1000 / 9999-12-31	AAAA-MM-DD	Los valores de fecha
TIEMPO	3	"-838: 59: 59 '/' 838: 59: 59 "	HH: MM: SS	Valor de tiempo o la duración
AÑO	1	1901/2155	AAAA	Valor año
DATETIME	8	1000-01-0100: 00: 00 / 9999-12-31 23:59:59	AAAA-MM-DD HH: MM: SS	Mezcla los valores de fecha y hora
TIMESTAMP	4	En algún momento 00/2037 Año: 1970-01-01 00:00	AAAAMMDD HHMMSS	La mezcla y la fecha valor de tiempo, una marca de tiempo



Confirmar como permanentes las modificaciones realizadas.
Deshacer todas las modificaciones realizadas desde la última confirmación.

Base de datos

Diseño físico - ***Ejemplo sentencias***

Lenguaje de Definición de Datos, DDL .

Permite crear, borrar o actualizar las tablas de la BBDD.

Ejemplo: `CREATE DATABASE juegos;`

```
CREATE TABLE juego(  
  id_juego smallint(5),  
  titulo varchar(50),  
  genero varchar(50),  
)
```

Base de datos

Diseño físico - ***Ejemplo sentencias***

Lenguaje de Manipulación de Datos, DML.

Permite insertar, borrar , actualizar o borrar datos de la BBDD.

Ejemplo:

```
INSERT INTO consola(nombre, marca) VALUES('PS4','Sony');
```

```
UPDATE juego SET titulo='The Legend of Zelda' WHERE titulo='Metroid Prime'
```

```
DELETE FROM juego WHERE id_juego=5
```

Base de datos

Diseño físico - ***Ejemplo sentencias***

Lenguaje de control de Datos, DCL/TCL. .

Permite gestionar permisos de usuario y transacciones de la BBDD.

Ejemplo:

En el siguiente ejemplo se concede el permiso `CREATE TABLE` para la base de datos `juegos` al usuario `Rafa`.

`USE juegos;`

`GRANT CREATE TABLE TO Rafa;`

Base de datos

Diseño físico

Introducción

Modelo físico

Lenguaje SQL

MySQL

Data Definition Language (DDL)

Base de datos

Diseño físico - ***Mysql***

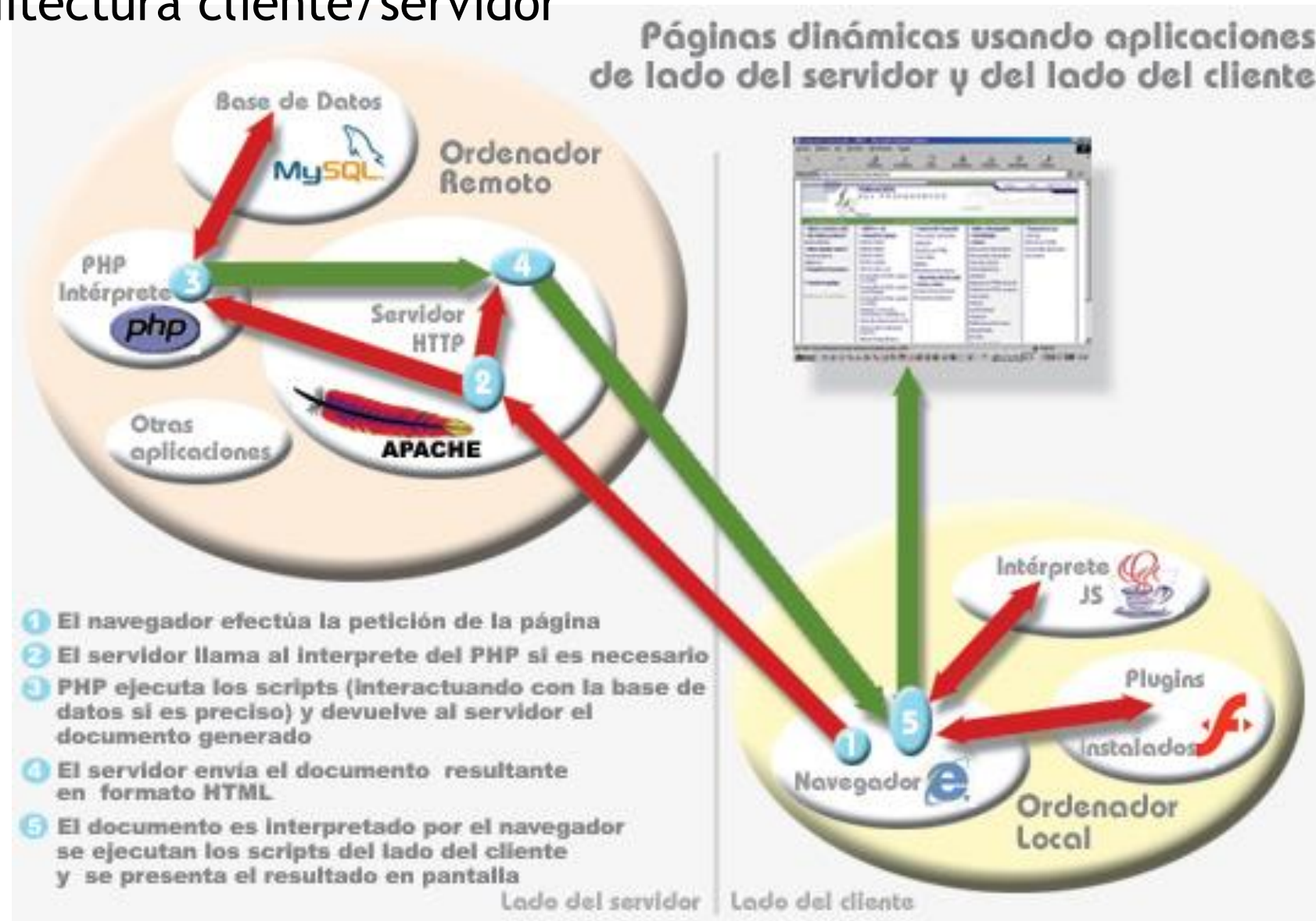
MySQL es un sistema de gestión de bases de datos relacionales de código abierto (RDBMS, por sus siglas en inglés) con un modelo cliente-servidor.

RDBMS **Relational Database** Management System es un software o servicio utilizado para crear y administrar bases de datos basadas en un modelo relacional. Utilizada en aplicaciones web como wordpress, prestashop, drupal, etc..



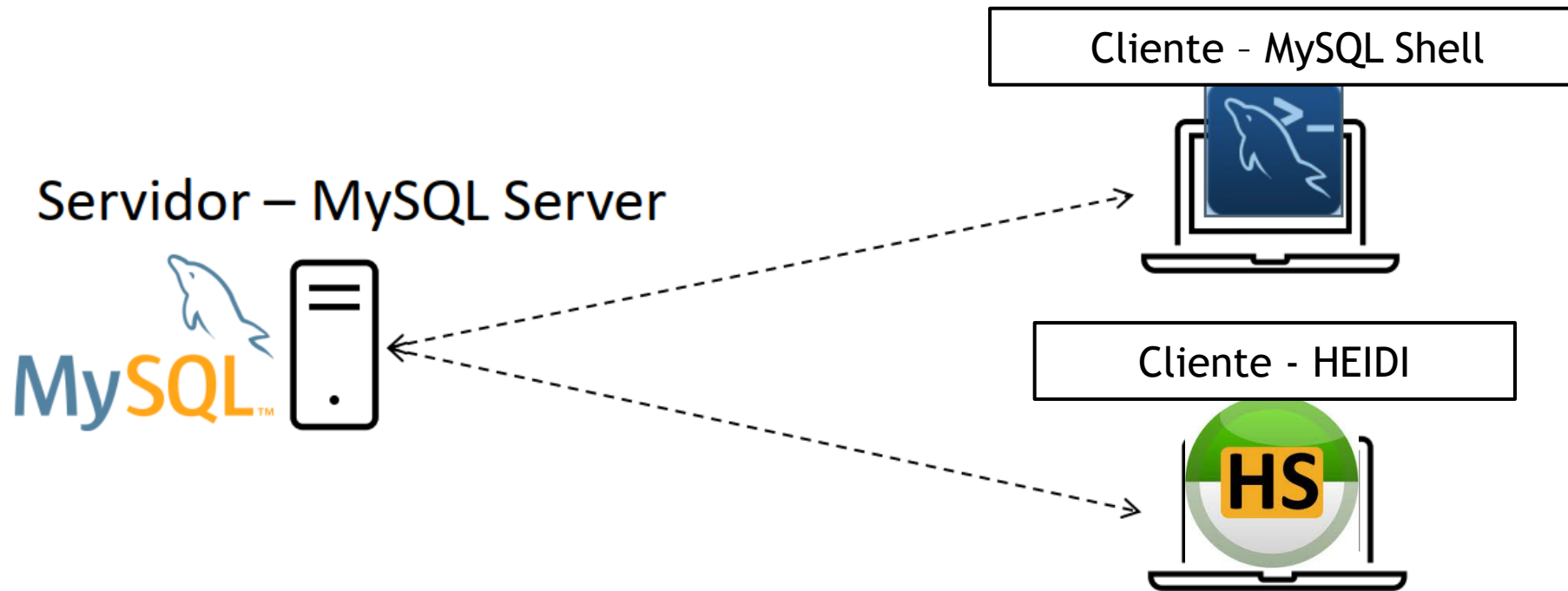
Base de datos Diseño físico - **Mysql**

Arquitectura cliente/servidor



Base de datos Diseño físico - **MySQL**

Arquitectura cliente/servidor



[MySQL :: Download MySQL Shell](#)

[MySQL :: Download MySQL Installer](#)

[HeidiSQL - MariaDB, MySQL, MSSQL, PostgreSQL and SQLite made easy](#)

Base de datos

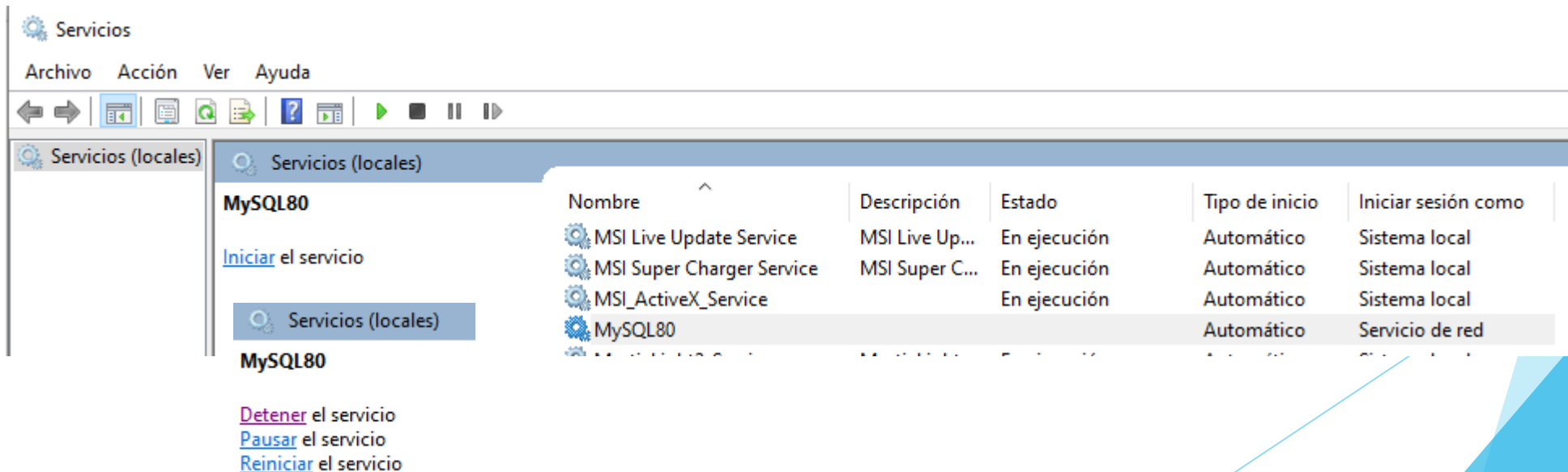
Diseño físico - **Mysql**

Instalación Mysql

¿Como iniciamos nuestro servidor?

Como lo hemos configurado como servicio se iniciara automáticamente

(Windows / Servicios)



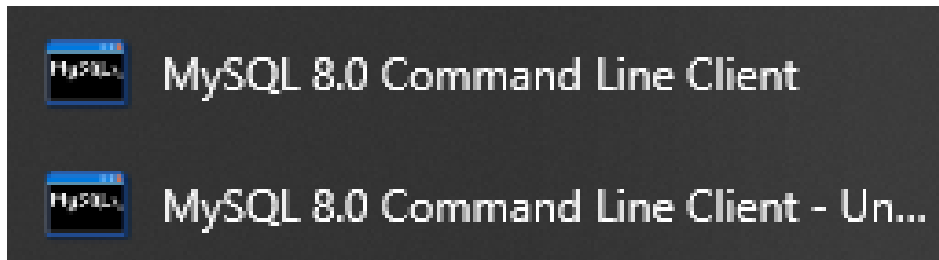
Base de datos

Diseño físico - **Mysql**

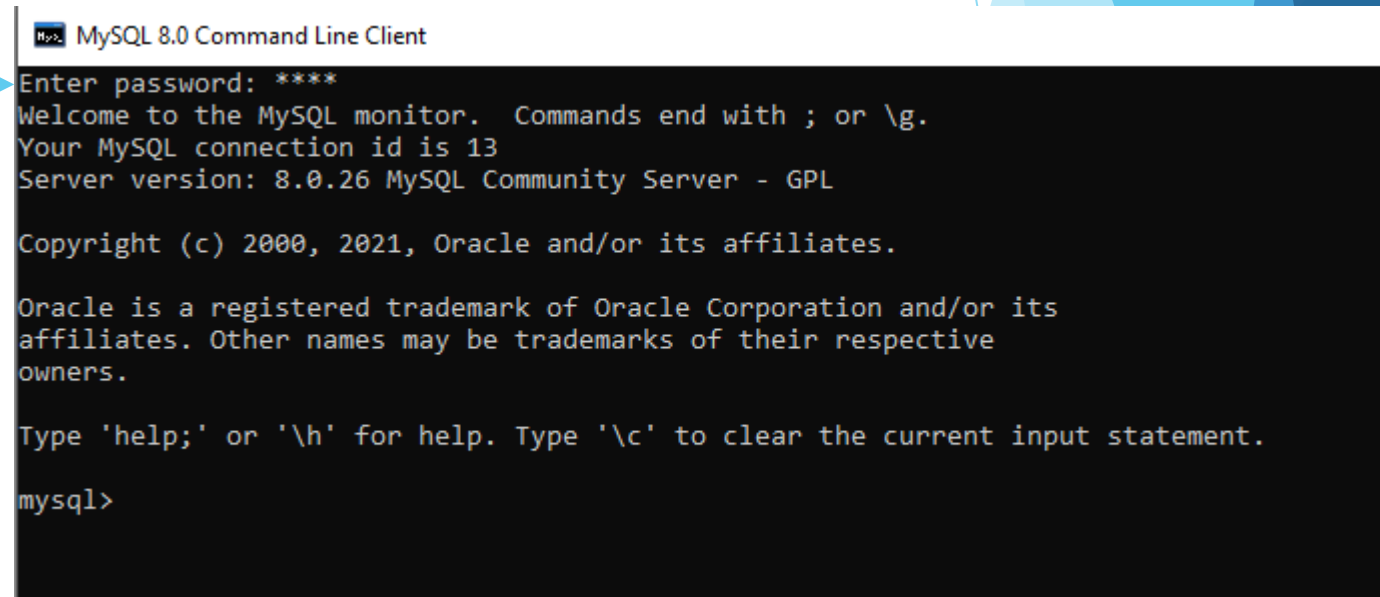
Instalación Mysql

¿Como accedemos a nuestro servidor?

Inicio



Estas dos opciones nos pedirán contraseña

A screenshot of the MySQL 8.0 Command Line Client interface. The window title is 'MySQL 8.0 Command Line Client'. The text inside the window reads: 'Enter password: ****', 'Welcome to the MySQL monitor. Commands end with ; or \g.', 'Your MySQL connection id is 13', 'Server version: 8.0.26 MySQL Community Server - GPL', 'Copyright (c) 2000, 2021, Oracle and/or its affiliates.', 'Oracle is a registered trademark of Oracle Corporation and/or its affiliates. Other names may be trademarks of their respective owners.', 'Type \'help;\' or \'h\' for help. Type \'c\' to clear the current input statement.', and 'mysql>'. A blue arrow points from the text 'Estas dos opciones nos pedirán contraseña' to the 'Enter password: ****' prompt.

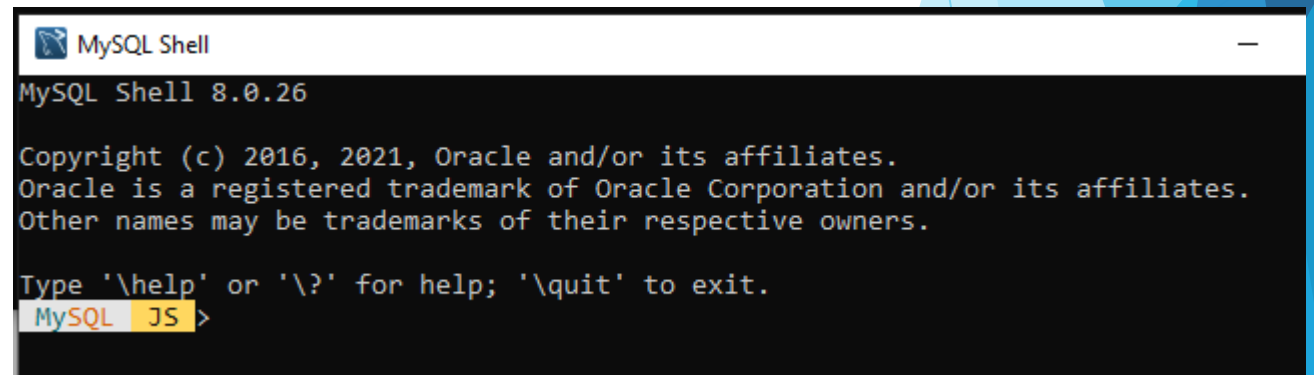
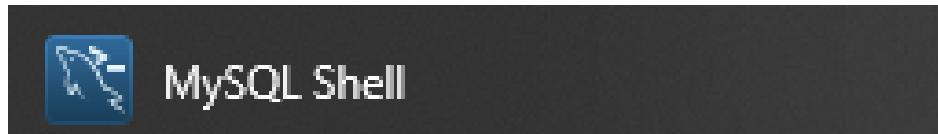
Base de datos

Diseño físico - **Mysql**

Instalación Mysql

¿Como accedemos a nuestro servidor?

Inicio

A screenshot of a terminal window titled "MySQL Shell". The window has a black background with white text. It displays the version "MySQL Shell 8.0.26", a copyright notice for Oracle, and instructions to type '\help' or '\?' for help and '\quit' to exit. The prompt "MySQL JS >" is visible at the bottom.

```
MySQL Shell
MySQL Shell 8.0.26

Copyright (c) 2016, 2021, Oracle and/or its affiliates.
Oracle is a registered trademark of Oracle Corporation and/or its affiliates.
Other names may be trademarks of their respective owners.

Type '\help' or '\?' for help; '\quit' to exit.
MySQL JS >
```

Aquí utilizamos comando de Shell

Base de datos

Diseño físico - **Mysql**

Comandos de MySQL Shell

Funciones, instalación y comandos de MySQL Shell 8.0 - programador clic (programmerclick.com)

Comando	atajo	Descrpcion
\help	\h or \?	Ayuda
\connect	\c	Conecta a nuestro servicio
\status	\s	Muestra el estado actual de MySQL Shell.
\js		Cambie el modo de ejecución a JavaScript.
\py		Cambie el modo de ejecución a Python
\sql		Cambie el modo de ejecución a SQL.
\quit	\q	Salir de MySQL Shell

Base de datos

Diseño físico - **Mysql**

\connect root@localhost:3306

Usuario servidor puerto

\connect root@127.0.0.1:3306

Usuario servidor puerto

Usuario: usuario con el cual nos queremos conectar al servicio, lo haremos con root, usuario por defecto con permisos.

Servidor: dirección del servidor en este caso utilizaremos 127.0.0.1 o localhost.

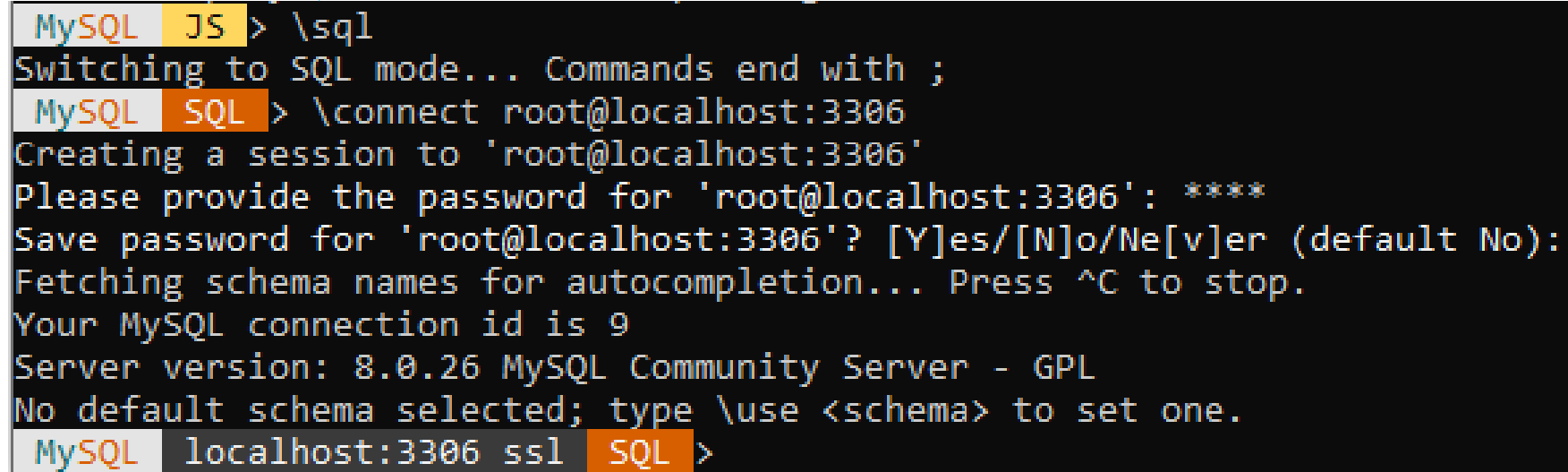
Puerto: puerto especificado en la instalación del servio, normalmente 3306.

Base de datos

Diseño físico - **MySQL**

Debemos asegurarnos que estamos en mod SQL

Tecleamos \sql



```
MySQL JS > \sql
Switching to SQL mode... Commands end with ;
MySQL SQL > \connect root@localhost:3306
Creating a session to 'root@localhost:3306'
Please provide the password for 'root@localhost:3306': ****
Save password for 'root@localhost:3306'? [Y]es/[N]o/[N]ever (default No):
Fetching schema names for autocompletion... Press ^C to stop.
Your MySQL connection id is 9
Server version: 8.0.26 MySQL Community Server - GPL
No default schema selected; type \use <schema> to set one.
MySQL localhost:3306 ssl SQL >
```

\connect root@localhost:3306

Usuario

servidor

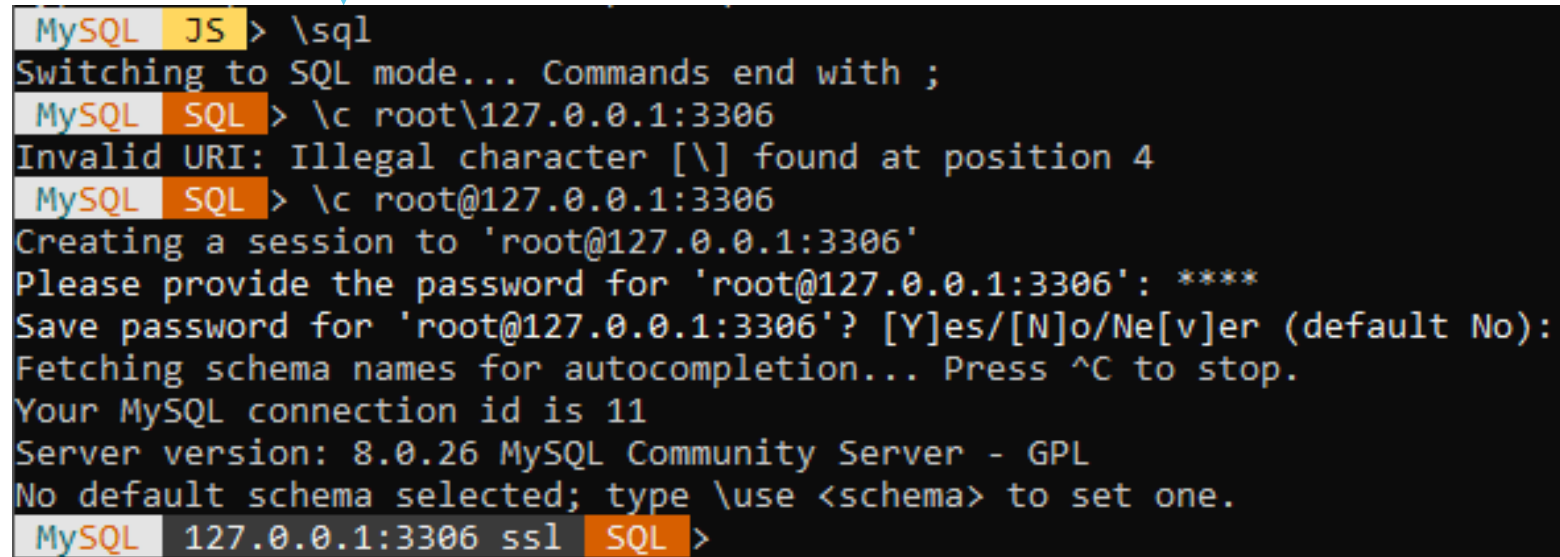
puerto

Base de datos

Diseño físico - **Mysql**

Debemos asegurarnos que estamos en mod SQL

Tecleamos \sql



```
MySQL JS > \sql
Switching to SQL mode... Commands end with ;
MySQL SQL > \c root\127.0.0.1:3306
Invalid URI: Illegal character [\] found at position 4
MySQL SQL > \c root@127.0.0.1:3306
Creating a session to 'root@127.0.0.1:3306'
Please provide the password for 'root@127.0.0.1:3306': ****
Save password for 'root@127.0.0.1:3306'? [Y]es/[N]o/[e]ver (default No):
Fetching schema names for autocompletion... Press ^C to stop.
Your MySQL connection id is 11
Server version: 8.0.26 MySQL Community Server - GPL
No default schema selected; type \use <schema> to set one.
MySQL 127.0.0.1:3306 ssl SQL >
```

\connect root@127.0.0.1:3306

Usuario

servidor

puerto

Base de datos

Diseño físico - ***Mysql***

Comandos de MySQL Shell

Funciones, instalación y comandos de MySQL Shell 8.0 - programador clic (programmerclick.com)

Comando	Descrpcion
show databases;	Muestra las base de datos del servidor
use BBDD;	Selecciona la BBDD
show tables;	Muestra tablas de la BBDD seleccionada.
describe nombre_tabla	Muestra el formato de los campos de la tabla.

Base de datos

Diseño físico

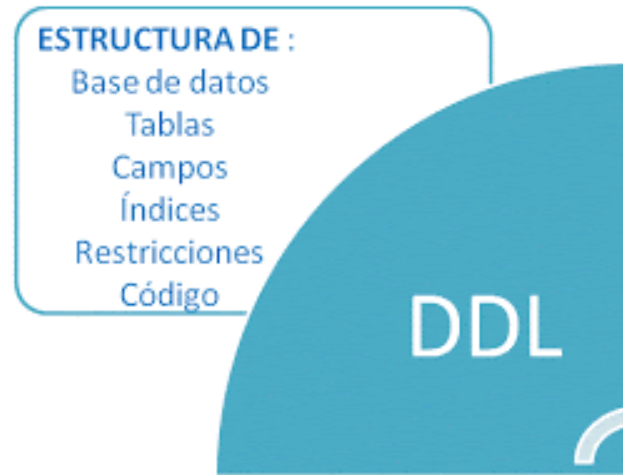
Introducción

Modelo físico

Lenguaje SQL

MySQL

Data Definition Language (DDL)



Lenguaje de Definición de Datos, DDL

Create Drop y Alter

Base de datos

Diseño físico

Lenguaje de Definición de Datos, DDL

Create

```
CREATE database tienda;
```

```
use tienda;
```

```
CREATE TABLE <tabla> (  
  <campo1> <tipo>[(<longitud>)] [NOT NULL][UNIQUE][PRIMARY KEY]  
  [CHECK <condición>] [DEFAULT <valor>], [  
  <campo2> <tipo>[(<longitud>)] [NOT NULL][UNIQUE][PRIMARY KEY]  
  CHECK <condición>][DEFAULT <valor>], ]  
  ...
```

restricciones

Los valores entre corchetes son opcionales

ESTRUCTURA DE :
Base de datos
Tablas
Campos
Índices
Restricciones
Código

DDL

Base de datos

Diseño físico

Lenguaje de Definición de Datos, DDL

- Consideraciones a la hora de nombrar las tablas
- No usar tildes ni caracteres especiales, tipo @#~€, etc...
- En el caso de palabras compuestas NO utilizar espacios , en este caso (_)
Ejemplo: tipo_juego
- Intentar que tengan una lógica con el contenido de las mismas.

ESTRUCTURA DE :
Base de datos
Tablas
Campos
Índices
Restricciones
Código

DDL

Base de datos

Diseño físico

Lenguaje de Definición de Datos, DDL

Create .- sentencia que nos permite crear

```
CREATE TABLE Nombre_Tabla  
(  
Nombre_Columna1 Tipo_de_Dato (longitud),  
Nombre_Columna2 Tipo_de_Dato (longitud),  
Nombre_Columna3 Tipo_de_Dato (longitud),  
....  
);
```

```
CREATE TABLE juegos  
(  
nombre VARCHAR(30),  
tipo VARCHAR(20) DEFAULT 'rol',  
precio int(3)  
);
```

Field	Type	Null	Key	Default	Extra
nombre	varchar(30)	YES		NULL	
tipo	varchar(25)	YES		NULL	
precio	int	YES		NULL	

ESTRUCTURA DE :
Base de datos
Tablas
Campos
Índices
Restricciones
Código

DDL

Base de datos

Diseño físico

ESTRUCTURA DE :
Base de datos
Tablas
Campos
Índices
Restricciones
Código

DDL

Lenguaje de Definición de Datos, DDL

Drop .- sentencia que nos permite borrar

DROP database tienda;

DROP TABLE nombre_tabla;

DROP TABLE IF EXISTS nombre_tabla

DROP TABLE IF EXISTS nombre_tabla, nombre_tabla2, nombre_tabla3

```
MySQL localhost:3306 ssl juegos SQL > drop database juegos;
Query OK, 1 row affected (0.0237 sec)
MySQL localhost:3306 ssl juegos SQL > show databases;
+-----+
| Database |
+-----+
| information_schema |
| mysql |
| performance_schema |
| sys |
+-----+
```

Base de datos

Diseño físico

ESTRUCTURA DE :
Base de datos
Tablas
Campos
Índices
Restricciones
Código

DDL

Lenguaje de Definición de Datos, DDL

Alter .- sentencia que nos permite modificar tabla, columnas, añadir y eliminar restricciones

```
ALTER [ONLINE | OFFLINE] [IGNORE] TABLE nombre_tabla  
[alter_specification [, alter_specification] ...]
```

Base de datos

Diseño físico

Lenguaje de Definición de Datos, DDL

ESTRUCTURA DE :
Base de datos
Tablas
Campos
Índices
Restricciones
Código

DDL

Alter – listado de modificaciones

```
# alter_specification:
# table_options
| ADD [COLUMN] col_name column_definition
    [FIRST | AFTER col_name ]
| ADD [COLUMN] (col_name column_definition,...)
| ADD {INDEX|KEY} [index_name]
    [index_type] (index_col_name,...) [index_option] ...
| ADD [CONSTRAINT [symbol]] PRIMARY KEY
    [index_type] (index_col_name,...) [index_option] ...
| ADD [CONSTRAINT [symbol]]
    UNIQUE [INDEX|KEY] [index_name]
    [index_type] (index_col_name,...) [index_option] ...
| ADD FULLTEXT [INDEX|KEY] [index_name]
    (index_col_name,...) [index_option] ...
| ADD SPATIAL [INDEX|KEY] [index_name]
    (index_col_name,...) [index_option] ...
| ADD [CONSTRAINT [symbol]]
    FOREIGN KEY [index_name] (index_col_name,...)
    reference_definition
```

```
| ALTER [COLUMN] col_name {SET DEFAULT literal | DROP DEFAULT}
| CHANGE [COLUMN] old_col_name new_col_name column_definition
    [FIRST|AFTER col_name]
| MODIFY [COLUMN] col_name column_definition
    [FIRST | AFTER col_name]
| DROP [COLUMN] col_name
| DROP PRIMARY KEY
| DROP {INDEX|KEY} index_name
| DROP FOREIGN KEY fk_symbol
| DISABLE KEYS
| ENABLE KEYS
| RENAME [TO|AS] new_tbl_name
| ORDER BY col_name [, col_name] ...
| CONVERT TO CHARACTER SET charset_name [COLLATE collation_name]
| [DEFAULT] CHARACTER SET [=] charset_name [COLLATE [=]
collation_name]
```

Base de datos

Diseño físico

Lenguaje de Definición de Datos, DDL

Alter – Ejemplos

Modificar el nombre de la tabla

```
ALTER TABLE nombre_tabla RENAME nombre_nuevo_tabla;
```

Modificar el juego de caracteres

```
ALTER TABLE nombre_tabla CHARACTER SET latin1;
```

Eliminar una columna de la tabla:

```
ALTER TABLE nombre_tabla DROP COLUMN nombre_columna;
```

Eliminar varias columnas de la tabla:

```
ALTER TABLE nombre_tabla DROP COLUMN nombre_columna, DROP COLUMN nombre_columna2;
```

ESTRUCTURA DE :
Base de datos
Tablas
Campos
Índices
Restricciones
Código

DDL

Base de datos

Diseño físico

Lenguaje de Definición de Datos, DDL

Alter – Ejemplos

Asignar como clave primaria a una columna:

```
ALTER TABLE nombre_Tabla ADD PRIMARY KEY(nombre_columna);
```

Eliminar una clave primaria y clave externa (FOREING KEY y PRIMARY KEY):

#Eliminar clave primaria

```
ALTER TABLE nombre_tabla DROP PRIMARY KEY;
```

#Eliminar clave externa

```
ALTER TABLE nombre_tabla DROP FOREIGN KEY nombre_columna;
```

ESTRUCTURA DE :
Base de datos
Tablas
Campos
Índices
Restricciones
Código

DDL

Base de datos

Diseño físico

Lenguaje de Definición de Datos, DDL

ESTRUCTURA DE :
Base de datos
Tablas
Campos
Índices
Restricciones
Código

DDL

Alter – Ejemplos

Insertar una nueva columna al final de la tabla:

```
ALTER TABLE nombre_tabla ADD fecha_nacimiento date;
```

Añadir una nueva columna después de otra:

```
ALTER TABLE nombre_tabla ADD nombre_columna VARCHAR(5) AFTER nombre_columna_anterior;
```

Añadir una nueva columna en la primera posición de la tabla:

```
ALTER TABLE nombre_tabla ADD nombre_columna VARCHAR(5) INT FIRST;
```

Renombrar columnas de una tabla:

```
ALTER TABLE nombre_tabla RENAME COLUMN nombre1 TO nombre2;
```

Modificar la definición de una columna

```
ALTER TABLE nombre_tabla CHANGE direccion direccion VARCHAR(100);
```

Base de datos

Diseño físico

Lenguaje de Definición de Datos, DDL

ESTRUCTURA DE :
Base de datos
Tablas
Campos
Índices
Restricciones
Código

DDL

Alter – Ejemplos

Podemos utilizar esta sentencia para indicar las restricciones de columna

```
ALTER TABLE nombre_tabla CHANGE COLUMN direccion direccion VARCHAR(100) NOT NULL UNIQUE;
```

La cláusula **CONSTRAINT** se usa en las instrucciones ALTER TABLE y CREATE TABLE para crear o eliminar restricciones

```
ALTER TABLE persona ADD PRIMARY KEY(id);
```

```
ALTER TABLE persona ADD CONSTRAINT PRIMARY KEY(id);
```

```
ALTER TABLE persona ADD CONSTRAINT persona_id_pk PRIMARY KEY(id)
```



No podemos cambiar el tipo de una columna si forma parte de una clave ajena ya que se dejaría de cumplir la restricción de integridad referencial

Base de datos

Diseño físico

Lenguaje de Definición de Datos, DDL

Valores por defecto

Cuando no introducimos un valor para un registro.

Si el valor es una cadena de texto o una fecha, están se delimitan con comillas

Ejemplos:

```
CREATE TABLE juegos (  
  nombre VARCHAR(25),  
  tipo VARCHAR(30) DEFAULT 'Rol');
```

```
CREATE TABLE personaje (  
  nombre VARCHAR(25),  
  nivel INT DEFAULT 1);
```


Base de datos

Diseño físico

Lenguaje de Definición de Datos, DDL

Valores por defecto

Ejemplos:

```
CREATE TABLE juegos (  
  nombre VARCHAR(25),  
  precio DECIMAL(3,1) DEFAULT 24.5,  
  tipo VARCHAR(25) DEFAULT 'Rol'  
);
```

```
CREATE TABLE cliente (  
  nombre VARCHAR(25),  
  apellidos VARCHAR(25),  
  alta DATE DEFAULT '2021/0/11'  
);
```

Base de datos

Diseño físico

Lenguaje de Definición de Datos, DDL - Restricciones

Una restricción es una condición de obligado cumplimiento para una o más columnas de la tabla.

- NOT NULL: Para que la columna tenga valores NULL.
- UNIQUE: Permite que todos los valores de una columna determinada sean valores distintos .
- PRIMARY KEY: Las restricciones NOT NULL y UNIQUE permite identificar de forma unívoca cada fila de la tabla.
- FOREIGN KEY: Para que detecte o identifique de forma única una fila o registro de otra tabla.
- CHECK: Permite asegurar que todos los valores de una columna cumplen una condición determinada.

Base de datos

Diseño físico

Lenguaje de Definición de Datos, DDL

Añadir Restricciones

- Restricción de NOT NULL y UNIQUE. Autor (dni, nombre, pais)

```
CREATE TABLE autor (  
  dni INTEGER PRIMARY KEY,  
  nombre VARCHAR(50),  
  pais VARCHAR(30) NOT NULL,  
  UNIQUE (nombre)  
);
```

Base de datos

Diseño físico

Lenguaje de Definición de Datos, DDL

Añadir Restricciones

- Restricción de clave primaria: PRIMARY KEY.
- Restricción de columna. Asignamos PRIMARY KEY a la columna codigo

Autor (código, nombre)

```
CREATE TABLE libro (  
codigo INTEGER PRIMARY KEY,  
titulo VARCHAR(50)  
) ; Restricción de columna
```

```
CREATE TABLE editoriales(  
codigo number(3) not null,  
nombre varchar2(30),  
constraint PK_editoriales primary key (codigo)  
);
```

Base de datos

Diseño físico

Lenguaje de Definición de Datos, DDL

Añadir Restricciones

Creamos la tabla "autores" con una restricción "primary key", una "unique" y una "check":

```
CREATE TABLE autores(  
codigo number(4) not null constraint CK_autores_codigo check (codigo>=0),  
nombre varchar2(30) not null,  
constraint PK_autores_codigo primary key (codigo),  
constraint UQ_autores_nombre unique (nombre)  
);
```

Base de datos

Diseño físico

Lenguaje de Definición de Datos, DDL

Añadir Restricciones

Creamos la tabla “pedidos” con una restricción “Foreign key” a la columna PersonalID

```
CREATE TABLE Pedidos (  
  PedidoID int NOT NULL PRIMARY KEY,  
  NumeroPedido int NOT NULL,  
  PersonalID int FOREIGN KEY REFERENCES Personas(PersonalID) );
```

Tabla



Campo

Base de datos

Diseño físico

Lenguaje de Definición de Datos, DDL

Añadir Restricciones

Creamos la tabla “pedidos” con una restricción “Foreign key” a la columna PersonalID

```
CREATE TABLE Pedidos (  
    PedidoID int NOT NULL,  
    NumeroPedido int NOT NULL,  
    PersonalID int,  
    PRIMARY KEY (PedidoID), CONSTRAINT FK_PedidoPersona  
    FOREIGN KEY (PersonalID) REFERENCES Personas(PersonalID) );
```

Base de datos

Diseño físico

Lenguaje de Definición de Datos, DDL

Añadir Restricciones

Modificar una tabla para añadir una restricción tipo FOREIGN KEY

Pedidos (Numero, cantidad, PersonalID)

Personas (Cliente_ID, Nombre, Direccion)

```
ALTER TABLE Pedidos ADD FOREIGN KEY (PersonalID)  
REFERENCES Personas(Cliente_ID);
```


Base de datos

Diseño físico

Lenguaje de Definición de Datos, DDL

Añadir Restricciones CHECK

Son restricciones que establecen una condición que deben cumplir los datos de una columna.

- La condición se expresa entre paréntesis y debe devolver un resultado booleano (true o false).
- Podemos utilizar los operadores lógicos y relacionales:
>, <, >=, !=, <=, =, IS NULL, IS NOT NULL, AND, NOT, OR.
- Un campo puede tener varias restricciones CHECK.

Base de datos

Diseño físico

Lenguaje de Definición de Datos, DDL

Añadir Restricciones CHECK -- Ejemplos

Restricción a nivel de tabla en el cual el importe debe ser mayor a 0

```
CREATE TABLE ingresos (  
  cod INT PRIMARY KEY,  
  concepto VARCHAR (40) NOT NULL,  
  importe INT,  
  CONSTRAINT ingresos_importe_ck CHECK (importe > 0)  
);
```

Base de datos

Diseño físico

Lenguaje de Definición de Datos, DDL

Añadir Restricciones CHECK -- Ejemplos

Restricción a nivel de columna en el cual el importe debe ser mayor a 0

```
CREATE TABLE ingresos (  
  cod INT PRIMARY KEY,  
  concepto VARCHAR (40) NOT NULL,  
  importe INT CHECK (importe > 0)  
);
```

Base de datos

Diseño físico

Lenguaje de Definición de Datos, DDL

Añadir Restricciones CHECK -- Ejemplos

Restricción a nivel de tabla en el cual el importe debe estar entre los valores

```
CREATE TABLE ingresos (  
  cod INT PRIMARY KEY,  
  concepto VARCHAR (40) NOT NULL,  
  importe INT,  
  CONSTRAINT ingresos_importe_1_ck  
  CHECK (importe > 0),  
  CONSTRAINT ingresos_importe_2_ck  
  CHECK (importe < 5000)  
);
```

Base de datos

Diseño físico

Lenguaje de Definición de Datos, DDL

Añadir Restricciones CHECK -- Ejemplos

Restricción a nivel de tabla en el cual el importe debe estar entre los valores
Utilizando operadores lógicos.

```
CREATE TABLE ingresos (  
  cod INT PRIMARY KEY,  
  concepto VARCHAR (40) NOT NULL,  
  importe INT,  
  CONSTRAINT ingresos_importe_ck  
  CHECK (importe > 0 AND importe < 8000)  
);
```

Base de datos

Diseño físico

Lenguaje de Definición de Datos, DDL

Añadir Restricciones CHECK -- Ejemplos

Restricción a nivel de columna en el cual el importe debe estar entre los valores
Utilizando operadores lógicos.

```
CREATE TABLE ingresos (  
  cod INT PRIMARY KEY,  
  concepto VARCHAR (40) NOT NULL,  
  importe INT CHECK (importe > 0)  
             CHECK (importe < 5000)  
);
```

```
CREATE TABLE ingresos (  
  cod INT PRIMARY KEY,  
  concepto VARCHAR (40) NOT NULL,  
  importe INT CHECK (importe > 0  
             AND importe < 8000)  
);
```

Base de datos

Diseño físico

Lenguaje de Definición de Datos, DDL

Añadir Restricciones **AUTO_INCREMENT** -- Ejemplos

Genera automáticamente valores numéricos secuenciales cada vez que se inserta una fila.

Muy útil para identificar filas como únicas.

```
CREATE TABLE autor (  
  codigo INTEGER PRIMARY KEY AUTO_INCREMENT,  
  nombre VARCHAR(50)  
);
```

Codigo	nombre
1	Rafa
2	Daniel