



# MODELOS DE DATOS

*D.D.L. (Data Definition Language)*

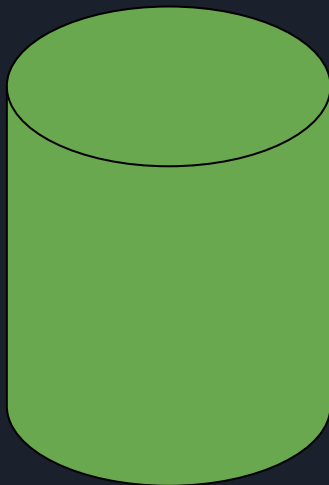


RECORDERIS...





# ¿QUÉ APRENDEREMOS HOY?



- DDL (Data Definition Language)
- Cómo escribir el DDL a partir de un modelo relacional



# Recordenis!!!

CLIENTES		
ID	NOMBRE	DIRECCIÓN
12345	BILL GATES	CALLE 50 # 40 - 30
45678	STEVE JOBS	CARRERA 1 SUR # 10 - 15
54321	JEFF BEZOS	

- Nombre de la tabla: **CLIENTES**
- Campos:
  - **ID (int) not null**
  - **NOMBRE varchar(100) not null**
  - **DIRECCION varchar(100) null**
- Cantidad de registros: 3

Lo que está en **verde** se llama la **estructura de la tabla**. Básicamente es el nombre, los campos y los tipos de dato de cada campo.



# Qué significa DDL?

DDL

Data Definition Language (lenguaje de definición de datos)

Permite definir **la estructura** de los objetos de la base de datos

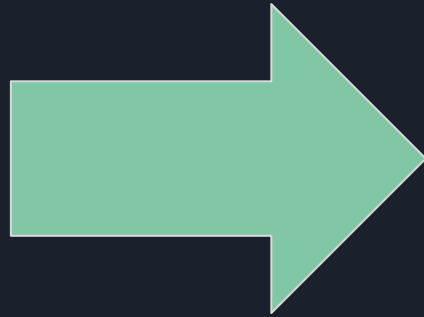
No tiene nada que ver con el manejo de los datos

Se usa para **crear, alterar o eliminar** objetos



# *Materializando el M.R.*

Modelo  
Relacional



SQL



# Ejemplo 1: algo simple

M.R.

CUSTOMERS	
ID	VARCHAR(20)
ID_TYPE	VARCHAR(3)
FULL_NAME	VARCHAR(100)

SQL

```
create table customers(  
    id          varchar(20),  
    id_type     varchar(3),  
    full_name   varchar(100)  
);
```



## Ejemplo 2: con clave primaria y obligatorios / opcionales

M.R.

CUSTOMERS		
PK	ID	VARCHAR(20)
*	ID_TYPE	VARCHAR(3)
*	FULL_NAME	VARCHAR(100)

SQL

```
create table customers(  
    id          varchar(20)  not null,  
    id_type     varchar(3)   not null,  
    full_name   varchar(100) not null,  
    primary key(id)  
);
```



# Ejemplo 3: clave primaria compuesta

M.R.

CUSTOMERS		
PK	ID	VARCHAR(20)
PK	ID_TYPE	VARCHAR(3)
*	FULL_NAME	VARCHAR(100)

SQL

```
create table customers(  
    id          varchar(20)  not null,  
    id_type     varchar(3)   not null,  
    full_name   varchar(100) not null,  
    primary key(id, id_type)  
);
```

# Ejemplo 4: con clave foránea

M.R.

CUSTOMERS		
PK	ID	VARCHAR(20)
*	ID_TYPE	VARCHAR(3)
*	FULL_NAME	VARCHAR(100)

SALES		
PK	ID_INVOICE	VARCHAR(20)
*	VALUE	FLOAT
FK	ID_CUSTOMER	VARCHAR(20)

SQL

```
create table customers(  
    id          varchar(20)  not null,  
    id_type     varchar(3)   not null,  
    full_name   varchar(100) not null,  
    primary key (id)  
);  
  
create table sales(  
    id_invoice  varchar(20)  not null,  
    value       float        not null,  
    id_customer varchar(20)  not null,  
    primary key (id_invoice),  
    constraint foreign key (id_customer)  
    references customers(id)  
);
```

# Ejemplo 5: Consultar lo existente

```
show databases;
```

Database
information_schema
mysql
performance_schema
sakila
sys
world

6 rows in set (0,012 sec)

```
show tables;
```

Tables_in_world
city
country
countrylanguage

3 rows in set (0.00 sec)

```
desc customers;
```

Field	Type	Null	Key	Default	Extra
id	varchar(20)	NO	PRI	NULL	
id_type	varchar(3)	NO		NULL	
full_name	varchar(100)	NO		NULL	

3 rows in set (0.03 sec)

```
desc sales;
```

Field	Type	Null	Key	Default	Extra
id_invoice	varchar(20)	NO	PRI	NULL	
value	float	NO		NULL	
id_customer	varchar(20)	NO	MUL	NULL	

3 rows in set (0.00 sec)

# Ejemplo 6: Modificar una tabla (+)

M.R.

CUSTOMERS		
PK	ID	VARCHAR(20)
PK	ID_TYPE	VARCHAR(3)
*	FULL_NAME	VARCHAR(100)
*	ADDRESS	VARCHAR(100)

SQL

```
alter table customers  
add column address varchar(100) not null ;  
  
desc customers; (...)
```

# Ejemplo 7: Modificar una tabla (-)

M.R.

CUSTOMERS		
PK	ID	VARCHAR(20)
PK	ID_TYPE	VARCHAR(3)
*	ADDRESS	VARCHAR(100)

SQL

```
alter table customers  
drop column full_name ;  
  
desc customers; (...)
```

# Ejemplo 8: Modificar una tabla (+ / -)

M.R.

CUSTOMERS		
PK	ID	VARCHAR(20)
PK	ID_TYPE	VARCHAR(3)
*	FULL_NAME	VARCHAR(100)

SQL

```
alter table customers
add column full_name varchar(100) not null,
drop column address ;

desc customers; (...)
```

# Ejemplo 9: Eliminar una tabla

```
show databases;
```

```
+-----+  
| Database |  
+-----+  
| bd2020   |  
| tabd2020 |  
+-----+  
2 rows in set (0.00 sec)
```

```
show tables;
```

```
+-----+  
| Tables_in_bd2020 |  
+-----+  
| customers         |  
| sales             |  
+-----+  
2 rows in set (0.00 sec)
```

```
drop table sales;  
drop table customers;
```

Ejercicio: Crear las tablas nuevamente y borrar primero "customers" y luego "sales". ¿Que ocurre y porqué?

# Ejemplo 10: Opciones adicionales

```
create table customers (  
    id int not null auto_increment,  
    id_type varchar(3) not null,  
    full_name varchar(100) null,  
    primary key (id,id_type)  
);
```

```
create table customers (  
    id int not null auto_increment,  
    id_type varchar(3) not null,  
    full_name varchar(100) null,  
    email varchar(100) not null unique,  
    primary key (id,id_type)  
);
```

```
create table customers (  
    id int not null auto_increment,  
    id_type varchar(3) not null default 'CC',  
    full_name varchar(100) null,  
    primary key (id,id_type)  
);
```

```
create table customers (  
    id int not null auto_increment,  
    id_type varchar(3) not null default 'CC',  
    full_name varchar(100) null,  
    email varchar(100) not null unique,  
    genre enum('F','M') not null,  
    primary key (id,id_type)  
);
```





# Primero lo primero: crear BD

Se debe crear una base de datos y luego situarse en ella para que MariaDB/MySQL sepan donde ejecutar las operaciones.

## SQL

```
create database my_database;  
use my_database;
```



# Ejercicios (1 / 2)

Ejecutar uno por uno los ejercicios descritos en las diapositivas previas.



## Ejercicios ( 2 / 2 )

Revisar el modelo relacional de la pizzería en uvirtual.  
Posteriormente escribir las sentencias ddl para crear las tablas acorde a dicho modelo.

Ambos modelos en uvirtual son iguales, fueron generados con herramientas diferentes. Usar el que más claro le parezca.



# ¿PREGUNTAS?

