

MySQL

Tipos de Datos

Numéricos

Texto (String)

Fecha / Hora

Tipos de datos numéricos

- INT (INTEGER)
 - 2,147,483,648 - 2,147,483,647
 - Unsigned 0 - 4,294,967,295
- TINYINT
 - 128 - 127
 - Unsigned 0 - 255
- BOOLEAN equivale a TINYINT(1), donde 0 → false, <>0 → true
 - Explicitarlo como TINYINT UNSIGNED
- SMALLINT, MEDIUMINT, BIGINT
- Uso de **AUTO_INCREMENT**

SELECT con datos numéricos

```
SELECT *
```

```
FROM peliculas
```

```
WHERE anio = 2000;
```

```
SELECT *
```

```
FROM peliculas
```

```
WHERE anio >= 2000;
```

Tipos de datos numéricos

DECIMAL(10,2) → 10 dígitos en total (precisión), 2 decimales

FLOAT → No recomendado. Puede perder precisión (7.99999999)

DOUBLE → Tiene el doble de precisión que FLOAT (hasta 15 decimales)

SELECT con numéricos

```
SELECT *
```

```
FROM peliculas
```

```
WHERE rating = 2.0;
```

```
SELECT *
```

```
FROM peliculas
```

```
WHERE rating >= 2.5;
```

Tipo de datos de texto (String)

CHAR(100) → 100 caracteres (ocupa 100 bytes en memoria)

VARCHAR(100) → 100 caracteres (ocupa *hasta* 100 bytes)

TEXT → Texto libre (post de un blog)

ENUM ('mal', 'regular', 'bien', 'muy bien', 'sobresaliente')

Tipo de datos de Fecha / Hora

DATE

TIME

DATETIME

Práctica

CONSTRAINTS (restricciones)

Son reglas que deben cumplir los datos de una tabla. Se especifican con DDL.

Se verifican ante cada operación INSERT, UPDATE o DELETE

- PRIMARY KEY
- UNIQUE KEY
- DEFAULT
- Datos inválidos según el tipo de dato / ENUM / NOT NULL
- FOREIGN KEY

NOT NULL (nulos)

NULL significa que no se conoce el valor para ese campo

NULL **NO** es vacío ' ' (blank) ni cero (0)

Cuando se arma la estructura de la tabla, se especifica con NOT NULL los campos que deben tener algún valor.

UNIQUE KEY

UNIQUE KEY asegura que el campo tenga un valor único en la tabla (o NULL)

Ejemplos:

- DNI
- ISBN (libros)
- Número de vuelo + Fecha/hora

PRIMARY KEY (PK) - Clave Primaria

Una PRIMARY KEY es una combinación de NOT NULL y UNIQUE KEY.

Una PK es un campo (o a una combinación de campos) que **identifica a cada fila** de una tabla de forma única.

Cada tabla debiera tener su PK ya que facilita la búsqueda de sus registros.

Unique Keys - Si el campo DNI es nullable

id	DNI	Apellido
1	30653013	Valdez
2	38222555	Prado
3	38222555	Portero

id	DNI	Apellido
1		Valdez
2		Prado
3		Portero

DEFAULT VALUE

Valor por default que se le asigna al crear un registro, en caso de no especificarlo

Un ejemplo típico es tener un campo con la fecha de creación de registro

Los campos de tipo NULL, sin DEFAULT VALUE, se le asigna NULL.

DDL

DDL - Creación de tablas

```
CREATE TABLE nombre_de_la_tabla (  
    columna1  TIPO_DE_DATO,  
    ... columna2 TIPO_DE_DATO  
);
```

DDL - Creación de tablas

```
CREATE TABLE nombre_de_la_tabla (  
    columna1    TIPO_DE_DATO [NULL | NOT NULL] [DEFAULT valor] [PRIMARY KEY]  
    ...  
);
```

DDL - Creación de tablas

```
CREATE TABLE post (  
    id          INT UNSIGNED PRIMARY KEY AUTO_INCREMENT,  
    titulo      VARCHAR(200) NOT NULL  
);
```

DDL - Creación de tablas

```
CREATE TABLE peliculas (  
    id                INT UNSIGNED PRIMARY KEY AUTO_INCREMENT,  
    titulo            VARCHAR(500) NOT NULL,  
    rating            DECIMAL(3,1) UNSIGNED NOT NULL,  
    premios           INT UNSIGNED DEFAULT 0,  
    fecha_de_estreno  DATE NOT NULL,  
    duracion          INT UNSIGNED NULL  
);
```

DDL - Borrado de tablas

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

DDL - Modificación de tablas

ALTER TABLE nombre_de_tabla

ADD columna3 TIPO_DE_DATO... [FIRST | AFTER columna2]

MODIFY columna1 TIPO_NUEVO_DE_DATO

DROP columna4

DDL - Modificación de tablas

```
ALTER TABLE peliculas  
  ADD rating DECIMAL(3,1) UNSIGNED NOT NULL  
;
```

DDL - Modificación de tablas

```
ALTER TABLE peliculas  
    MODIFY rating DECIMAL(4,1) UNSIGNED NOT NULL  
;
```


DDL - Modificación de tablas

```
ALTER TABLE peliculas  
    DROP rating  
;
```

Práctica

DML

INSERT

UPDATE

DELETE

DML - Insert

```
INSERT INTO nombre_de_tabla (campo1, campo2, campo4)  
VALUES (valor_campo_1, valor_campo_2, valor_campo4);
```

DML - Insert

```
INSERT INTO peliculas (titulo, rating, premios, fecha_de_estreno, duracion)  
VALUES ('Avatar', 2.9, 2, '2003-10-04', 120);
```

DML - Insert

```
INSERT INTO nombre_de_tabla
```

```
VALUES (valor_campo_1, valor_campo_2, valor_campo3, valor_campo4);
```

Todos los campos de la tabla tienen que estar especificados

DML - Insert

```
INSERT INTO peliculas  
VALUES (99, 'Avatar', 2.9, 2, '2003-10-04', 120);
```

DML - Update

UPDATE nombre_de_tabla

SET

campo1 = nuevo_valor_1

[, campo2 = nuevo_valor_2]

[WHERE condicion]

DML - Update

```
UPDATE peliculas  
SET rating = 9.9  
WHERE id = 15;
```

```
UPDATE peliculas  
SET rating = 9.9  
WHERE  
    id = 15  
    AND titulo = 'La vida es bella'  
;
```

DML - Delete

```
DELETE FROM nombre_de_tabla
```

```
[WHERE condicion]
```

DML - Delete

```
DELETE FROM peliculas  
WHERE fecha_de_estreno < '1990-01-01'
```

```
DELETE FROM peliculas;
```

Práctica