

# Práctico SQL

## Tipos de datos

Tipo de dato	Ejemplo
SMALLINT [2 bytes]	
INT [4 bytes]	
BIGINT [8 bytes]	
BIT	
DECIMAL (TAMAÑO TOTAL, NUM DECIMALES)	DECIMAL(5,2) → 300,25 DECIMAL(6,1) → 40000,5
NUMERIC (TAMAÑO TOTAL, NUM DECIMALES) <i>* Es lo mismo que DECIMAL</i>	NUMERIC(5,2) → 300,25 NUMERIC(6,1) → 40000,5
REAL [4 bytes]	
FLOAT [8 bytes]	
DATE	
TIME	
DATETIME	
CHAR (MAX)	CHAR(1) → A CHAR(2) → AB
VARCHAR (MAX)	VARCHAR(9) → 12345678A
TEXT	
SMALLMONEY	
MONEY	

## Fechas y horas

### Interval

Se expresa como: INTERVAL *expr unit*, **donde:**

unit	expr
DAY	DAYS
DAY_HOUR	'DAYS HOURS'
DAY_MICROSECOND	'DAYS HOURS:MINUTES:SECONDS.MICROSECONDS'
DAY_MINUTE	'DAYS HOURS:MINUTES'
DAY_SECOND	'DAYS HOURS:MINUTES:SECONDS'
HOUR	HOURS
HOUR_MICROSECOND	'HOURS:MINUTES:SECONDS.MICROSECONDS'
HOUR_MINUTE	'HOURS:MINUTES'
HOUR_SECOND	'HOURS:MINUTES:SECONDS'
MICROSECOND	MICROSECONDS
MINUTE	MINUTES
MINUTE_MICROSECOND	'MINUTES:SECONDS.MICROSECONDS'
MINUTE_SECOND	'MINUTES:SECONDS'
MONTH	MONTHS
QUARTER	QUARTERS
SECOND	SECONDS
SECOND_MICROSECOND	'SECONDS.MICROSECONDS'

```

FLUSH PRIVILEGES;
# Otorgar privilegios específicos
GRANT <privilegio> ON <db>.<table> TO 'david'@'localhost';
# y privilegio puede ser: CREATE, DROP, DELETE, INSERT, SELECT,
UPDATE, GRANT OPTION [para modificar permisos de otros usuarios]

# Actualizar la contraseña de un usuario
ALTER USER 'david'@'%' IDENTIFIED BY 'password';

# Eliminar un usuario si existe
DROP USER IF EXISTS 'david'@'%';

```

## Creación y borrado de una base de datos

```

CREATE DATABASE IF NOT EXISTS <db> CHARACTER SET utf8 COLLATE
utf8_general_ci;

USE <db>;

DROP DATABASE IF EXISTS <db>;

```

## Creación de tablas

```

CREATE TABLE CUSTOMERS (
    ID      INT                NOT NULL,
    NAME    VARCHAR (20)       NOT NULL,
    AGE     INT                NOT NULL UNIQUE,
    ADDRESS CHAR (25),
    SALARY  DECIMAL (18, 2) DEFAULT 20000.00,
    CONSTRAINT pk_customers PRIMARY KEY (ID),
    CONSTRAINT chck_customers_age CHECK (AGE >= 18)
);

```

- Si quisiéramos añadir una columna a posteriori:

```

ALTER TABLE auditoria ADD fechahora DATETIME NOT NULL DEFAULT
CURRENT_TIMESTAMP;
... ó ...
ALTER TABLE auditoria ADD fechahora DATETIME NOT NULL DEFAULT
NOW();

```

- Si quisiéramos añadir una restricción modificando la columna:

# Normalización

## Dependencia funcional

Uno de los conceptos fundamentales en la normalización es el de *dependencia funcional*. Una dependencia funcional es una *relación entre atributos de una misma tabla*. Si  $x$  e  $y$  son atributos de la relación  $R$ , se dice que  $y$  es funcionalmente dependiente de  $x$  (se denota por  $x \longrightarrow y$ ) si cada valor de  $x$  tiene asociado un solo valor de  $y$  ( $x$  e  $y$  pueden constar de uno o varios atributos). A  $x$  se le denomina *determinante*, ya que  $x$  determina el valor de  $y$ . Se dice que el atributo  $y$  es *completamente dependiente* de  $x$  si depende funcionalmente de  $x$  y no depende de ningún subconjunto de  $x$ .

## Primera forma normal (1FN)

### Primera forma normal

Una tabla está en primera forma normal (1FN) si, y sólo si, todos los dominios de sus atributos contienen valores atómicos, es decir, no hay *grupos repetitivos*. Un grupo repetitivo es *un atributo que puede tener múltiples valores para cada fila de la relación*. Son los atributos que tienen forma de tabla.

```
PRODUCTO(codprod, nombre, VERSIÓN(número, fecha, ventas))
```



```
PRODUCTO(codprod, nombre)
```

```
VERSIÓN(codprod, número, fecha, ventas)
```

```
VERSIÓN.codprod es una clave ajena a PRODUCTO
```