

# BBDD

## Bases de Datos – Modelo Físico



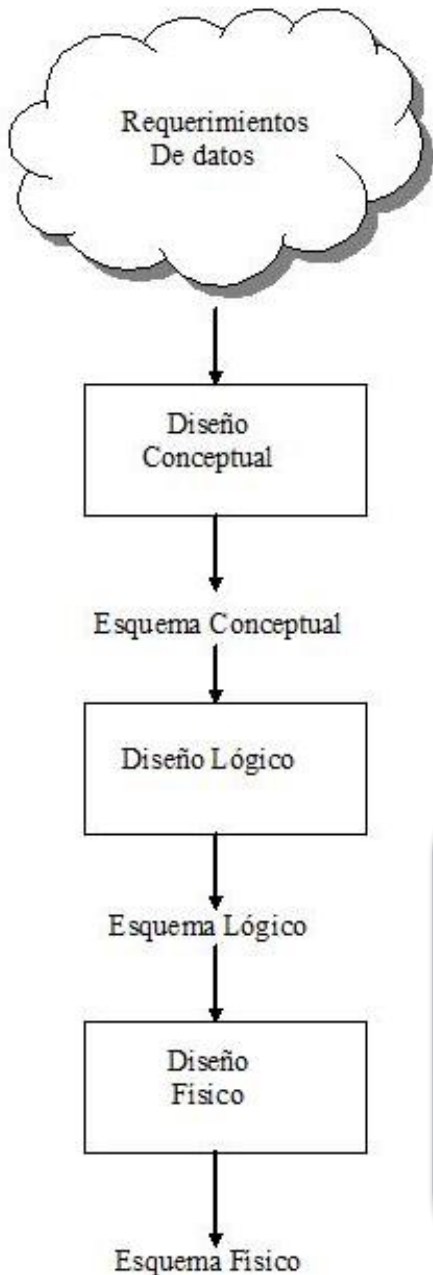
# Modelización de BD

El modelado transcurre en tres fases:

**1) Modelo conceptual:** Modelo expresivo y entendible por el usuario experto en negocio (pero inexperto en informática). Representa el dominio del problema tal y como el usuario lo concibe. Nosotros trabajamos el **modelo E/R**.

**2) Modelo lógico:** Más técnico y que tienen traducción directa al modelo físico que entiende el futuro SGBD. Este modelo depende del tipo de BD (no es igual modelar una BD orientada a objetos que una relacional). Nosotros veremos el **Modelo relacional**.

**3) Modelo físico:** Resultado de aplicar el modelo lógico a un SGBD concreto. Habitualmente se expresa en un lenguaje de programación de BBDD como SQL. La transformación del Modelo Relacional al modelo físico se lleva a cabo mediante el sub-lenguaje DDL (lenguaje de definición de datos) de SQL. Nosotros lo veremos orientado al **SGBD MySQL**, sobre el que finalmente implementaremos la BD.



# Objetivos

- Manejar el lenguaje de definición de datos para crear, modificar y borrar bases de datos y tablas tanto desde la interfaz de línea de comandos como desde una interfaz gráfica.
- Implementar restricciones durante el proceso de creación de tablas PRIMARY KEY, ON DELETE y ON UPDATE.
- Conocer los tipos de herramientas gráficas proporcionadas por los SGBD.



# Introducción

El **modelo físico** es un modelo específico dependiente del SGBD (y por tanto también del tipo de base de datos) sobre el que se implanta la base de datos y que determina cómo se almacenarán los datos identificados durante el modelo lógico.

El modelo físico, por tanto, se trata de una descripción detallada de la implementación de una BD y se expresará mediante un lenguaje de definición de datos.

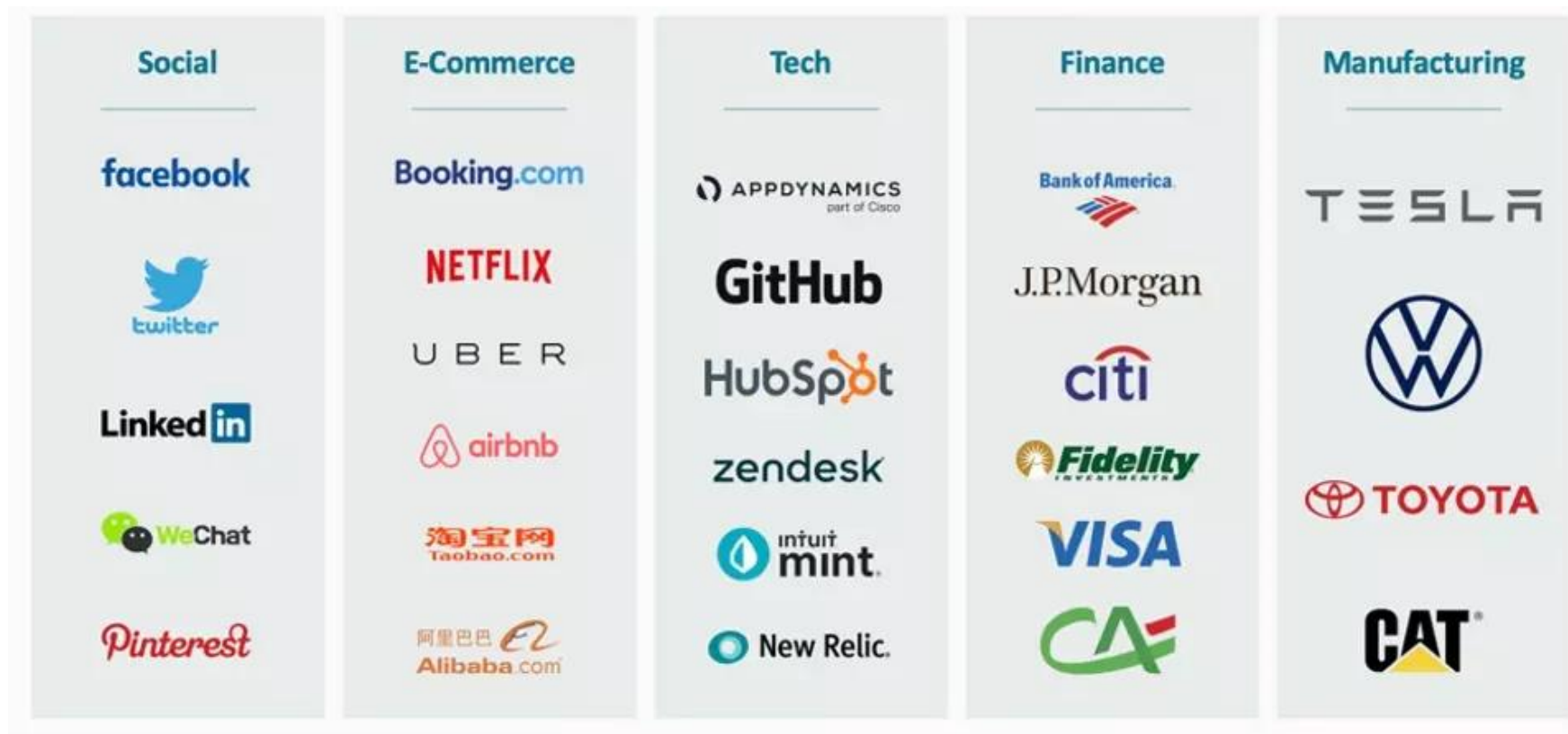
En nuestro caso (BD relacional), el SGBD que usaremos será **MySQL** y el lenguaje será **SQL**.



<https://dev.mysql.com/doc/refman/8.0/en/>

# Introducción

## ¿Por qué MySQL?



<https://dev.mysql.com/doc/refman/8.0/en/>

# Lenguaje SQL

El lenguaje de programación **SQL** es el **lenguaje fundamental de los SGBD** relacionales y los elementos que lo componen son:

**a) DDL (*Data Definition Language*):** permite la **definición, modificación y eliminación de las estructuras básicas** (BD, tablas, etc.) en un SGBD.

**b) DML (*Data Manipulation Language*):** es el lenguaje que consulta o manipula los datos ya existentes de nuestra BD.

**a) DCL (*Data Control Language*):** administra a los usuarios de la BD, concediendo o denegando los permisos oportunos.

**b) TCL (*Transaction Control Language*):** lenguaje que controla el procesamiento de las transacciones de la BD.

# Lenguaje SQL

## **Normas básicas** sobre las instrucciones en lenguaje **SQL**:

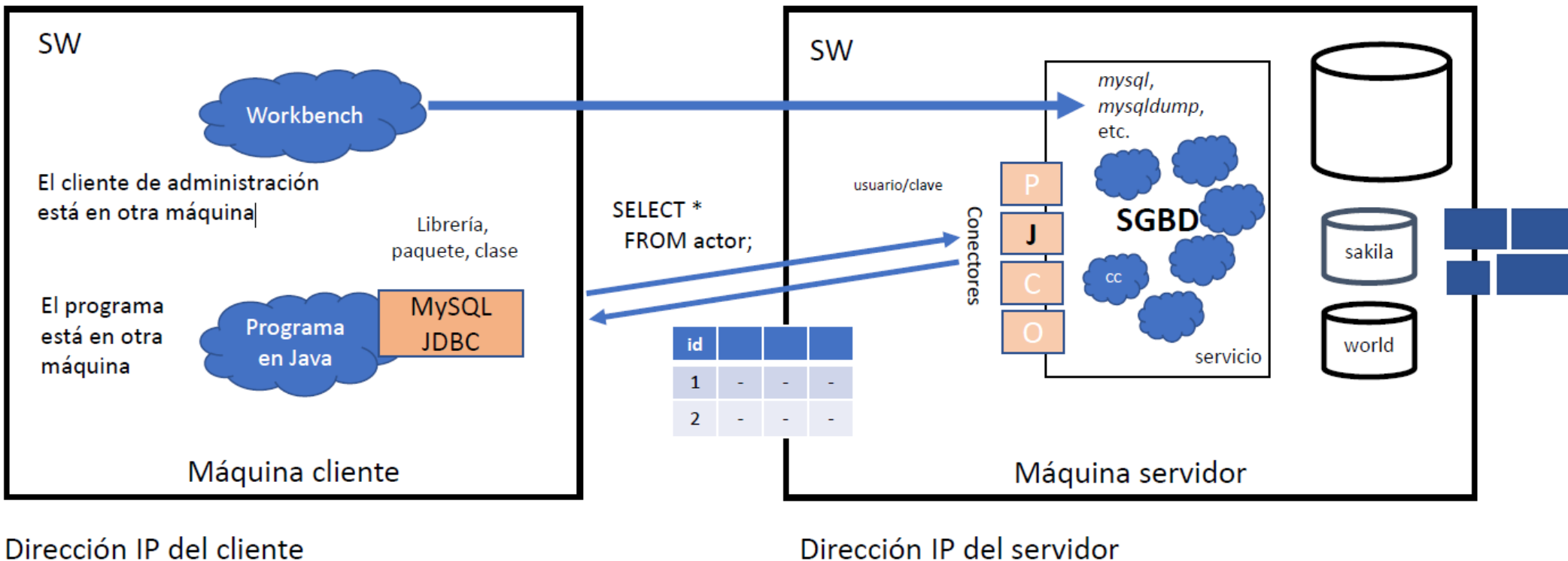
1. No se distingue entre mayúsculas y minúsculas.
2. Las instrucciones deben terminar con el carácter “;” (punto y coma).
3. Antes de finalizar la instrucción, cualquier comando puede ir seguido por: un espacio en blanco o un salto de línea.
4. Se recomienda usar mayúsculas para las sentencias y palabras reservadas, y las minúsculas para los nombres definidos por el usuario.



# Opciones de despliegue de BBDD

## Opción 1 (ámbito empresarial)

SGDB: Despliegue en servidor y cliente por separado

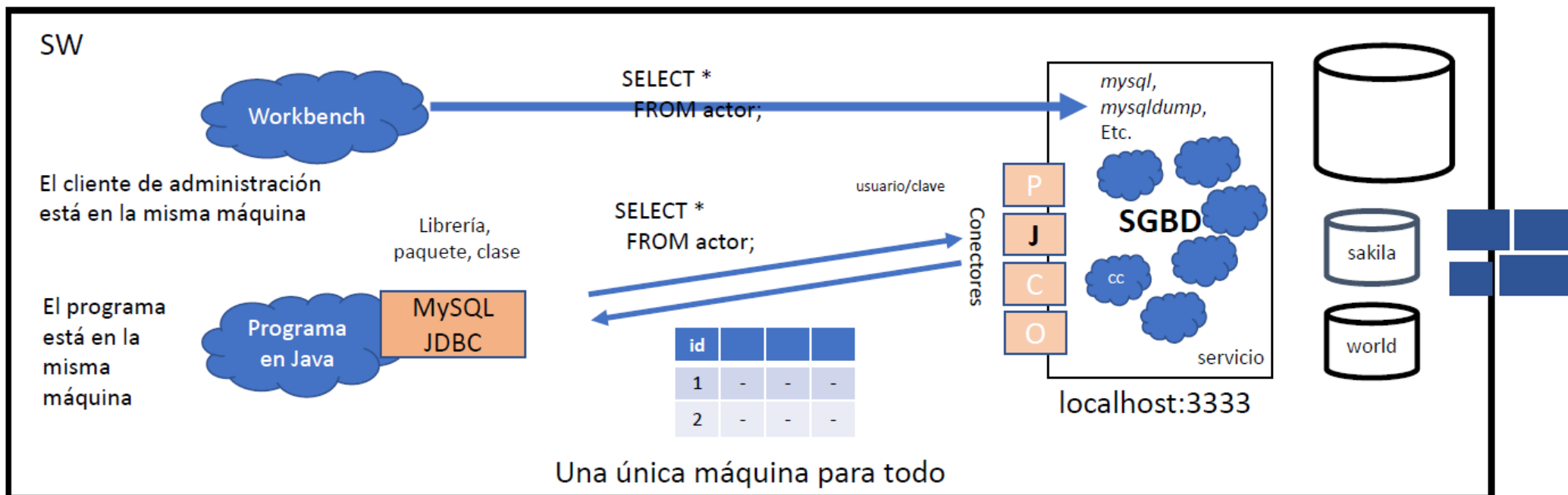




# Opciones de despliegue de BBDD

## Opción 2 (ámbito local/docente)

SGDB: Despliegue “stand-alone”



Dirección IP local: 127.0.0.1 o localhost

# SGBD MySQL: interacción desde cliente

- Interfaz de línea de comandos (consola)

```
Símbolo del sistema - mysql -uroot -p

C:\Users\jsala>mysql -uroot -p
Enter password: ****
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 77
Server version: 8.0.31 MySQL Community Server - GPL

Copyright (c) 2000, 2022, 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.

mysql>
```

```
Windows PowerShell

Windows PowerShell
Copyright (C) Microsoft Corporation. Todos los derechos reservados.

Prueba la nueva tecnología PowerShell multiplataforma https://aka.ms/pscore6

PS C:\Users\jsala> mysql -uroot -p
Enter password: ****
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 78
Server version: 8.0.31 MySQL Community Server - GPL

Copyright (c) 2000, 2022, 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.

mysql>
```

- Interfaz gráfica (aplicaciones gráficas)



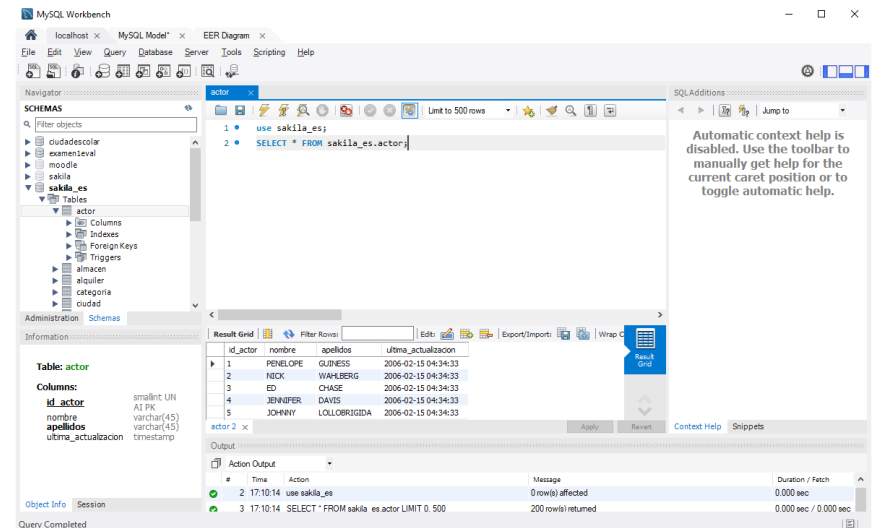
dbeaver



MySQL  
Workbench



HeidiSQL



# SGBD MySQL: Consola (admón. BDs)

Establecer conexión con el servidor (SGBD):

```
mysql -P3333 -uroot -p
```

```
mysql -uroot
```

```
mysql -uroot -p
```

Listar bases de datos disponibles:

```
SHOW DATABASES;
```

Seleccionar base de datos con la que interactuar:

```
USE nombre_bd;
```

Listar tablas disponibles en una BD:

```
SHOW TABLES;
```

Ver detalles de una tabla:

```
DESCRIBE nombre_tabla;
```

```
DESC nombre_tabla;
```

Ver SQL de creación de una BD:

```
SHOW CREATE DATABASE nombre_bd;
```

Ver SQL de creación de una tabla:

```
SHOW CREATE TABLE nombre_tabla;
```

Ejecutar script SQL desde CLI:

```
SOURCE fichero.sql
```

# SGBD MySQL: Creación de BD

## Creación de bases de datos

```
CREATE DATABASE nombre_bd;
```

```
CREATE DATABASE IF NOT EXISTS nombre_bd;
```

```
CREATE DATABASE nombre_bd CHARACTER SET juego COLLATE cotejamiento;
```

- **CHARACTER SET**: Determina el juego de caracteres que se usarán en la BD. Nosotros usaremos **UTF8mb4** (Unicode con hasta 4 bytes) que se fija por defecto en MySQL para evitar problemas de acentos, eñes, etc.
- **COLLATE**: Determina el criterio de ordenación de cadenas de caracteres que se usará la BD. Por defecto para UTF8mb4, MySQL establece **utf8mb4\_0900\_ai\_ci**  
El cotejamiento más común es:
  - accent-sensitive (**\_as**): Los caracteres a y á son diferentes.
  - accent-insensitive (**\_ai**): Los caracteres a y á son iguales.
  - case-sensitive (**\_cs**): Los caracteres a y A son diferentes.
  - case-insensitive (**\_ci**): Los caracteres a y A son iguales.
- **IF NOT EXISTS**: Clausula que evita error durante la creación de una BD que ya exista.

```
SHOW COLLATION;
```

```
SHOW CHARACTER SET;
```

```
SELECT schema_name AS 'database',  
       default_character_set_name AS 'charset',  
       default_collation_name AS 'collation'  
FROM   information_schema.SCHEMATA  
WHERE  schema_name = "nombre_bd";
```

# SGBD MySQL: Borrado y modificación BD

## Borrado de bases de datos

```
DROP DATABASE nombre_bd;
```

```
DROP DATABASE IF EXISTS nombre_bd;
```

- **IF NOT EXISTS:** Clausula que evita error durante la creación de una BD que ya exista.
- Borraremos la BD y también todos los datos almacenados. ¡Es irreversible!

## Modificación de bases de datos

```
ALTER DATABASE nombre_bd  
    CHARACTER SET charset_name  
    | COLLATE collation_name  
    | ENCRYPTION {'Y' | 'N'}  
    | READ ONLY {0 | 1};
```

# SGBD MySQL: Restricciones

## Creación de tablas (versión simplificada)

Elementos de una tabla: Nombre de la tabla, tipos de datos y **restricciones**

La definición de una columna consta del nombre de la columna, un tipo de datos predefinido, un conjunto de definiciones por defecto y restricciones de columna.

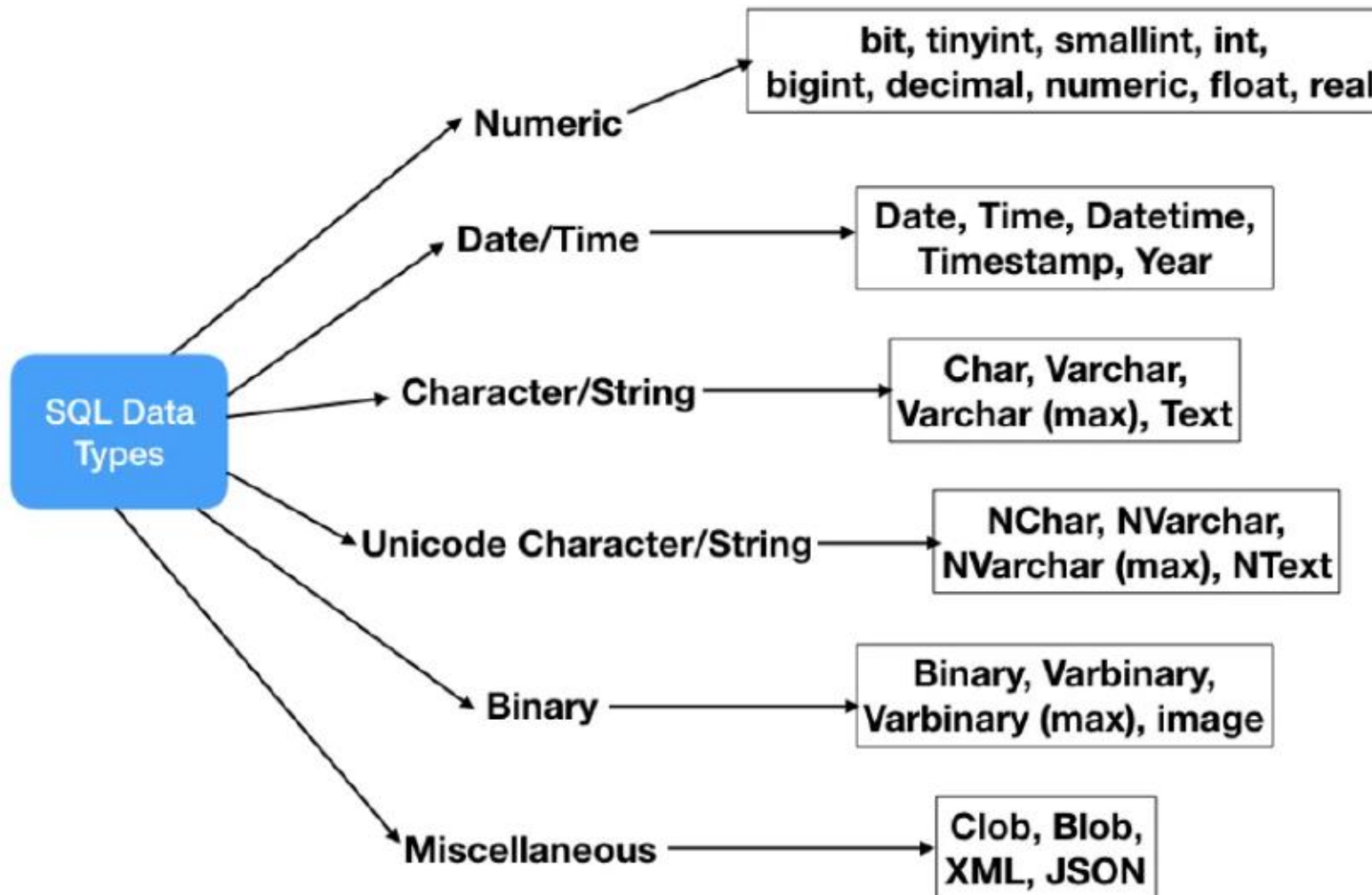
### Restricciones sobre las columnas de una tabla:

- **NOT NULL:** Evita que un campo pueda tener valores vacíos.
- **UNIQUE:** Se asegura de que todos los valores de una columna sean distintos.
- **PRIMARY KEY:** Clave primaria. Combina NOT NULL y UNIQUE. Tiene un índice asociado.
- **FOREIGN KEY:** Clave foránea. Identifica de forma única una fila de otra tabla.
- **CHECK:** Garantiza que todos los valores de una columna cumplen determinada condición.
- **DEFAULT:** Fija el valor por defecto para un campo cuando no se especifica uno.
- **AUTO\_INCREMENT:** Columna autonumérica. Su valor se incrementa automáticamente en cada inserción de una fila. Sólo se utiliza en campos de tipo entero.

# SGBD MySQL: Tipos de datos

## Creación de tablas (versión simplificada)

Elementos de una tabla: Nombre de la tabla, tipos de datos y restricciones



# SGBD MySQL: Tipos de datos

## Ejemplos con tipos de datos en MySQL

### Cadenas de caracteres:

- nombre CHAR(20)
- Inscrito ('s','n') CHAR(1)
- cp CHAR(5)
- descripcion VARCHAR (500)
- declaracion VARCHAR (2000)
- cod\_país CHAR(3)
- activo ('X','') CHAR(1)

Value	CHAR ( 4)	Storage Required	VARCHAR ( 4)	Storage Required
' '	' '	4 bytes	' '	1 byte
'ab'	'ab '	4 bytes	'ab'	3 bytes
'abcd'	'abcd'	4 bytes	'abcd'	5 bytes
'abcdefgh'	'abcd'	4 bytes	'abcd'	5 bytes



# SGBD MySQL: Tipos de datos

## Ejemplos con tipos de datos en MySQL

### Números:

- edad TINYINT UNSIGNED
- num\_ciudades SMALLINT UNSIGNED
- Saldo INT
- peso DECIMAL (5,2) /\* desde 0 hasta 999.99 \*/
- peso NUMERIC (5,2) /\* desde 0 hasta 999.99 \*/
- cantidad\_grande FLOAT (15,5) /\*FLOAT Y DOUBLE no son exactos, son aproximaciones\*/
- cantidad\_binaria BIT(3)

Type	Storage (Bytes)	Minimum Value Signed	Minimum Value Unsigned	Maximum Value Signed	Maximum Value Unsigned
TINYINT	1	-128	0	127	255
SMALLINT	2	-32768	0	32767	65535
MEDIUMINT	3	-8388608	0	8388607	16777215
INT	4	-2147483648	0	2147483647	4294967295
BIGINT	8	$-2^{63}$	0	$2^{63}-1$	$2^{64}-1$



# SGBD MySQL: Tipos de datos

## Ejemplos con tipos de datos en MySQL

### Fechas:

- fecha DATE `/* '1000-01-01' a '9999-12-31' */`
- fecha\_hora DATETIME `/* 'YYYY-MM-DD hh:mm:ss' */`
- fecha\_hora TIMESTAMP `/* '1970-01-01 00:00:01' a '2038-01-19 03:14:07' */`
- Hora TIME `/* ' hh:mm:ss' */`

### Valores booleanos:

- activo BOOL `/* en MySQL no existe aunque se permite -> TINYINT*/`
- activo BOOLEAN `/* en MySQL no existe aunque se permite -> TINYINT */`
- activo TINYINT(1) `/* 0 falso, <>0 verdadero */`

### Valores enumerados y conjuntos:

- nota ENUM ('suspenseo', 'aprobado', 'notable', 'sobresaliente')  
`/* cadena de texto elegida de una lista de valores permitidos*/`
- Idioma SET ('italiano', 'español', 'francés', 'alemán')  
`/* puede almacenar más de un valor en el campo */`


# SGBD MySQL: Creación de tablas

## Creación de tablas (versión simplificada)

Opción 1: Creación de tablas: claves y restricciones en columnas.

```
CREATE TABLE [IF NOT EXISTS] nombre_tabla (  
col1 type1 PRIMARY KEY,  
col2 type2 CHECK (col2 BETWEEN 18 AND 72),  
col3 type3 NOT NULL,  
col4 type4 REFERENCES nom_tab2(colX) ON DELETE CASCADE,  
col5 type5 UNIQUE  
);
```

No soportado en  
MySQL aunque no  
dé error



Opción 2: Creación de tablas: claves y restricciones a nivel de tabla.

```
CREATE TABLE [IF NOT EXISTS] nombre_tabla (  
col1 type1,  
col2 type2,  
col3 type3 NOT NULL,  
col4 type4,  
CONSTRAINT pk_col1 PRIMARY KEY (col1),  
CONSTRAINT fk_col4 FOREIGN KEY (col4) REFERENCES nom_tab2(colX)  
ON DELETE CASCADE,  
CONSTRAINT cv_col2 CHECK(col2 BETWEEN 18 AND 72),  
CONSTRAINT uv_codigo UNIQUE(col5)  
);
```

Nombre explícito  
de las  
restricciones

# SGBD MySQL: Creación de tablas

## Creación de tablas (versión simplificada)

Opción 3: Creación de tablas: claves y restricciones a nivel de tabla.

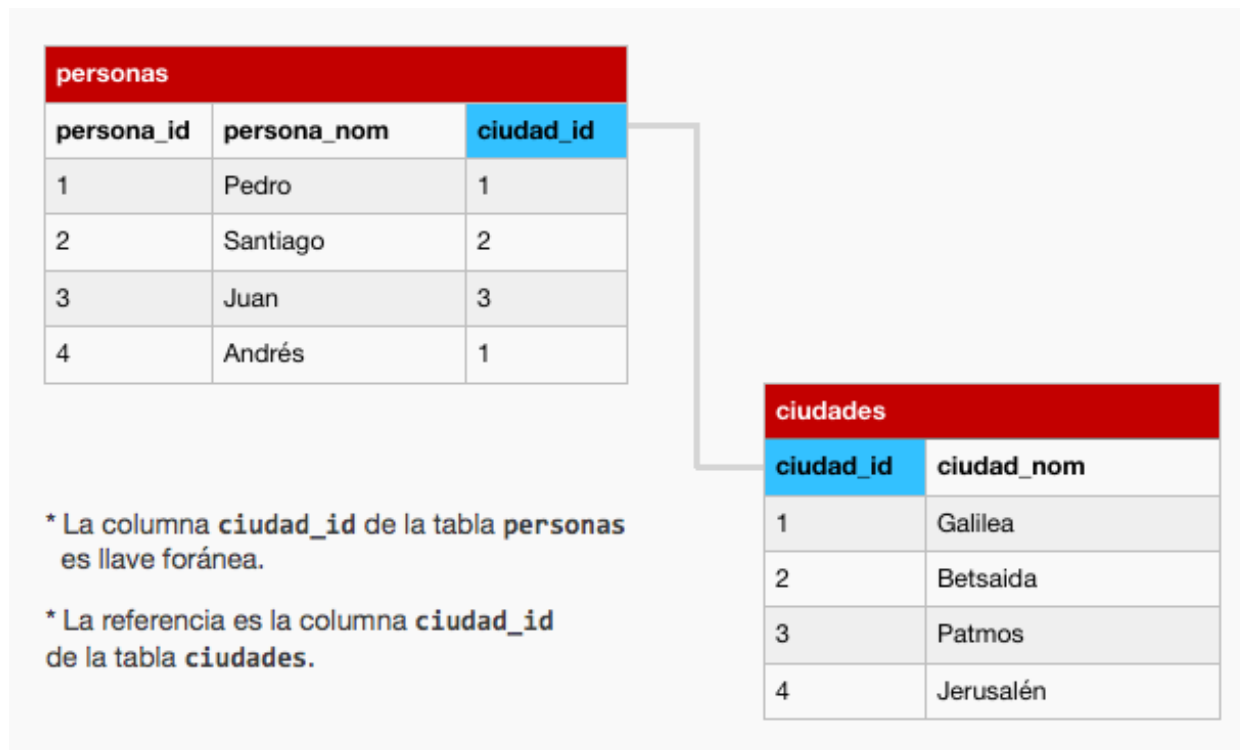
```
CREATE TABLE [IF NOT EXISTS] nombre_tabla (  
  col1 type1,  
  col2 type2,  
  col3 type3 NOT NULL,  
  col4 type4,  
  PRIMARY KEY (col1),  
  FOREIGN KEY (col4) REFERENCES nom_tab2(colX) ON DELETE CASCADE,  
  CHECK(col2 BETWEEN 18 AND 72),  
  UNIQUE(col5)  
);
```

El nombre de las  
restricciones lo  
fija el SGBD

# SGBD MySQL: Restricciones de borrado

## Creación de tablas: claves y restricciones de borrado

**Caso de partida:** definimos *ciudad\_id* en la tabla *personas* como clave foránea que referencia a la columna *ciudad\_id* en la tabla *ciudades*.

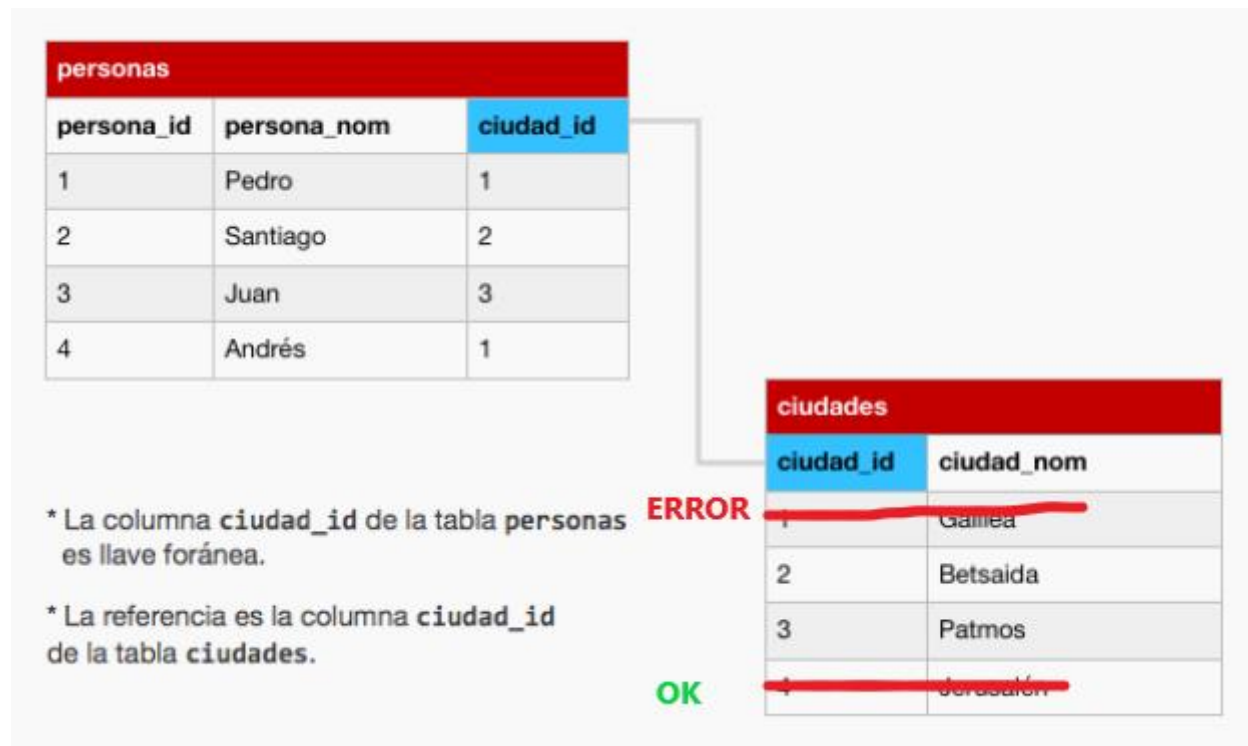


**CONSTRAINT** fk\_ciudad\_id **FOREIGN KEY** (*ciudad\_id*) **REFERENCES** ciudades (*ciudad\_id*)

# SGBD MySQL: Restricciones de borrado

## Creación de tablas: claves y restricciones de borrado

ON DELETE RESTRICT | ON DELETE NO ACTION | *{equivale a no poner nada}*



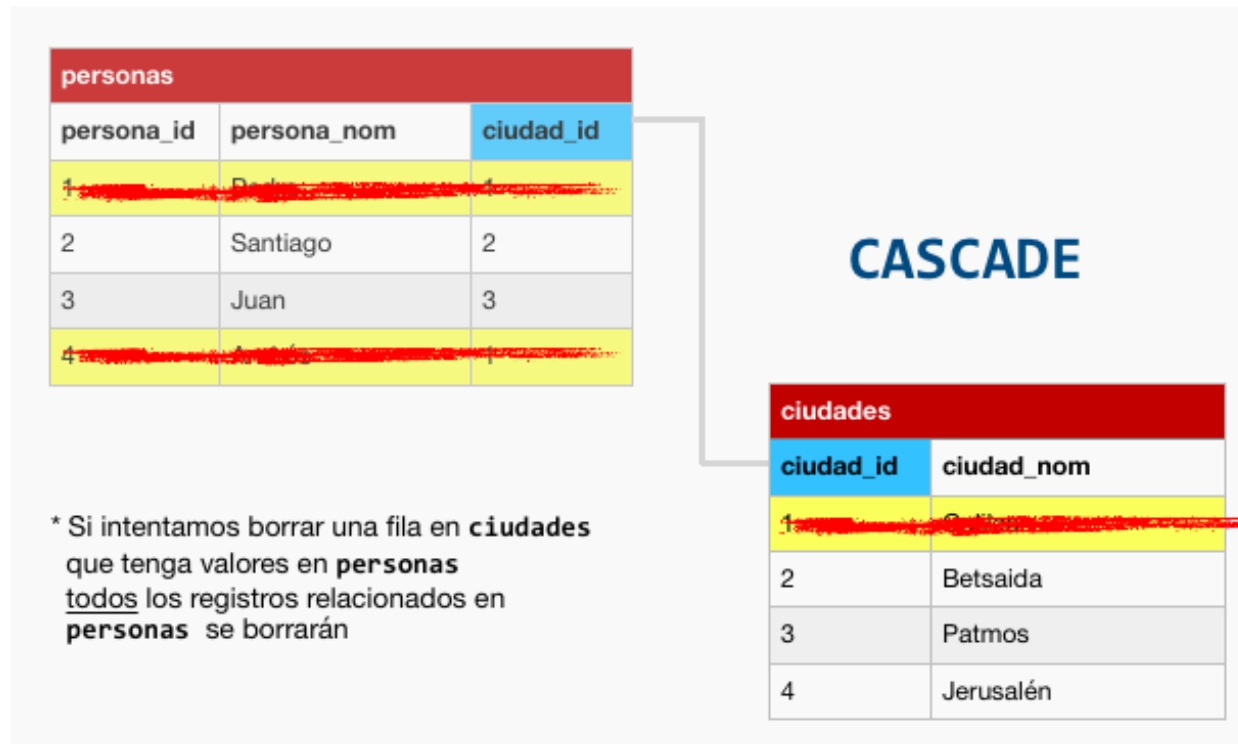
Si tratamos de borrar la ciudad de Galilea, nos dará error, por estar referenciada en otra tabla como clave foránea. Si tratáramos de borrar una no referenciada, sí nos dejaría.

```
CONSTRAINT fk_ciudad_id FOREIGN KEY (ciudad_id) REFERENCES ciudades (ciudad_id) ON DELETE RESTRICT;  
CONSTRAINT fk_ciudad_id FOREIGN KEY (ciudad_id) REFERENCES ciudades (ciudad_id) ON DELETE NO ACTION;  
CONSTRAINT fk_ciudad_id FOREIGN KEY (ciudad_id) REFERENCES ciudades (ciudad_id);
```

# SGBD MySQL: Restricciones de borrado

## Creación de tablas: claves y restricciones de borrado

### ON DELETE CASCADE



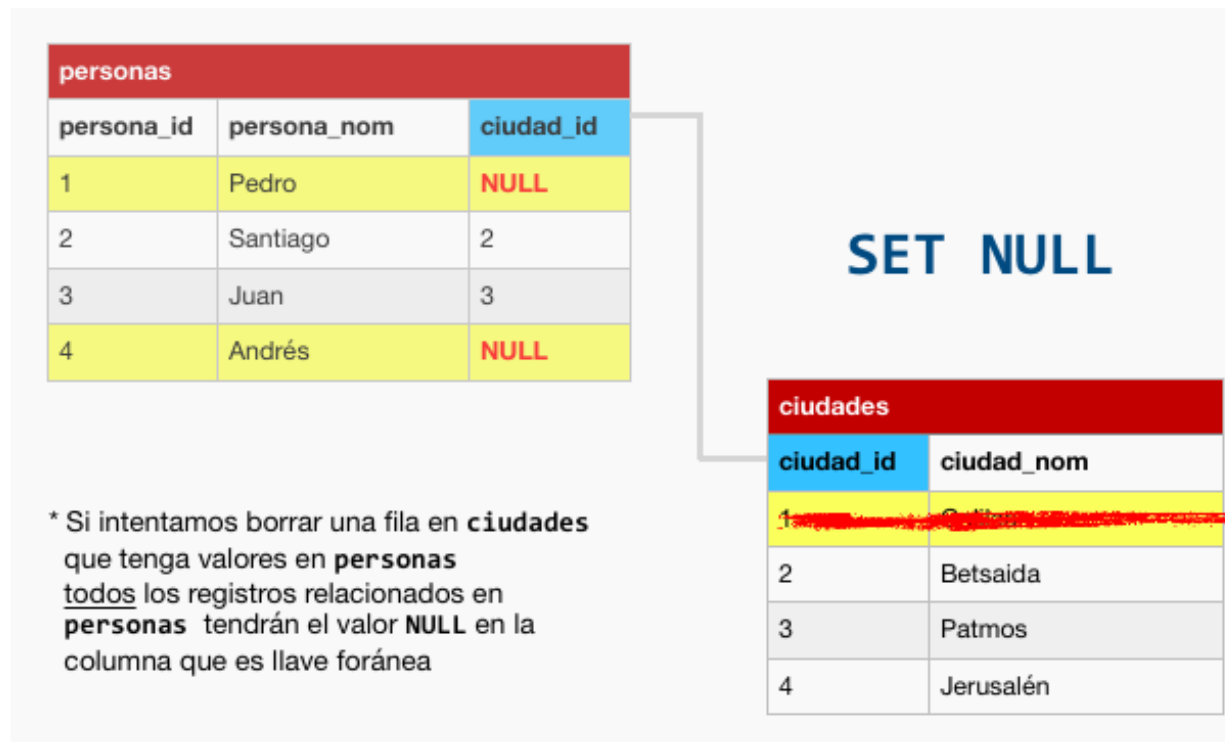
Si tratamos de borrar la ciudad de Galilea, borrará ese registro y todas las referencias en otras tablas. En este caso, las personas nacidas en Galilea.

```
CONSTRAINT fk_ciudad_id FOREIGN KEY (ciudad_id) REFERENCES ciudades (ciudad_id) ON DELETE CASCADE;
```

# SGBD MySQL: Restricciones de borrado

## Creación de tablas: claves y restricciones de borrado

### ON DELETE SET NULL



Si tratamos de borrar la ciudad de Galilea, pondrá a NULL todas las referencias en otras tablas. En este caso, las personas nacidas en Galilea pasarán a tener valor NULL en esa columna.

```
CONSTRAINT fk_ciudad_id FOREIGN KEY (ciudad_id) REFERENCES ciudades (ciudad_id) ON DELETE SET NULL;
```



# SGBD MySQL: Restricciones de actualización

## Creación de tablas: claves y restricciones de actualización

ON UPDATE RESTRICT | ON UPDATE NO ACTION | {*equivale a no poner nada*}

personas		
persona_id	persona_nom	ciudad_id
1	Pedro	1
2	Santiago	2
3	Juan	3
4	Andrés	1

\* La columna `ciudad_id` de la tabla `personas` es llave foránea.

\* La referencia es la columna `ciudad_id` de la tabla `ciudades`.

ciudades	
ciudad_id	ciudad_nom
1	Galilea
2	Betsaida
3	Patmos
4	Jerusalén

error → 6

OK → 5

Si tratamos de actualizar la ciudad de Jerusalén, nos dará error, por estar referenciada en otra tabla como clave foránea. Si tratáramos de actualizar una no referenciada, sí nos dejaría.

```
CONSTRAINT fk_ciudad_id FOREIGN KEY (ciudad_id) REFERENCES ciudades (ciudad_id) ON UPDATE RESTRICT ;  
CONSTRAINT fk_ciudad_id FOREIGN KEY (ciudad_id) REFERENCES ciudades (ciudad_id) ON UPDATE NO ACTION;  
CONSTRAINT fk_ciudad_id FOREIGN KEY (ciudad_id) REFERENCES ciudades (ciudad_id);
```

# SGBD MySQL: Restricciones de actualización

## Creación de tablas: claves y restricciones de actualización

### ON UPDATE CASCADE

personas		
persona_id	persona_nom	ciudad_id
1	Pedro	1 → 5
2	Santiago	2
3	Juan	3
4	Andrés	1 → 5

\* La columna `ciudad_id` de la tabla `personas` es llave foránea. **OK**

\* La referencia es la columna `ciudad_id` de la tabla `ciudades`.

ciudades	
ciudad_id	ciudad_nom
1 → 5	Galilea
2	Betsaida
3	Patmos
4	Jerusalén

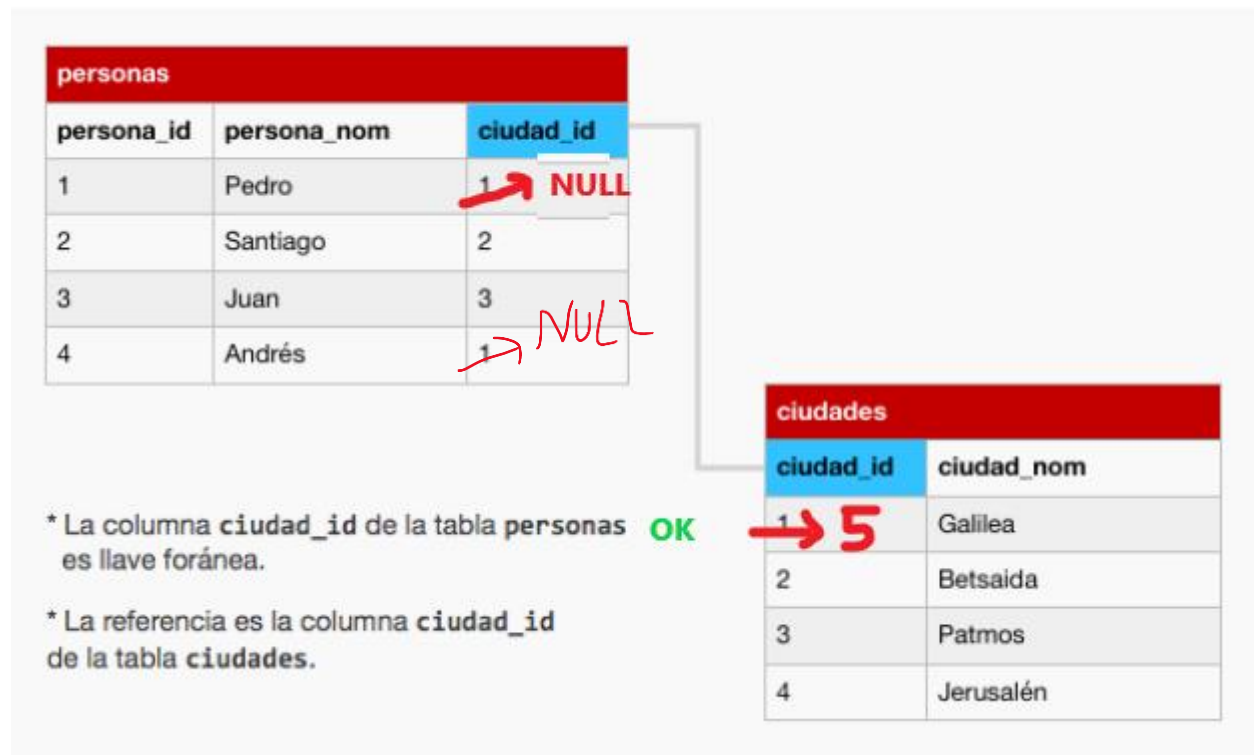
Si tratamos de actualizar la ciudad de Galilea, actualizará de forma automática las referencias que tenga, por haber indicado la cláusula **ON UPDATE CASCADE**.

```
CONSTRAINT fk_ciudad_id FOREIGN KEY (ciudad_id) REFERENCES ciudades (ciudad_id) ON UPDATE CASCADE ;
```

# SGBD MySQL: Restricciones de actualización

## Creación de tablas: claves y restricciones de actualización

### ON UPDATE SET NULL



Si tratamos de actualizar la ciudad de Galilea, actualizará de forma automática las referencias que tenga fijando su valor a NULL, por haber indicado la cláusula **ON UPDATE SET NULL**.

```
CONSTRAINT fk_ciudad_id FOREIGN KEY (ciudad_id) REFERENCES ciudades (ciudad_id) ON UPDATE SET NULL ;
```



# SGBD MySQL: Borrado de tablas

## Borrado de tablas

```
DROP TABLE nombre_tabla;
```

```
DROP TABLE IF EXISTS nombre_tabla;
```

- **IF EXISTS:** Clausula que evita error durante el borrado de una tabla que no exista.
- **¡OJO!** No se borran los permisos asociados a la tabla. Si posteriormente se crea una tabla con ese mismo nombre, los permisos previos no borrados, seguirán en vigor.

## Borrado del contenido de una tabla

```
DELETE FROM nombre_tabla;
```

```
TRUNCATE TABLE nombre_tabla;      /* Borra y recrea la tabla */
```

# SGBD MySQL: Modificación de tablas

## Añadir o eliminar restricciones de integridad de datos

Eliminar o añadir restricciones de integridad de datos:

```
ALTER TABLE nombre_tabla ADD CONSTRAINT fk_name  
FOREIGN KEY (col) REFERENCES nombre_tabla2 (col);
```

```
ALTER TABLE nombre_tabla DROP CONSTRAINT fk_name;
```

Añadir, renombrar, modificar o borrar una columna:

```
ALTER TABLE nombre_tabla ADD nombre_columna tipo_dato;
```

```
ALTER TABLE nombre_tabla RENAME COLUMN nombre_columna_original  
TO nombre_columna_nueva;
```

```
ALTER TABLE nombre_tabla MODIFY nombre_columna tipo_dato_nuevo;
```

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

Modificar restricciones a nivel de columna (default, null)

```
ALTER TABLE nombre_tabla ALTER nombre_columna SET DEFAULT "N/A";
```

```
ALTER TABLE nombre_tabla ALTER nombre_columna DROP DEFAULT;
```

```
ALTER TABLE nombre_tabla MODIFY nombre_columna tipo_dato_nuevo NOT NULL;
```

# SGBD MySQL: Comentarios en SQL

- Los comentarios nos sirven para describir el código, y así que sea más fácil de entender.
- También usamos comentarios para ignorar una cierta parte del código mientras analizamos las consultas SQL (no permita que se ejecute esa parte del código).
  - Desde un carácter '`#`' hasta el fin de la línea.
  - Desde una secuencia '`--`' hasta el final de la línea.
  - Desde una secuencia '`/*`' hasta la próxima secuencia '`*/`'.

`/*`

La siguiente consulta SQL creará una tabla llamada 'alumnos' donde se almacenará la información básica de los estudiantes. Esta información incluye nombre completo, expediente, email y año de matricula

`*/`

`-- Esta columna me permitirá guardar el Código de curso`

`# La siguiente consulta SQL nos devolverá todos los alumnos matriculados este curso`

# SGBD MySQL: Metadatos

INFORMATION\_SCHEMA es la base de datos de información (metadatos), que almacena información acerca de todas las bases de datos que mantiene el servidor MySQL. Solo es de lectura (no puedes insertar, actualizar o borrar su contenido).

Tablas creadas:

```
SELECT table_name FROM information_schema.tables;
```

Columnas referenciadas (FK):

```
SELECT table_name, column_name,  
referenced_table_name, referenced_column_name  
FROM information_schema.key_column_usage;
```

Restricciones (FK):

```
SELECT *  
FROM information_schema.table_constraints  
WHERE constraint_type="FOREIGN KEY";
```

Privilegios y permisos:

```
SELECT * FROM mysql.user;
```

Operaciones en cascada (FK):

```
SELECT constraint_name,  
unique_constraint_name,  
update_rule,  
delete_rule,  
table_name,  
referenced_table_name  
FROM information_schema.  
referential_constraints;
```