

## Contenido

1	NORMALIZACIÓN DE BASES DE DATOS.....	2
2	SENTENCIAS DE MYSQL .....	5
2.1	Crear Base de Datos .....	5
2.2	Usar la Base de Datos .....	5
2.3	Para eliminar la base de Datos .....	5
2.4	Creación de tablas .....	5
2.5	Ver las tablas de un B.D.....	6
2.6	Ver las columnas de una tabla:.....	7
2.7	Ver la instrucción (query) de cómo se creó una tabla:.....	7
2.8	Renombrar una tabla.....	7
2.9	Agregar columnas a una tabla .....	7
2.10	Eliminar una columna de una tabla.....	8
2.11	Cambiar el nombre (renombrar) una columna de una tabla .....	8
2.12	Eliminar una tabla.....	8
3	RELACIÓN ENTRE TABLAS .....	9
3.1	Uno a uno (1:1).....	9
3.2	Uno a muchos (1:N).....	9
3.3	Muchos a muchos (1:N).....	10
3.4	Ejercicio .....	11
3.4.1	Solución .....	12
4	CREACIÓN DE RELACIONES O CLAVES FORÁNEAS Y RESTRICCIONES.....	15
4.1	RESTRICCIONES.....	16
4.1.1	Restric .....	16
4.1.2	Cascade.....	17
4.1.3	Set Null.....	17
4.1.4	No Action .....	17
4.2	Borrar relaciones o clave foránea.....	20
5	MANEJO DE DATOS.....	20
5.1	Inserción de datos .....	20
5.2	Visualizar los datos de una tabla .....	21

# 1 NORMALIZACIÓN DE BASES DE DATOS

La normalización es el proceso de simplificación de datos con el objetivo de:

- ✓ Reducir el espacio de almacenamiento.
- ✓ Eliminar datos repetidos
- ✓ Eliminar errores lógicos
- ✓ Tener los datos ordenados
- ✓ Optimizar la base de datos y tener solo los datos necesarios y relacionados entre si.

Hay varios niveles para la normalización:

- ✓ Primera forma Normal
- ✓ Segunda forma Normal
- ✓ Tercera forma Normal
- ✓ Forma Normal de Boyce Codd
- ✓ cuarta forma Normal
- ✓ Quinta forma Normal

Esta simplificación se debe realizar sin que haya pérdida de información. Por ejemplo:

## Primera forma normal:

Matrícula	Nombre	Dirección	Teléfono	Materia	Num Materia	Carrera
1	Sergio	Puebla 22	56565656	Base de datos	123	Sistemas
1	Sergio	Puebla 22	56565656	Programación web	234	Sistemas
1	Sergio	Puebla 22	56565656	Programación visual	234	Sistemas
2	Ana	Reforma 1	23232323	Base de datos	123	Sistemas

La primera forma normal son los datos iniciales que nos dan, en el ejemplo están los datos del alumno, la matrícula, los datos de la carrera, las materias matriculadas al igual que su nombre, dirección, y teléfono. Al alumno se le está identificando un número de matrícula. Si observamos hay redundancia de datos, pues hay datos repetidos: El Nombre se repite 3 veces, la dirección 3 veces, el teléfono 3 veces, materia 2 veces, número de materia 2 veces, carrera 4 veces. La primera forma normal se trata de identificar los datos que

se repiten. Luego mirar cuales son los campos que nos están generando la redundancia, en este caso es la materia y el número de materia. Por lo tanto estos datos se deben separar en dos tablas de la siguiente forma:

### Alumno

Matrícula	Nombre	Dirección	Teléfono	Carrera
1	Sergio	Puebla 22	56565656	Sistemas
2	Ana	Reforma 1	23232323	Sistemas

### Alumno Materia

Matrícula	Materia	Num Materia
1	Base de datos	123
1	Programación web	234
1	Programación visual	234
2	Base de datos	123

Segunda forma normal

Alumno

Matrícula	Nombre	Dirección	Teléfono	Carrera
1	Sergio	Puebla 22	56565656	Sistemas
2	Ana	Reforma 1	23232323	Sistemas

Matri\_Mater

Matrícula	Num Materia
1	123
1	234
1	234
2	123

Materia

Materia	Num Materia
Base de datos	123
Programación web	234
Programación visual	234
Base de datos	123

Tercera forma normal

Alumno

Matrícula	Nombre	Dirección	Teléfono	No Carrera
1	Sergio	Puebla 22	56565656	1234
2	Ana	Reforma 1	23232323	1234

Carrera

No Carrera	Carrera
1234	Sistemas
6789	Mecatrónica

Matri\_Mater

Matrícula	Num Materia
1	123
1	234
1	234
2	123


Materia

Materia	Num Materia
Base de datos	123
Programación web	234
Programación visual	234
Base de datos	123

## 2 SENTENCIAS DE MYSQL

### 2.1 Crear Base de Datos

Se puede escribir la instrucción tanto en mayúscula como en minúscula, para este caso se escriben en mayúsculas para que se diferencie entre la sintaxis y los demás parámetros

**CREATE DATABASE colegio ;**  
  
Sentencia      Nombre de la BD

Para que la BD reconozca el idioma del país, en este caso el español, en el momento de crear la BD se le deben agregar los parámetros siguientes

**CREATE DATABASE colegio CHARACTER SET utf8 COLLATE utf8\_spanish\_ci;**

### 2.2 Usar la Base de Datos

Para trabajar con la base de datos se debe primero ingresar a la misma

**USE colegio;**

### 2.3 Para eliminar la base de Datos

Se da la instrucción y el nombre de la BD

**DROP DATABASE colegio;**

### 2.4 Creación de tablas

Para ello debemos estar dentro de la B.D a la que se le crearán las tablas, luego se abre paréntesis y dentro de este se escriben las columnas de la tabla y se definen otros parámetros como tipo de datos, clave primaria, etc

**CREATE TABLE** nombre\_de\_la\_tabla;

**CREATE TABLE** productos(

-> idProducto **int auto\_increment primary Key,**

-> nombre **varchar(50) not null,**

-> precio **decimal(10,2) not null,**

-> fechaVencimiento **date**

-> );

**Int:** Va a contener números enteros

**Auto\_increment** incrementará de 1 en 1 automáticamente

**primary Key** se asigna como clave primaria

**varchar ( )** Contiene texto de 50 caracteres

**decimal(10,2)** va a contener un número con decimales de longitud 10 y 2 decimales

**not null** No puede estar vacía

**date** Tipo fecha

## 2.5 Ver las tablas de un B.D

**SHOW TABLES;**

```
+-----+
| Tables_in_colegio |
+-----+
| productos          |
+-----+
```

## 2.6 Ver las columnas de una tabla:

**SHOW COLUMNS FROM productos;**

Field	Type	Null	Key	Default	Extra
idProducto	int(11)	NO	PRI	NULL	auto_increment
nombre	varchar(50)	NO		NULL	
precio	decimal(10,2)	NO		NULL	
fechaVencimiento	date	YES		NULL	

## 2.7 Ver la instrucción (query) de cómo se creó una tabla:

**SHOW CREATE TABLE productos;**

```
productos | CREATE TABLE `productos` (  
  `idProducto` int(11) NOT NULL AUTO_INCREMENT,  
  `nombre` varchar(50) COLLATE utf8_spanish_ci NOT NULL,  
  `precio` decimal(10,2) NOT NULL,  
  `fechaVencimiento` date DEFAULT NULL,  
  PRIMARY KEY (`idProducto`)  
) ENGINE=MyISAM DEFAULT CHARSET=utf8 COLLATE=utf8_spanish_ci |
```

## 2.8 Renombrar una tabla

**RENAME TABLE productos to product;**

Con esta instrucción se le está cambiando el nombre de la tabla “**productos**” por el nombre “**product**” sin alterarle las columnas.

## 2.9 Agregar columnas a una tabla

A la tabla productos se le agrega la columna existencia de tipo entero y no nula.

**ALTER TABLE product ADD existencias INT NOT NULL;**

```
mysql> SHOW COLUMNS FROM product;
+-----+-----+-----+-----+-----+-----+
| Field          | Type          | Null | Key | Default | Extra          |
+-----+-----+-----+-----+-----+-----+
| idProducto     | int(11)       | NO   | PRI | NULL    | auto_increment |
| nombre         | varchar(50)   | NO   |     | NULL    |                |
| precio         | decimal(10,2) | NO   |     | NULL    |                |
| fechaVencimiento | date         | YES  |     | NULL    |                |
| existencias    | int(11)       | NO   |     | NULL    |                |
+-----+-----+-----+-----+-----+-----+
5 rows in set (0.00 sec)
```

## 2.10 Eliminar una columna de una tabla

Se coloca el nombre de la tabla y de la columna

**ALTER TABLE product DROP COLUMN existencias;**

## 2.11 Cambiar el nombre (renombrar) una columna de una tabla

El ejemplo cambia la columna precio de la tabla product por precios, además por un número de 18 caracteres y dos decimales

**ALTER TABLE product CHANGE precio precios DECIMAL(18,2);**

```
mysql> SHOW COLUMNS FROM product;
+-----+-----+-----+-----+-----+-----+
| Field          | Type          | Null | Key | Default | Extra          |
+-----+-----+-----+-----+-----+-----+
| idProducto     | int(11)       | NO   | PRI | NULL    | auto_increment |
| nombre         | varchar(50)   | NO   |     | NULL    |                |
| precios        | decimal(18,2) | YES  |     | NULL    |                |
| fechaVencimiento | date         | YES  |     | NULL    |                |
+-----+-----+-----+-----+-----+-----+
4 rows in set (0.00 sec)
```

## 2.12 Eliminar una tabla

**DROP TABLE product;**



### 3 RELACIÓN ENTRE TABLAS

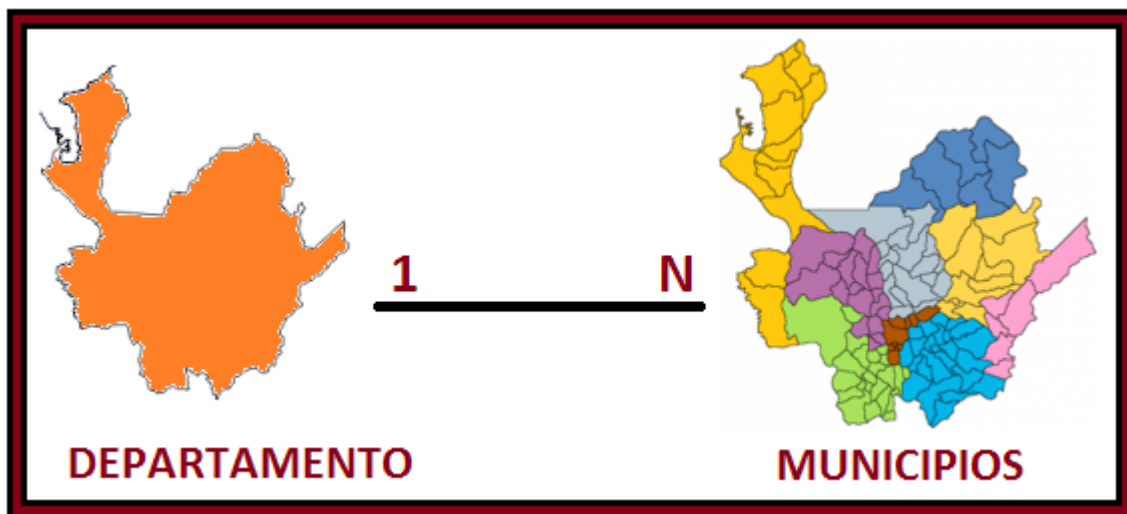
#### 3.1 Uno a uno (1:1)

Cuando un registro de una tabla está relacionado uno y solamente un registro de una segunda tabla, y a su vez este segundo estará relacionado solamente con el primero. Por ejemplo, un producto solo puede tener un código de barras y un código de barras solo puede pertenecer a un producto.



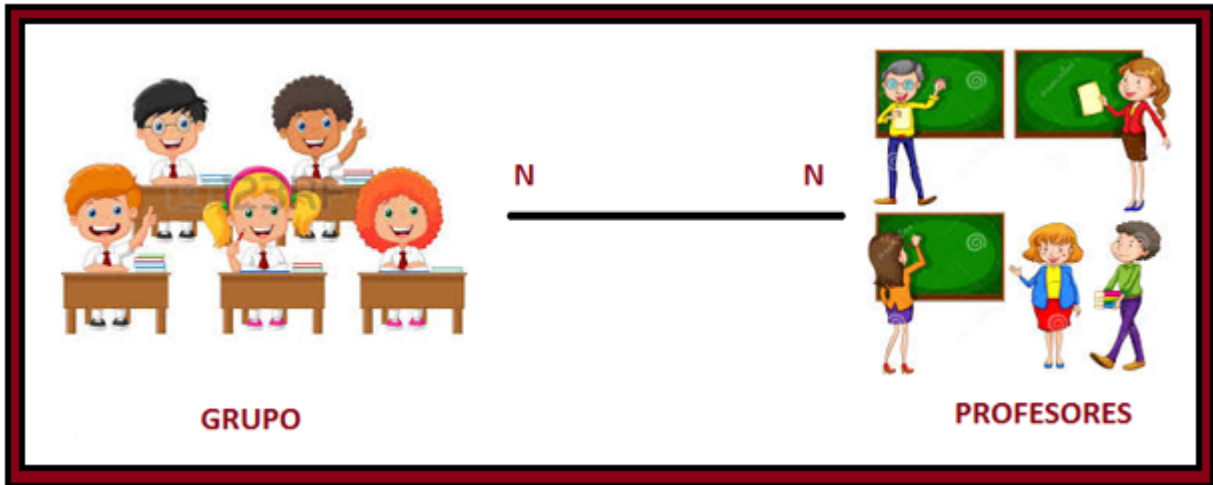
#### 3.2 Uno a muchos (1:N)

Un registro de la tabla uno se relacionará con uno y solo un registro de una segunda tabla, pero el registro de la segunda tabla se relaciona con varios de la primera. Por ejemplo, un municipio pertenece a un solo departamento, pero un departamento tiene muchos municipios.



### 3.3 Muchos a muchos (1:N)

En esta relación cada registro de la tabla uno se puede relacionar con muchos registros de la tabla dos y cada registro de la tabla dos se puede relacionar con muchos registros de la tabla 1. Por ejemplo, un grupo tiene varios profesores y un profesor tiene varios grupos.



Las relaciones más utilizadas son las de uno a muchos y muchos a muchos.

### 3.4 Ejercicio

Hacer una base de Datos con la siguiente información del Tour de Francia

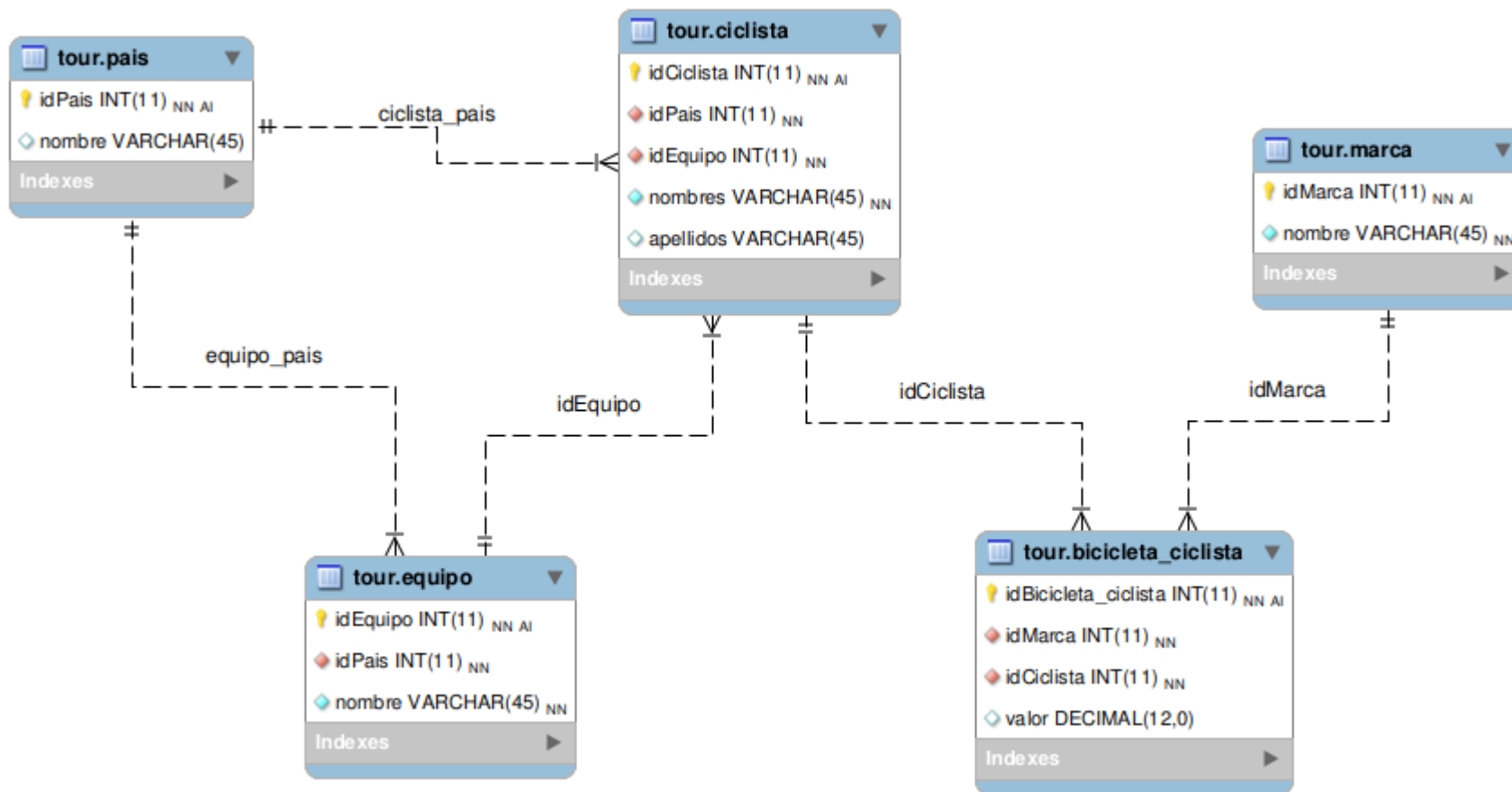
No.	CICLISTA	PAIS DEL CICLISTA	EQUIPO	BICICLETA	COSTO DE LA BICICLETA
1	Chris Froome	Gran Bretaña	Team Sky (Gran Bretaña)	Pinarello	25.000.000
2	Jakob Fuglsang	Dinamarca	Astana (Kazajstán)	Argon 18	27.000.000
3	Nairo Quintana	Colombia	Movistar Team (España)	Canyon	28.000.000
4	Christian Knees	Alemania	Team Sky (Gran Bretaña)	Pinarello	28.500.000
5	Jonathan Castoviejo	España	Movistar Team (España)	Canyon	25.000.000
6	Carlos Alberto Betancur	Colombia	Movistar Team (España)	Canyon	28.000.000
7	Alexis Vuillermoz	Francia	AG2R La Mondiale (Francia)	Factor	27.000.000
8	Dmitriy Gruzdev	(Kazajstán)	Astana (Kazajstán)	Argon 18	26.500.000
9	Rigoberto Urán	Colombia	Cannondale Drapac (Estados Unidos)	Cannondale	26.500.000
10	Taylor Phinney	Estados Unidos	Cannondale Drapac (Estados Unidos)	Cannondale	26.500.000
11	Pierre Rolland	Francia	Cannondale Drapac (Estados Unidos)	Cannondale	26.500.000

Información basada en:

<http://www.elperiodico.com/es/noticias/ciclismo/tour-francia-analisis-equipos-estrellas-guia-6121380>

### 3.4.1 Solución

#### Diagrama entidad relación



## Creación de la base de datos y las tablas

```
mysql> create database tour;
Query OK, 1 row affected (0.00 sec)

mysql> create table pais(
  -> idPais int auto_increment primary key,
  -> nombre varchar(45) not null
  -> );
Query OK, 0 rows affected (0.09 sec)

mysql> create table ciclista(
  -> idCiclista int auto_increment primary key,
  -> idPais int,
  -> idEquipo int,
  -> nombre varchar(45),
  -> apellidos varchar(45)
  -> );
Query OK, 0 rows affected (0.05 sec)
```

```
mysql> create table marca(
  -> idMarca int not null auto_increment primary key,
  -> nombre varchar(45)
  -> );
Query OK, 0 rows affected (0.05 sec)

mysql> create table equipo(
  -> idEquipo int not null primary key,
  -> idPais int,
  -> nombre varchar(45)
  -> );
Query OK, 0 rows affected (0.03 sec)
```

```
mysql> create table bici_ciclista(
  -> idBici_ciclista int not null primary key,
  -> idMarca int,
  -> idCiclista int,
  -> valor decimal(12,0)
  -> );
Query OK, 0 rows affected (0.07 sec)
```

### Visualización de las tablas creadas

```
mysql> show columns from bici_ciclista;
+-----+-----+-----+-----+-----+-----+
| Field          | Type          | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+-----+
| idBici_ciclista | int(11)       | NO   | PRI | NULL    |      |
| idMarca        | int(11)       | YES  |     | NULL    |      |
| idCiclista     | int(11)       | YES  |     | NULL    |      |
| valor          | decimal(12,0) | YES  |     | NULL    |      |
+-----+-----+-----+-----+-----+-----+
4 rows in set (0.07 sec)

mysql> show columns from ciclista;
+-----+-----+-----+-----+-----+-----+
| Field          | Type          | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+-----+
| idCiclista     | int(11)       | NO   | PRI | NULL    | auto_increment |
| idPais         | int(11)       | YES  |     | NULL    |      |
| idEquipo       | int(11)       | YES  |     | NULL    |      |
| nombre         | varchar(45)   | YES  |     | NULL    |      |
| apellidos      | varchar(45)   | YES  |     | NULL    |      |
+-----+-----+-----+-----+-----+-----+
5 rows in set (0.01 sec)

mysql> show columns from pais;
+-----+-----+-----+-----+-----+-----+
| Field | Type          | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+-----+
| idPais | int(11)       | NO   | PRI | NULL    | auto_increment |
| nombre | varchar(45)   | NO   |     | NULL    |      |
+-----+-----+-----+-----+-----+-----+
2 rows in set (0.00 sec)

mysql> show columns from marca;
+-----+-----+-----+-----+-----+-----+
| Field | Type          | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+-----+
| idMarca | int(11)       | NO   | PRI | NULL    | auto_increment |
| nombre  | varchar(45)   | YES  |     | NULL    |      |
+-----+-----+-----+-----+-----+-----+
2 rows in set (0.00 sec)

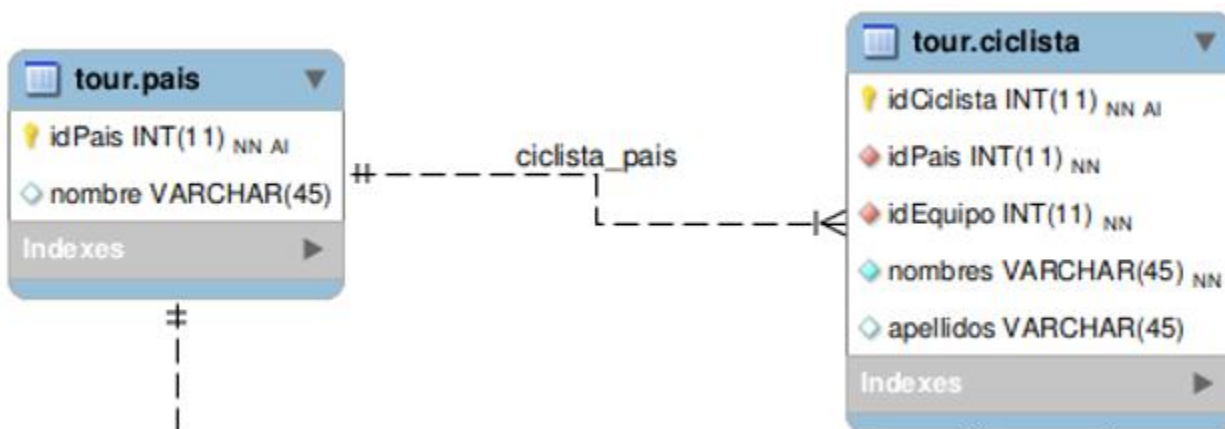
mysql> show columns from equipo;
+-----+-----+-----+-----+-----+-----+
| Field | Type          | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+-----+
| idEquipo | int(11)       | NO   | PRI | NULL    |      |
| idPais   | int(11)       | YES  |     | NULL    |      |
| nombre   | varchar(45)   | YES  |     | NULL    |      |
+-----+-----+-----+-----+-----+-----+
3 rows in set (0.00 sec)
```

```
mysql> show tables;
+-----+
| Tables_in_tour |
+-----+
| bici_ciclista  |
| ciclista       |
| equipo         |
| fabricante     |
| marca          |
| pais           |
+-----+
6 rows in set (0.00 sec)
```

## 4 CREACIÓN DE RELACIONES O CLAVES FORÁNEAS Y RESTRICCIONES

Cada relación tiene un nombre, ésta por regla debe comenzar con las iniciales **fk (foreign key)**. En la imagen es **ciclista\_pais**, esta se crea con la instrucción:

### **CONSTRAINT fk\_nombre\_realación**



Si la tabla ya ha sido creada previamente se debe modificar con alter table, así:

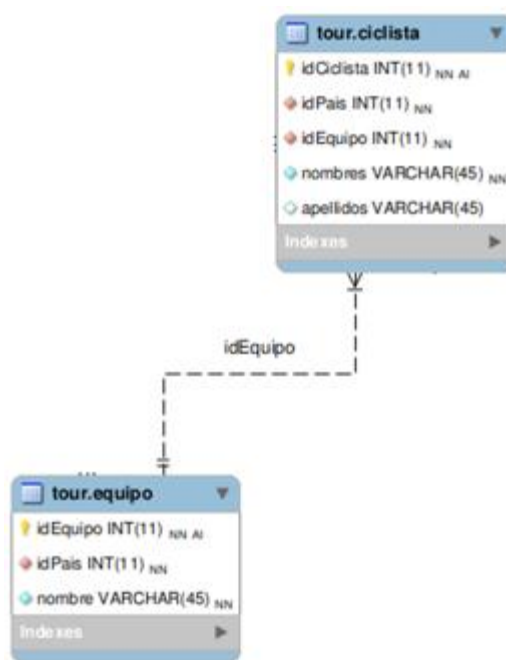
**ALTER TABLE tabla1 ADD CONSTRAINT fk\_nombre\_realación FOREIGN KEY (nombre\_colum\_tabla1) REFERENCES tabla2 (nombre\_colum\_tabla2);**

```
mysql> alter table ciclista
-> add constraint fk_ciclista_pais
-> foreign key (idPais)
-> references pais (idPais);
Query OK, 0 rows affected (0.09 sec)
Records: 0 Duplicates: 0 Warnings: 0
```

Ver las características de creación de la tabla ciclista

```
mysql> show create table ciclista;
+-----+
| ciclista | CREATE TABLE `ciclista` (
  `idCiclista` int(11) NOT NULL AUTO_INCREMENT,
  `idPais` int(11) DEFAULT NULL,
  `idEquipo` int(11) DEFAULT NULL,
  `nombre` varchar(45) COLLATE utf16_spanish2_ci DEFAULT NULL,
  `apellidos` varchar(45) COLLATE utf16_spanish2_ci DEFAULT NULL,
  PRIMARY KEY (`idCiclista`),
  KEY `fk_ciclista_pais` (`idPais`)
) ENGINE=MyISAM DEFAULT CHARSET=utf16 COLLATE=utf16_spanish2_ci |
+-----+
```

Creación de relación idEquipo



```
mysql> alter table ciclista
-> add constraint fk_idEquipo
-> foreign key (idEquipo)
-> references equipo (idEquipo);
Query OK, 0 rows affected (0.06 sec)
Records: 0 Duplicates: 0 Warnings: 0
```

## 4.1 RESTRICCIONES

Estas tienen que ver con las actualizaciones o eliminación de datos, que pasaría con una tabla que tiene relación con otra se actualiza o eliminan los campos de la otra tabla.

### 4.1.1 Restric

Comportamiento por defecto cuando se crea una tabla con su clave foránea que impide hacer modificaciones que atenten contra la integridad referencial. Si borramos el departamento **sistemas**, este no se podrá eliminar, ya que la tabla empleados contiene registros asociados con ella. O sea, la tabla padre no se podrá eliminar o modificar si tiene registros hijos.



Departamento	
id	nombre
1	Sistemas
2	Comercio
3	Cientes

Empleados			
id	nombre	edad	idDpto
1	Carlos	25	3
2	Pedro	32	2
3	Sandra	51	2
4	Jimena	38	2
5	Carmen	26	1
6	Marcos	27	1

#### 4.1.2 Cascade

Permite eliminar o actualizar un registro padre, afectando automáticamente todos sus hijos, por ejemplo si elimino el departamento comercio, automáticamente eliminará sus registros asociados en la tabla empleados.

Departamento	
id	nombre
1	Sistemas
2	Comercio
3	Cientes

Empleados			
id	nombre	edad	idDpto
1	Carlos	25	3
2	Pedro	32	2
3	Sandra	51	2
4	Jimena	38	2
5	Carmen	26	1
6	Marcos	27	1

#### 4.1.3 Set Null

Se establece a NULL en valor de la clave secundaria (hijos) cuando se elimina o modifica el registro padre

Departamento	
id	nombre
1	Sistemas
2	Comercio
3	Cientes

Empleados			
id	nombre	edad	idDpto
1	Carlos	25	3
2	Pedro	32	2
3	Sandra	51	2
4	Jimena	38	2
5	NULL	NULL	NULL
6	NULL	NULL	NULL

#### 4.1.4 No Action

Inhabilita el efecto de la restricción, si eliminamos de la tabla departamento a **comercio**, no va a pasar nada con los registros hijos de la tabla asociada empleados, se queda intacta.

Departamento	
id	nombre
1	Sistemas
2	Comercio

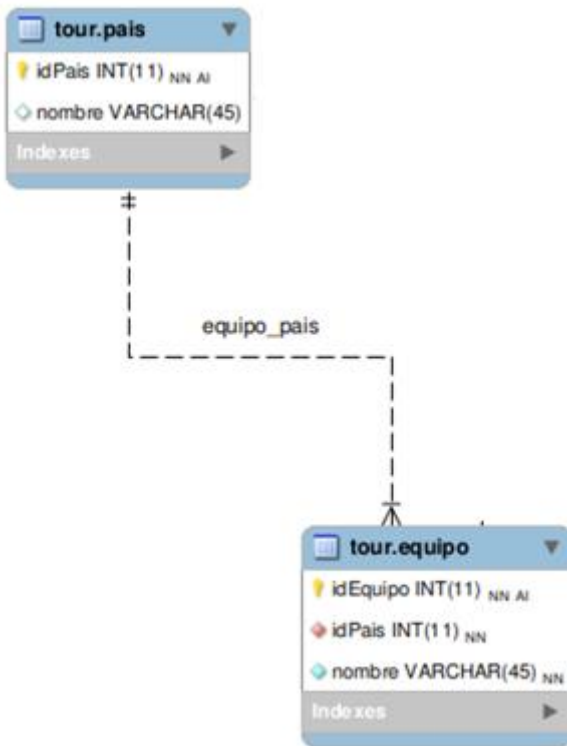
Empleados			
id	nombre	edad	idDpto
1	Carlos	25	3
2	Pedro	32	2

3	Cientes
---	---------

3	Sandra	51	2
4	Jimena	38	2
5	NULL	NULL	NULL
6	NULL	NULL	NULL

Las restricciones más utilizadas son **Restric y cascade**

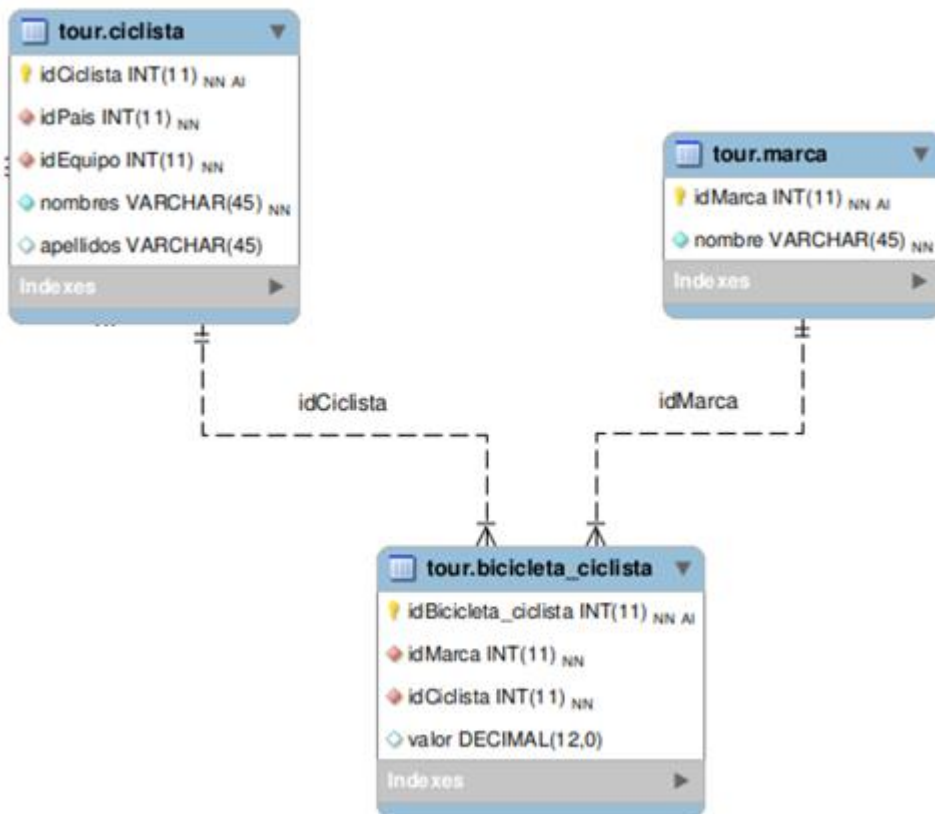
### Creación de la relación equipo\_pais con restricciones



```
mysql> alter table equipo
-> add constraint fk_equipo_pais
-> foreign key (idPais)
-> references pais (idPais)
-> on delete cascade
-> on update cascade;
Query OK, 0 rows affected (0.06 sec)
Records: 0 Duplicates: 0 Warnings: 0
```

```
mysql> show columns from equipo;
+-----+-----+-----+-----+-----+-----+
| Field      | Type      | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+-----+
| idEquipo   | int(11)   | NO   | PRI | NULL    |       |
| idPais     | int(11)   | YES  | MUL | NULL    |       |
| nombre     | varchar(45) | YES  |     | NULL    |       |
+-----+-----+-----+-----+-----+-----+
3 rows in set (0.00 sec)
```

### Creación de la relación idCiclista y la relación idMarca



```
mysql> alter table bici_ciclista
-> add constraint fk_idMarca
-> foreign key (idMarca)
-> references marca (idMarca)
-> on delete cascade
-> on update cascade;
Query OK, 0 rows affected (0.11 sec)
Records: 0 Duplicates: 0 Warnings: 0

mysql> alter table bici_ciclista
-> add constraint fk_idCiclista
-> foreign key (idCiclista)
-> references ciclista (idCiclista)
-> on delete cascade
-> on update cascade;
Query OK, 0 rows affected (0.07 sec)
Records: 0 Duplicates: 0 Warnings: 0
```

## 4.2 Borrar relaciones o clave foránea

**ALTER TABLE nombre\_de\_la\_tabla DROP FOREIGN fk\_nombre\_realación;**

El nombre de la tabla hace referencia a la tabla que posee la clave foránea. Por ejemplo borrar la relación `fk_ciclista_pais`:

**ALTER TABLE ciclista DROP FOREIGN KEY fk\_ciclista\_pais;**

## 5 MANEJO DE DATOS

### 5.1 Inserción de datos

**INSERT INTO tabla VALUES (dato1, dato2, ..., daton);**

Los datos tipo **texto y fecha** deben ir en comillas **doble o simples**, y si es un dato autoincremento se le coloca Null en vez del dato. Por ejemplo, insertar los datos en la tabla país:

```
mysql> show columns from pais;
+-----+-----+-----+-----+-----+-----+
| Field | Type          | Null | Key | Default | Extra          |
+-----+-----+-----+-----+-----+-----+
| idPais | int(11)       | NO   | PRI | NULL    | auto_increment |
| nombre | varchar(45)   | NO   |     | NULL    |                |
+-----+-----+-----+-----+-----+-----+
2 rows in set (0.01 sec)

mysql> insert into pais values (null, "Colombia");
Query OK, 1 row affected (0.04 sec)

mysql> insert into pais values (null, "inglaterra");
Query OK, 1 row affected (0.00 sec)

mysql> insert into pais values (null, "España");
Query OK, 1 row affected (0.00 sec)
```

En ocasiones se puede especificar los datos a insertar en la columna de una tabla, así:

**INSERT INTO tabla (columna) VALUES ("dato");**

Se pueden insertar varios datos a la vez de la siguiente forma:

**INSERT INTO tabla (columna) VALUES ("dato\_fila1"), ("dato\_fila2"), ... ("dato\_filaN");**

**INSERT INTO** tabla (columna1, columna2, ..., columnaN) **VALUES** (“dato1”, “dato2”, ... datoN);

## 5.2 Visualizar los datos de una tabla

**SELECT \*FROM** tabla;

**SELECT \*FROM** pais;

```
mysql> select *from pais;
+-----+-----+
| idPais | nombre |
+-----+-----+
|      1 | Colombia |
|      2 | Inglaterra |
|      3 | España |
+-----+-----+
3 rows in set (0.00 sec)
```

<https://www.youtube.com/watch?v=bO18omSzeR4>