



# Diseño Físico de BBDD

BASES DE DATOS

# Creación de una base de datos

- Creación de la base de datos

```
create database test;
```

```
use test;
```

- Creación de una tabla

```
create table alumno (
```

```
    ->nombre varchar(20),
```

```
    ->apellido1 varchar(20),
```

```
    ->apellido2 varchar(20)
```

```
);
```

```
desc alumno;
```



# Campos Clave



- Un campo clave no admite dos registros o tuplas con el mismo valor en este campo. Ejemplo: NIF, N° Seguridad Social, Pasaporte, Referencia catastral, etc. Puede ser el conjunto de 2 o más campos.
- Tipos:
  - Primary Key: Clave principal de la clave
  - Unique Key; Clave alternativa
  - Foreign Key: Sus valores deben existir en otra tabla denominada principal o padre.
- Foreign key update:
  - Acción a realizar si se borra o se actualiza la tabla principal
  - Se puede restringir la modificación o borrado en la tabla principal si el valor está siendo utilizado.

# Incluir un campo clave

- Modificación de la base de datos para incluir un campo nuevo

alter table **alumno** add **nia** varchar(5);

desc **alumno**;

- Indicar que el nuevo campo es clave primaria

alter table **alumno** modify **nia** varchar(5) primary key;

show create table **alumno**;

```
mysql> desc alumno;
```

Field	Type	Null	Key	Default	Extra
nombre	varchar(20)	YES		NULL	
apellido1	varchar(20)	YES		NULL	
apellido2	varchar(20)	YES		NULL	
nia	varchar(5)	NO	PRI	NULL	

```
4 rows in set (0.00 sec)
```

# Incluir un campo clave secundario

- Modificación de la base de datos para incluir un campo nuevo

```
alter table alumno add nif varchar(9);
```

```
desc alumno;
```

- Indicar que el nuevo campo es clave secundaria

```
alter table alumno modify nif varchar(9) unique key;
```

```
show create table alumno;
```

```
mysql> desc alumno;
```

Field	Type	Null	Key	Default	Extra
nombre	varchar(20)	YES		NULL	
apellido1	varchar(20)	YES		NULL	
apellido2	varchar(20)	YES		NULL	
nia	varchar(5)	NO	PRI	NULL	
nif	varchar(9)	YES	UNI	NULL	

```
5 rows in set (0.00 sec)
```



# Restricciones

- Campos obligatorios

`alter table alumno modify nombre varchar(20) not null;`

- Valores de defecto

`alter table alumno modify apellido1 varchar(20) default "";`

- Es muy recomendó utilizar uno de los dos. Se pueden utilizar los dos.



# Clave Ajena

- Creación de una tabla denominada **matrícula** que hace referencia a la tabla alumnos
- Comando SQL:

```
create table matricula (  
    ->alumno varchar(5),  
    ->asignatura varchar (20),  
    ->curso year,  
    ->foreign key (alumno) references alumno(nia)  
    ->);
```

```
show create table matricula;
```





# Indices



- Se utilizan cuando se realizan habitualmente búsquedas sobre un campo, o sea, cuando el campo es usado habitualmente en la cláusula WHERE.
- El comando SELECT busca primero en el índice que le indica qué registros se deben mostrar y donde están en la tabla.
- Se pueden añadir varios índices en la misma tabla.
- Implica que los datos ocupen más en el disco pero aumentan la velocidad en las lecturas. Las escrituras y modificaciones son más lentas.
- Puede haber valores repetidos. NO es una clave.





# Ejemplo de creación de un Índice

- Creación del índice

create index **niaindex** on **alumno** (nia);

- Comprobar los índices de una tabla (también salen las claves)

show index from **alumno**;

show create table **alumno**;

- Borrado de un índice

drop index **niaindex** on **alumno**;

# Inserción de Datos

- Comando SQL INSERT

**insert into** alumno (nombre, apellido1, apellido2, nia, nif) **values** ("Juan", "Perez", "Gonzalez", "15123", "3451287G");

- Se especifican los campos y, en el mismo orden, los valores.

- Si no se especifican los campos se asume, para los valores, el orden en el que están los campos en la definición de la tabla

**insert into** alumno **values** ("Luis", "Gonzalez", "Suarez", "15118", "7651287L");

- Antiguamente era necesario realizar el comando COMMIT. Actualmente las bases de datos tienen un modo, activo por defecto, que se denomina AUTOCOMMIT.

# Inserción de varios registros

- Se pueden insertar varios registros en el mismo comando

**insert into** alumno (nombre, apellido1, apellido2, nia, nif) **values** ("Juan", "Perez", "Gonzalez", "15123", "3451287G"), ("Luis", "Gonzalez", "Perez", "15124", "3352217L");

- En estos casos es mejor preparar un fichero de texto y, a continuación, copiar y pegar.

- Si no se especifican los campos, como en el caso anterior, se asume el orden en el que están los campos en la definición de la tabla

**insert into** alumno **values** ("Luis", "Gonzalez", "Suarez", "15118", "7651287L") ", ("Luis", "Gonzalez", "Perez", "15124", "3352217L");

# Datos tabla Alumnos

Nombre	Apellido1	Apellido2	NIA	NIF
Juan	Alvarez	Enriquez	15214	34112435R
Luis	Marchena	Martinez	15216	23452222G
Pedro	Sanchez	Barcelo	15218	7221569K
Natalia	Gonzalez	Ostos	14893	5331356L
Julian	Gutierrez	Iglesias	15219	6754523J
Ruben	Huerta	Sanchez	14894	18254429K
Maria	Lopez	Galvez	15221	21234123M
Cristina	Estevez	Echevarria	15224	8973452F
Carolina	Montero	Suarez	15234	22657342S

# Datos tabla Matricula

➤ Datos a ingresar

Alumno	Asignatura	Year
15214	Gestión de Bases de Datos	2019
15214	Redes Locales	2019
15216	Gestión de Bases de Datos	2019
15216	Implantación de Sistemas Operativos	2019
15216	Formación y Orientación Laboral	2019

➤ Ingresar 6 asignaturas por cada alumno para el año 2019

# Campo AUTO\_INCREMENT

- Es una característica de un campo como NOT NULL

```
alter table alumno modify nif varchar(5) auto_increment;
```

- No es necesario mencionarlo en el comando INSERT, se asigna un nuevo valor cada vez que se inserta un registro.

```
insert into alumno (nombre, apellido1, apellido2, nif) values ("Alberto",  
"Guillamon", "Alvarez", "34452287L"), ("Fernando", "Contador", "Sanchez",  
"3352327M");
```

# NUEVA TABLA CON DATOS DE OTRA

- El proceso de creación de una tabla con datos de otra tabla se puede realizar en una sola operación.

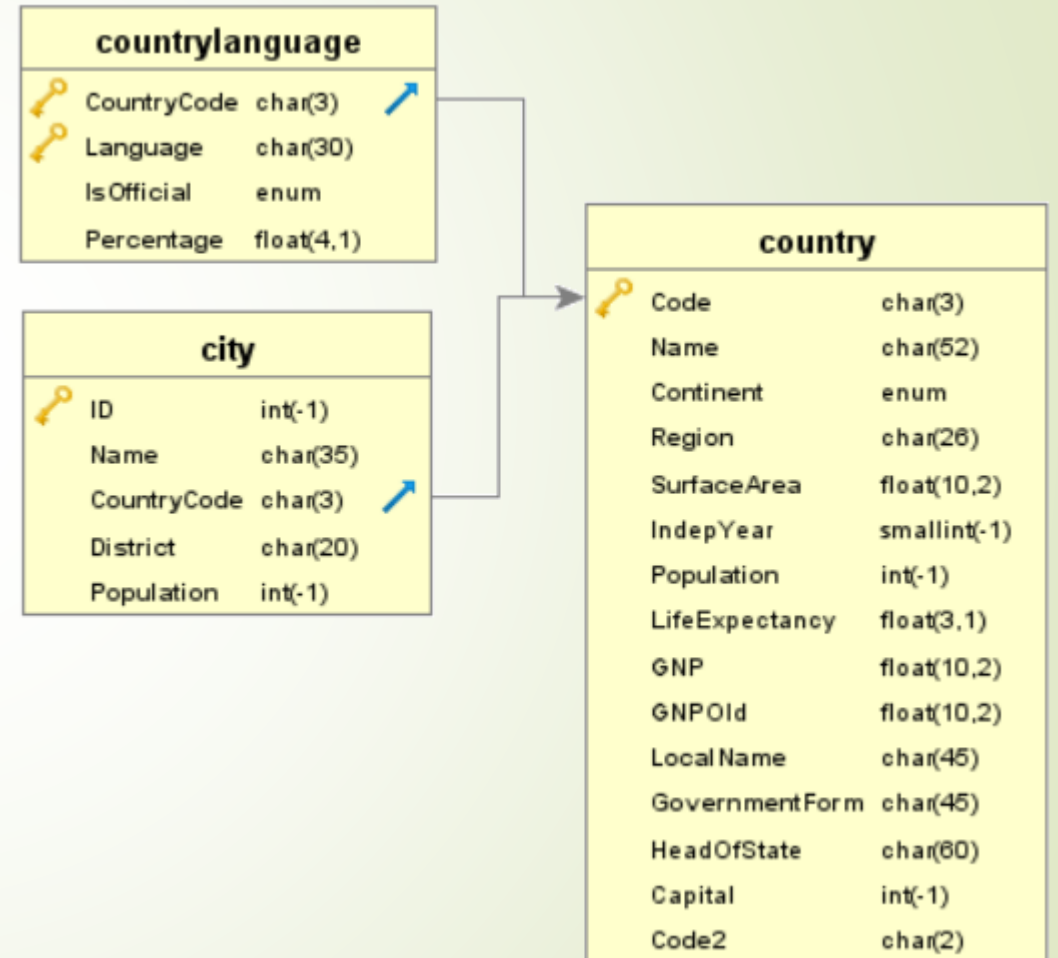
```
create table cityesp as select * from world.city where countrycode = 'ESP';
```

- Crea la tabla con los formatos de los campos de la tabla city.
- Introduce los valores que cumplen la condición
- NO crea primary keys ni índices ni foreign keys.



# Foreign Keys

- El campo **Code** es clave en la tabla **country**
- En la tabla **countrylanguage**, **CountryCode** es foreign key. Sus valores deben estar en el campo **Code** de la tabla **country**
- En la tabla **city**, **CountryCode** es foreign key. Sus valores deben estar en el campo **Code** de la tabla **country**



# Influencia de las Foreign Keys

- Foreign key definida en la tabla city
- Utilizando el comando **show create table** city;  
**constraint** `city\_ibfk\_1` **foreign key** (`CountryCode`) REFERENCES `country` (`Code`)
- El campo "CountryCode" de la tabla "City" solo puede contener valores que existan en el campo "Code" de la table "Country".
- Afectan a los comandos UPDATE, INSERT y DELETE
- No afecta de ninguna manera al comando SELECT

# Valores no permitidos

- Inserción de datos erróneos

```
Insert into city values (4080, 'Reus','KKK','Katalonia',104373);
```

- Modificación de datos erróneos

```
update city set countrycode = 'KKK' WHERE name = 'Madrid';
```

- En los dos casos indica un error: Cannot add or update a child row: a foreign key constraint fails

# Opciones de foreign key

- Por defecto restringe cualquier cambio en el registro referenciado (País) que afecte a los registros de la tabla que tiene la foreign key
- Se puede cambiar independientemente para el borrado y para la actualización
- Ejemplo: Restringir que se pueda borrar pero permitir la actualización

```
alter table city drop constraint city_ibfk_1;  
alter table countrylanguage drop constraint countryLanguage_ibfk_1;  
alter table city add foreign key (CountryCode) references country(Code) on delete  
restrict on update cascade;
```
- No permite CASCADE con dos FK al mismo campo. Probar las distintas alternativas.

# Ejercicio

- Crear una nueva BBDD llamada examen. Crear la tabla lideres.

*	Field	Type	Null	Key	Default	Extra
1	Id	int	NO	PRI	(null)	auto_increment
2	nombre	varchar(20)	YES		(null)	
3	apellidos	varchar(40)	YES		(null)	
4	partId	int	YES		(null)	

- En la BBDD examen crear la tabla partidos con las siguientes características

*	Field	Type	Null	Key	Default	Extra
1	Id	int	NO	PRI	(null)	auto_increment
2	Nombre	varchar(20)	YES		(null)	
3	Siglas	varchar(10)	YES		(null)	



# Ejercicio

- Añadir una clave ajena coherente en la tabla lideres
- Añadir un índice sobre el campo nombre de la tabla lideres
- Modificar la tabla lideres para que los campos nombre y apellido1 de la tabla lideres sean obligatorios
- Modificar la tabla lideres para que, el valor de defecto del campo apellido2, sea un texto vacío
- Modificar la tabla partidos para que los campos nombre y siglas sean obligatorios
- Añadir un índice en el campo nombre de la tabla partidos





GRACIAS