

UT4: LDD EN MYSQL

PRÁCTICA GUIADA 2

ASIR1
DAM 1

Gestión de Bases de Datos
Bases de Datos

UT4: LDD en MySQL Práctica Guiada 2

CONTENIDOS

1. SISTEMA DE INFORMACIÓN	3
1.1 Descripción informal	3
1.2 Diseño de la base de datos.....	3
2. DEFINICIÓN DE LA BASE DE DATOS	5
2.1 Lenguaje De Definición De Datos (L.D.D.).....	5
2.2 Creación de la base de datos: CREATE DATABASE	6
2.2.1 Borrado de la base de datos	6
2.2.2 Creación de una BD sencilla sin parámetros	6
2.2.3 Consulta existencia de la BD previa a su creación	6
2.2.4 Ejemplo 1: Creación de la base de datos BDFormacion.....	6
2.3 Creación de tablas: CREATE TABLE	6
2.3.1 Recordatorio de tipos de datos.....	6
2.3.2 Tipos de restricciones de columna	7
2.3.3 Especificación de restricciones de columna	8
2.4 Práctica de CREATE TABLE	9
2.4.1 Ejemplo 3: Creación Tabla Alumnos	9
2.4.2 Ejemplo 4: Creación Tabla Profesores.....	9
2.4.3 Ejemplo 5: Creación Tabla Cursos	10
2.4.4 Ejemplo 6: Creación Tabla AlumnosCursos.....	10
2.5 Borrado de una tabla: DROP TABLE.....	10
2.6 Modificación de la estructura de una tabla: ALTER TABLE.....	11
2.6.1 Ejemplo 7: Creación/Eliminación de un campo en tabla existente	11
2.6.2 Ejemplo 8: Modificación de un campo	12
2.6.3 Ejemplo 9: Añadir/Eliminar una restricción	12
2.7 EJERCICIO: AÑADIR LA TABLA DEPARTAMENTOS A LA BD	12
2.8 AMPLIACIÓN: Creación/ELIMINACIÓN de índices (CREATE/DROP INDEX)	13
2.8.1 Ejemplo 11: Crear un índice para un campo	13
2.8.2 Ejemplo 12: Crear un índice único para un campo.....	13
2.8.3 Ejemplo 13: Crear un índice único con dos campos.....	14

2.8.4 Ejemplo 14: Borrado de un índice: DROP INDEX.....	14
2.9 AMPLIACIÓN: Consulta de índices	14

Práctica Guiada T-SQL (LDD)

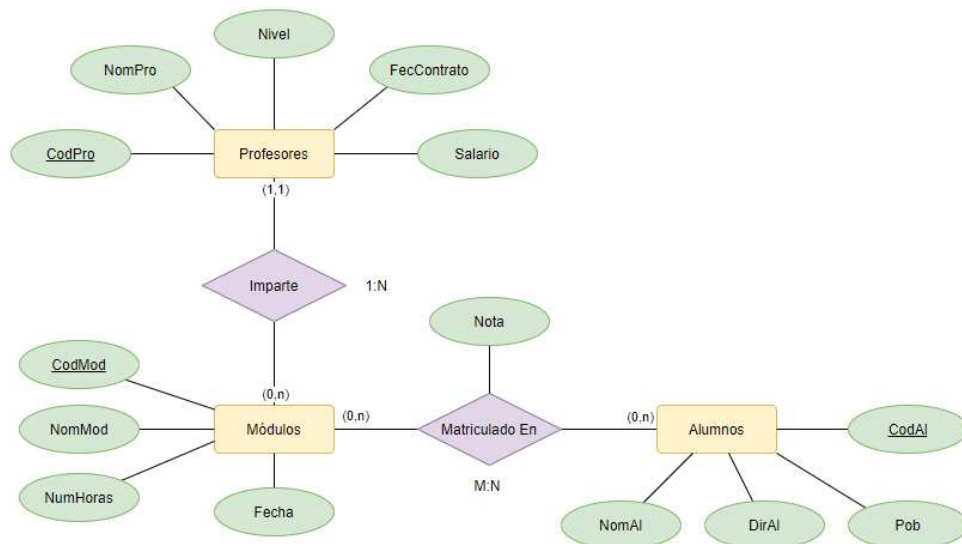
1. SISTEMA DE INFORMACIÓN

1.1 DESCRIPCIÓN INFORMAL

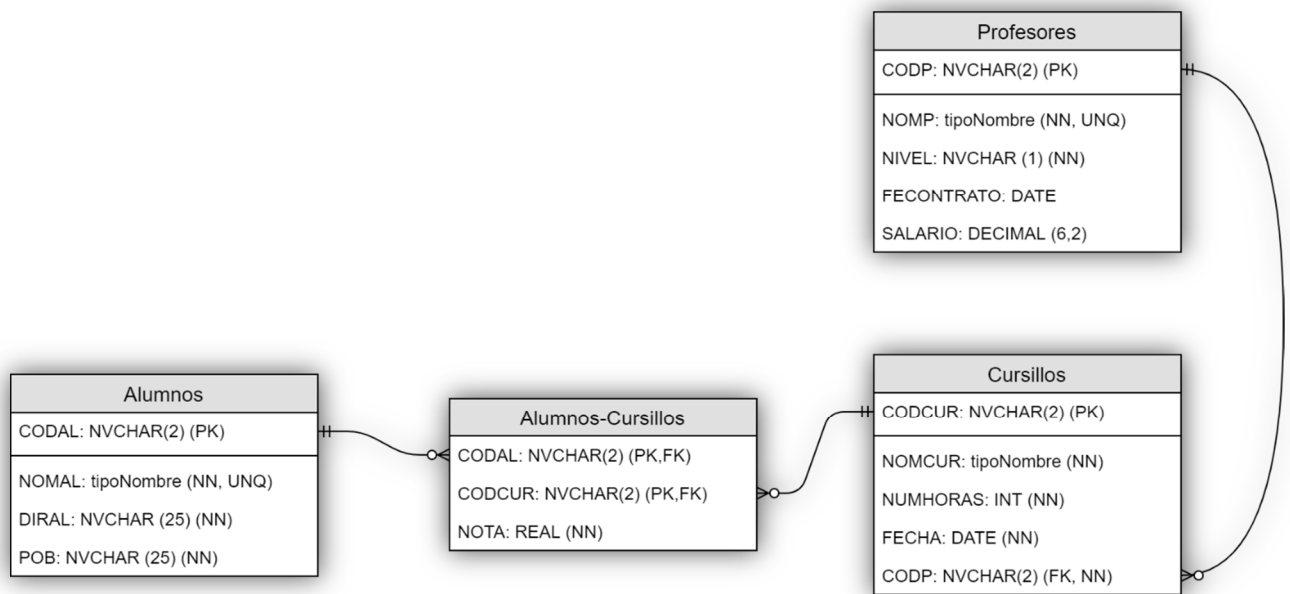
- ✓ Se desea almacenar la información de una empresa de formación que organiza cursillos.
- ✓ La información que es necesario almacenar sobre dichos cursillos es el nombre, número de horas (entre 15 y 350) y fecha de comienzo.
- ✓ Además los cursillos serán impartidos por profesores de los cuales también se desea almacenar el nombre, nivel (que será un valor A, B o C), fecha de contrato, nombre y planta del departamento al que está adscrito.
- ✓ Los alumnos para matricularse en los cursos (puede hacerlo en los que quiera) rellenarán una ficha con el nombre, dirección y población.
- ✓ Un cursillo estará impartido por un único profesor.

1.2 DISEÑO DE LA BASE DE DATOS

ESQUEMA CONCEPTUAL (E/R)



ESQUEMA RELACIONAL



NORMALIZACIÓN

Aplicar reglas de normalización: las tablas están en 3FN.

- ✓ Están en 1FN ya que todos los atributos tienen un único valor en cada fila de datos.
- ✓ Están en 2FN porque están en 1FN y además en cada tabla, cada atributo no principal (que no pertenece a la clave primaria) depende funcionalmente de la clave completa y no de una parte de ella.
 - *Dependencia funcional: Un atributo Y es funcionalmente dependiente de otro X ($X \rightarrow Y$) cuando a cada valor del atributo X le corresponde un único valor del atributo Y.*
- ✓ Están en 3FN porque están en 2FN y además en cada tabla, los atributos que no forman parte de la clave no dependen funcionalmente de otros atributos que no son clave.

DATOS DE LAS TABLAS (NO LOS INCLUYÁIS AÚN)

ALUMNOS

ALUMNO-PC.form...o - dbo.ALUMNOS				
	CODAL	NOMAL	DIRAL	POB
	A1	Antonio Antúnez	C/La Ermita s/n	PAMPLONA
	A2	Ane Aranburu	Avda. Central 5	Avda. Central 5
	A3	Amalia Naya	C/Magdalena 3	PAMPLONA

PROFESORES

ALUMNO-PC.forma...dbo.PROFESORES					
	CODP	NOMP	NIVEL	FECONTRATO	SALARIO
	P1	Patricia Pére	A	2008-10-05	1500,00
	P2	Paula Puerta	B	2010-03-19	1300,00
	P3	Peio Pagoaga	A	2011-10-25	1000,00

CURSILLOS

ALUMNO-PC.forma... - dbo.CURSILLOS					
	CODCUR	NOMCUR	NUMHORAS	FECHA	CODP
	C1	SO	300	2016-04-24	P1
	C2	HW	150	2016-03-21	P3
	C3	HTML	200	2015-11-12	P1
	C4	EIEM	200	2015-12-12	P2
	C5	FOL	200	2015-12-12	P2

ALUMNOSCURSILLOS

ALUMNO-PC.for...MNOSCURSILLOS			
	CODAL	CODCUR	NOTA
	A1	C1	9
	A2	C2	8
	A3	C1	9
	A3	C2	9

2. DEFINICIÓN DE LA BASE DE DATOS

2.1 LENGUAJE DE DEFINICIÓN DE DATOS (L.D.D.)

El lenguaje de definición de datos es el conjunto de instrucciones SQL que nos permiten crear y modificar tablas (campos, tamaños, restricciones de integridad etc.), crear índices, vistas etc.

- ✓ Creación de la base de datos: **CREATE DATABASE**
- ✓ Borrado de la base de datos: **DROP DATABASE**
- ✓ Creación del diseño de una tabla: **CREATE TABLE**
- ✓ Borrado de una tabla: **DROP TABLE**
- ✓ Modificación de la estructura de una tabla: **ALTER TABLE**

Para practicar con el LDD vamos a crear las tablas diseñadas anteriormente, en una base de datos que crearemos primero, de nombre *BDFormacion*.

2.2 CREACIÓN DE LA BASE DE DATOS: CREATE DATABASE

2.2.1 Borrado de la base de datos

```
DROP DATABASE IF EXISTS BDFormacion;
```

2.2.2 Creación de una BD sencilla sin parámetros

```
CREATE DATABASE BDFormacion;
```

2.2.3 Consulta existencia de la BD previa a su creación

```
CREATE DATABASE IF NOT EXISTS BDFormacion;
```

- También podemos preguntar si existe la BD y en ese caso, borrarla y volverla a crear:

```
DROP DATABASE IF EXISTS BDFormacion;
```

```
CREATE DATABASE BDFormacion;
```

2.2.4 Ejemplo 1: Creación de la base de datos BDFormacion

- ✓ Crea la base de datos BDFORMACION con parámetros personalizados y comprobando si existe previamente.
- ✓ En caso de existir una con el mismo nombre, bórrala previamente.

2.3 CREACIÓN DE TABLAS: CREATE TABLE

Se especifica el nombre de la tabla a crear, los campos (columnas) con su tipo y tamaño y las restricciones que tendrán dichas columnas.

2.3.1 Recordatorio de tipos de datos

Algunos de los diferentes tipos de datos que se pueden especificar en MySQL (consultar la documentación), son:

Numéricos	
INTEGER o INT	Número entero
DECIMAL (m, d)	Número decimal donde se especifica la cantidad total de dígitos (m) y la cantidad de dígitos decimales (d).

Fecha y hora

DATE	Fecha con formato "YYYY-MM-DD"
TIME	Hora con formato es "HH:MM:SS"
DATETIME	Fecha y hora, con formato "YYYY-MM-DD HH:MM:SS"
YEAR	Año representado por 4 dígitos ("YYYY")

Lista de valores

ENUM ('valor1', 'valor2,...')	Conjunto de valores definidos
--	-------------------------------

Cadena de caracteres Unicode

VARCHAR (X)	Cadena de tamaño adaptable con un límite de X
--------------------	---

Booleano

BOOLEAN	Solo dos valores posibles: 0 ó 1.
----------------	-----------------------------------

2.3.2 Tipos de restricciones de columna

Las restricciones en MySQL son:

- ✓ **NOT NULL**: indica que la columna no puede contener un valor nulo.
- ✓ **UNIQUE**: sirve para definir un índice único (sin valores duplicados) sobre un campo.
- ✓ **DEFAULT**: sirve para poner un valor por defecto en la columna cuando se inserta una nueva fila (INSERT).
- ✓ **PRIMARY KEY**: define la clave principal de una tabla; esto supone las características de NOT NULL y UNIQUE.
- ✓ **CHECK**: permite especificar una expresión booleana que cumplan los valores introducidos en la columna.
- ✓ **FOREIGN KEY ... REFERENCES**: define un campo como clave ajena que contiene un valor que hace referencia a una clave primaria de una fila de otra tabla, o valor nulo.
 - Opcionalmente podemos especificar el comportamiento en caso de modificación de la clave primaria (ON UPDATE) o borrado (ON DELETE) de las filas de la tabla a la que se referencia:

NO ACTION | CASCADE | SET NULL | SET DEFAULT

ACCIONES PARA MANTENER LA INTEGRIDAD REFERENCIAL

Acción que tiene lugar en las filas de la tabla creada si dichas filas tienen una relación referencial y la fila a la que se hace referencia se elimina/modifica de la tabla primaria. El valor predeterminado es NO ACTION.

- ✓ **NO ACTION:** El Motor de base de datos genera un error y se revierte la acción de eliminación/modificación de la fila de la tabla primaria.
- ✓ **CASCADE:** Si esa fila se elimina (modifica) de la tabla primaria, las filas correspondientes se eliminan/modifican de la tabla de referencia. Ídem.
- ✓ **SET NULL:** Todos los valores que forman la clave externa se establecen en NULL si se elimina/modifica la fila correspondiente de la tabla primaria. Para ejecutar esta restricción, las columnas de clave externa deben admitir valores NULL.

2.3.3 Especificación de restricciones de columna

Podemos especificar las restricciones de tres maneras:

- A continuación de la columna (no sirve para especificar una clave primaria compuesta de más de un campo):


```
USE BDFORMACION;
CREATE TABLE ALUMNOS
(CODAL VARCHAR(2) PRIMARY KEY,
NOMAL VARCHAR(25) NOT NULL UNIQUE,
DIRAL VARCHAR(25) NOT NULL,
POB VARCHAR(25) NOT NULL);
```
- Una vez declaradas las columnas (no sirve para NOT NULL, DEFAULT y la cláusula COLLATE):


```
USE BDFORMACION;
CREATE TABLE ALUMNOS
(CODAL VARCHAR(2),
NOMAL VARCHAR(25) NOT NULL,
DIRAL VARCHAR(25) NOT NULL,
POB VARCHAR(25) NOT NULL,
PRIMARY KEY(CODAL),
UNIQUE(NOMAL));
```
- Con la cláusula CONSTRAINT que permite dar un nombre a la restricción para luego poder referirnos a ella:

```
USE BDFORMACION;

CREATE TABLE ALUMNOS
(CODAL VARCHAR(2),
NOMAL VARCHAR(25) NOT NULL,
DIRAL VARCHAR(25) NOT NULL,
POB VARCHAR(25) NOT NULL DEFAULT ('PAMPLONA'),
CONSTRAINT pk_codal PRIMARY KEY (CODAL),
CONSTRAINT u_nomal UNIQUE (NOMAL)
);
```

Esta opción no es válida para NOT NULL, DEFAULT y la cláusula COLLATE.

Para especificar una clave compuesta de más de un campo es necesario indicar la restricción una vez declaradas las columnas.

2.4 PRÁCTICA DE CREATE TABLE

Crear las tablas ALUMNOS, CURSILLOS, PROFESORES Y ALUMNOSCURSILLOS de la base de datos BDFormacion:

- ✓ Primero se crean las tablas "1" y luego las "N" por la integridad referencial.

2.4.1 Ejemplo 3: Creación Tabla Alumnos

```
USE BDFORMACION;

DROP TABLE IF EXISTS ALUMNOS;
CREATE TABLE ALUMNOS
(CODAL VARCHAR(2) PRIMARY KEY,
NOMAL VARCHAR(25) NOT NULL UNIQUE,
DIRAL VARCHAR(25) NOT NULL,
POB VARCHAR(25) NOT NULL);
```

2.4.2 Ejemplo 4: Creación Tabla Profesores

```
USE BDFORMACION;

DROP TABLE IF EXISTS PROFESORES;
CREATE TABLE PROFESORES
(CODP VARCHAR(2) PRIMARY KEY,
NOMP VARCHAR(25) NOT NULL UNIQUE,
NIVEL VARCHAR(1) NOT NULL,
FECCONTRATO DATE NOT NULL,
SALARIO DECIMAL(10,2) NOT NULL,
CONSTRAINT ck_nivel CHECK (NIVEL = 'A' OR NIVEL = 'B'));
```

También podemos crear la restricción UNIQUE con la cláusula CONSTRAINT:

```
USE BDFORMACION;

DROP TABLE IF EXISTS PROFESORES;
CREATE TABLE PROFESORES
(CODP VARCHAR(2) PRIMARY KEY,
NOMP VARCHAR(25) NOT NULL,
NIVEL VARCHAR(1) NOT NULL,
FECONTRATO DATE NOT NULL,
SALARIO DECIMAL(10,2) NOT NULL,
CONSTRAINT uq_nomp UNIQUE(NOMP),
CONSTRAINT ck_nivel CHECK (NIVEL = 'A' OR NIVEL = 'B')
);
```

2.4.3 Ejemplo 5: Creación Tabla Cursosillos

```
USE BDFORMACION;

DROP TABLE IF EXISTS CURSILLOS;
CREATE TABLE CURSILLOS
(CODCUR VARCHAR(2) PRIMARY KEY,
NOMCUR VARCHAR(25) NOT NULL UNIQUE,
NUMHORAS INT NOT NULL,
FECHA date NOT NULL,
CODP VARCHAR(2),
CONSTRAINT ck_numhoras CHECK (NUMHORAS >=5 AND NUMHORAS <=1000),
CONSTRAINT fk_prof FOREIGN KEY (CODP)
REFERENCES PROFESORES (CODP) ON UPDATE CASCADE);
```

2.4.4 Ejemplo 6: Creación Tabla AlumnosCursosillos

```
USE BDFORMACION;

DROP TABLE IF EXISTS ALUMNOSCURSILLOS;
CREATE TABLE ALUMNOSCURSILLOS
(CODAL VARCHAR(2),
CODCUR VARCHAR(2),
NOTA DECIMAL(4,2) NOT NULL,
CONSTRAINT pk_al_cur PRIMARY KEY(CODAL,CODCUR),
CONSTRAINT fk_al FOREIGN KEY (CODAL)
REFERENCES ALUMNOS (CODAL) ON UPDATE CASCADE,
CONSTRAINT fk_cur FOREIGN KEY (CODCUR)
REFERENCES CURSILLOS (CODCUR) ON UPDATE CASCADE);
```

- ✓ Crear el diagrama de relaciones desde el entorno gráfico (Database -> Reverse Engineer) y comprobar que al agregar las tablas se reflejan las integridades referenciales.

2.5 BORRADO DE UNA TABLA: DROP TABLE

Permite borrar una tabla de la base de datos; la tabla no se podrá eliminar si el borrado infringe las reglas de integridad referencial.

```
DROP TABLE nombre_tabla;
```

Si queremos una comprobación previa para evitar el error podemos usar:

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

En caso de no existir la tabla, en vez de un error tan solo nos mostrará un warning, y el resto de las sentencias siguientes se ejecutarán con normalidad.

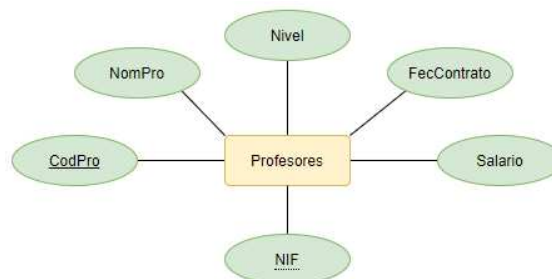
2.6 MODIFICACIÓN DE LA ESTRUCTURA DE UNA TABLA: ALTER TABLE

Qué se puede modificar:

- ✓ Añadir una columna: ADD
- ✓ Borrar una columna: DROP COLUMN
- ✓ Añadir una restricción: ADD CONSTRAINT
- ✓ Borrar una restricción: DROP CONSTRAINT
- ✓ Modificar un campo: ALTER COLUMN

2.6.1 Ejemplo 7: Creación/Eliminación de un campo en tabla existente

Crear un nuevo campo en la tabla PROFESORES que sea el NIF.



```
USE BDFORMACION;
```

```
ALTER TABLE PROFESORES ADD NIF VARCHAR(9) NOT NULL UNIQUE;
```

Resultado:

Column	Type
◇ CODY	varchar(2)
◇ NOMP	varchar(25)
◇ NIVEL	varchar(1)
◇ FECCONTRATO	date
◇ SALARIO	decimal(10,2)
◇ NIF	varchar(9)

Si quisiéramos borrar el campo creado anteriormente:

```
USE BDFORMACION;
```

```
ALTER TABLE PROFESORES DROP COLUMN FECONTRATO;
```

2.6.2 Ejemplo 8: Modificación de un campo

No permitir que la columna DIRAL de la tabla ALUMNOS admita valores nulos:

```
USE BDFORMACION;
```

```
ALTER TABLE ALUMNOS MODIFY COLUMN DIRAL VARCHAR(25) NOT NULL;
```

2.6.3 Ejemplo 9: Añadir/Eliminar una restricción

Añadir una restricción: El campo DIRAL toma valores diferentes (no se pueden repetir).

```
USE BDFORMACION;
```

```
ALTER TABLE ALUMNOS ADD CONSTRAINT uq_diral_uni UNIQUE(DIRAL);
```

Si quisiera borrar la restricción creada anteriormente:

```
USE BDFORMACION;
```

```
ALTER TABLE ALUMNOS DROP CONSTRAINT uq_diral_uni;
```

2.7 EJERCICIO: AÑADIR LA TABLA DEPARTAMENTOS A LA BD

Descripción informal

Se quiere almacenar la información detallada a continuación, sobre los departamentos a los que pertenecen los profesores. Los departamentos están relacionados con los profesores de tal manera que todo profesor pertenece a un departamento pero sólo a uno. No hay departamentos en los que no haya ningún profesor.

Características de la tabla departamentos:

- ✓ Los campos son: CODDEP, NOMDEP, PLANTA.
- ✓ La clave principal es CODDEP, dos caracteres, no podrá tener valores nulos ni duplicados.

- ✓ El NOMDEP será como máximo de 25 caracteres, no podrá repetirse y siempre se le asignará valor y se creará un índice.
- ✓ La PLANTA es un valor numérico entre 1 y 5 y no podrá tener valores nulos.

Realizar las modificaciones necesarias en el esquema conceptual, lógico (relacional) y en la implementación de la base de datos. En concreto:

- a) Añadir la entidad Departamentos al esquema conceptual.
 - Estará relacionada con Profesores con 1:N. Es decir, cada profesor tiene obligatoriamente un departamento y un departamento puede tener varios profesores.
- b) Reflejar el cambio en el esquema lógico.
- c) Crear la tabla departamentos.
- d) Añadir el campo CodDep a la tabla Profesores como clave ajena que apunta a Departamentos.
- e) Crear el índice para el campo NOMDEP.

2.8 AMPLIACIÓN: CREACIÓN/ELIMINACIÓN DE ÍNDICES (CREATE/DROP INDEX)

- ✓ Un índice es una estructura de disco asociada con una tabla o una vista que acelera la recuperación de filas de la tabla o de la vista.
- ✓ Contiene claves generadas a partir de una o varias columnas de la tabla o vista y permite que el gestor busque de forma rápida y eficiente la fila o filas asociadas a los valores de cada clave.
- ✓ Los índices se crean automáticamente cuando las restricciones PRIMARY KEY y UNIQUE se definen en las columnas de tabla.

2.8.1 Ejemplo 11: Crear un índice para un campo

- ✓ Crear un índice con valores duplicados para el campo CODP de la tabla CURSILLOS.

```
USE BDFORMACION;
```

```
CREATE INDEX ix_codp ON CURSILLOS (CODP);
```

2.8.2 Ejemplo 12: Crear un índice único para un campo

Crear un índice único para el campo DIR de la tabla ALUMNOS (suponiendo que las direcciones no se pueden repetir).

```
USE BDFORMACION;
```

```
CREATE UNIQUE INDEX uq_dir ON ALUMNOS (DIRAL);
```

2.8.3 Ejemplo 13: Crear un índice único con dos campos

Suponiendo que en la tabla ALUMNOS tenemos dos campos independientes, nombre (NOMAL) y la población del alumno (POB), escribir cómo sería la instrucción para crear un índice sin duplicados para los dos campos.

```
USE BDFORMACION;
```

```
CREATE UNIQUE INDEX uq_nom_pob ON ALUMNOS (NOMAL, POB);
```

2.8.4 Ejemplo 14: Borrado de un índice: DROP INDEX

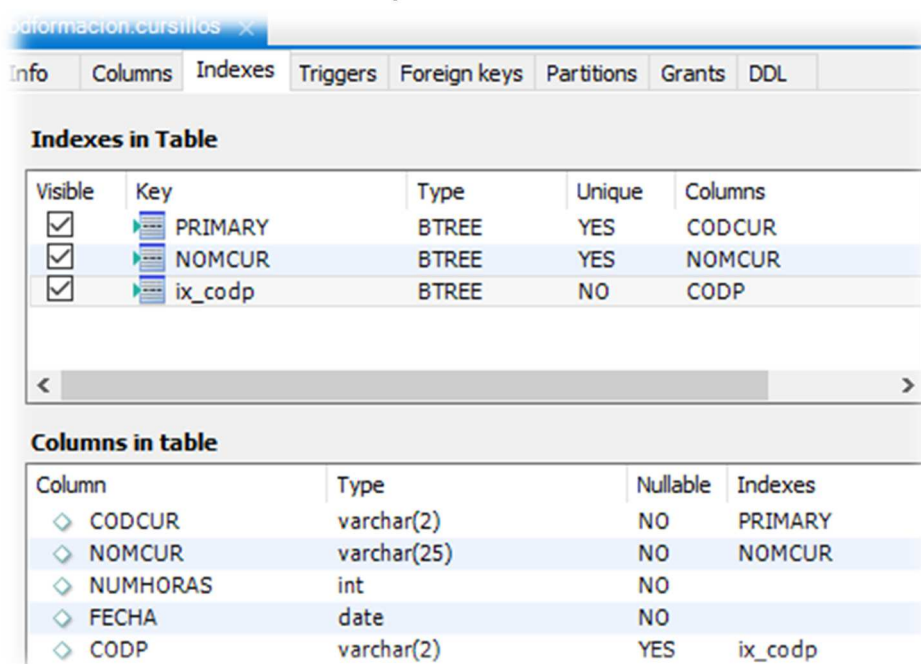
Eliminar el índice único para el campo DIR de la tabla ALUMNOS

```
USE BDFORMACION;
```

```
DROP INDEX uq_dir ON ALUMNOS;
```

2.9 AMPLIACIÓN: CONSULTA DE ÍNDICES

Podemos ver los índices o claves creados desde el entorno gráfico
Botón derecho en la tabla -> Tabla Inspector -> Pestaña



Visible	Key	Type	Unique	Columns
<input checked="" type="checkbox"/>	PRIMARY	BTREE	YES	CODCUR
<input checked="" type="checkbox"/>	NOMCUR	BTREE	YES	NOMCUR
<input checked="" type="checkbox"/>	ix_codp	BTREE	NO	CODP

Column	Type	Nullable	Indexes
◇ CODCUR	varchar(2)	NO	PRIMARY
◇ NOMCUR	varchar(25)	NO	NOMCUR
◇ NUMHORAS	int	NO	
◇ FECHA	date	NO	
◇ CODP	varchar(2)	YES	ix_codp