

## Solución a la tarea de la UD3 - XBD

### Ejercicio 1

#### Enunciado

Partiendo del siguiente modelo relaciona que corresponde a un restaurante crea una base de datos en MySQL con las siguientes tablas:

EMPLEADOS (DNI, Nombre, Apellidos, Dirección, Ciudad, Teléfono, CP, FechaAlta, Categoría, Salario)  
COCINEROS (DNI(fk), Puesto, Especialidad)  
ADMINISTRACIÓN (DNI(fk), Cargo)  
CAMAREROS (DNI(fk), Turno, años, DNI\_encargado(fk))  
COMEDORES (CodigoC, Nombre, Capacidad, Nmesas, Localización)  
MESAS (CodigoM, CodigoC(fk), Nasientos, DNI\_camarero(fk))  
Hacen ((CodigoM, CodigoC)(fk), Nreserva(fk), NasientosR)  
RESERVAS (Nreserva, Fhreserva, Nombre, Fhreservada, Comida/Cena, Npersonas, Varios)  
FACTURAS (NFactura, FechaF, (CodigoM, CodigoC)(fk))  
Incluyen (NFactura(fk), CodigoPL(fk), Unidades)  
PLATOS (CodigoPl, Nombre, Descripción, Tipo, Precio)  
Contienen (CodigoPL(fk), CodigoPr(fk), Cantidad)  
PRODUCTOS (CodigoPr, Descripción, Stock, UnidadBase, PrecioU, Categoría, CodigoPro(fk))  
PROVEEDORES (CodigoPro, Dirección, CP, Teléfono, Fax, Contacto)

Notas:

- Las claves primarias se indican en subrayado y las ajenas se indican en cursiva y con (fk)
- Debes asignar un tipo de dato apropiado al contenido de cada columna
- Aplica las restricciones de columna que veas apropiadas.
- En las claves foráneas, justifica el comportamiento que indiques en cuanto a borrados y modificaciones.

#### Solución

Nota: en algunos casos, la solución es orientativa respecto al tipo de datos y tamaño elegidos para las columnas.

```
DROP DATABASE IF EXISTS RestauranteMouro;
CREATE DATABASE RestauranteMouro;
USE RestauranteMouro;

CREATE TABLE EMPLEADOS (
  DNI VARCHAR(9) NOT NULL,
  Nombre VARCHAR(25),
  Apellidos VARCHAR(50),
  Direccion VARCHAR(50),
  Ciudad VARCHAR(25),
  Telefono VARCHAR(9),
  CP VARCHAR(5),
  FechaAlta DATE,
  Categoria VARCHAR(25),
  Salario DECIMAL(10,2),
  PRIMARY KEY(DNI)
```

```

)ENGINE=InnoDB;

CREATE TABLE COCINEROS (
  DNI VARCHAR(9) NOT NULL,
  Puesto VARCHAR(25),
  Especialidad VARCHAR(25),
  PRIMARY KEY(DNI)
)ENGINE=InnoDB;
CREATE TABLE ADMINISTRACION (
  DNI VARCHAR(9) NOT NULL,
  Cargo VARCHAR(25),
  PRIMARY KEY(DNI)
)ENGINE=InnoDB;

CREATE TABLE CAMAREROS (
  DNI VARCHAR(9) NOT NULL,
  Turno VARCHAR(25),
  Años TINYINT,
  DNI_encargado VARCHAR(9),
  PRIMARY KEY(DNI)
)ENGINE=InnoDB;

CREATE TABLE COMEDORES (
  CodigoC VARCHAR(5) NOT NULL,
  Nombre VARCHAR(25),
  Capacidad TINYINT,
  Nmesas TINYINT,
  Localizacion VARCHAR(25),
  PRIMARY KEY(CodigoC)
)ENGINE=InnoDB;

CREATE TABLE MESAS (
  CodigoM VARCHAR(5) NOT NULL,
  CodigoC VARCHAR(5) NOT NULL,
  Nasientos TINYINT,
  DNI_camarero VARCHAR(9),
  PRIMARY KEY(CodigoM, CodigoC)
)ENGINE=InnoDB;

CREATE TABLE Hacen (
  CodigoM VARCHAR(5) NOT NULL,
  CodigoC VARCHAR(5) NOT NULL,
  Nreserva INT NOT NULL,
  NasientosR INT,
  PRIMARY KEY(CodigoM, CodigoC, Nreserva)
)ENGINE=InnoDB;

CREATE TABLE RESERVAS (
  Nreserva INT NOT NULL,
  Fhreserva DATETIME,
  Nombre VARCHAR(25) NOT NULL,
  Fhreservada DATETIME,
  ComidaCena ENUM("Comida", "Cena"),
  Npersonas SMALLINT,
  Varios VARCHAR(100),
  PRIMARY KEY(Nreserva)
)ENGINE=InnoDB;

CREATE TABLE FACTURAS (
  NFactura INT NOT NULL,
  FechaF DATETIME,
  CodigoM VARCHAR(5),
  CodigoC VARCHAR(5),
  PRIMARY KEY(NFactura)
)ENGINE=InnoDB;

CREATE TABLE Incluyen (
  NFactura INT NOT NULL,
  CodigoPl VARCHAR(5) NOT NULL,
  Unidades TINYINT,
  PRIMARY KEY(NFactura, CodigoPl)
)ENGINE=InnoDB;

CREATE TABLE PLATOS (
  CodigoPl VARCHAR(5) NOT NULL,
  Nombre VARCHAR(25),
  Descripcion VARCHAR(50),

```

```

Tipo VARCHAR(25),
Precio DECIMAL(6,2),
PRIMARY KEY(CodigoPl)
)ENGINE=InnoDB;

CREATE TABLE Contienen (
CodigoPl VARCHAR(5) NOT NULL,
CodigoPr VARCHAR(5) NOT NULL,
Cantidad TINYINT,
PRIMARY KEY(CodigoPl, CodigoPr)
)ENGINE=InnoDB;

CREATE TABLE PRODUCTOS (
CodigoPr VARCHAR(5) NOT NULL,
Descripcion VARCHAR(50),
Stock TINYINT,
UnidadBase VARCHAR(25),
PrecioU DECIMAL(6,2),
Categoria VARCHAR(25),
CodigoPro VARCHAR(5),
PRIMARY KEY(CodigoPr)
)ENGINE=InnoDB;

CREATE TABLE PROVEEDORES (
CodigoPro VARCHAR(5) NOT NULL,
Direccion VARCHAR(50),
CP VARCHAR(5),
Telefono VARCHAR(9),
Fax VARCHAR(9),
Contacto VARCHAR(25),
PRIMARY KEY(CodigoPro)
)ENGINE=InnoDB;

ALTER TABLE COCINEROS
ADD CONSTRAINT COC_FK_DNI FOREIGN KEY(DNI) REFERENCES EMPLEADOS(DNI)
ON DELETE CASCADE ON UPDATE CASCADE;

ALTER TABLE ADMINISTRACION
ADD CONSTRAINT ADM_FK_DNI FOREIGN KEY(DNI) REFERENCES EMPLEADOS(DNI)
ON DELETE CASCADE ON UPDATE CASCADE;

ALTER TABLE CAMAREROS
ADD CONSTRAINT CAM_FK_DNI FOREIGN KEY(DNI) REFERENCES EMPLEADOS(DNI)
ON DELETE CASCADE ON UPDATE CASCADE,
ADD CONSTRAINT CAM_FK_DNI_encargado FOREIGN KEY(DNI_encargado) REFERENCES
CAMAREROS(DNI)
/* Así si borro un camarero que sea encargado evito borrar los camareros a los que
supervisa */
ON DELETE SET NULL ON UPDATE CASCADE;

ALTER TABLE MESAS
ADD CONSTRAINT MES_FK_CodC FOREIGN KEY(CodigoC) REFERENCES COMEDORES(CodigoC)
ON DELETE CASCADE ON UPDATE CASCADE,
ADD CONSTRAINT MES_FK_DNI_C FOREIGN KEY(DNI_camarero) REFERENCES CAMAREROS(DNI)
/* Así si borro un camarero que hubiese sido asignado a una mesa evito borrar a dicha
mesa*/
ON DELETE SET NULL ON UPDATE CASCADE;

ALTER TABLE Hacen
ADD CONSTRAINT Hac_FK_CodMCodC FOREIGN KEY(CodigoM,CodigoC) REFERENCES
MESAS(CodigoM,CodigoC)
/* no permite borrar una mesa que haya sido asignada a una reserva, solo permite
actualizaciones en cascada */
ON DELETE NO ACTION ON UPDATE CASCADE,
ADD CONSTRAINT Hac_FK_NRes FOREIGN KEY(Nreserva) REFERENCES RESERVAS(Nreserva)
/* si se borrara una reserva se borraría el correspondiente registro aquí */
ON DELETE CASCADE ON UPDATE CASCADE;

ALTER TABLE FACTURAS
ADD CONSTRAINT FAC_FK_CodMCodC FOREIGN KEY(CodigoM,CodigoC) REFERENCES
FACTURAS(CodigoM,CodigoC)
ON DELETE SET NULL ON UPDATE CASCADE;

ALTER TABLE Incluyen
ADD CONSTRAINT Inc_FK_NFac FOREIGN KEY(NFactura) REFERENCES FACTURAS(NFactura)
ON DELETE CASCADE ON UPDATE CASCADE,
ADD CONSTRAINT Inc_FK_CodPl FOREIGN KEY(CodigoPl) REFERENCES PLATOS(CodigoPl)

```

```

/* Si quiero tener información de facturas que contenían platos que ya no están en
carta*/
ON DELETE NO ACTION ON UPDATE NO ACTION;

ALTER TABLE Contienen
ADD CONSTRAINT Con_FK_CodPl FOREIGN KEY(CodigoPl) REFERENCES PLATOS(CodigoPl)
/* no permite borrar un plato que ya se hubiese diseñado */
ON DELETE NO ACTION ON UPDATE NO ACTION,
ADD CONSTRAINT Con_FK_CodPr FOREIGN KEY(CodigoPr) REFERENCES PRODUCTOS(CodigoPr)
ON DELETE CASCADE ON UPDATE CASCADE;

ALTER TABLE PRODUCTOS
ADD CONSTRAINT PRO_FK_CodPro FOREIGN KEY(CodigoPro) REFERENCES PROVEEDORES(CodigoPro)
/* Así si borro un proveedor que suministre un cierto producto evito borrar el
producto en sí, en tal caso el código del proveedor se pone a valor nulo*/
ON DELETE SET NULL ON UPDATE CASCADE;

```

## Ejercicio 2

### Enunciado

Sobre la base de datos Restaurante Mouro realiza las siguientes modificaciones:

#### Tabla EMPLEADOS

- 2.1. Añade un índice que facilite búsquedas frecuentes por Apellidos y Nombre sin duplicados.
- 2.2. El Restaurante se ha inaugurado el día 15 de Julio de 2010. Comprueba que la fecha de alta de los empleados no sea anterior a esa fecha.

#### Tabla CAMAREROS

- 2.3. El turno de trabajo sólo puede tomar 3 valores: mañana, tarde y noche. Añade esa restricción teniendo en cuenta que un camarero puede tener más de un turno (no utilices CHECK).

#### Tabla PROVEEDORES

- 2.4. Añade las columnas Apellidos y Nombre entre el código y la dirección.
- Tabla MESAS
- 2.5. El valor por defecto del número de asientos en las mesas es 4.

#### Tabla PRODUCTOS

- 2.6. Añade un índice por Categoría. Muestra todos los índices de la tabla.
- 2.7. Añade una restricción en la tabla, de forma que el Stock sea entero de 4 cifras, sin signo y que no admita nulos.
- 2.8. Borra el índice que acabas de crear.

#### Tabla PLATOS

- 2.9. Borra la tabla PROVEEDORES. ¿Qué ocurre?. Borra previamente las claves ajenas.

#### BASE DE DATOS Restaurante Mouro

- 2.10. Borra la base de datos

### Solución

```

use RestauranteMouro;
/*
TABLA EMPLEADOS:
    2.1 Añadir un índice para facilitar búsquedas por Apellidos y Nombres sin duplicados
    2.2 Comprobar que fecha de alta de empleados no sea anterior al 15 de Julio de 2010
*/
ALTER TABLE EMPLEADOS
ADD UNIQUE INDEX key_EMP_ApeNom (Apellidos,Nombre),

```

```

ADD CONSTRAINT check_EMP_FechaAlta CHECK (FechAlta >= "2010-07-10");
/*
TABLA CAMAREROS:
    2.3 Añadir restricción: Turno de trabajo solo puede tomar 3 valores: mañana, tarde y noche
    (sin usar check)
*/
ALTER TABLE CAMAREROS
MODIFY Turno ENUM("Mañana","Tarde","Noche");
/*
TABLA PROVEEDORES:
    2.4 Añadir columnas "Apellidos" y "Nombre" entre el código y la dirección
*/
ALTER TABLE PROVEEDORES
ADD COLUMN Apellidos VARCHAR(50) AFTER CodigoPro,
ADD COLUMN Nombre VARCHAR(25) AFTER Apellidos;
/*
TABLA MESAS:
    2.5 El valor por defecto del número de asientos en las mesas es 4
*/
ALTER TABLE MESAS
ALTER COLUMN Nasientos SET DEFAULT 4;
/*
TABLA PRODUCTOS:
    2.6 Añadir un índice por Categoría. Mostrar todos los índices de la tabla.
    2.7 Añadir una restricción en la tabla de manera que Stock sea entero de 4 cifras, sin signo y que no
    admita nulos.
    2.8 Borrar el índice que acabo de crear.
*/
-- 2.6
ALTER TABLE PRODUCTOS
ADD UNIQUE INDEX key_PRO_Cat(Categoria);
SHOW INDEX FROM PRODUCTOS;

-- 2.7
ALTER TABLE PRODUCTOS
CHANGE Stock Stock SMALLINT(4) UNSIGNED NOT NULL; -- INT(N) y semejantes quedarán obsoletos
en versiones posteriores a esta.

-- 2.8
DROP INDEX key_PRO_Cat ON PRODUCTOS;

/* 2.9 Intentar borrar la tabla proveedores, ¿qué pasa?*/
DROP TABLE PROVEEDORES;
-- no deja borrarla por que hay una clave ajena que hace referencia a esta tabla: en tabla productos hay
una clave ajena
-- que referencia a la de proveedores. Así que debemos borrar primero esta clave ajena:
ALTER TABLE PRODUCTOS
DROP FOREIGN KEY PRO_FK_CodPro;
-- ahora sí nos dejará borrar la tabla productos
DROP TABLE PROVEEDORES;

/* 2.10 Borrar la base de datos*/
DROP DATABASE RestauranteMouro;

```