



Preparador Informática

PRÁCTICA 11

BASES DE DATOS

(SQL BÁSICO)

(SOLUCIONES Ejercicios 27 al 32)

EJERCICIO 27. EMPLEADO-DEPARTAMENTO

27. En vista de las siguientes tablas debes:

- Crear la base de datos "Departamento".
- Crea la tabla EMPLEADO y la tabla DEPARTAMENTO con las columnas que vemos en la imagen y utilizando los tipos de datos y restricciones que consideres adecuados.

	codigo_c	nombre	edad	oficio	dir	fecha_alt	salario	comision	depto_no
▶	281-160483-0005F	Rocha Vargas Hector	27	Vendedor	Leon	12/05/1983	12000	0	40
	281-040483-0056P	López Hernandez Julio	27	Analista	Chinandega	14/07/1982	13000	1500	20
	081-130678-0004S	Esquivel José	31	Director	Juigalpa	05/06/1981	16700	1200	30
	281-160473-0009Q	Delgado Carmen	37	Vendedor	Leon	02/03/1983	13400	0	40
	281-160493-0005F	Castillo Montes Luis	17	Vendedor	Masaya	12/08/1982	16309	1000	40
	281-240784-0004Y	Esquivel Leonel Alfonso	26	Presidente	Nagarote	12/09/1981	15000	0	30
	281-161277-0008R	Perez Luis	32	Empleado	Managua	02/03/1980	16890	0	10

	depto_no	nombre_depto	localizacion
▶	10	Desarrollo Software	El Coyolar
	20	Analisis Sistema	Guadalupe
	30	Contabilidad	Subtiava
	40	Ventas	San Felipe
*	0		

- Inserta en las tablas todos los datos que se ven en las imágenes anteriores.
- Realiza las siguientes actualizaciones de la base de datos:

- Insertar en la tabla EMPLEADO un empleado con código 081-220678-0008U, nombre 'Pérez Luis Carlos' de 32 años, oficio Analista, vive en Málaga, fecha de alta en la empresa el 22-06-2001, salario de 15600 euros, no tiene comisión y pertenece al departamento 20.
- Insertar en la tabla DEPARTAMENTO un departamento cuyo número sea 50, de nombre 'GENERAL' y cuya localización sea 'Laborio'
- Insertar en la tabla DEPARTAMENTO un departamento cuyo número sea 60 y de nombre 'Pruebas'
- Doblar el salario a todos los empleados del departamento 30.
- Cambiar todos los empleados del departamento número 30 al departamento número 20.
- Incrementar en un 10% el sueldo de los empleados del departamento 10.
- Cambiar la localidad del departamento número 10 a 'Zaragoza'.



- En la tabla DEPARTAMENTO borrar el departamento número 40.
- En la tabla EMPLEADO borrar todos los empleados que sean del departamento 20 y sean 'Analista'.
- En la tabla DEPARTAMENTO borrar todos los empleados que no tengan comisión.

EJERCICIO 27. EMPLEADO-DEPARTAMENTO. SOLUCIÓN PROPUESTA

a) Crear la base de datos "Departamento".

```
CREATE DATABASE Departamento;  
USE Departamento;
```

b) Crea la tabla EMPLEADO y la tabla DEPARTAMENTO con las columnas que vemos en la imagen y utilizando los tipos de datos y restricciones que consideres adecuados.

```
CREATE TABLE Departamento  
(  
    depto_no INT PRIMARY KEY,  
    nombre_depto varchar (20) NOT NULL,  
    localizacion varchar (30)  
);  
  
CREATE TABLE Empleado  
(  
    codigo_c VARCHAR(16) NOT NULL PRIMARY KEY,  
    nombre VARCHAR(30) NOT NULL ,  
    edad INT (2),  
    oficio VARCHAR(15) NOT NULL,  
    direccion VARCHAR(15) NOT NULL,  
    fecha_alt DATE NOT NULL,  
    salario FLOAT NOT NULL,  
    comision FLOAT,  
    depto_no INT NOT NULL,  
    CONSTRAINT FK_Dpto FOREIGN KEY (depto_no) REFERENCES  
    Departamento (depto_no) on delete cascade on update cascade  
);
```

c) Inserta en las tablas todos los datos que se ven en las imágenes anteriores.

```
INSERT INTO Departamento (depto_no, nombre_depto, localizacion)
VALUES
```

```
    ('10', 'Desarrollo Software', 'El Coyolar'),
    ('20', 'Análisis Sistema', 'Guadalupe'),
    ('30', 'Contabilidad', 'Subtiava'),
    ('40', 'Ventas', 'San Felipe');
```

```
INSERT INTO Empleado VALUES
```

```
    ('281-160483-0005F', 'Rocha Vargas Hector', 27, 'Vendedor',
    'Leon', '1983-05-12', 12000, null, 40),

    ('281-040483-0056P', 'López Hernandez Julio', 27, 'Analista',
    'Chinandega', '1982-07-14', 13000, 1500, 20),

    ('081-130678-0004S', 'Esquivel Jose', 31, 'Director',
    'Juigalpa', '1981-06-05', 16700, 1200, 30),

    ('281-160473-0009Q', 'Delgado Carmen', 37, 'Vendedor', 'Leon',
    '1983-03-02', 13400, null, 40),

    ('281-160493-0005F', 'Castillo Montes Luis', 17, 'Vendedor',
    'Masaya', '1982-08-12', 16309, 1000, 40),

    ('281-240784-0004Y', 'Esquivel Leonel Alfonso', 26,
    'Presidente', 'Nagarote', '1981-09-12', 15000, null, 30),

    ('281-161277-0008R', 'Perez Luis', 32, 'Empleado', 'Managua',
    '1980-03-02', 16890, null, 10) ;
```

d) Realiza las siguientes actualizaciones de la base de datos:

- Insertar en la tabla EMPLEADO un empleado con código 081-220678-0008U, nombre 'Pérez Luis Carlos' de 32 años, oficio Analista, vive en Málaga, fecha de alta en la empresa el 22-06-2001, salario de 15600 euros, no tiene comisión y pertenece al departamento 20.

```
INSERT INTO Empleado VALUES
```

```
    ('220678-0008U', 'Pérez Luis Carlos', 32, 'Analista',
    'Málaga', '2001-06-22', 15600, null, 20);
```

- Insertar en la tabla DEPARTAMENTO un departamento cuyo número sea 50, de nombre 'GENERAL' y cuya localización sea 'Laborio'

```
INSERT INTO Departamento VALUES (50, 'General',
    'Laborio');
```



- Insertar en la tabla DEPARTAMENTO un departamento cuyo número sea 60 y de nombre 'Pruebas'

```
INSERT INTO Departamento (depto_no, nombre_depto)
VALUES (60, 'Pruebas');
```

- Doblar el salario a todos los empleados del departamento 30.

```
UPDATE Empleado SET salario=salario*2 where
depto_no=30;
```

- Cambiar todos los empleados del departamento número 30 al departamento número 20.

```
UPDATE Empleado SET depto_no= 20 where depto_no=30;
```

- Incrementar en un 10% el sueldo de los empleados del departamento 10.

```
UPDATE Empleado SET salario=salario*1.10 where
depto_no=10;
```

- Cambiar la localidad del departamento número 10 a 'Zaragoza'.

```
UPDATE Departamento SET localizacion='Zaragoza' where
depto_no=10;
```

- En la tabla DEPARTAMENTO borrar el departamento número 40.

```
DELETE FROM Departamento WHERE depto_no=40;
```

- En la tabla EMPLEADO borrar todos los empleados que sean del departamento 20 y sean 'Analista'.

```
DELETE FROM Empleado WHERE depto_no=20 and
oficio='Analista';
```

- En la tabla DEPARTAMENTO borrar todos los empleados que no tengan comisión.

```
DELETE FROM Empleado WHERE comision is null;
```

EJERCICIO 28. ATRACCIONES

28. Dado el esquema:

Trabaja (DNI, atracción, fecha, horas)

DiasApertura (fecha, horaApertura, horaCierre, entradasVendidas)

VentasOnline (tarjeta, fecha, numEntradas, fechaCompra, descuento)

Crear las tablas con SQL incluyendo integridad referencial y las siguientes restricciones:

- Si se decide cerrar un día por mal tiempo se borran en cascada las asignaciones de trabajo y VentasOnline de dicho día.
- No se puede asignar una atracción a un encargado más de 8 horas al día

EJERCICIO 28. ATRACCIONES. SOLUCIÓN PROPUESTA

Aunque no lo pida expresamente el ejercicio vamos a crear una base de datos nueva para este ejercicio llamada "Feria" por ejemplo:

```
CREATE DATABASE Feria;
USE Feria;

Create table DiasApertura
(
    fecha DATE,
    HoraApertura TIME,
    HoraCierre TIME,
    EntradasVendidas INT(5),
    CONSTRAINT tra_PK PRIMARY KEY (fecha)
);

Create table Trabaja
(
    DNI INT(8),
    atraccion char(25),
    fecha DATE,
    horas INT(1),
    CONSTRAINT tra_PK PRIMARY KEY (dni, atraccion, fecha),
    CONSTRAINT tra_ct_FK FOREIGN KEY (fecha)
    REFERENCES DiasApertura(fecha) on delete cascade,
    CONSTRAINT tra_maximo CHECK(horas<=8)
);
```

```
Create table VentasOnline
(
    tarjeta CHAR (10),
    fecha DATE,
    fechaCompra DATE,
    numEntradas INT(4),
    Descuento FLOAT,
    CONSTRAINT tra_PK PRIMARY KEY (tarjeta, fecha),
    CONSTRAINT tra_ct_FK2 FOREIGN KEY (fecha)
    REFERENCES DiasApertura(fecha) on delete cascade
);
```

EJERCICIO 29. ATRACCIONES (II)

29. Dado el esquema:

Trabaja (DNI, atracción, fecha, horas, aforoAtracción)

¿Está en Tercera Forma Normal? En caso negativo entonces descomponer “Trabaja” hasta llegar a 3FN subrayando las claves.

EJERCICIO 29. ATRACCIONES. SOLUCIÓN PROPUESTA

TRABAJA (DNI, atracción, fecha, horas)

ATRACCIÓN (atracción, aforoAtracción)

Preparador Informática

EJERCICIO 30. AGENDA

30. Se dispone de la tabla AGENDA con los datos de aquellos empleados que trabajan en una serie de hospitales.

cod_hospital	dni	apellidos	funcion	salario	localidad
1	12345678	García Hernández, Eladio	CONSERJE	1200	LORCA
1	87654321	Fuentes Bermejo, Carlos	DIRECTOR	2000	MURCIA
2	55544433	González Marín, Alicia	CONSERJE	1200	MURCIA
1	66655544	Castillo Montes, Pedro	MEDICO	1700	MURCIA
2	22233322	Tristán García, Ana	MEDICO	1900	MURCIA
3	55544411	Ruiz Hernández, Caridad	MEDICO	1900	LORCA
3	99988333	Serrano Díaz, Alejandro	DIRECTOR	2400	CARTAGENA
4	33222111	Mesa del Castillo, Juan	MEDICO	2200	LORCA
2	22233333	Martínez Molina, Andrés	MEDICO	1600	CARTAGENA
4	55544412	Jiménez Jiménez, Dolores	CONSERJE	1200	MURCIA
4	22233311	Martínez Molina, Gloria	MEDICO	1600	MURCIA

- Crea en SQL la base de datos AGENDA.
- Crea la tabla anterior y observando los datos deduce las posibles restricciones que debe llevar la tabla.
- Introduce todos los valores anteriores en la tabla creada.
- Realiza las siguientes modificaciones en la tabla:
 - Añade los campos teléfono y email.
 - Modifica la localidad de todos los que tengan código de hospital 1. Ahora su localidad será Almería.
 - Elimina de la base de datos a todas las personas que cobren más de 2000 euros.

EJERCICIO 30. AGENDA. SOLUCIÓN PROPUESTA

- Crea en SQL la base de datos AGENDA.

```
CREATE DATABASE bd_agenda;
USE bd_agenda;
```



- b) Crea la tabla anterior y observando los datos deduce las posibles restricciones que debe llevar la tabla.

```
create table personas
(
    cod_hospital int(3),
    dni int(8) primary key,
    apellidos varchar(50),
    funcion varchar(30),
    salario float,
    localidad varchar(20)
);
```

- c) Introduce todos los valores anteriores en la tabla creada.

```
insert into personas values (1,12345678,'García
Hernández, Eladio','CONSERJE',1200,'LORCA');

insert into personas values (1,87654321,'Fuentes
Bermejo, Carlos','DIRECTOR',2000,'MURCIA');

insert into personas values (2,55544433,'González
Marín, Alicia','CONSERJE',1200,'MURCIA');

insert into personas values (1,66655544,'Castillo
Montes, Pedro','MEDICO',1700,'MURCIA');

insert into personas values (2,22233322,'Tristán
García, Ana','MEDICO',1900,'MURCIA');

insert into personas values (3,55544411,'Ruiz
Hernández, Caridad','MEDICO',1900,'LORCA');

insert into personas values (3,99988333,'Serrano Díaz,
Alejandro','DIRECTOR',2400,'CARTAGENA');

insert into personas values (4,33222111,'Mesa del
Castillo, Juan','MEDICO',2200,'LORCA');

insert into personas values (2,22233333,'Martínez
Molina, Andrés','MEDICO',1600,'CARTAGENA');

insert into personas values (4,55544412,'Jiménez
Jiménez, Dolores','CONSERJE',1200,'MURCIA');

insert into personas values (4,22233311,'Martínez
Molina, Gloria','MEDICO',1600,'MURCIA');
```

d) Realiza las siguientes modificaciones en la tabla:

- **Añade los campos teléfono y email.**

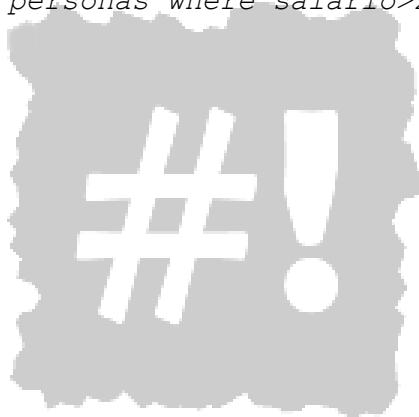
```
Alter table personas add column (telefono int(9),email  
varchar(30));
```

- **Modifica la localidad de todos los que tengan código de hospital 1. Ahora su localidad será Almería.**

```
Update personas set localidad = 'Almeria' where  
cod_hospital = 1;
```

- **Elimina de la base de datos a todas las personas que cobren más de 2000 euros.**

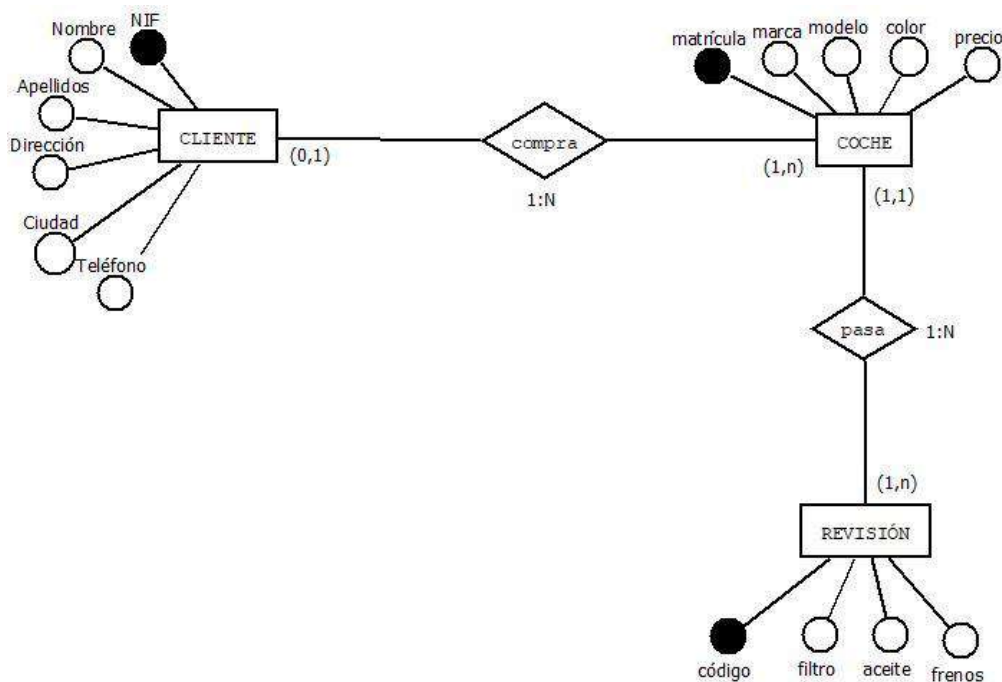
```
Delete from personas where salario>2000;
```



Preparador Informática

EJERCICIO 31. COCHES

31. Dado el siguiente diagrama entidad-relación.



Se pide:

- Realizar el paso al modelo relacional.
- Comprueba que las tablas estén en 3FN.
- Realiza el grafo relacional.
- Crea en SQL la base de datos COCHES.
- Crea las tablas necesarias en SQL con las restricciones necesarias.
- Introduce todos los valores siguientes en las tablas creadas.

nif	nombre	direccion	telefono
111	Ignacio	Murcia	555-11-22-33
222	Julia	Burgos	555-00-33-44
333	Pedro	Cádiz	555-99-88-00
444	David	Vizcaya	555-00-88-77

matricula	marca	modelo	color	precio	nif_cliente
2222-BBB	seat	ibiza	blanco	12000.00	111
3333-CCC	seat	toledo	gris	18000.00	222
4444-DDD	renault	clio	rojo	15000.00	222
6666-FFF	renault	megane	blanco	19000.00	444

codigo	matricula_coche	filtro	aceite	frenos
1	2222-BBB	1	0	1
2	3333-CCC	1	1	1
3	4444-DDD	0	0	1
4	2222-BBB	1	0	1

EJERCICIO 31. COCHES. SOLUCIÓN PROPUESTA

a) Realizar el paso al modelo relacional.

Para realizar el paso al modelo relacional se siguen los siguientes criterios:

- Todas las entidades se convierten en tabla y todo atributo se transforma en columna dentro de la tabla. En nuestro ejercicio se crearán tablas de las entidades COCHE, CLIENTE y REVISIÓN con sus correspondientes atributos.
- Respecto a las relaciones de este diagrama he seguido los siguientes criterios:
 - o Para las relaciones 1:N que no tienen atributos propios lo recomendable es propagar el identificador de la entidad A cuya cardinalidad máxima es 1 a la tabla de la entidad B haciéndolo como clave ajena y no crear la tabla para la relación. Así por ejemplo en nuestro ejercicio tenemos la relación “pasa” para la cual no crearemos tabla y lo que haremos será propagar la clave de la entidad COCHE a la entidad REVISIÓN como clave ajena y, también tenemos la relación “compra” para la cual no crearemos tabla y lo que haremos será propagar la clave de la entidad CLIENTE a la entidad COCHE como clave ajena.

De esta forma, se obtienen las siguientes tablas donde aparecen subrayados los atributos clave:

CLIENTE (NIF, Nombre, Apellidos, Dirección, Ciudad, Teléfono)

COCHE (Matrícula, Marca, Modelo, Color, Precio, NIFCliente)

REVISIÓN (Código, Filtro, Aceite, Frenos, MatriculaCoche)

b) Comprueba que las tablas estén en 3FN.

NORMALIZACIÓN

El proceso de normalización es un proceso importante en nuestra base de datos. Los objetivos del mismo son los siguientes:

- Controlar la redundancia de la información.
- Evitar pérdidas de información.
- Capacidad para representar toda la información.
- Mantener la consistencia de los datos.



1ª Forma Normal:

Una relación está en 1FN si no existen atributos multivaluados ni relaciones anidadas. Dicho de otra forma, estará en 1FN si los atributos no clave, dependen funcionalmente de la clave.

Comprobando las tablas obtenidas en el paso a tablas se observa que ya se encuentran en 1ª Forma Normal.

2ª Forma Normal

Una relación está en 2FN si está en 1FN y todos los atributos no clave tienen dependencia funcional completa de la clave primaria. De acuerdo con esta definición, cada tabla que tiene un atributo único como clave, está en segunda forma normal.

Comprobando las tablas obtenidas en el paso a tablas se observa que ya se encuentran en 2ª Forma Normal.

3ª forma normal

Una relación está en 3FN si y solo si está en 2FN y todos sus atributos no clave dependen no transitivamente de la clave primaria. Dicho de otro modo, si y sólo si los atributos no clave son mutuamente independientes y son dependientes por completo de la clave primaria.

Comprobando las tablas obtenidas en el paso a tablas se observa que ya se encuentran en 3ª Forma Normal.

Por tanto, las tablas obtenidas en el paso a tablas se encuentran normalizadas hasta la 3ªFN.

c) Realiza el grafo relacional.

El grafo relacional es un grafo compuesto de un conjunto de nodos, donde cada nodo representa un esquema de relación, es decir, una tabla de la BD con su nombre y atributos. Se dibuja, además, un conjunto de arcos que conectan los atributos que constituyen la clave ajena de una tabla respecto a la clave primaria de la otra tabla referenciada. Los arcos están direccionados de modo que el arco parta de la clave ajena y la flecha señale a la tabla referenciada.

CLIENTE (NIF, Nombre, Apellidos, Dirección, Ciudad, Teléfono)

COCHE (Matrícula, Marca, Modelo, Color, Precio, NIFCliente)

REVISIÓN (Código, Filtro, Aceite, Frenos, MatriculaCoche)

d) Crea en SQL la base de datos COCHES.

```
CREATE DATABASE bd_automoviles;  
USE bd_automoviles;
```

e) Crea las tablas necesarias en SQL con las restricciones necesarias.

```
CREATE TABLE cliente  
(  
    nif VARCHAR(9) PRIMARY KEY,  
    nombre VARCHAR(20) NOT NULL,  
    apellidos VARCHAR(30),  
    direccion VARCHAR(50),  
    ciudad VARCHAR(20),  
    telefono INT(9)  
);  
  
CREATE TABLE coche  
(  
    matricula VARCHAR(8) PRIMARY KEY,  
    marca VARCHAR(20),  
    modelo VARCHAR(20),  
    color VARCHAR(20),  
    precio FLOAT,  
    nif_cliente VARCHAR(9),  
    FOREIGN KEY (nif_cliente) REFERENCES clientes(nif)  
    ON DELETE SET NULL ON UPDATE CASCADE  
);  
  
CREATE TABLE revision  
(  
    codigo INT AUTO_INCREMENT PRIMARY KEY,  
    matricula_coche VARCHAR(8),  
    filtro BOOL DEFAULT FALSE,  
    aceite BOOL DEFAULT FALSE,  
    frenos BOOL DEFAULT FALSE,  
    FOREIGN KEY (matricula_coche) REFERENCES  
    coches(matricula) ON DELETE CASCADE ON UPDATE CASCADE  
);
```

f) Introduce todos los valores siguientes en las tablas creada.

nif	nombre	direccion	telefono
111	Ignacio	Murcia	555-11-22-33
222	Julia	Burgos	555-00-33-44

333 Pedro Cádiz 555-99-88-00
444 David Vizcaya 555-00-88-77

INSERT INTO cliente VALUES

('111', 'Ignacio', 'Murcia', 555112233),
('222', 'Julia', 'Burgos', 555003344),
('333', 'Pedro', 'Cadiz', 555998800),
('444', 'David', 'Vizcaya', 555008877);

matricula	marca	modelo	color	precio	nif_cliente
2222-BBB	seat	ibiza	blanco	12000.00	111
3333-CCC	seat	toledo	gris	18000.00	222
4444-DDD	renault	clio	rojo	15000.00	222
6666-FFF	renault	megane	blanco	19000.00	444

INSERT INTO coche VALUES

('2222-BBB', 'seat', 'ibiza', 'blanco', 12000, '111'),
('3333-CCC', 'seat', 'toledo', 'gris', 18000, '222'),
('4444-DDD', 'renault', 'clio', 'rojo', 15000, '222'),
('6666-FFF', 'renault', 'megane', 'blanco', 19000, '444');

codigo	matricula_coche	filtro	aceite	frenos
1	2222-BBB	1	0	1
2	3333-CCC	1	1	1
3	4444-DDD	0	0	1
4	2222-BBB	1	0	1

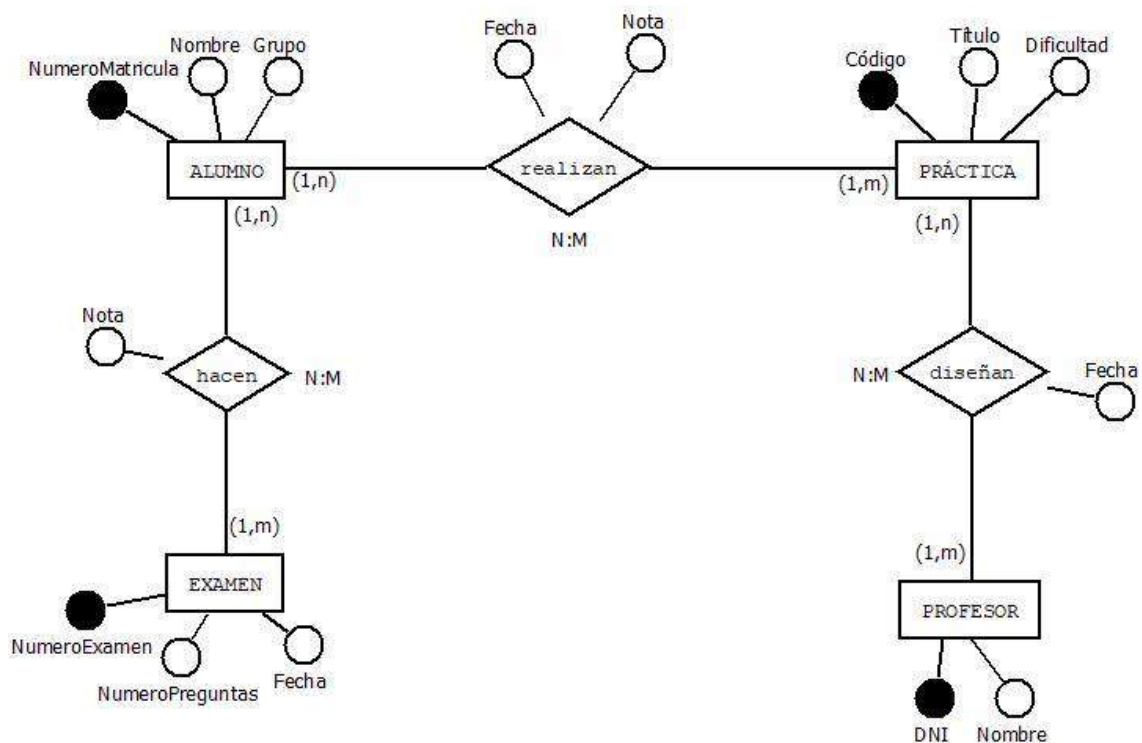
Preparador Informática

INSERT INTO revision VALUES

(1, '2222-BBB', 1, 0, 1),
(2, '3333-CCC', 1, 1, 1),
(3, '4444-DDD', 0, 0, 1),
(4, '2222-BBB', 1, 0, 1);

EJERCICIO 32. EXÁMENES

32. Dado el siguiente diagrama entidad-relación y dado su grafo relacional.



EXAMEN (NumeroExamen, NumeroPreguntas, Fecha)

HACEN (NumeroExamen, NumeroMatricula, Nota)

ALUMNO (NumeroMatricula, Nombre, Grupo)

REALIZAN (NumeroMatricula, Alumno, Código, Práctica, Fecha, Nota)

PRÁCTICA (Código, Título, Dificultad)

DISEÑAN (Código, Práctica, DNI, Profesor, fecha)

PROFESOR (DNI, Nombre)

Se pide:

a) Crea en SQL la base de datos EXÁMENES.

```
CREATE DATABASE examenes;
```

```
USE examenes;
```

b) Crea las tablas necesarias en SQL con las restricciones necesarias.

```
CREATE TABLE examen
(
    NumeroExamen int(5) PRIMARY KEY,
    NumeroPreguntas int(3) NOT NULL,
    Fecha DATE
);
```

```
CREATE TABLE alumno
(
    NumeroMatricula int(10) PRIMARY KEY,
    Nombre VARCHAR (30) NOT NULL,
    Grupo VARCHAR (10)
);
```

```
CREATE TABLE hacen
(
    NumeroExamen int(5) ,
    NumeroMatricula int(10),
    Nota FLOAT,
    CONSTRAINT PK1 PRIMARY KEY (NumeroExamen, NumeroMatricula),
    CONSTRAINT FK1 FOREIGN KEY (NumeroExamen) REFERENCES examen
        (NumeroExamen) on delete cascade on update cascade,
    CONSTRAINT FK2 FOREIGN KEY (NumeroMatricula) REFERENCES alumno
        (NumeroMatricula) on delete cascade on update cascade
);
```

```
CREATE TABLE practica
(
    codigo int(5) PRIMARY KEY,
    Titulo VARCHAR (30) NOT NULL,
    dificultad int (1)
);
```

```
CREATE TABLE realizan
(
    NumeroMatricula int(10),
   CodigoPractica int(5) ,
    Fecha DATE,
    Nota FLOAT,
    CONSTRAINT PK2 PRIMARY KEY (NumeroMatricula,CodigoPractica),
    CONSTRAINT FK3 FOREIGN KEY (NumeroMatricula) REFERENCES
alumno (NumeroMatricula) on delete cascade on update
cascade,
    CONSTRAINT FK4 FOREIGN KEY (CodigoPractica) REFERENCES
practica (codigo) on delete cascade on update cascade
);
```

```
CREATE TABLE profesor
(
    DNI int(8) PRIMARY KEY,
    Nombre VARCHAR (30) NOT NULL
);
```

```
CREATE TABLE diseñan
(
    CodigoPractica int(5) ,
    DNIProfesor int(8),
    fecha DATE,
    CONSTRAINT PK1 PRIMARY KEY (CodigoPractica, DNIProfesor),
    CONSTRAINT FK5 FOREIGN KEY (CodigoPractica) REFERENCES
practica (codigo) on delete cascade on update cascade,
    CONSTRAINT FK6 FOREIGN KEY (DNIProfesor) REFERENCES profesor
(DNI) on delete cascade on update cascade
);
```