



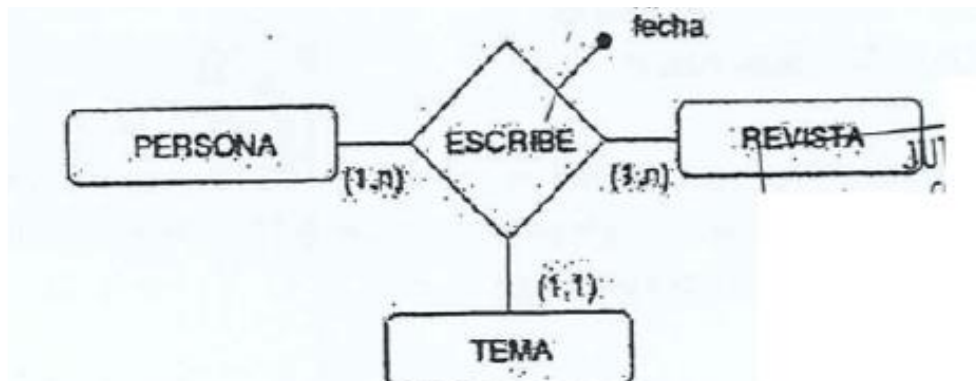
Preparador Informática

www.preparadorinformatica.com

PRÁCTICA 10
BASES DE DATOS
(Ejercicios 25 y 26)
OPOSICIONES SAI 2018
ANDALUCÍA

EJERCICIO 25. Pregunta Tipo Test

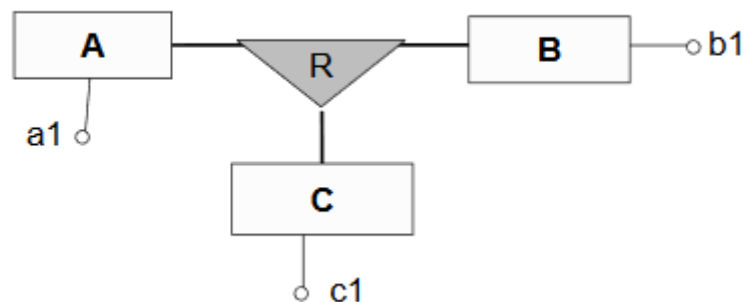
25. Indica cuál de las siguientes afirmaciones es cierta teniendo en cuenta la transformación del siguiente esquema E/R al esquema relacional:



- a) La relación ESCRIBE (persona, revista, fecha)
- b) La relación ESCRIBE (persona, revista, tema, fecha)
- c) La relación ESCRIBE (persona, revista, tema, fecha)
- d) La relación ESCRIBE (persona, revista, tema, fecha)

EJERCICIO 25. Pregunta Tipo Test. SOLUCIÓN PROPUESTA

La solución correcta es la d. Para argumentarlo podéis ver el siguiente esquema:



A (a1, ...)

B (b1, ...)

C (c1, ...)

1:N:N → R (a1, b1, c1)

EJERCICIO 26. Examen EJERCICIO 3

26. Enunciado: Se desea crear una base de datos que contenga la información de las distintas convocatorias de oposiciones que se realizan, así como de los/as opositores/as que se presentan en cada especialidad y la calificación final que obtienen.

- 1. Realiza el Modelo Entidad Relación y el Modelo Relacional que contemple el supuesto anterior teniendo en cuenta las siguientes consideraciones:**
 - Se deberán registrar las distintas convocatorias, así como los opositores y las especialidades.
 - En cada convocatoria se podrán convocar una o varias especialidades con un número de plazas determinado.
 - Un opositor opoita a una especialidad de una convocatoria.
 - Puede ocurrir que una persona opoite en una convocatoria a una especialidad y en otra convocatoria a otra.
 - Se desea almacenar la nota final que obtiene el opositor.
- 2. Escribe las sentencias SQL necesarias para:**
 - Crear y usar la base de datos “Oposiciones” usando codificación utf8.
 - Crear al menos dos tablas del modelo anteriormente desarrollado. Selecciona el tipo de dato que consideres más eficiente para almacenar los datos.
 - Establecer las claves primarias de las tablas anteriores.

Nota:

- Para acceder a mysql no hace falta contraseña. Usa: `sudo mysql`

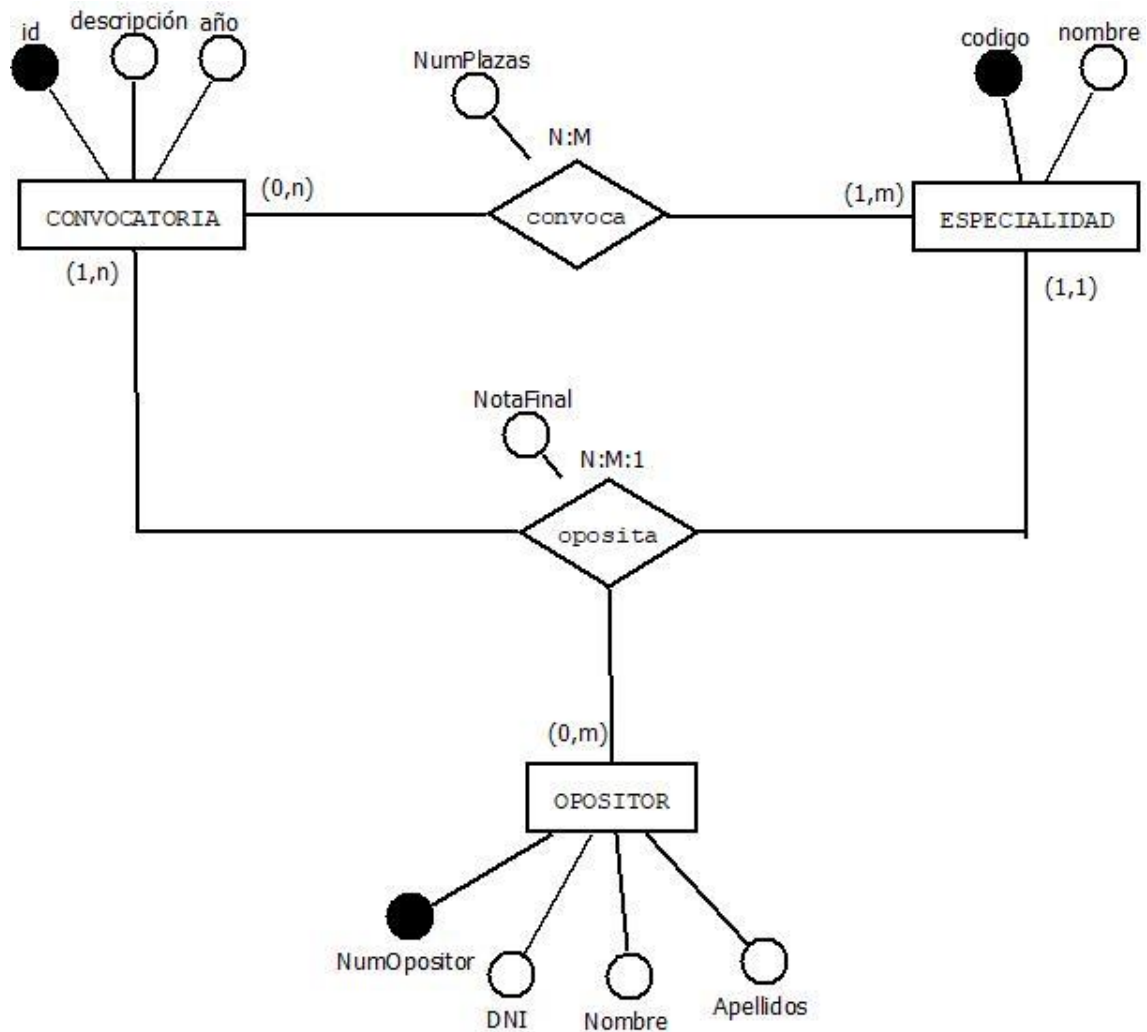
Modo de entrega:

- **Apartado 1:** Escribe la solución en la hoja de respuesta del ejercicio 3.
- **Apartado 2:**
 - Copia todas las sentencias SQL a un archivo `ejercicio3.sql` y grábalo en el pendrive.
 - Escribe las sentencias SQL en la hoja de respuesta del ejercicio 3.



EJERCICIO 26. Examen EJERCICIO 3. SOLUCIÓN PROPUESTA

El **diagrama entidad-relación** para los requerimientos anteriores es el siguiente:



ACLARACIONES:

(Mis aclaraciones las podéis escuchar en el video)

(Una vez hecho el Modelo Entidad Relación también nos pide realizar al paso al Modelo Relacional. En el examen no nos especifica que hagamos la normalización ni el grafo relacional. Lo hemos hecho para realizar el ejercicio más completo.)

PASO A MODELO RELACIONAL

Para realizar el paso a tablas 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 CONVOCATORIA, ESPECIALIDAD y OPOSITOR con sus correspondientes atributos.
- Respecto a las relaciones de este diagrama he seguido los siguientes criterios:
 - Las relaciones N:M se convierten a tabla, teniendo como atributos los propios de la relación y los atributos claves de las entidades que relacionan. Tendrán como clave primaria la concatenación de los atributos principales de cada una de las entidades que relacionan que serán clave ajena respecto a cada una de las tablas donde ese atributo es clave primaria. En este caso tenemos la relación “convoca”.
 - Además, en este ejercicio encontramos la relación ternaria “oposita” que al ser del tipo N:M:1, debemos crear la tabla correspondiente con las claves principales de las entidades que relaciona y tendrá como clave principal la concatenación de las claves principales de cada una de las entidades que se relacionan con cardinalidad muchos, éstas actuarán como clave ajena y le añadiremos el atributo “NotaFinal” propio de la relación.

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

CONVOCATORIA (id, descripción, año)

CONVOCA (idConvocatoria, CódigoEspecialidad, NumPlazas)

ESPECIALIDAD (código, nombre)

OPOSITA (idConvocatoria, NumOpositor, CódigoEspecialidad, NotaFinal)

OPOSITOR (NumOpositor, DNI, Nombre, Apellidos)

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

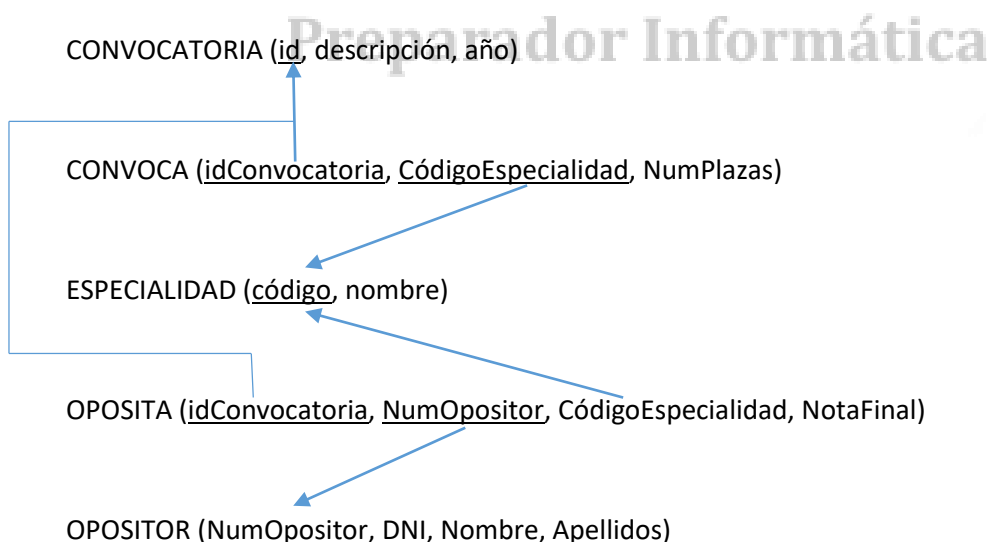
Una relación está en 3FN si y solo si está en 2FN y todos sus atributos no clave dependen no transitivamente de la clave primaria. Dicho de otro modo, si y sólo si los atributos no clave son mutuamente independientes y son dependientes por completo de la clave primaria.

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.

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.



SQL. Escribe las sentencias SQL necesarias para:

- Crear y usar la base de datos “Oposiciones” usando codificación utf8.
- Crear al menos dos tablas del modelo anteriormente desarrollado. Selecciona el tipo de dato que consideres más eficiente para almacenar los datos.
- Establecer las claves primarias de las tablas anteriores.

(Creamos la base de datos que nos pide)

```
CREATE DATABASE Oposiciones CHARACTER SET utf8;  
USE Oposiciones;
```

(El examen pedía al menos dos tablas, nosotros hemos hecho todas las tablas)

```
CREATE TABLE convocatoria  
(  
    id VARCHAR(10) PRIMARY KEY,  
    descripcion VARCHAR(100) NOT NULL,  
    año YEAR  
);
```

```
CREATE TABLE especialidad  
(  
    codigo INT PRIMARY KEY,  
    nombre VARCHAR(30) NOT NULL  
);
```

```
CREATE TABLE opositor  
(  
    NumOpositor INT PRIMARY KEY,  
    DNI INT NOT NULL,  
    nombre VARCHAR(20) NOT NULL,  
    apellidos VARCHAR(40) NOT NULL  
);
```



```
CREATE TABLE convoca
(
    idConvocatoria VARCHAR(10) NOT NULL,
    CodigoEspecialidad INT NOT NULL,
    NumPlazas SMALLINT UNSIGNED NOT NULL,
    CONSTRAINT PKconvoca PRIMARY KEY (idConvocatoria,
    CodigoEspecialidad),
    CONSTRAINT FKconv1 FOREIGN KEY (idConvocatoria)
    REFERENCES convocatoria(id) ON UPDATE CASCADE ON DELETE
    CASCADE,
    CONSTRAINT FKesp1 FOREIGN KEY (CodigoEspecialidad)
    REFERENCES especialidad(codigo) ON UPDATE CASCADE ON
    DELETE CASCADE
);
```

```
CREATE TABLE oposita
(
    idConvocatoria VARCHAR(10) NOT NULL,
    NumOpositor INT NOT NULL,
    CodigoEspecialidad INT NOT NULL,
    NotaFinal DOUBLE UNSIGNED NOT NULL,
    CONSTRAINT PKoposita PRIMARY KEY (idConvocatoria,
    NumOpositor),
    CONSTRAINT FKconv2 FOREIGN KEY (idConvocatoria)
    REFERENCES convocatoria(id) ON UPDATE CASCADE ON DELETE
    CASCADE,
    CONSTRAINT FKopositor FOREIGN KEY (NumOpositor)
    REFERENCES opositor(NumOpositor) ON UPDATE CASCADE ON
    DELETE CASCADE,
    CONSTRAINT FKesp2 FOREIGN KEY (CodigoEspecialidad)
    REFERENCES especialidad(codigo) ON UPDATE CASCADE ON
    DELETE CASCADE
);
```