

Actividad de aprendizaje 3-05

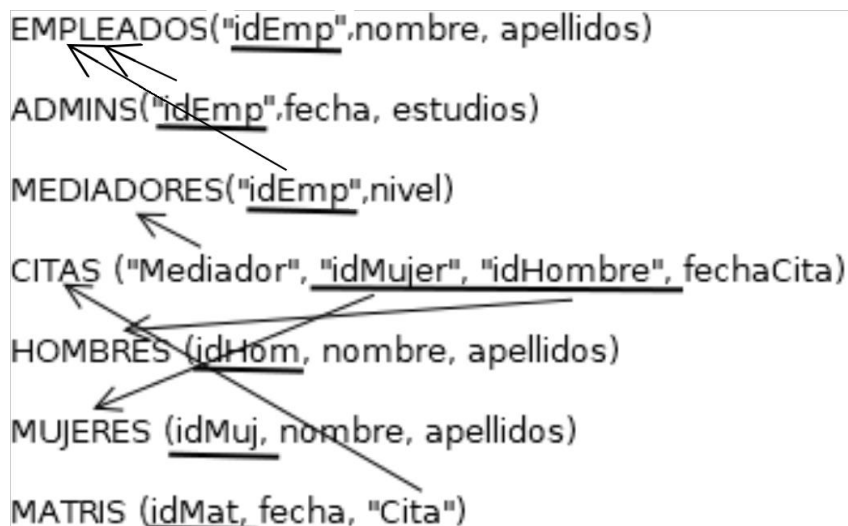
Se trata de:

- Realizar el diseño físico de una base de datos para gestionar una agencia matrimonial a partir de su diseño lógico y de la descripción de funcionalidad.

Objetivos:

- Interpretar el diseño lógico de una base de datos para implementar su diseño físico.
- Crear las tablas a partir del diseño lógico.
- Definir columnas y tipos de datos a partir del diseño lógico y de la descripción de funcionalidad.
- Establecer restricciones de columnas a partir del diseño lógico y de la descripción de funcionalidad.
- Comprobar que una base de datos se ha implementado correctamente.

Procedimiento



A partir del diseño lógico mostrado anteriormente debes tener en cuenta las siguientes restricciones:

- En la tabla empleados los nombres y apellidos deben ser obligatorios. Ambos deben ser cadenas de caracteres. El Id será numérico y auto incrementado.
- En la tabla Admins estudios es una cadena de caracteres. Fecha es obligatorio.
- En la tabla mediadores nivel es un valor numérico obligatorio con valor por defecto 0.
- En la tabla citas fechaCita será obligatorio.
- Las tablas hombres y mujeres tendrán los mismos tipos de datos que la tabla empleados.
- En la tabla Matris, idMat será un valor numérico obligatorio y autoincremental.

- Todas las claves ajenas no deben permitir borrado ni modificación en caso de ser referenciadas en otra tabla salvo las relacionadas con idEmp que serán en cascada en ambos casos.

Escribe las instrucciones SQL necesarias para:

1.- Crear la base de datos.

```
create database empresa;
```

2.- Crear cada una de las tablas.

```
create table empleados (idEmp INT NOT NULL AUTO_INCREMENT, nombre  
VARCHAR(25) NOT NULL, apellidos VARCHAR(25) NOT NULL, PRIMARY KEY (idEmp));
```

```
create table admins (idEmp INT NOT NULL, fecha DATE NOT NULL, estudios  
VARCHAR(25), PRIMARY KEY (idEmp), CONSTRAINT fk_admins_emp FOREIGN KEY  
(idEmp) REFERENCES empleados(idEmp) ON DELETE CASCADE ON UPDATE CASCADE);
```

```
create table mediadores (idEmp INT NOT NULL, nivel TINYINT, PRIMARY KEY (idEmp),  
CONSTRAINT fk_meds_emp FOREIGN KEY (idEmp) REFERENCES empleados(idEmp) ON  
DELETE CASCADE ON UPDATE CASCADE);
```

```
create table hombres (idHom INT NOT NULL AUTO_INCREMENT, nombre  
VARCHAR(25) NOT NULL, apellidos VARCHAR(25) NOT NULL, PRIMARY KEY (idHom));
```

```
create table mujeres (idMuj INT NOT NULL AUTO_INCREMENT, nombre VARCHAR(25)  
NOT NULL, apellidos VARCHAR(25) NOT NULL, PRIMARY KEY (idMuj));
```

```
create table citas (Mediador INT NOT NULL, idMujer INT NOT NULL, idHombre INT NOT  
NULL, fechaCita DATE NOT NULL, PRIMARY KEY (idMujer, idHombre), CONSTRAINT  
fk_meds_citas FOREIGN KEY (Mediador) REFERENCES mediadores(idEmp) ON DELETE  
CASCADE ON UPDATE CASCADE, CONSTRAINT fk_hombres_citas FOREIGN KEY  
(idHombre) REFERENCES hombres(idHom), CONSTRAINT fk_mujeres_citas FOREIGN  
KEY (idMujer) REFERENCES mujeres(idMuj));
```

```
create table matris (idMat INT NOT NULL AUTO_INCREMENT, fecha DATE, idMujer INT  
NOT NULL, idHombre INT NOT NULL, PRIMARY KEY (idMat), CONSTRAINT  
fk_citas_matris1 FOREIGN KEY (idMujer,idHombre) REFERENCES  
citas(idMujer,idHombre));
```

3.- Añadir en la tabla de las citas dos columnas que permitan guardar la valoración de la cita por parte de los implicados.

```
alter table citas ADD valoracion_muj INT;
```

`alter table citas ADD valoracion_hom INT;`

4.- En la tabla citas crea un índice con la fecha.

`alter table citas ADD INDEX ind_fecha (fechaCita);`

5.- En la tabla matris el conjunto fecha+cita no se puede repetir.

`alter table matris ADD UNIQUE uni_matris (fecha,idMujer,idHombre);`