

## Práctica Modelo de Examen: DEPARTAMENTOS y EDIFICIOS

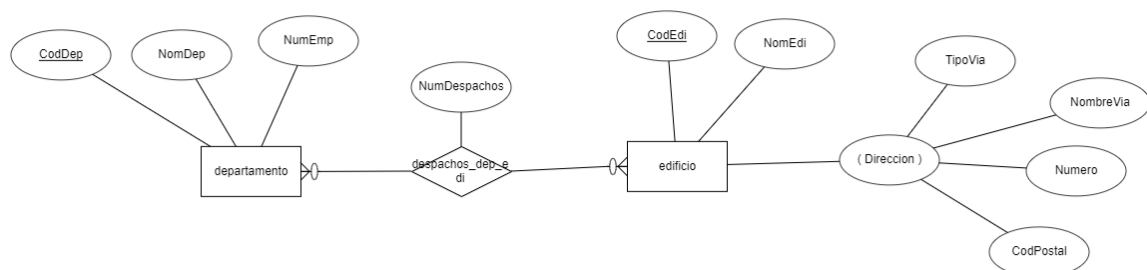
- Una empresa de Málaga desea registrar la distribución de sus departamentos. Un departamento puede estar distribuido en varios edificios. Del departamento tenemos su código, nombre y el número de empleados que lo integran. De los edificios sabemos su código, nombre, dirección completa (tipo vía, nombre vía, número, código postal) y el número de despachos que tienen ocupados. En cada edificio pueden ubicarse distintos departamentos. Debemos poder registrar el número de despachos que cada departamento tiene en cada edificio.

Debes tener en cuenta:

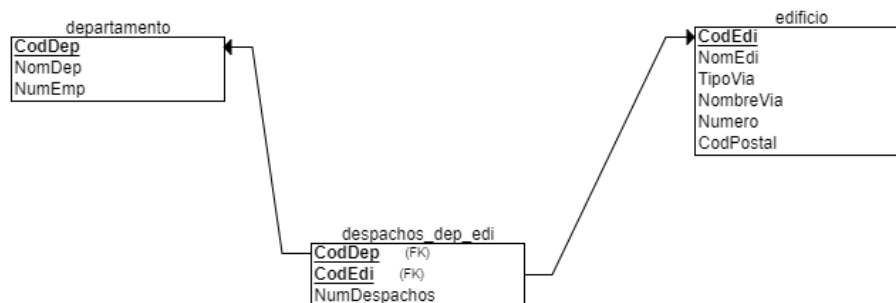
- Podemos tener dado de alta un departamento y aún no haber indicado en que edificios se encuentra.
- Podemos tener dados de alta un edificio y aún no haber indicado los departamentos que allí se ubican.
- Ninguna Foreign Key podrá admitir nulos.
- Si detectas algún atributo calculado no debes ponerlo. Recuerda que un atributo calculado es aquel que su valor se podría obtener mediante una consulta SELECT. Además como el resultado de esa consulta podría variar a lo largo del tiempo, habría que encargarse de mantenerlo siempre actualizado.

Se pide:

- Realizar el Diagrama Entidad – Relación extendido. Utilizar la notación adecuada para que el modelo sea lo más expresivo posible. (Realizar en papel)



- Convertir el modelo anterior a un modelo relacional (DED). Indicar los atributos que son clave primaria y los que son claves foráneas. Indica los atributos que no admiten nulo poniendo NN y los que sí admiten poniendo NULL. (Realizar en papel). (Antes de entregar esta primera parte, debes quedarte con una copia de este apartado)



2. Crear una base de datos. Crea todas las tablas obtenidas en el ejercicio anterior. Deben cumplir lo siguiente:

- En cada tabla debes definir la clave primaria.
- El código del edificio debes definirlo para que se autoincremente automáticamente empezando en 10 y se incremente de 10 en 10.
- Independientemente de cómo lo hayas modelado anteriormente, debes poner la Primary Key compuesta en la entidad obtenida del tipo de correspondencia muchos a muchos. Llama a esta tabla "despachos\_dep\_edi".
- Las clave foránea debes definirla en la sentencia de creación de la tabla, junto con sus correspondientes reglas de modificación y borrado. Decide tú cuales pones y justifica en un comentario tu decisión.
- Ningún campo admite valor nulo.
- Por defecto el número de despachos ocupados en un edificio por un departamento debe tomar el valor 1.
- Utiliza las restricciones adecuadas para verificar que el atributo tipo vía toma alguno de estos valores: Calle, Avenida, Bulevar o Pasaje.

```
CREATE DATABASE EDIFICIO
```

```
GO
```

```
USE EDIFICIO
```

```
GO
```

```
CREATE TABLE departamento
```

```
(  
    CodDep INT NOT NULL,  
    NomDep VARCHAR(50) NOT NULL,  
    NumEmp SMALLINT NOT NULL,  
    PRIMARY KEY (CodDep)  
);
```

```
CREATE TABLE edificio
```

```
(  
    CodEdi INT NOT NULL IDENTITY(10,10),  
    NomEdi VARCHAR(50) NOT NULL,  
    TipoVia VARCHAR(7) NOT NULL CHECK (TipoVia IN ('Calle', 'Avenida', 'Bulevar', 'Pasaje')),  
    NombreVia VARCHAR(80) NOT NULL,  
    Numero SMALLINT NOT NULL,  
    CodPostal CHAR(9) NOT NULL,  
    PRIMARY KEY (CodEdi)  
);
```

```
CREATE TABLE despachos_dep_edi
```

```
(  
    CodEdi INT NOT NULL,  
    CodDep INT NOT NULL,  
    NumDespachos TINYINT NOT NULL DEFAULT 1,  
    PRIMARY KEY (CodEdi, CodDep),  
    FOREIGN KEY (CodEdi) REFERENCES edificio(CodEdi) ON DELETE NO ACTION ON UPDATE NO ACTION,  
    FOREIGN KEY (CodDep) REFERENCES departamento(CodDep) ON DELETE CASCADE ON UPDATE NO ACTION  
);
```

```
/*Supongamos que después de consultar con la persona que aporta la especificación de la  
aplicación nos ha dicho que debe funcionar de la siguiente manera:*/
```

```
/*FK CoDep: ON DELETE CASCADE
```

```
Si borramos el departamento deben borrarse automáticamente todos los depachos que tenga  
ocupado  
en los diferentes edificios donde se ubique*/
```

```
/*FK CodDep: ON UPDATE NO ACTION
```

```
Si se decidiera modificar el CodDep en la tabla departamento y este tiene despachos en algún  
edificio, no se va a permitir la modificación*/
```

```
/*FK CodEdi: ON DELETE NO ACTION
```

```
Si se intenta eliminar un edificio que tiene despachos de algún departamento  
no se debe permitir, se debe mostrar un mensaje al usuario indicándole  
que el edificio que intenta elimniar tiene despachos y por tanto, para poder eliminar  
el edificio debe reubicar los departamentos en otros edificios.*/
```

```
/*FK CodEdi: ON UPDATE NO ACTION
```

```
Si se decidiera modificar el CodEdi en la tabla edificio y este tiene despachos de algún departamento, no se va a permitir la modificación*/
```

3. Utilizando sentencias del LMD de SQL (solo tres sentencias INSERT), inserta filas en las tablas de manera que tengamos: Cuatro departamentos y dos edificios. En la tabla depachos\_dep\_edi los dos primeros departamentos tienen despachos tanto en el primer como en el segundo edificio, y el tercer departamento solo tiene despachos en el segundo edificio.

Los despachos están distribuidos así:

- primer edificio – primer departamento – 2 despachos
- primer edificio – segundo departamento – 3 despachos
- segundo edificio – primer departamento – 1 despacho
- segundo edificio – segundo departamento – 4 despachos
- segundo edificio – tercer departamento – 1 despacho

```
INSERT INTO departamento
VALUES (1, 'Dep1', 6), (2, 'Dep2', 15), (3, 'Dep3', 2), (4, 'Dep4', 1)
```

```
INSERT INTO edificio
VALUES ('Edificio1', 'Calle', 'Álamo', 10, 29006),
      ('Edificio2', 'Avenida', 'Olmo', 20, 29010),
      ('Edificio3', 'Bulevar', 'Peral', 30, 29590)
```

```
INSERT INTO despachos_dep_edi
VALUES(10,1,2), (10,2,3), (20,1,1), (20,2,5), (20,3,1)
```

4. Utilizando la sentencia adecuada del LMD de SQL borra (elimina filas completas) los despachos del segundo edificio que has introducido si cumple que el departamento que se ubica allí ocupa solo un despacho. Solo conoces el nombre del edificio. La sentencia debe estar dentro de una transacción y cuando compruebes que se han borrado los despachos correctamente, debes deshacerla.

```
BEGIN TRANSACTION
```

```
DELETE
FROM despachos_dep_edi
WHERE NumDespachos=1 AND CodEdi = (SELECT CodEdi FROM edificio WHERE NomEdi LIKE
'Edificio2')
```

```
DELETE FROM despachos_dep_edi
FROM despachos_dep_edi DE JOIN edificio e ON (DE.CodEdi=E.CodEdi)
WHERE NumDespachos=1 AND NomEdi LIKE 'Edificio2'
```

```
SELECT * FROM despachos_dep_edi
```

```
ROLLBACK TRANSACTION
```

5. Utilizando la sentencia adecuada del LMD de SQL borra un departamento que tenga algún despacho en un edificio y comprueba que la regla que has puesto de eliminación funciona correctamente. Explica en un comentario que ha ocurrido según la regla que elegiste. La sentencia debe estar dentro de una transacción y cuando hayas hecho la comprobación, debes deshacerla.

```
BEGIN TRANSACTION
```

```
SELECT * FROM departamento
SELECT * FROM despachos_dep_edi
```

```
DELETE
FROM departamento
WHERE CodDep=1
```

```
ROLLBACK TRANSACTION
```

```
/*Se ha eliminado el departamento 1 y los depachos que tenía en los edificios 10 y 20*/
```

6. Utilizando la sentencia adecuada del LMD de SQL modifica el número de empleados del primer departamento introducido. Solo conoces el nombre del departamento.

```
UPDATE departamento
SET NumEmp=7
WHERE NomDep LIKE 'Dep1'

SELECT * FROM departamento
```

7. Realiza las siguientes consultas:

- A. Obtener cuantos despachos tiene ocupado cada edificio. Debe aparecer el nombre del edificio y la cantidad de despachos que tiene ocupados por los diferentes departamentos (el total). Utiliza la función de agregado SUM y alias para las columnas.

```
/*SELECT *
FROM edificio E JOIN despachos_dep_edi D ON (E.CodEdi=D.CodEdi)*/

SELECT NomEdi AS 'Nombre del Edificio', SUM(NumDespachos) AS 'Numero de Despachos Ocupados'
FROM edificio E JOIN despachos_dep_edi D ON (E.CodEdi=D.CodEdi)
GROUP BY NomEdi

/* Ten en cuenta que en el enunciado dice "De los edificios sabemos su código, nombre,
dirección
completa (tipo vía, nombre vía, número, código postal) y el número de despachos que tienen
ocupados."
El número de despachos que tienen ocupados es el atributo calculado que no debías poner.
Aquí ves que este dato se obtiene con una consulta*/
```

- B. Obtener el nombre de cada departamento junto con el nombre del edificio donde tiene despachos, así como el número de despachos que ocupa. Si a un departamento aún no se le ha asignado despachos en un edificio, el nombre del departamento debe salir igualmente, y en el nombre del edificio debe aparecer "edificio sin asignar" y en el número de despachos aparecerá 0. Utiliza alias para el nombre de las columnas.

```
/*
SELECT *
FROM (departamento D LEFT JOIN despachos_dep_edi DE ON (D.CodDep=DE.CodDep))
LEFT JOIN edificio E ON (E.CodEdi=DE.CodEdi)
*/

SELECT NomDep AS 'Departamento', ISNULL(NomEdi,'edificio sin asignar') AS
'Edificio',ISNULL(NumDespachos,0) AS 'Numero de despachos'
FROM (departamento D LEFT JOIN despachos_dep_edi DE ON (D.CodDep=DE.CodDep))
LEFT JOIN edificio E ON (E.CodEdi=DE.CodEdi)
```