

UNIVERSIDAD AUTÓNOMA METROPOLITANA
UNIDAD IZTAPALAPA



Casa abierta al tiempo

Práctica Semana 8

Profesor
Dr. Orlando Muñoz Texzocotetla

NOMBRE DE LOS INTEGRANTES

Carreón González Saúl Horacio
Martínez Buenrostro Jorge Rafael
Morales Hernández José Manuel

Bases de Datos

Actividades

Antes de comenzar con las actividades de esta semana mostraremos el diagrama que tenemos originalmente para poder ver cómo cambia después de todas las actividades



1. Investigar cómo se lleva a cabo la forma normal Boyce-Codd y dar 2 ejemplos en el sistema en el que están trabajando en equipo.

La forma normal de Boyce-Codd es una versión más estricta de la tercera forma normal (3NF). Se utiliza en la normalización de bases de datos con el objetivo de eliminar todas las dependencias funcionales no triviales de los atributos que no sean un conjunto de la clave candidata. A continuación, veremos los pasos a seguir para realizar esta normalización:

- 1) **Cumplir con la tercera forma normal (3NF):** Todas las dependencias transitivas deben eliminarse y todas las columnas no clave deben depender completamente de la clave primaria.
- 2) **Cumplir con la condición de Boyce-Codd:** Establece que, para cada dependencia funcional no trivial $X \rightarrow Y$ en la tabla (donde X y Y son conjuntos de atributos), X debe ser una superclave de la tabla, es decir, si X determina Y, entonces X debe ser suficiente para identificar de manera única cada fila de la tabla.

En el caso de nuestras tablas ya cumplen con la forma normal Boyce-Codd por lo que modificaremos las tablas para crear dos ejemplos en los que esta forma normal nos puede ayudar a normalizar las tablas

Ejemplo 1

empleadoBC	
id	int
nombre	varchar
ap_paterno	varchar
ap_materno	varchar
fecha_nacimiento	date
direccion	varchar
fecha_alta	date
fecha_baja	date
salario	float
id_puesto	int
nombre_puesto	varchar

Con esta tabla tenemos dos dependencias:

$id \rightarrow \text{nombre}, \text{ap_paterno}, \dots, \text{salario}$

$id_puesto \rightarrow \text{nombre} - \text{puesto}$

En esta tabla *id* es una llave primaria lo que significa que los atributos: *nombre*, *ap_paterno*, ..., *salario* son atributos primarios. Mientras que *id-puesto* es un atributo primario, *nombre-puesto* no es un atributo no primario, lo que no cumple con las reglas de la normalización de Boyce-Codd

Para poder cumplir con las condiciones de la normalización de Boyce-Codd, descompondremos la tabla *empleadoBC* en dos tablas: *empleado* y *puesto*. Como se puede ver en la siguiente imagen.

empleadoBC		puestoBC	
id	int	id_puesto	int
nombre	varchar	nombre_puesto	varchar
ap_paterno	varchar		
ap_materno	varchar		
fecha_nacimiento	date		
direccion	varchar		
fecha_alta	date		
fecha_baja	date		
salario	float		

Ejemplo 2

proyectoBC	
id	int
descripcion	text
fecha_inicio	date
fecha_fin	date
id_departamento	int
departamento	varchar
presupuesto	double

Con esta tabla tenemos dos dependencias:

$id \rightarrow \text{descripcion}, \text{fecha_inicio}, \text{fecha_fin}$
 $id_departamento \rightarrow \text{departamento}, \text{presupuesto}$

En esta tabla *id* es una llave primaria lo que significa que los atributos: *descripción*, *fecha_inicio* y *fecha_fin* son atributos primarios. Mientras que *id_departamento* es un atributo primario, *departamento* y *presupuesto* no es un atributo no primario, lo que no cumple con las reglas de la normalización de Boyce-Codd

Para poder cumplir con las condiciones de la normalización de Boyce-Codd, descompondremos la tabla *proyectoBC* en dos tablas: *proyecto* y *departamento*. Como se puede ver en la siguiente imagen.

proyectoBC		departamentoBC	
id	int	id_departamento	int
descripcion	text	departamento	varchar
fecha_inicio	date	presupuesto	double
fecha_fin	date		

2. Aplicar las primeras 4 formas normales en el sistema que están trabajando en equipo.

- **Primera forma normal (1NF)**, los valores en cada columna de una tabla son atómicos.

empleado	
id	int
nombre	varchar
apellidos	varchar
fecha_nacimiento	date
direccion	varchar
fecha_alta	date
fecha_baja	date
salario	float
departamento	int
puesto	int

Para la tabla empleado podemos ver que el único campo que no es atómico es apellidos por lo que necesitamos “separar” sus valores en dos campos nuevos **ap_paterno** y **ap_materno**. En este caso el campo nombre aunque una persona puede tener más de un nombre en nuestro contexto podemos considerar como atómico todos los nombres de una persona

departamento	
id	int
nombre	varchar
descripcion	text
presupuesto	int

Como podemos ver todos los valores de los campos de la tabla **departamento** son atómicos, es decir, esta tabla ya está en **1NF**

proyecto	
id	int
descripcion	text
fecha_inicio	date
fecha_fin	date
presupuesto	float
departamento	int

La tabla **proyecto** no cuenta con valores que no sean atómicos, por lo que podemos decir que ya está en **1NF**.

empleadoproyecto	
id	int
empleado	int
proyecto	int

Esta tabla es una tabla relación entre **empleado** y **proyecto**. Por lo que sus atributos son totalmente atómicos. Esto nos dice que ya se encuentra en la **1NF**

perfil	
id	int
empleado	int
rol	int
usuario	int

Los valores de esta tabla son atómicos, es decir, se encuentra en la **1NF**

usuario	
id	int
nombre	varchar
correo	varchar
contraseña	varchar

Los valores de esta tabla son atómicos, es decir, se encuentra en la **1NF**. En este caso el nombre de usuario puede ser un nickname, no necesariamente es el nombre completo del empleado lo que nos permite decir que es atómico.

puesto	
id	int
nombre	varchar
descripcion	text
salario_base	float
departamento	int

Los valores de esta tabla son atómicos, es decir, se encuentra en la **1NF**

- **Segunda forma normal (2NF)**, debemos asegurarnos de que todas las columnas no clave dependan completamente de la clave principal.

Para esta forma normal podemos que todas las tablas que solo tengan un solo atributo como llave primaria cumplen con las condiciones.

- **Tercera forma normal (3NF)**, debemos asegurarnos de que no haya dependencias transitivas entre las columnas no clave.

Para esta forma normal buscamos que no haya dependencia entre atributos no clave, como podemos ver en los diagramas, ninguna tabla presenta este tipo de dependencias por lo que podemos decir que todas las tablas están en la tercera forma normal **3NF**

- **Cuarta forma normal (4NF)**, debemos asegurarnos de que no haya dependencias multivaluadas independientes.

Para esta forma normal se busca que no haya dependencias multivaluadas, una dependencia multivaluada ocurre cuando un atributo no clave depende de dos o más atributos clave, de forma que puede tener varios valores posibles para cada combinación de claves. Como podemos ver en las tablas no tenemos este tipo de dependencias dentro de las tablas, por lo que podemos decir que todas las tablas se encuentran en **4NF**

empleado	
id	int
nombre	varchar
ap_paterno	varchar
ap_materno	varchar
fecha_nacimiento	date
direccion	varchar
fecha_alta	date
fecha_baja	date
salario	float
departamento	int
puesto	int

proyecto	
id	int
descripcion	text
fecha_inicio	date
fecha_fin	date
presupuesto	float
departamento	int

empleadoproyecto	
id	int
empleado	int
proyecto	int

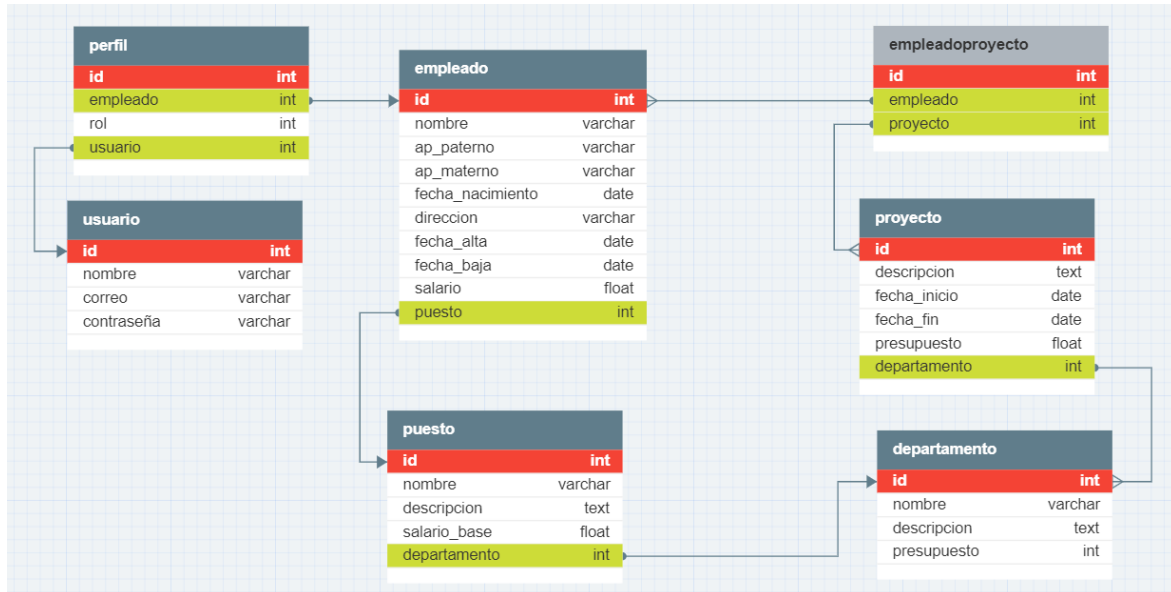
perfil	
id	int
empleado	int
rol	int
usuario	int

usuario	
id	int
nombre	varchar
correo	varchar
contraseña	varchar

puesto	
id	int
nombre	varchar
descripcion	text
salario_base	float
departamento	int

3. Implementar las interfaces gráficas, interfaces (clases abstractas), clases y métodos necesarios para aplicar el patrón DAO a 3 entidades del sistema trabajado en equipo. En esta parte sólo implementar la inserción, pero el diseño de la interface del menú debe contemplar el borrado y la actualización de esos datos.

Primero mostraremos el diagrama de la base final del sistema



Se seleccionaron las entidades:

- Usuario
- Empleado
- Proyecto