Facultad de Ciencias
UNAM

Fundamentos de Bases de datos

Gerardo Avilés Rosas

PROYECTO FINAL

CENTRO DE CONTACTO

Uribe García Zurisadai	318223197
Rodríguez García Ulises	318042202
Ramírez Plascencia Tania	318038830
Illescas Coria Janet	318219309
Cruz Hernández Victor Emiliano	318081166

Diciembre 07, 2022

Utilizamos SMBD PostgreSQL para la implementación de la base de datos en este proyecto.

Justificación del modelo E/R

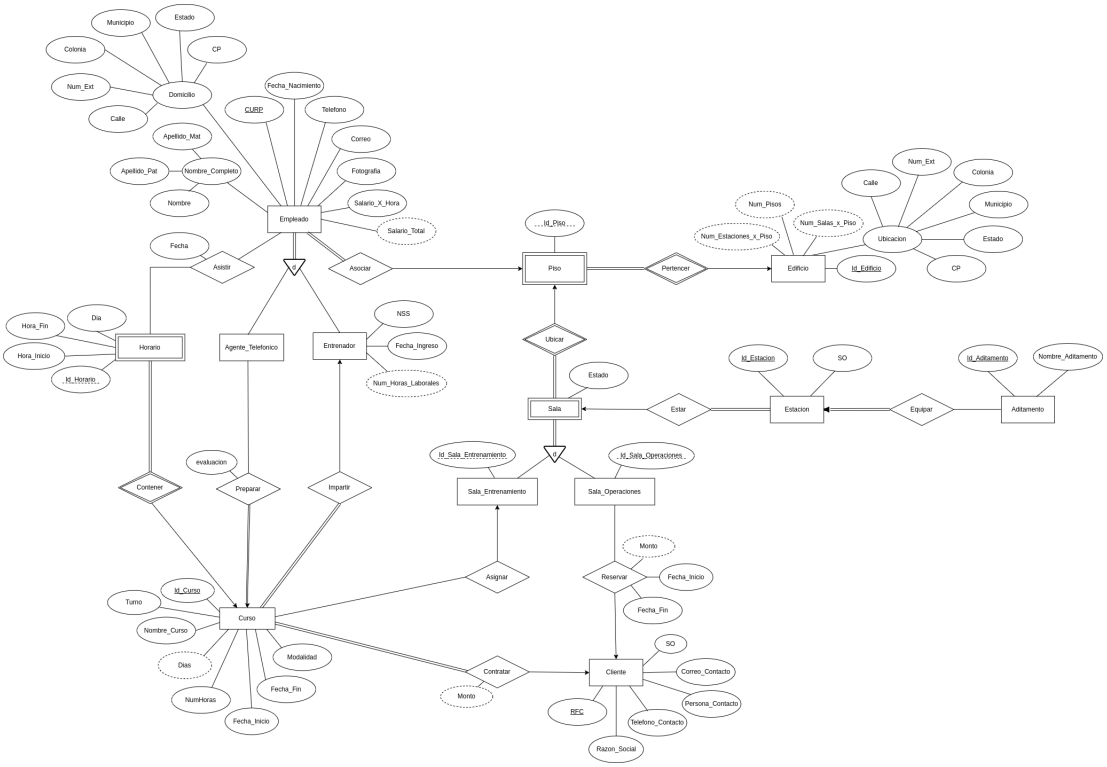

Figura 1: Modelo Entidad Relación

Entidades

- **Empleado:** Nos piden guardar la información relevante tanto de los entrenadores como de los agentes telefónicos, sin embargo notamos que la información que nos piden de ambos es la misma por lo que creamos una entidad *Empleado* que tiene como atributos los datos solicitados. Es importante mencionar que agregamos como atributo calculado $Salario\_Total$ ya que este puede variar con el tiempo y se puede obtener el resultado mediante el valor del atributo $Salario\_X\_Hora$.

- **Agente_Telefonico:** Como mencionamos en el punto anterior, nos interesa almacenar la información de estos empleados, a pesar de que en la entidad *Empleado* tenemos su información general es necesario especificar el rol que desempeñan.

- **Entrenador:** Análogamente al punto anterior, tenemos que especificar que algunos empleados desarrollan el rol de entrenador por lo que creamos una entidad *Empleado* para poder cumplir con dicho propósito. Adicional a esto, nos piden almacenar datos únicamente pertenecientes a este tipo de empleado, por lo que esta entidad tiene como atributos la información específica que se solicita, de igual manera agregamos como atributo calculado $Num\_Horas\_Laborales$ ya que dicho valor cambia constantemente.

- **Horario:** Nos interesa saber el día y horas de entrada/salida de cada empleado a su piso de operación, por lo que creamos la entidad *Horario* que tiene como atributos estos datos mencionados. Consideramos que es una entidad débil ya que no puede existir un horario sin un curso.

- **Curso:** Tenemos que la empresa imparte cursos de capacitación para preparar a los agentes telefónicos y nos interesa almacenar la información de cada uno, para lograr esto creamos la entidad *Cursos* que tiene como atributos los aspectos de interés de cada uno de ellos (nombre del curso, modalidad, entre otros). Agregamos como atributos calculados *dias*, *duracion* ya que estos datos se pueden obtener a partir de la información guardada en *Fecha_inicio* y *Fecha_Fin*.

- **Cliente:** De igual manera, necesitamos almacenar los datos de los clientes que hacen uso de los servicios del Centro de Contacto, por lo que diseñamos una entidad *Cliente* que contiene como atributos la información que nos piden almacenar.

- **Edificio:** Sabemos que el Centro de Contacto tiene a su disposición diversos edificios a lo largo de la CDMX, así que agregamos una entidad *Edificio* que tiene como atributos los datos que nos interesan saber de cada uno de estos (como dirección e ID del edificio). Adicional a esto, necesitamos saber el número de pisos, número de estaciones por piso y número de salas por piso de cada edificio, sin embargo esta información la podemos calcular a través de otras relaciones por lo que decidimos agregar estos atributos como calculados.

- **Piso:** Necesitamos llevar un control de cada piso y lo que en él se encuentra, para lograr esto decidimos añadir la entidad *Piso*; es importante destacar que es una entidad débil ya que no puede existir un piso sin un edificio.

- **Sala:** Análogamente al punto anterior, necesitamos tener un control de cada sala (si está ocupada o no) y saber de qué tipo de sala se habla por lo cual diseñamos una entidad débil *Sala* (es débil ya que no puede existir una sala sin un piso).

- **Sala_Entrenamiento:** Partiendo del punto anterior nos damos cuenta que las salas tienen algo en común, esto es el estado en el que se encuentran (ocupada o desocupada), sin embargo tenemos que existen dos tipos de sala. Para hacer esta distinción decidimos crear una entidad *Sala_Entrenamiento*.

- **Sala_Operaciones:** Es análogo al punto anterior.

- **Estacion:** Sabemos que cada edificio tiene un número determinado de estaciones, las cuales son computadoras de escritorio, nos interesa almacenar el sistema operativo de estas por lo que creamos una entidad *Estacion* que tiene como atributo este dato.

- **Aditamento:** Nos piden inventariar los elementos con los que está equipada cada estación, para lo cual creamos la entidad *Aditamento*.

Relaciones

- **Asistir:** Cada empleado tiene un horario en el cuál tiene que presentarse a trabajar, por lo que creamos la relación *Asistir* la cual tiene como atributo $fecha$ ya que el horario cambia dependiendo de la fecha.
  Tenemos que es una relación muchos a muchos ya que un empleado puede tener varios horarios (siempre y cuando no se superpongan) y un horario puede tener varios empleados.

- **Asociar:** Cada empleado tiene asignado un piso por lo que diseñamos la relación *Asociar* para simular esto, es una relación uno a muchos puesto que un piso puede tener varios empleador pero un empleado sólo puede tener un piso asignado.

- **Contener:** Puesto que cada curso tiene un horario asignado creamos una relación débil *Contener* la cuál tiene como atributo calculado Num_Horas ya que esto se puede calcular con algunos datos almacenados en la entidad *Horario* pero varían de acuerdo al curso. Esta relación es uno a muchos con participación total ya que todo curso debe tener un horario mientras que en un horario se pueden impartir varios cursos.

- **Preparar:** Nos indican que un $Agente\_Telefonico$ se prepara tomando un curso, para simular esto creamos la relación *Preparar* con un atributo *evaluacion* ya que debemos tomar en cuenta la evaluación de cada agente en cada curso para así poder determinar si el empleado puede pasar a piso de operaciones o no. Esta relación es muchos a uno con participación total puesto que un agente telefónico sólo puede tomar un curso mientras que un curso puede ser tomado por varios agentes telefónicos, además es obligatorio que en un curso haya agentes telefónicos.

- **Impartir:** Los cursos son impartidos por los entrenadores, diseñamos la relación *Impartir* para simular esta acción, es una relación uno a muchos con participación total puesto que un curso sólo puede ser impartido por un entrenador pero un entrenador puede impartir varios cursos, además es obligatorio que un curso tenga un entrenador asignado.

- **Pertenecer:** Dado que un piso pertenece a un edificio agregamos la relación *Pertenecer*, la cuál es débil ya que la existencia del piso depende de la existencia del edificio. Por otro lado, tenemos que es una relación uno a muchos con participación total ya que todos los pisos deben pertenecer a un edificio mientras que un edificio puede tener muchos pisos.

- **Ubicar:** Las salas se ubican en un piso en específico así que agregamos la relación *Ubicar* para plasmar esto, es una relación débil ya que la existencia de la sala depende de la existencia del piso; además es una relación uno a muchos con participación total ya que todas las salas deben pertenecer a un sólo piso pero un piso puede tener varias salas.

- **Estar:** Sabemos que cada estación se encuentra en una sala por lo que diseñamos la relación *Estar* para simular este comportamiento, es una relación uno a muchos con participación total puesto que todas las estaciones deben pertenecer a una sola sala mientras que una sala puede tener varias estaciones.

- **Equipar:** Creamos la relación *Equipar* para hacer referencia que una estación es equipada con ciertos aditamentos, la diseñamos como una relación muchos a uno con participación total ya que una estación debe estar equipada con algunos aditamentos y un aditamento sólo puede pertenecer a una estación.

- **Asignar:** Agregamos la relación *Asignar* para modelar que un curso tiene asignada una sala de entrenamiento para poder ser impartido, esta relación es uno a muchos con participación total puesto que todo curso debe estar asignado a una sala de entrenamiento mientras que una sala de entrenamiento puede o no tener un curso asignado.

- **Contratar:** Tenemos que los cursos son contratados por un cliente, para modelar esto diseñamos la relación *Contratar* que tiene un atributo calculado $Monto$ ya que dependiendo de las horas que se contraten será el costo del curso contratado por el cliente. Es una relación uno a muchos con participación total debido a que todos los cursos deben ser contratados por un cliente mientras que un cliente puede o no contratar un curso.

- **Reservar:** De manera similar al punto anterior, un cliente puede reservar una sala de operación, para esto creamos la relación *Reservar* la cual contiene como atributos $Fecha\_Inicio$, $Fecha\_Fin$ y como atributo calculado $Monto$ ya que estos datos varían dependiendo de cada reservación. En particular, *Reservar* es una relación uno a muchos ya que un cliente puede reservar varias salas de operaciones pero una sala de operación sólo puede ser reservada por un cliente.

Traducción del modelo relacional

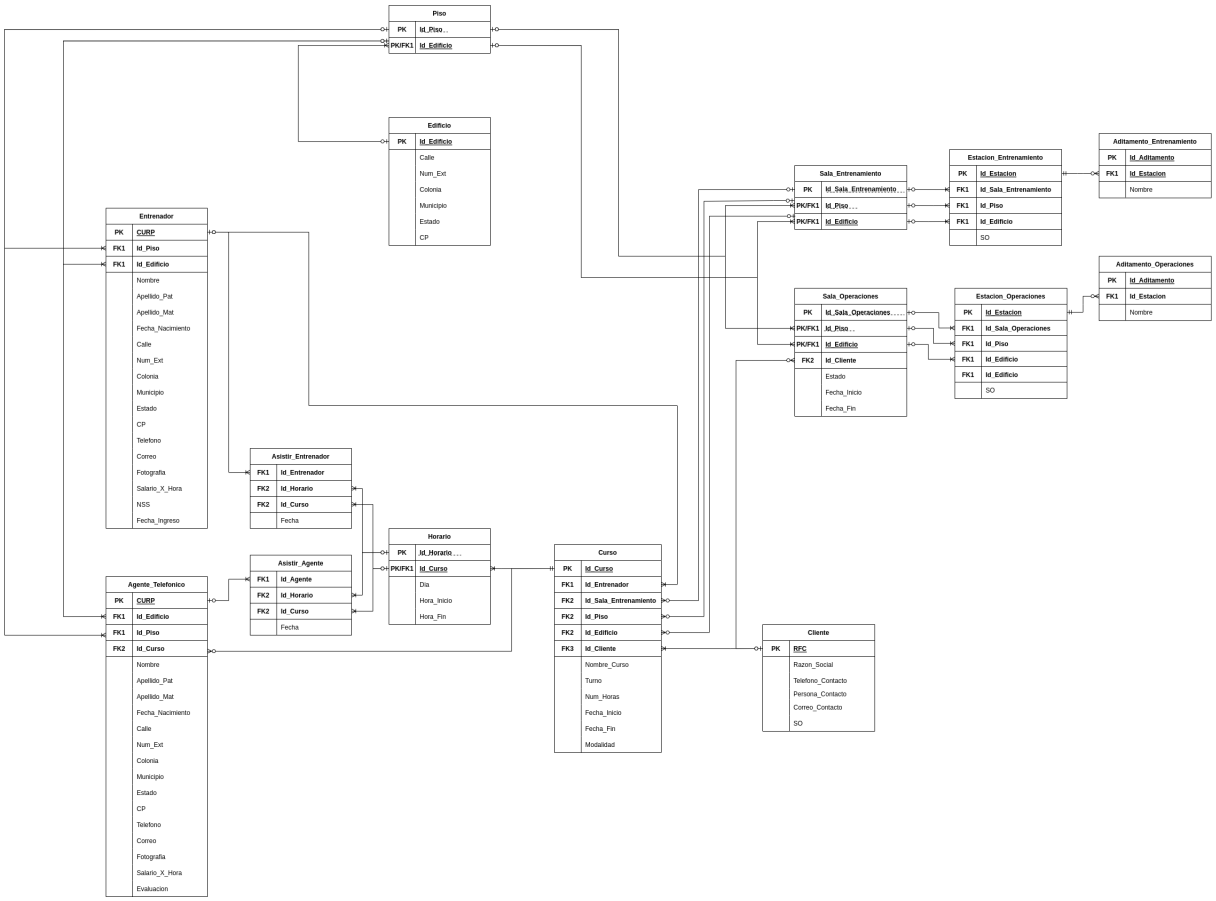

Figura 2: Modelo Relacional

Especificaciones y Restricciones

- **Entrenador**:
  - Llaves:
    - Llave primaria $CURP$.
    - Llave foránea Id_Piso, se relaciona con la llave Id_Piso de la tabla Piso.
    - Llave foránea $Id\_Edificio$, se relaciona con la llave $Id\_Edificio$ de la tabla Piso.
  - Dominio:
    - $Nombre$: Cadena de caracteres.
    - $Apellido\_Pat$: Cadena de caracteres.
    - $Apellido\_Mat$: Cadena de caracteres.
    - $Fecha\_Nacimiento$: Cadena de caracteres en el formato $aaaa - mm - dd$.
    - $Calle$: Cadena de caracteres.
    - Num_Ext : Campo numérico.
    - $Colonia$: Cadena de caracteres.

- $Municipio$: Cadena de caracteres.
- $Estado$: 'CDMX'.
- CP : Campo numérico de 5 dígitos.
- $Telefono$: Campo numérico.
- $Correo$: Cadena de caracteres.
- $Fotografia$: Cadena de caracteres.
- $Salario\_X\_Hora$: Campo numérico.
- NSS : Campo numérico.
- $Fecha\_Ingreso$: Cadena de caracteres en el formato $aaaa - mm - dd$.

- **Asistir_Entrenador:**
  - Llaves:
    - Llave foránea $Id\_Entrenador$, se relaciona con la llave $CURP$ de la tabla Entrenador.
    - Llave foránea $Id\_Horario$, se relaciona con la llave $Id\_Horario$ de la tabla Horario.
    - Llave foránea Id_Curso, se relaciona con la llave Id_Curso de la tabla Horario.
  - Dominio:
    - $Fecha$: Cadena de caracteres en el formato $aaaa - mm - dd$.

- **Agente_Telefonico**:
  - Llaves:
    - Llave primaria $CURP$
    - Llave foránea $Id\_Edificio$, se relaciona con la llave $Id\_Edificio$ de la tabla Piso.
    - Llave foránea Id_Piso, se relaciona con la llave Id de la tabla Piso.
    - Llave foránea Id_Curso, se relaciona con la llave Id_Curso de la tabla Curso.
  - Dominio:
    - $Nombre$: Cadena de caracteres.
    - $Apellido\_Pat$: Cadena de caracteres.
    - $Apellido\_Mat$: Cadena de caracteres.
    - $Fecha\_Nacimiento$: Cadena de caracteres en el formato $aaaa - mm - dd$.
    - $Calle$: Cadena de caracteres.
    - Num_Ext : Campo numérico.
    - $Colonia$: Cadena de caracteres.
    - $Municipio$: Cadena de caracteres.
    - $Estado$: Cadena de caracteres.
    - CP : Campo numérico de 5 dígitos.
    - $Telefono$: Campo numérico.
    - $Correo$: Cadena de caracteres.
    - $Fotografia$: Cadena de caracteres.
    - $Salario\_X\_Hora$: Campo numérico.
    - $Evaluacion$: Campo numérico.

- **Asistir_Agente :**
  - Llaves:
    - Llave foránea Id_Agente, se relaciona con la llave $CURP$ de la tabla Agente_Telefonico.
    - Llave foránea $Id\_Horario$, se relaciona con la llave $Id\_Horario$ de la tabla Horario.
    - Llave foránea Id_Curso, se relaciona con la llave Id_Curso de la tabla Horario.
  - Dominio:
    - $Fecha$: Cadena de caracteres en el formato $aaaa - mm - dd$.
- **Piso:**
  - Llaves:
    - Llave primaria Id_Piso
    - Llave primaria y foránea $Id\_Edificio$, se relaciona con la llave $Id\_Edificio$ de la tabla Edificio y la tabla Sala_Operaciones.
- **Edificio:**
  - Llaves:
    - Llave primaria $Id\_Edificio$
  - Dominio:
    - $Calle$: Cadena de caracteres.
    - Num_Ext : Campo numérico.
    - $Colonia$: Cadena de caracteres.
    - $Municipio$: Cadena de caracteres.
    - $Estado$: Cadena de caracteres.
    - CP : Campo numérico de 5 dígitos.
- **Horario:**
  - Llaves:
    - Llave primaria $Id\_Horario$.
    - Llave primaria y foránea Id_Curso, se relaciona con la llave Id_Curso de la tabla Curso.
  - Dominio:
    - Dia : Cadena de caracteres.
    - $Hora\_Inicio$: Cadena de caracteres en el formato $hh : mm : ss$.
    - $Hora\_Fin$: Cadena de caracteres en el formato $hh : mm : ss$.
- **Curso:**
  - Llaves:
    - Llave primaria Id_Curso.

- Llave foránea $Id\_Entrenador$, se relaciona con la llave $CURP$ de la tabla Entrenador.
- Llave foránea $Id\_Sala\_Entrenamiento$, se relaciona con la llave $Id\_Sala\_Entrenamiento$ de la tabla Sala_Entrenamiento.
- Llave foránea Id_Piso, se relaciona con la llave Id_Piso de la tabla Sala_Entrenamiento.
- Llave foránea $Id\_Edificio$, se relaciona con la llave $Id\_Edificio$ de la tabla Sala_Entrenamiento.
- Llave foránea $Id\_Cliente$, se relaciona con la llave $Id\_Cliente$ de la tabla Cliente.

- Dominio:
  - $Nombre\_Curso$: Cadena de caracteres.
  - $Turno$: Cadena de caracteres.
  - $Fecha\_Inicio$: Cadena de caracteres en el formato $aaaa - mm - dd$.
  - $Fecha\_Fin$: Cadena de caracteres en el formato $aaaa - mm - dd$.
  - $Modalidad$: Cadena de caracteres.
  - Num_Horas : Campo número.

- **Cliente:**
  - Llaves:
    - Llave primaria RFC.
  - Dominio:
    - $Razon\_Social$: Cadena de caracteres.
    - $Telefono\_Contacto$: Campo numérico.
    - $Persona\_Contacto$: Cadena de caracteres.
    - $Correo\_Contacto$: Cadena de caracteres.
    - SO : Cadena de caracteres.

- **Sala_Entrenamiento:**
  - Llaves:
    - Llave primaria $Id\_Sala\_Entrenamiento$.
    - Llave primaria y foránea Id_Piso, se relaciona con la llave Id_Piso de la tabla Piso.
    - Llave primaria y foránea $Id\_Edificio$, se relaciona con la llave $Id\_Edificio$ de la tabla Piso.

- **Estacion_Entrenamiento:**
  - Llaves:
    - Llave primaria $Id\_Estacion$.
    - Llave foránea $Id\_Sala\_Entrenamiento$, se relaciona con la llave $Id\_Sala\_Entrenamiento$ de la tabla Sala_Entrenamiento.
    - Llave foránea Id_Piso, se relaciona con la llave Id_Piso de la tabla Sala_Entrenamiento.

    - Llave foránea $Id\_Edificio$, se relaciona con la llave $Id\_Edificio$ de la tabla Sala_Entrenamiento.
  - Dominio:
    - SO : Cadena de caracteres.

- **Aditamento_Entrenamiento:**
  - Llaves:
    - Llave primaria $Id\_Aditamento$.
    - Llave foránea $Id\_Estacion$, se relaciona con la llave $Id\_Estacion$ de la tabla Estacion_Entrenamiento.
  - Dominio:
    - $Nombre$: Cadena de caracteres.

- **Sala_Operaciones:**
  - Llaves:
    - Llave primaria $Id\_Sala\_Operaciones$.
    - Llave primaria y foránea Id_Piso, se relaciona con la llave Id_Piso de la tabla Piso.
    - Llave primaria y foránea $Id\_Edificio$, se relaciona con la llave $Id\_Edificio$ de la tabla Piso.
    - Llave foránea $Id\_Cliente$, se relaciona con la llave RFC de la tabla Cliente.
  - Dominio:
    - $Estado$: Cadena de caracteres.
    - $Fecha\_Inicio$: Cadena de caracteres en el formato $aaaa - mm - dd$.
    - $Fecha\_Fin$: Cadena de caracteres en el formato $aaaa - mm - dd$.

- **Estacion_Operaciones:**
  - Llaves:
    - Llave primaria $Id\_Estacion$.
    - Llave foránea $Id\_Sala\_Operaciones$, se relaciona con la llave $Id\_Sala\_Operaciones$ de la tabla Sala_Operaciones.
    - Llave foránea Id_Piso, se relaciona con la llave Id_Piso de la tabla Sala_Operaciones.
    - Llave foránea $Id\_Edificio$, se relaciona con la llave $Id\_Edificio$ de la tabla Sala_Operaciones.
  - Dominio:
    - SO : Cadena de caracteres.

- **Aditamento_Operaciones:**
  - Llaves:
    - Llave primaria $Id\_Aditamento$.
    - Llave foránea $Id\_Estacion$, se relaciona con la llave $Id\_Estacion$ de la tabla Estacion_Operaciones.
  - Dominio:
    - $Nombre$: Cadena de caracteres.

Dependencias Funcionales

Nos sirven para especificar formalmente cuando un diseño es correcto.

Definición

Relación unidireccional entre 2 atributos de tal forma que en un momento dado, para cada valor único de A, sólo un valor de B se asocia con él a través de la relación

Curso(Id_Curso,Id_Entrenador, Id_Sala_Entrenamiento, Id_Piso, Id_Edificio, Id_Cliente,Nombre_Curso, Turno, Num_Horas, Fecha_Inicio, Fecha_Fin, Modalidad)

El curso se compone del entrenador que lo dará.
El piso, el edificio y turno deben ser concordantes para determinar el lugar en el que se impartirá o no el curso.
El cliente determina quien pretende llevar a cabo el curso con sus fechas iniciales y finales.

Asistir_Agente (Id_Agente, Id_Horario,Id_Curso,Fecha)

Asistir_Entrenador (Id_Entrenador, Id_Horario,Id_Curso,Fecha)

Sala_Operaciones (Id_Sala_Operaciones, Id_Piso,Id_Edificio,Id_Cliente,Estado, Fecha_Inicio, Fecha_Fin)

Estacion_Entrenamiento (Id_Estacion, Id_Sala_Entrenamiento, Id_Piso, Id_Edificio,SO)

Estacion_Operaciones (Id_Estacion, Id_Sala_Operaciones, Id_Piso, Id_Edificio,SO)

Poblaciones

Para la generación de los datos nos apoyamos en la herramienta web**Mockaroo**

Nuestras tablas base fueron **edificio, piso, entrenador, agente_telefonico** y **cliente**, aunque esto no significa que las demás no tengan importancia, sino que estas fueron usadas como dataset para completar con concordancia algunos de los datos requeridos en otras tablas, ya que si quisieramos insertar tuplas en alguna tabla que tenga atributos ligados con una de estas tablas, no sería posible ya que son indispensables para completarlo.

Ahora bien, los datos vienen ya en el *script SQL*, hicimos un dataset para los estados y solo colocamos los estados que consideramos más cercanos a la CDMX (Morelos, Tlaxcala, Estado de México, Puebla) en los que actualmente pudieran vivir los entrenadores o agentes. Por otro lado para los datos como el CURP (ya sea del *agente* o *entrenador*) hicimos que los datos sean lo más cercano a la realidad conforme al nombre y apellidos, fecha de nacimiento, estado y el género(aleatorios), para que todos esos datos concordaran. Para ello creamos un dataset con los datos requeridos para que una persona tenga su CURP, posteriormente agregamos una fórmula que nos permitiera tomar las dos primeras letras del apellido paterno, la primer letra del apellido materno y del nombre, los últimos dos dígitos del año, el mes (mm), día(dd), genero(dato aleatorio), entidad y finalmente letras y dígitos aleatorios (ya que naturalmente esos datos no estaban a nuestro alcance). Por lo que cada CURP es una clave única y cumple su función de ser *PRIMARY KEY*.

Figura 3: Parte del esquema **entrenador**

En los atributos `calle, colonia, municipio` y `estado` usamos como tipo el Dataset Column teniendo ya descargados archivos con extensión csv con el contenido de varias calles, colonias, municipios y estados de México. Pero debido a que el formato del archivo que contenía las calles esta completamente en mayúsculas, tuvimos que usar una formula para primero convertirlas en minuscula (`lower(this)`) y luego capitalizar la primera letra (`this.capitalize`) de tal forma que si la calle está guardada como MAGNOLIA, termina apareciendo como Magnolia.

Para el atributo `cp` (Código Postal) usamos el tipo Character Sequence con el que podemos crear secuencias de caracteres, digitos o símbolos. En este caso es una secuencia de 5 números aleatorios.

Consultas diseñadas

1. **Número de agentes que se tiene en capacitación**
   La consulta "SELECT COUNT (curp) FROM (curso NATURAL JOIN agente_telefonico) WHERE (curso.fecha_fin <(SELECT NOW()));

   devuelve el número de agentes que se están capacitando.

2. **Información de los cursos que se tienen activos**
   La consulta "SELECT * FROM curso WHERE fecha_fin <(SELECT NOW())

   devuelve los siguientes datos de los cursos que se encuentran activos.

   - *id_curso*
   - *id_entrenador*
   - *id_sala_entrenamiento*
   - *id_piso*
   - *id_edificio*
   - *id_cliente*
   - *nombre_curso*
   - *turno*
   - *fecha_inicio*
   - *fecha_fin*
   - *modalidad*
   - *num_horas*

3. **Número de entrenadores que están dando una capacitación**
   La consulta "SELECT COUNT(curp) FROM (entrenador LEFT JOIN curso ON entrenador.curp=curso.id_entrenador) WHERE (id_curso IS NOT NULL);

   devuelve el número de entrenadores que están impartiendo un curso en la fecha de la consulta.

4. **El historial de cursos de un cliente**
   Lo siguiente crea una tabla historial_curso que contiene los datos que nos interesan almacenar
   ÇREATE TABLE historial_cursos(id_curso int, id_cliente varchar(13), nombre_curso text, turno varchar(10), fecha_inicio date, fecha_fin date);

   La consulta "SELECT * FROM (SELECT * FROM historial_cursos WHERE id_cliente='CUHE020323JM0') AS A UNION (SELECT id_curso,id_cliente,nombre_curso,turno,fecha_inicio,fecha_fin FROM curso WHERE id_cliente='CUHE020323JM0');

   regresa el historial de cursos del cliente.

5. **Número máximo de horas que dura un curso en telellamadas**
   La consulta "SELECT MAX (num_horas) FROM curso;

   regresa el número máximo de

6. **El cliente que más paga por servicios del centro de contacto**

7. **Los agentes que han sido dados de baja por baja calificación**
   "SELECT nombre FROM agente_telefonico WHERE (agente_telefonico.evaluacion <8);

8. **Los agentes que no cumplieron con su cuota de horas de entrenamiento**

9. **Los pisos de entrenamiento disponibles**
   La consulta "SELECT id_piso FROM sala_operaciones WHERE sala_operaciones.estado = 0;
   devuelve el ID de los pisos disponibles.
10. **Los pisos asignados a un cliente del centro de contacto**
11. **Las salas donde se imparte un curso**
12. **El total de ingresos por pisos del mes pasado**
13. **El total de ingresos que se ha ganado este mes por piso**
14. **Los 5 entrenadores que más horas han impartido**
15. **El edificio que más ingresos ha generado**