

Universidad Nacional Autónoma de México

Facultad de Ciencias

Fundamentos de Bases de Datos

Práctica 04. Modelo Relacional

Equipo : eSosQLones

Estrada Garcia Luis Gerardo	319013832
Jiménez Hernández Allan	420003478
Mancera Quiroz Javier Alejandro	319274831
Mora Hernández Dulce Julieta	319236448
Peña Nuñez Axel Yael	318279754

Reporte

Actividad

Deberán realizar un documento PDF donde especifiquen el dominio de sus atributos y sus restricciones si es que existen. Y deberán especificar las llaves foráneas, compuestas y primarias de cada relación. Lo llamaran Práctica04.

Dominio y restricciones

- Para *Trabajar* tenemos los atributos
 1. RFC con dominio *varchar* para manejar cadenas de longitud variable, que además tendrá como restricción el ser de doce dígitos, donde tenemos tres primeras letras de las tres primeras palabras de la denominación. Seguidos de año, mes y día y finalmente tres caracteres diferenciadores únicos. (para personas morales)
 2. Tipo con dominio *varchar* ya que al ser una cadena de longitud variable nos permite modificar el tipo de trabajo que realiza cada veterinario.
- Para *Trabajador* tenemos los atributos
 1. RFC con dominio *varchar* que justo como se menciona arriba, nos permite manejar cadenas de longitud variable y nos facilitara el proceso de manejar el RFC.
 2. Fecha_Nac con dominio *date* el cual nos permite almacenar una fecha, lo que hace que el manejo de este atributo se facilite
 3. FechaIn_Con, con dominio *date*, al igual que con la fecha de nacimiento, el hecho de utilizar este tipo de dominio nos permite, almacenar las fechas de manera eficaz
 4. FechaFin_Con, este dominio puede contener valores del tipo *date*, que nos indican la fecha de finalización del contrato.
- Para *Persona* tenemos los atributos
 1. Id_Persona es el ID asociado a una persona, como se trata de un tipo numérico que no contiene decimales, ocupamos *int*; se debe verificar que no se repita.
 2. Nombre, este atributo almacena el nombre de una persona, por lo que para representarlo es suficiente con utilizar *text*.
 3. ApellidoPat es el atributo encargado de almacenar el apellido paterno de una persona, como puede variar el tamaño del apellido, es conveniente utilizar *varchar*.

4. ApellidoMat, de manera similar al punto anterior, como es un atributo que almacena el apellido materno de una persona, conviene utilizar *varchar*.
 5. Genero, como el genero puede ser representado mediante una unica letra, es coveniente usar *char*.
 6. Fecha_Nac con dominio *date* el cual nos ayuda a representar fechas.
 7. FechaIn_Con, con dominio *date*, que nos permite representar fechas.
 8. FechaFin_Con, este dominio puede contener valores que son fechas, por lo que pueden ser representadas mediante el tipo *date*, que nos indican la fecha de finalizacion del contrato.
 9. Estado, este atributo puede almacenar los nombres de los estados, pero como no tienen una longitud especifica, podemos utilizar *varchar*.
 10. Num_Int como se trata de un atributo que representa números, es conveniente utilizar *int* ya que los números interiores no cuentan con decimales, por lo que es suficiente un int para representarlo.
 11. Num_Ext, justo como en el punto anterior, es conveniente utilizar *int*.
 12. Colonia, como la colonia es expresada mediante palabras, podemos utilizar *text*, pues es una cadena de texto de longitud variable.
 13. Calle, para este caso, al tener calles con nombres diferentes y de longitudes variables, podemos utilizar *varchar*.
- Para *Veterinario* tenemos los atributos
 1. RFC con dominio *varchar* que facilita el proceso de manejar un RFC dadas las restricciones que este atributo presenta.
 2. Fecha_Nac con dominio *date* el cual nos ayuda a representar fechas.
 3. FechaIn_Con, con dominio *date*, que nos permite representar fechas.
 4. FechaFin_Con, este dominio puede contener valores que son fechas, por lo que pueden ser representadas mediante el tipo *date*, que nos indican la fecha de finalizacion del contrato.
 5. Salario el cual es un atributo que representa cuanto gana un veterinario, como puede ser una cifra numerica que tenga centavos , ocupamos el tipo *double*.
 - Para *Cuidador* tenemos los atributos
 1. RFC con dominio *varchar* que facilita el proceso de manejar un RFC.
 2. Fecha_Nac con dominio *date* el cual nos ayuda a representar fechas.

3. FechaIn_Con, con dominio *date*, que nos permite representar fechas.
 4. FechaFin_Con, este dominio puede contener valores que son fechas, por lo que pueden ser representadas mediante el tipo *date*, que nos indican la fecha de finalización del contrato.
 5. Animal_Enc, se refiere al animal a cargo de un cuidador, para este atributo requerimos texto, por lo que podemos ocupar *varchar*, el cual nos permite tener cadenas de longitud variable.
 6. Dias_Trab se refiere al número de días que trabaja un cuidador, como los días son un numero entero, lo representamos a traves de un *int*.
 7. Salario es un atributo que representa cuanto gana un cuidador, como se trata de un valor numérico que puede tener centavos , ocupamos el tipo *double*.
- Para *Proveedor* tenemos los atributos
 1. RFC con dominio *varchar* que facilita el proceso de manejar un RFC.
 2. Fecha_Nac con dominio *date* el cual nos ayuda a representar fechas.
 3. FechaIn_Con, con dominio *date*, que nos permite representar fechas.
 4. FechaFin_Con, este dominio puede contener valores que son fechas, por lo que pueden ser representadas mediante el tipo *date*, que nos indican la fecha de finalizacion del contrato.
 5. Costo_Serv, como se trata de la cantidad a pagar por un servicio y sabemos que podemos tener centavos, conviene utilizar un *double*.
 6. Producto, los proveedores ofrecen diferentes tipos de productos, por lo que para poder saber de cual se trata podemos ocupar un *text*.
 7. Frecuencia, este atributo se refiere a la frecuencia con la que un proveedor da un producto, por lo que podemos representarlo mediante un *int*.
 - Para *Correo* tenemos los atributos
 1. Id_Persona, al tratarse de un ID que es representado mediante numeros, utilizaremos *int*, se debe verificar que el ID sea único.
 2. Correo, sabemos que los correos cuentan con diferentes caracteres y que pueden variar en longitud, por lo que podemos representarlo mediante *text*; además sabemos que deben de tener alguna terminación, por lo que podríamos verificar el dominio para confirmar que se trate de un correo.
 - Para *Teléfono* tenemos los atributos

1. Id_Persona, como se trata de un ID que es representado mediante valores enteros, utilizar *int* es una opción útil.
 2. Teléfono, sabemos que los teléfonos están conformados únicamente por números enteros, por lo que para representarlo ocuparemos *int*
- Para *Atender* tenemos los atributos
 1. RFC con dominio *varchar* que facilita el proceso de manejar un RFC.
 2. Id_Animal, como se trata de un ID que es representado mediante valores enteros, utilizaremos *int*.
 3. mismoBioma, para este atributo solo se debe conocer si el Veterinario y el Animal pertenecen al mismo Bioma, por lo que podemos utilizar un valor de tipo *bool*.
 - Para *Especialidad* tenemos los atributos
 1. RFC, para este atributo podemos utilizar *varchar* que facilita el proceso de manejar un RFC.
 2. Especialidad, como se trata de un atributo que nos permite identificar que tipo de especialidad tiene un veterinario, y es expresado mediante una cadena, podemos utilizar *varchar*.
 - Para *Bioma* tenemos los atributos
 1. Tipo, este atributo se refiere al tipo de bioma del que hablamos, para representarlo ocuparemos *text* y además debemos verificar que el tipo de bioma escrito sea un tipo de Bioma existente en nuestro zoológico.
 2. RFC, como es un atributo que tiene diferentes caracteres ocuparemos *varchar*
 3. Id_Animal como se trata de un ID, además de verificar que sea único, tendremos que almacenarlo, y para ello lo haremos en un *int*.
 4. Num_Cuid este atributo se refiere al número de cuidadores que hay en un bioma, utilizamos *int* para representarlo.
 5. Num_Vet notemos que este atributo se refiere al número de veterinarios que hay en un bioma, por lo que ocuparemos *int*.
 6. Num_Anim este atributo se refiere al número de animales que hay en un bioma, como es un número, para representarlo ocupamos *int*.
 7. Servicio, este atributo se refiere al servicio de un bioma, para representarlo ocuparemos *text*.

8. Num_Jaulas este atributo se refiere al numero de jaulas que hay en un bioma, para representarlo ocupamos *int*.
- Para *Animal* tenemos los atributos
 1. Id_Animal es el atributo que funciona como nuestro identificador para un animal, para este atributo utilizaremos un *int* y debemos verificar que sea único.
 2. RFC, como se trata de un atributo que almacena el RFC, ocuparemos *varchar*.
 3. Num_Jaula, como es un atributo que guarda valores numéricos enteros, ocuparemos un *int*.
 4. Alimentación, para este atributo ocupamos el tipo de dato *text* para recibir texto; además debemos verificar que se trate de una alimentación permitida por el caso de uso.
 5. Sexo, como el sexo puede ser representado mediante una letra podemos ocupar *char*, sin embargo, debemos realizar una verificación para validar que la letra ingresada haga referencia a un valor válido.
 6. Altura, este atributo es representado mediante un *float* ya que el rango proporcionado por un float es suficiente para almacenar dicho valor.
 7. Peso, justo como en el punto anterior, ocupamos un *float*.
 8. Especie, para este atributo ocupamos *text* ya que buscamos un tipo de dato que permita almacenar texto.
 - Para *Distribuir* tenemos los atributos
 1. Id_Insumo, al tratarse de un ID que es representado por valores enteros, utilizaremos *int*
 2. Tipo, este atributo puede contener que tipo de insumo se distribuye, por lo que para utilizarlo ocuparemos *text*.
 - Para *Insumo* tenemos los atributos
 1. Id_Insumo, este atributo es un ID por lo que además de verificar que sea único lo almacenaremos en un *int*.
 2. fechaCad como este atributo se refiere a una fecha, es conveniente utilizar *date* ya que nos permite expresar fechas de la forma (Año, Mes, Día).
 3. Proveedor, como se refiere a quien es el proveedor del insumo, podemos expresarlo mediante una cadena, por lo que utilizaremos *varchar*.

4. *seRefrigera*, este atributo se refiere a si el tipo de Insumo se refrigera o no, por lo que conviene usar un *boolean* ya que nos da un valor de *true* o de *false*.
 5. *Nombre*, como se trata de una palabra, podemos utilizar *text*.
 6. *Cantidad*, este atributo se refiere a la cantidad de insumos, como hablamos de cantidad, hablamos de números, por lo que podemos utilizar *double* para representarlo.
- Para *Alimento* tenemos los atributos
 1. *Id_Insumo*, este atributo es un ID por lo que además de verificar que sea único lo almacenaremos en un *int*.
 2. *tipoAlimento* es un atributo que expresa el tipo de alimento del que se habla, para ello lo almacenaremos en un tipo de dato *text* y verificaremos que sea un tipo de alimento valido.
 - Para *Medicina* tenemos los atributos
 1. *Id_Insumo*, este atributo es un ID por lo que además de verificar que sea único lo almacenaremos en un *int*.
 2. *labProcedencia*, como se trata de un atributo donde debemos almacenar palabras, ocuparemos *text* y verificaremos que sea un laboratorio de procedencia valido.
 3. *lote*, este atributo acepta valores numéricos, por lo que para representarlo ocuparemos *int*.
 - Para *IndicacionMed* tenemos los atributos
 1. *Id_Animal* veamos que tenemos el caso de un ID por lo que tenemos la certeza de que sera un número y que además no tendrá decimales, a partir de esto, decidimos utilizar un *int*; además requerimos verificar que no se repita.
 2. *IndicacionMed*, como se trata de un atributo cuya intención es expresar las indicaciones médicas de un animal, utilizaremos *text*.
 - Para *Cliente* tenemos los atributos
 1. *Id_Cliente*, como se trata de un identificador, tenemos la certeza de que sera un número y que además no tendrá decimales, por lo que podemos ocupar un *int*; además requerimos verificar que no se repita

- Para *Notificar* tenemos los atributos
 1. Id_Cliente, al tratarse de un ID, que sabemos que no tendrá decimales, utilizaremos *int*; se debe verificar que no haya repetidos
 2. Id_Evento, como se trata de un ID que es representado mediante valores enteros, utilizaremos *int* y ocupamos verificar que no exista ya con anterioridad ya que debe ser único
 3. Promoción, en el caso de las promociones, como pueden requerir palabras para las especificaciones, ocuparemos un tipo de dato que permita expresar texto, por lo que utilizaremos *text*
 4. Descuento, para representar un descuento, ocuparemos *int* dado que es un tipo de dato que nos permite expresar números enteros
 5. Noti_Evento, como se trata de una notificación que se les da a los clientes, dicha debe ser expresada mediante palabras, por lo que para representarla ocuparemos *text*
- Para *Evento* tenemos los atributos
 1. Id_Evento, notemos que se trata de un ID, el cual es representado mediante valores enteros, para ello utilizaremos *int*, además debemos verificar que el valor no se repita
 2. listaAsistentes, como se trata de un atributo que contiene a los asistentes, debemos de guardar estos datos en un tipo de dato que permita almacenar texto, por lo que ocuparemos *text*
 3. Capacidad, el atributo hace referencia a la cantidad de personas, por lo que para representarlo utilizaremos *int*
 4. tipoServicio, como contamos con tres tipos de servicios podemos utilizar *text*; sin embargo, debemos considerar que hay que verificar que el tipo de servicio ingresado corresponda a los disponibles
 5. Fecha, para este atributo podemos utilizar *date* ya que nos permite escribir la fecha de la forma año, mes y día
- Para *Ticket* tenemos los atributos
 1. NumTicket, los tickets tienen un número asociado, este número no tiene decimales, lo que nos permite saber que cuenta con valores enteros, por lo que para representarlos se utilizara *int*
 2. Id_Servicio, como se trata de un ID que es representado mediante valores enteros, utilizaremos *int*, además debemos verificar que el valor no se repita

3. costoTotal, como estamos trabajando sobre un ticket, podemos utilizar *money* el cual nos permite expresar la cantidad de moneda
 4. costoUnitario, de manera similar al punto anterior, podemos utilizar *money*
 5. Descuento, como se trata de una cantidad del 1 al 100, podemos utilizar *int*, además debemos verificar que este dentro de los límites de un descuento
 6. tipoServicio para identificarlos ocupamos palabras, por lo que utilizar *text*; se debe verificar que sea un servicio existente además
 7. Fecha, para representar fechas, *date* es útil
 8. Id_Cliente, como se trata de un ID, *int* es la opción a utilizar, además debemos verificar que cumpla ser un ID único.
- Para *Servicio* tenemos los atributos
 1. Id_Servicio, como se trata de un ID que es representado mediante valores enteros, ocuparemos *int*, al tratarse de un ID debemos verificar que su valor no se repita
 2. listaAsistentes, este atributo almacena los asistentes, por lo que buscamos un tipo que nos permita guardar texto, para ello podemos utilizar *text*
 3. Capacidad, se refiere a la cantidad de personas, por lo que para representar utilizaremos *int*
 4. tipoServicio, como contamos con tres tipos de servicios podemos utilizar *text*; además debemos verificar que el tipo de servicio ingresado corresponda a los disponibles
 5. Fecha, para este atributo podemos utilizar *date* ya que nos permite escribir la fecha con año, mes y día
 - Para *Registrar* tenemos los atributos
 1. NumTicket, los tickets tienen un número asociado, este número no tiene decimales, lo que nos permite saber que cuenta con valores enteros, por lo que para representarlos se utilizara *int*
 2. Id_Servicio, como se trata de un ID que es representado mediante valores enteros, utilizaremos *int*, dicho entero no puede repetirse

Llaves

En este apartado se hablara de las llaves con las que cuenta cada tabla y porque se eligieron a esos atributos como llaves

- **Trabajar**

Trabajar, en el esquema E/R, no es una entidad sino una relación, pero al contar con atributos, es necesario que tenga su tabla.

- Llave foránea (RFC): Esta llave es esencial, ya que no puede existir un trabajo sin un trabajador (por ello la existencia de una participación obligatoria).
- Llave foránea (Tipo): Esta llave forma parte de esta tabla, ya que obligatoriamente debe de haber un lugar de trabajo (en este caso, los biomas).

- **Veterinario**

- Llave foránea (RFC): Como el Veterinario solo es una especialización del trabajador, es suficiente con tener una llave foránea que nos haga referencia a la tabla trabajador y tener acceso a los atributos compartidos.

- **Cuidador**

- Llave foránea (RFC): Como el Cuidador solo es una especialización del trabajador, es suficiente con tener una llave foránea que nos haga referencia a la tabla trabajador y tener acceso a los atributos compartidos.

- **Proveedor**

- Llave foránea (RFC): Como el Proveedor solo es una especialización del trabajador, es suficiente con tener una llave foránea que nos haga referencia a la tabla trabajador y tener acceso a los atributos compartidos.

- **Atender**

- Llave foránea (RFC): Necesitamos la llave foránea para tener acceso a los datos del trabajador que atenderá a que animal en el Zoológico.
- Llave foránea (IdAnimal): También necesitamos saber qué animal será atendido.

- **Especialidad**

- Llave foránea (RFC): Necesitamos la llave foránea para tener acceso a los datos del trabajador que tendrá una especialidad.
- Llave primaria (Especialidad): También necesitamos saber cuál es la especialidad.

- **Bioma**

- Llave primaria (Tipo): Se ha seleccionado el Tipo como llave primaria porque cada bioma es distinto y único.

- Llave foránea (RFC): Esta llave nos permite conocer qué trabajador está asignado a cada bioma.
 - Llave foránea (idAnimal): Nos permite saber qué animales se encuentran en cada bioma.
- **Distribuir**
 - Llave foránea (idInsumo): Esencial para identificar qué insumo se está distribuyendo.
 - Llave foránea (Tipo): Necesaria para saber a qué bioma se distribuye el insumo.
- **Insumo**
 - Llave primaria (idInsumo): Cada insumo es único y por ello se utiliza su identificador como llave primaria.
- **Alimento**
 - Llave foránea (idInsumo): Al ser el alimento un tipo de insumo, esta llave foránea nos permite vincular ambos.
- **IndicacionMed**
 - Llave foránea (idAnimal): Necesaria para saber a qué animal está dirigida la indicación médica.
 - Llave primaria (indicacionMed): Cada indicación médica es única y se utiliza como llave primaria.
- **Cliente**
 - Llave primaria (idCliente): El identificador del cliente es único y se usa como llave primaria para distinguir a cada cliente.
- **Medicina**
 - Llave foranea (idInsumo): Esta llave foránea permite asociar la medicina como un tipo de insumo, proporcionando una relación directa con los atributos del insumo.
- **Registrar**
 - Llave foranea (NumTicket): Es esencial para identificar el registro de cada transacción y mantener un historial de los servicios solicitados.
 - Llave foranea (idServicio): Esta llave permite vincular los servicios registrados, asegurando que cada uno se relacione correctamente.

- **Servicio**

- Llave primaria (idServicio): El idServicio es único para cada servicio, por ello se utiliza como llave primaria para distinguir cada servicio ofrecido.

- **Ticket**

- Llave primaria (NumTicket): El NumTicket es único y se utiliza como llave primaria para diferenciar cada ticket.
- Llave foranea (idCliente): Necesaria para identificar a qué cliente pertenece cada ticket.

- **Notificar**

- Llave foranea (idCliente): Esta llave es crucial para asociar la notificación con el cliente correspondiente.
- Llave primaria (idEvento): Cada evento es único, por lo que se utiliza su identificador como llave primaria.

- **Persona**

- Llave foranea (idPersona): Se utiliza para asociar la persona con sus datos correspondientes en otras tablas.
- Llave primaria (idEvento): Cada persona es única, por lo que se utiliza su identificador como llave primaria.

- **Trabajador**

- Llave primaria (RFC): El RFC es único para cada trabajador, por eso se utiliza como llave primaria.
- Llave foranea (idPersona): Esta llave foránea vincula al trabajador con sus datos personales en la tabla Persona.

- **Telefono**

- Llave foranea (idPersona): Esta llave foránea vincula el número de teléfono o los numeros de telefono con la persona correspondiente.

- **Correo**

- Llave foranea (idPersona): Esta llave foránea vincula el correo o los correos con la persona correspondiente.