|  |  |
| --- | --- |
|  | **Universidad Nacional Autónoma de México** |
|  | **Facultad de Ingeniería** |
|  | **División de Ingeniería Eléctrica** |
|  | **Laboratorio de**  **Bases de Datos** |

|  |  |  |  |
| --- | --- | --- | --- |
|  | Practica 5 | Diseño básico de modelos relacionales | |
|  |  |  |  | |
|  | Profesor: | Lugowski Rivero Czeslaw Kristofer |  | |
|  | Grupo: | 06 |  | |
|  | Alumnos: | Guerrero López Uriel Ivan 320046045  Muñoz San Agustin Victoria Monserrat 320094187 | | |
|  |  |  | Calificación | |
|  |  | Fecha de entrega | Jueves 20 de marzo de 2025 | |
|  |  |  | Semestre 2025-2 | |

**Objetivo:**

* El alumno comprenderá e implementará la construcción de modelos de datos relacionales básicos empleando herramientas CASE a partir de un diagrama ER.

**Introducción**

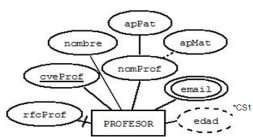
El diseño lógico de una base de datos relacional consiste en el desarrollo de un modelo relacional que se obtiene a partir del modelo ER así como del enunciado y las reglas de negocio del caso de estudio.

1. Transformación de un modelo Entidad Relación a un Modelo Relacional.

Para obtener un modelo relacional a partir de un modelo Entidad-Relación se tienen varias reglas

* 1. Toda entidad fuerte se transforma en una relación
* Se conservan los atributos y la llave primaria
* En claves candidatas se establece restricción de unicidad
* Atributos compuestos se colocan en forma individual
* Atributos multivalorados se crea una nueva relación propagando la llave primaria de la relación como foránea a la nueva relación.
* Atributos derivados o calculados se establece como atributo calculado
* Se establecen restricciones sobre atributos (\*CS1) como superíndice

Simbología a utilizar

* PRIMARY KEY (PK)
* UNIQUE (U)
* FOREIGN KEY (FK)
* CHECK (CK).
* Calculados (C)
* Discriminantes (D)
* No obligatorio (N)

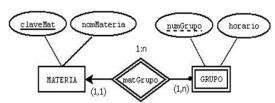
Ejemplo:

PROFESOR={cveProf(PK),nomProf,apPat,apMat(N),rfcProf(U),edad(C)\*CS1}

EMAILPROF={cveProf(PK,FK),email(U,PK)}

\*CS1 Se obtiene del rfc del profesor

* 1. Entidades Débiles
* Se crea la relación conservando todos sus atributos y se propaga la llave primaria de la entidad fuerte de la que depende como llave foránea.
* La llave principal se formará con la llave primaria de la entidad fuerte y el discriminante de la entidad débil

Ejemplo:

MATERIA= {claveMat(PK), nomMateria}

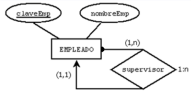
GRUPO= {[claveMat(FK), numGrupo(D)](PK), horario}

* 1. Relaciones
* 1:1 La llave primaria de una entidad se propaga a la otra entidad como llave foránea con la restricción de unicidad, dependiendo de la cardinalidad mínima, si la cardinalidad mínima es cero, se pasa el atributo de la entidad de cardinalidad uno, hacía la de cardinalidad cero, si en ambas es la misma cardinalidad, dependerá del contexto.
* m:1 ó 1:m La llave primaria de la entidad que tiene la cardinalidad uno se propaga a la relación de muchos como llave foránea.
* m: m Se genera una nueva relación que contendrá la llave primaria de cada una de las entidades que une, pero como llaves foráneas y juntas como llave primaria, además de sus atributos propios (descriptivos) de la relación.

En las relaciones (1:1, 1:m, m:1) que contengan atributos descriptivos, se tienen dos opciones:

* Generar una nueva relación que contendrá la llave primaria de cada una de las entidades que une como llaves foráneas, además de sus atributos descriptivos, conservando la unicidad en la nueva relación.
* Propagar la llave primaria de la entidad uno junto con todos los atributos de la relación a la entidad m
  1. Relaciones recursivas

Ejemplo:



La llave primaria de la entidad pasa como llave foránea de la misma entidad.

EMPLEADO= {claveEmp(PK), nombreEmp, claveSup(FK)}

1. Notaciones para diseño lógico de una base de datos

Existen varias notaciones para realizar la construcción de un modelo relacional. Las 2 notaciones comúnmente empleadas en la industria del desarrollo de software son:

* Notación Crow’s foot
* Notación IDEF1X

Ambas notaciones son muy similares. En esta sección se muestran las características de ambas haciendo énfasis en donde exista una diferencia entre ellas.

* 1. Representación de entidades, atributos y llaves primarias

|  |  |
| --- | --- |
| Notación crow’s foot e IDEF1X | |
|  | * Las entidades se representan por tablas. En el rectángulo superior se indica la llave primaria. * Tanto el nombre de la tabla como el de los atributos se especifican en mayúsculas. Las palabras compuestas se separan por guión bajo. |

* En lo posible omitir abreviaturas. Si se usan deben ser muy claras.
* Notar que se debe incluir el tipo de dato del atributo así como el constraint null o not null.
* Todos los atributos se consideran como obligatorios por default, por lo que para un atributo opcional se especifica con la palabra null.
  1. Tipos de llaves en el modelo relacional

|  |  |
| --- | --- |
| Clave principal en modelo ER | Tipo de llave primaria en modelo relacional |
| Clave principal | llave primaria natural  llave primaria artificial |
| Clave compuesta | llave primaria compuesta  llave primaria artificial o subrogada |
| Clave candidata | llave candidata o alternativa |

* En el modelo relacional se sugiere en la medida de lo posible el uso de una PK artificial principalmente por los beneficios que ofrece:
  + Mejor rendimiento. El índice que genera una llave primaria tiene un mejor desempeño en valores numéricos de preferencia consecutivos.
  + Sin problemas de duplicidad
  + Sin probabilidad de cambio de valores
* Para el caso de una PK compuesta, se debe tener cuidado con su selección. Una PK compuesta implica la generación de combinaciones de valores de cada atributo lo cual pudiera ser incorrecto para algunos escenarios.
  1. Representación de atributos compuestos

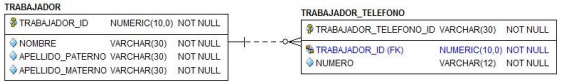
|  |  |
| --- | --- |
| Diseño conceptual | Diseño lógico Crow’s Foot e IDEF1X |
|  |  |

* En el diseño lógico, el atributo compuesto se omite, únicamente se incluyen los sub atributos.
  1. Representación de atributos multivalorados

|  |  |
| --- | --- |
| Diseño conceptual | Diseño lógico Crow’s Foot e IDEF1X |
|  |  |

* Para implementar este concepto se deberá crear una nueva tabla. A nivel general, la PK de la nueva tabla será una PK compuesta formada por la PK de la tabla, en este caso num\_trabajador y por el atributo multivalorado, en este caso, teléfono.
* El ejemplo anterior funciona correctamente. Sin embargo, la PK compuesta anterior no es del todo eficiente al ser compuesta y formada por un identificador más un número telefónico.
* Para resolver este detalle se puede especificar una PK artificial, dejando los atributos de la llave natural indicados juntos como únicos.

Cambiar



* 1. Representación de relaciones

Las relaciones en el modelo relacional se representan a través de llaves foráneas. Para este concepto existe diferencia entre las notaciones Crow’s foot e IDEF1X.

|  |  |
| --- | --- |
| Notación crow’s foot | Notación IDEF1X |
|  |  |

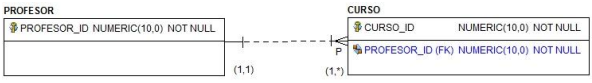
* + 1. Relaciones identificativas y no identificativas

Como se puede observar en el diagrama anterior, existe una diferencia importante entre el uso de una línea punteada o continua

* Punteada: Relación no identificativa, suave o débil
* Continua: Relación identificativa, dura, o fuerte
* En una relación identificativa la PK de la tabla padre se propaga como PK y FK en la tabla hija.
* En una relación no identificativa, la PK de la tabla padre se propaga únicamente como FK en la tabla hija

Ejemplo:

* Curso y profesor 1:m
* Un profesor imparte varios cursos. 1:m
* Un curso es impartido por un profesor. 1:1



* Para una relación 1:m se emplea una relación no identificativa. Notar que la PK profesor\_id se propaga únicamente como FK hacia la tabla hija curso.
* La FK siempre se debe ubicar en la tabla que tiene la cardinalidad muchos (m). En este ejemplo, corresponde con el valor \* . Un profesor imparte varios cursos, por lo tanto la FK deberá ubicarse en la tabla curso.
  + 1. Relaciones 1:1

En las relaciones 1:1 se tienen diferentes técnicas. La solución final depende en buena medida de las reglas de negocio.

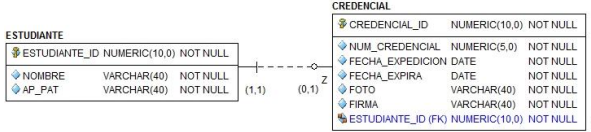
Ejemplo:

Estudiante y Credencial 1:1

A cada estudiante se le asigna su credencial 1:1

Una credencial se le asigna a un solo estudiante 1:1

Agregar unique



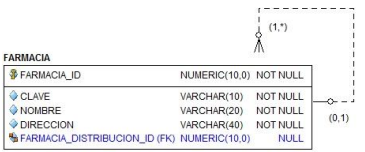
* + 1. Relaciones unarias o recursivas

El siguiente ejemplo muestra la representación de una relación unaria.

Cada farmacia cuenta con su centro de distribución a partir del cual se le proporcionan sus medicamentos. Un centro de distribución almacena grandes cantidades de medicamentos para distribuir a otras farmacias. El centro de distribución también se considera como farmacia ya que ahí mismo se atienden a los clientes.

De lo anterior:

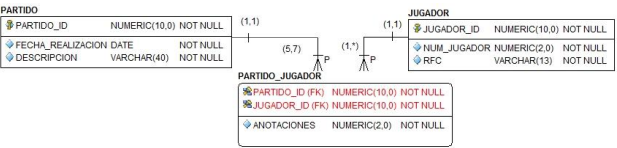
* Una farmacia cuenta con su centro de distribución (0:1)
* Una farmacia de distribución surte medicamentos a varias farmacias (1:m)



En este ejemplo la tabla farmacia representa a la entidad hija, y la farmacia de distribución que es en sí una farmacia representa a la tabla padre.

* La cardinalidad (0,1) indica que no todas las farmacias están asociadas a una farmacia de distribución. Las farmacias de distribución por ser consideradas como tal no cuentan a su vez con una farmacia de distribución. Para todas ellas, el valor de la FK deberá ser nula.
* A nivel general, una FK que proviene de la misma tabla siempre deberá ser marcada como null.
  + 1. Relaciones muchos a muchos

En el modelo relacional para las relaciones m:m, se requiere crear 2 relaciones 1:m haciendo uso de una tabla intermedia.

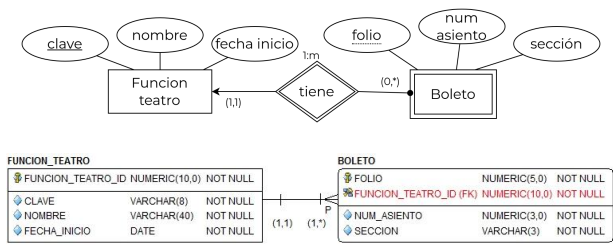


En una relación m:m la PK de la tabla intermedia es compuesta. Esto permite relacionar ambas tablas. Las combinaciones de la PK compuesta generan la relación m:m, es decir, un mismo partido puede asociarse con varios jugadores, y un mismo jugador puede asociarse con varios partidos.

Los atributos que se asignan a la tabla intermedia tienen sentido únicamente con la combinación de los valores de la PK. En el ejemplo, el número de anotaciones depende de ambos, del jugador que las realizó en un partido en específico.

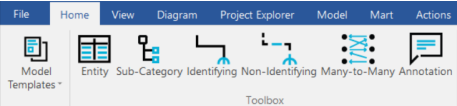
* 1. Representación de entidades débiles

En el modelo relacional, la entidad débil tiene una PK compuesta formada por la PK natural de la entidad débil y la llave primaria de la tabla padre.



Notar el uso de una relación identificativa cuando se propaga la PK de la tabla padre también como PK a la tabla hija.

1. Generación de modelos relacionales con ERwin

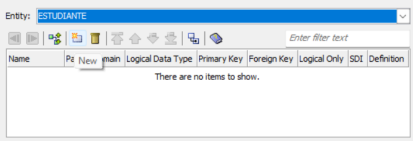
Para agregar elementos al modelo lógico se hará uso de la pestaña Home de Erwin.

* 1. Creación de entidades.

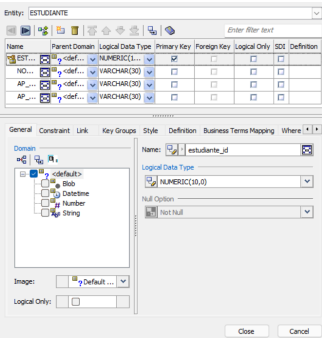
Presionar el icono correspondiente a Entity para crear una entidad, colocar dicha entidad en el espacio de trabajo y asignarle un nombre.

* 1. Atributos de una entidad

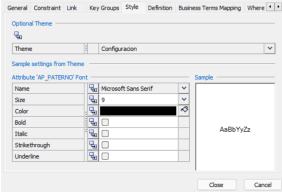
Una vez creada la entidad se realiza click derecho sobre la entidad y se selecciona la opción Attribute Properties, se mostrará una ventana como la siguiente



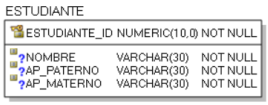
Crear un nuevo atributo haciendo click en el icono New, colocar el nombre del atributo, tipo de dato, null/not null, y si es llave primaria.



Es posible seleccionar el tipo y tamaño de letra de los atributos, para ello se selecciona el apartado Style y en la opción Theme seleccionamos la opción Configuración creada anteriormente.

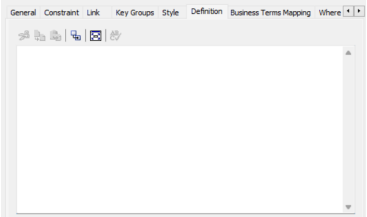


Obteniendo la siguiente entidad:



* + 1. Documentación de entidades y atributos.

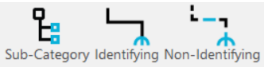
Al hacer click derecho en la entidad tanto en la opción de Entity Properties y Attribute Properties tienen un apartado de Definition en el cual se pueden documentar las entidades y cada uno de los atributos.



* 1. Asociación de entidades empleando relaciones.

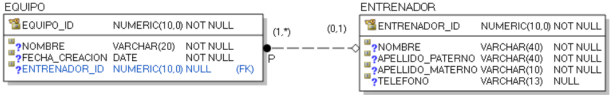
La asociación se realiza a partir de los tipos de relaciones:

* Relaciones identificativas si el atributo fk forma parte de la pk
* Relaciones no identificativas si el atributo fk no forma parte de la pk
  + Obligatorias
  + Opciónales
* Subtipos

Para seleccionar alguno de los tipos de relación, hacer clic en alguno de los íconos mostrados en la figura.

Para asociar 2 entidades:

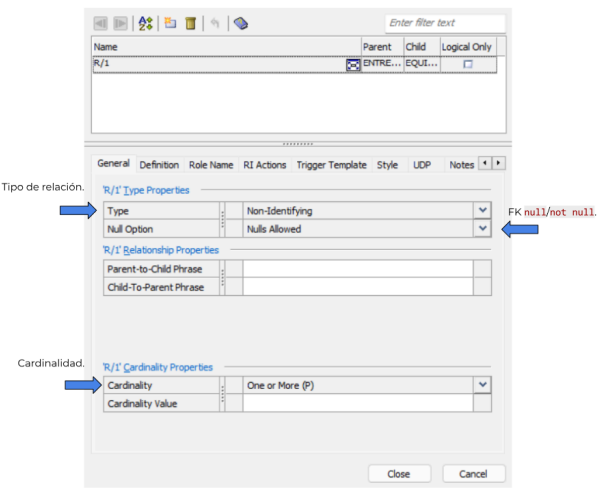
* Hacer clic en el icono que representa el tipo de relación a usar.
* Hacer clic sobre la entidad origen
* Hacer clic sobre la entidad destino (la tabla que contendrá la FK).
* Al final, se obtiene un modelo como el siguiente:



Observar que los valores de la cardinalidad se deben escribir de forma manual, empleando la herramienta de texto mostrada en la imagen.

* + 1. Opciones para relaciones.

Al hacer doble clic en la línea que representa la relación, aparecerá una ventana como la siguiente, en la que se puede personalizar o modificar las propiedades de la relación, como el nombre, tipo, cardinalidad entre otras. Usualmente se modifican las opciones

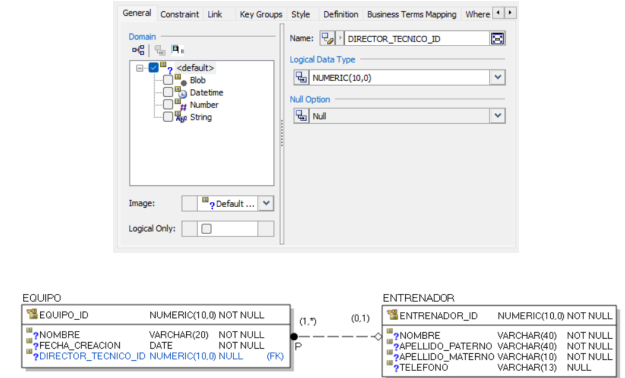
señaladas en la siguiente imagen.

* + 1. Cambio del nombre a la FK.

En algunas situaciones es conveniente cambiar el nombre a la FK para tener una mayor comprensión del modelo.

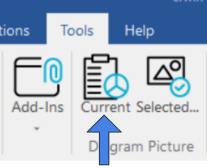
Suponer que se desea cambiar el nombre de la FK: entrenador\_id A director\_tecnico\_id.

Para realizar el cambio, basta con ir a la opción Attribute Properties y cambiar el nombre del atributo correspondiente a la FK.

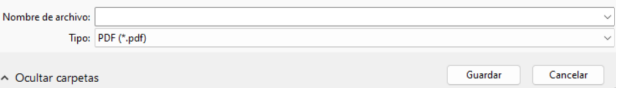


* + 1. Exportando diagramas.

Para exportar diagramas se utiliza la pestaña Tools de Erwin en la opción Current.



Cuando se selecciona la opción Current se abre la ventana de Guardar como del sistema operativo, es posible exportar el diagrama como imagen o como un archivo pdf seleccionando el tipo de archivo.



**Actividades propuestas por la academia**

1. A partir del modelo Entidad Relación realizado en la práctica anterior, realizar la transformación al modelo relacional generando los objetos necesarios en la herramienta CASE elegida.
2. Seguir los lineamientos y notaciones indicados en el previo de la práctica. Indicar en el modelo cardinalidad, valores opcionales y requeridos.

Llaves primarias

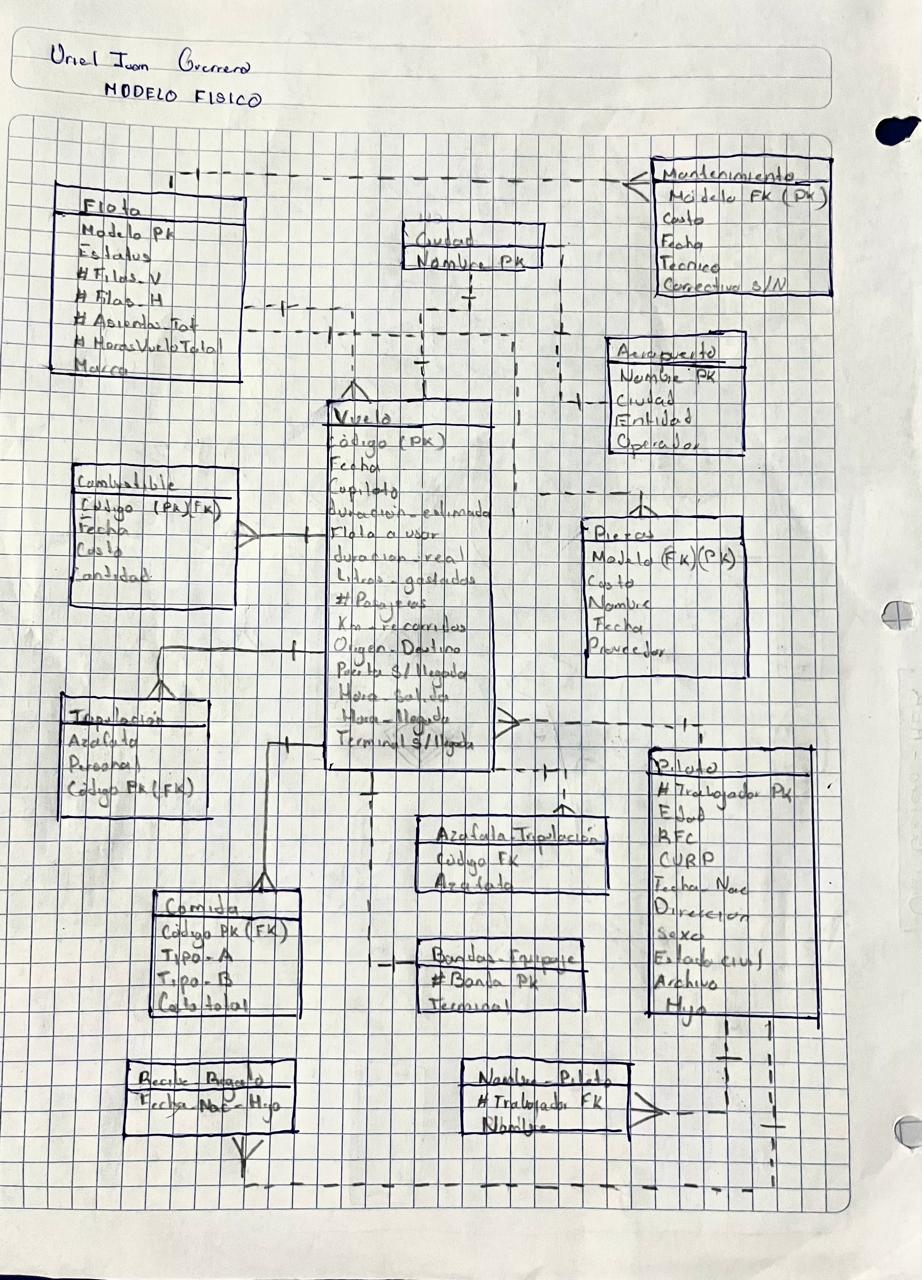
* #Trabajador
* Modelo
* Código
* Nombre\_C
* Nombre\_A

Multivaluados

* Azafata
* Nombre\_Piloto

1. De manera individual en clase, cada integrante realizará su propuesta de modelo Relacional de forma manual. Estos borradores se deberán incluir en el reporte (incluir el nombre).

Uriel Ivan Guerrero López

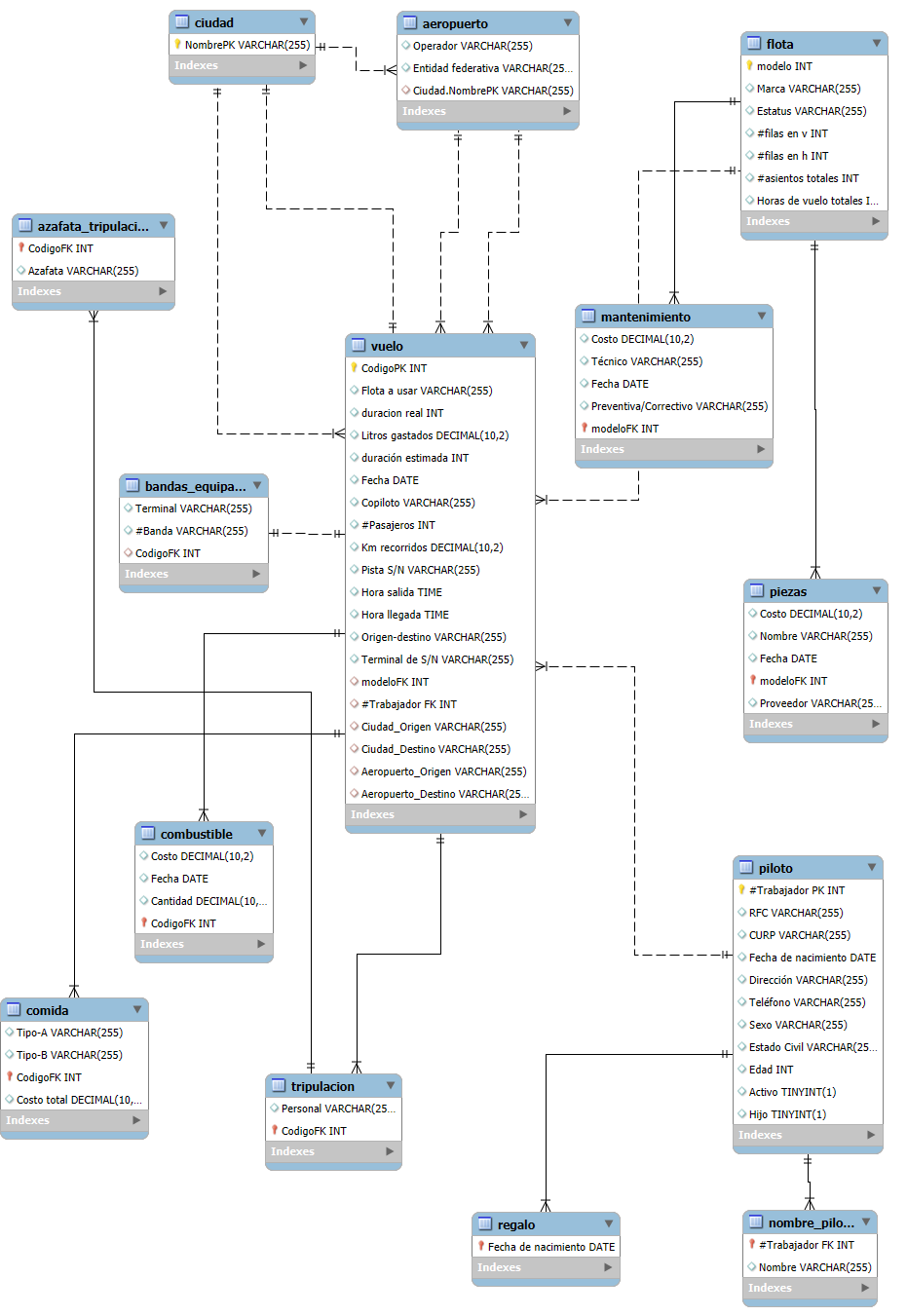


Muñoz San Agustin Victoria Monserrat



1. Generar una versión final de su modelo Relacional empleando la herramienta CASE de forma obligatoria. Incluir el modelo en su reporte.

Versión final generada en Workbench:

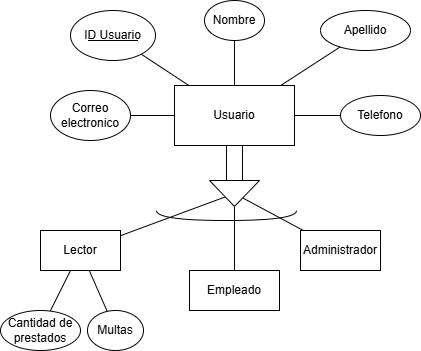


**Preguntas Previo 5**

1. **Generar el diseño conceptual de un caso de estudio donde se incluya la utilización del concepto de jerarquía de entidades.**

En una biblioteca pública, la administración de usuarios es esencial para garantizar un control adecuado de los préstamos y accesos a los recursos. Actualmente, el sistema no diferencia entre los distintos tipos de usuarios, lo que genera dificultades en la asignación de permisos y responsabilidades. Para solucionar este problema, se implementará un modelo basado en la jerarquía de entidades, permitiendo organizar a los usuarios según su función dentro de la biblioteca.

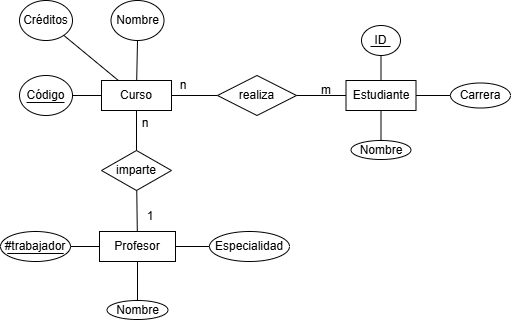
El sistema contará con una entidad principal llamada Usuario, que almacenará información general como nombre, apellido, id usuario, correo electrónico y teléfono. A partir de esta entidad, se derivarán tres subtipos: Lector, Empleado y Administrador. Los lectores serán los usuarios que pueden solicitar préstamos y devoluciones, teniendo restricciones en la cantidad de libros que pueden llevar y multas en caso de retrasos. Los empleados serán responsables de la gestión operativa de la biblioteca, incluyendo la organización del inventario y la supervisión de los préstamos. Los administradores, además de ser empleados, tendrán permisos especiales para modificar configuraciones y políticas del sistema.



1. **Generar el diseño conceptual de un caso de estudio donde se incluya la utilización del concepto agregación.**

En una universidad, la administración de los cursos es esencial para garantizar una correcta organización académica y el seguimiento del rendimiento estudiantil. Actualmente, el sistema no gestiona adecuadamente la relación entre los cursos y los estudiantes, lo que dificulta el control de inscripciones y la asignación de profesores.

El sistema contará con una entidad principal llamada Curso, que almacenará información general como código, nombre, cantidad de créditos y profesor a cargo. A partir de esta entidad, se establecerá una relación de agregación con la entidad Estudiante, lo que significa que un curso puede agrupar a múltiples estudiantes, pero estos pueden existir en el sistema sin estar inscritos en un curso en particular. Los Estudiantes representarán a los alumnos inscritos en la universidad y tendrán atributos como ID, nombre y carrera. A su vez, los cursos estarán asociados a un Profesor, quien será el responsable de impartir la materia y evaluar a los estudiantes.



**Conclusiones**

* Guerrero López Uriel Ivan

Una vez concluida la práctica logre comprender el uso de la herramienta de modelado de datos, en este caso se utilizó MySQL Workbench, fue muy importante el uso de esta herramienta ya que ayuda bastante al momento de pasar del Modelo Entidad-Relación, fue interesante darme cuenta de que podemos hacerlo directamente de forma gráfica con un modelo de tablas, permitiéndonos agregar atributo por atributo a cada una de las entidades, además de declarar que tipo de restricción tenia, ya sea una Primary o en su defecto una Foreing key.

A su vez fue interesante darme cuenta de que otra forma en la que se podía trabajar en la realización del modelo relacional, la cual era aplicar “Reverse Engineering”, la cual, por medio del código en SQL, nos permite obtener el modelo relacional, esto es de gran ayuda ya que, al programar y crear nuestras tablas por medio de código, se obtiene muy fácilmente y con gran exactitud.

* Muñoz San Agustin Victoria Monserrat

A lo largo de esta práctica he aprendido la importancia de comprender y construir modelos de datos relacionales a partir de diagramas Entidad-Relación (ER). El objetivo de este aprendizaje ha sido desarrollar las habilidades necesarias para representar de manera estructurada y eficiente la información en bases de datos, un aspecto fundamental en la ingeniería de software y en la gestión de sistemas de información. Utilizando herramientas CASE, he podido ver cómo estas facilitan la transición del modelo conceptual al modelo lógico, permitiéndome crear representaciones visuales claras y precisas. Esta metodología no solo ha reforzado mi comprensión teórica sobre las relaciones entre entidades, sino que también me ha permitido abordar de manera más eficiente el diseño de bases de datos, minimizando redundancias y mejorando la optimización del almacenamiento de datos. Además, este proceso me ha enseñado que el diseño de bases de datos no es solo una tarea técnica, sino también un ejercicio de resolución de problemas y toma de decisiones sobre cómo organizar y acceder a la información de la mejor manera posible. He comprendido que un buen diseño relacional no solo facilita la implementación de sistemas, sino que también impacta directamente en el rendimiento y la escalabilidad de las aplicaciones que dependen de esos datos.

**Referencias**

Oracle. (2025). ¿Qué es una base de datos relacional (sistema de gestión de bases de datos relacionales)? Recuperado el 24 de marzo de 2025 de https://www.oracle.com/mx/database/what-is-a-relational-database/

IBM (2021). Estructura de base de datos relacional. Recuperado el 24 de marzo de 2025 de https://www.ibm.com/docs/es/control-desk/7.6.1.1?topic=design-relational-database-structure