Fundamento de Base de Datos

TESH
HUIXQUILUCAN

2.- Diseño de base de datos con el modelo E-R

2.1 El proceso de Diseño

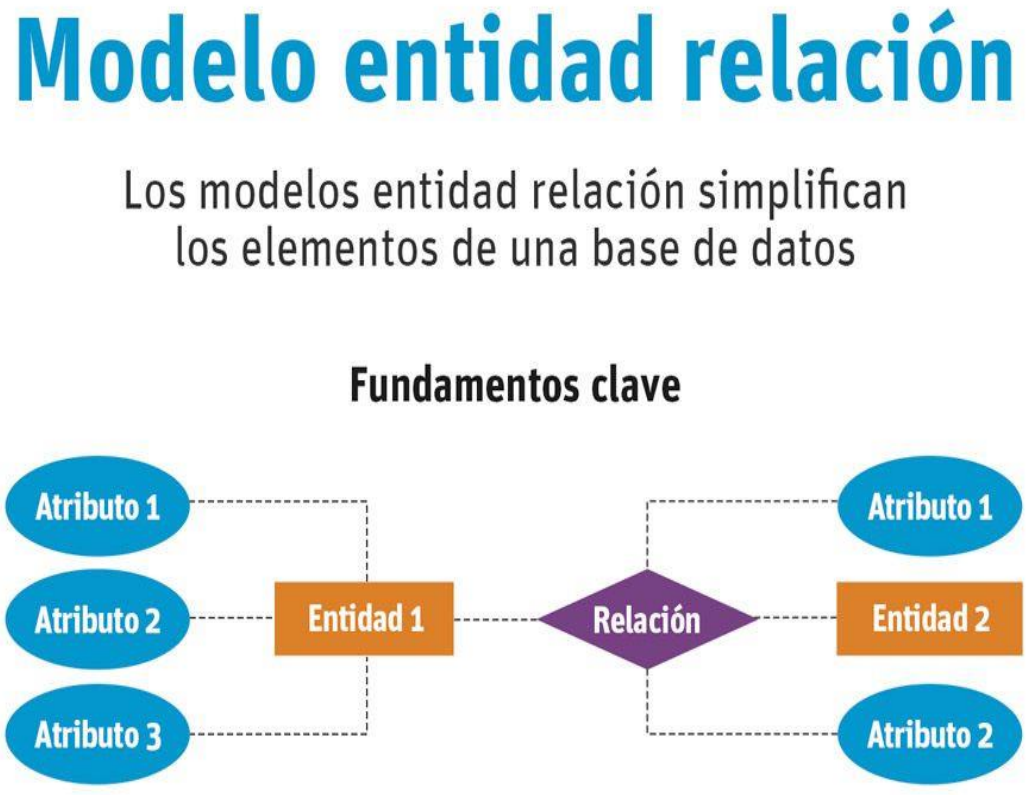
- **Determinar la finalidad de la base de datos**: Este paso inicial es crucial para definir el propósito de la base de datos y guiar el resto del proceso de diseño. Identificar claramente qué información se desea almacenar y cómo se utilizará ayuda a definir las entidades y relaciones necesarias.
- **Buscar y organizar la información necesaria**: Reúne todos los tipos de información que deseas registrar en la base de datos. Esto incluye nombres de productos, números de pedidos, y cualquier otro dato relevante para el propósito de la base de datos.
- **Crear el modelo Entidad-Relación**: Utiliza el modelo E-R para describir los datos, las relaciones entre los datos, y la semántica de los datos. Este modelo es útil para representar los significados e interacciones del mundo real en un esquema conceptual. Muchas herramientas de diseño de bases de datos se basan en los conceptos del modelo E-R debido a su utilidad.
- **Definir entidades y relaciones**: Las entidades representan objetos o conceptos del mundo real que son importantes para la base de datos, como productos o pedidos. Las relaciones describen cómo estas entidades se conectan entre sí, por ejemplo, un pedido puede estar relacionado con un producto.
- **Especificar atributos para las entidades**: Los atributos son las características o propiedades de las entidades. Por ejemplo, un producto puede tener atributos como nombre, precio, y cantidad en stock.
- **Establecer restricciones de dominio y consistencia**: Las restricciones de dominio especifican el tipo de datos que puede tener cada atributo (por ejemplo, entero, cadena de texto, fecha). Las restricciones de consistencia aseguran que los datos en la base de datos sean coherentes y válidos.
- **Considerar modelos E-R extendidos**: Para casos más complejos, puedes necesitar utilizar características extendidas del modelo E-R, como especialización, generalización,

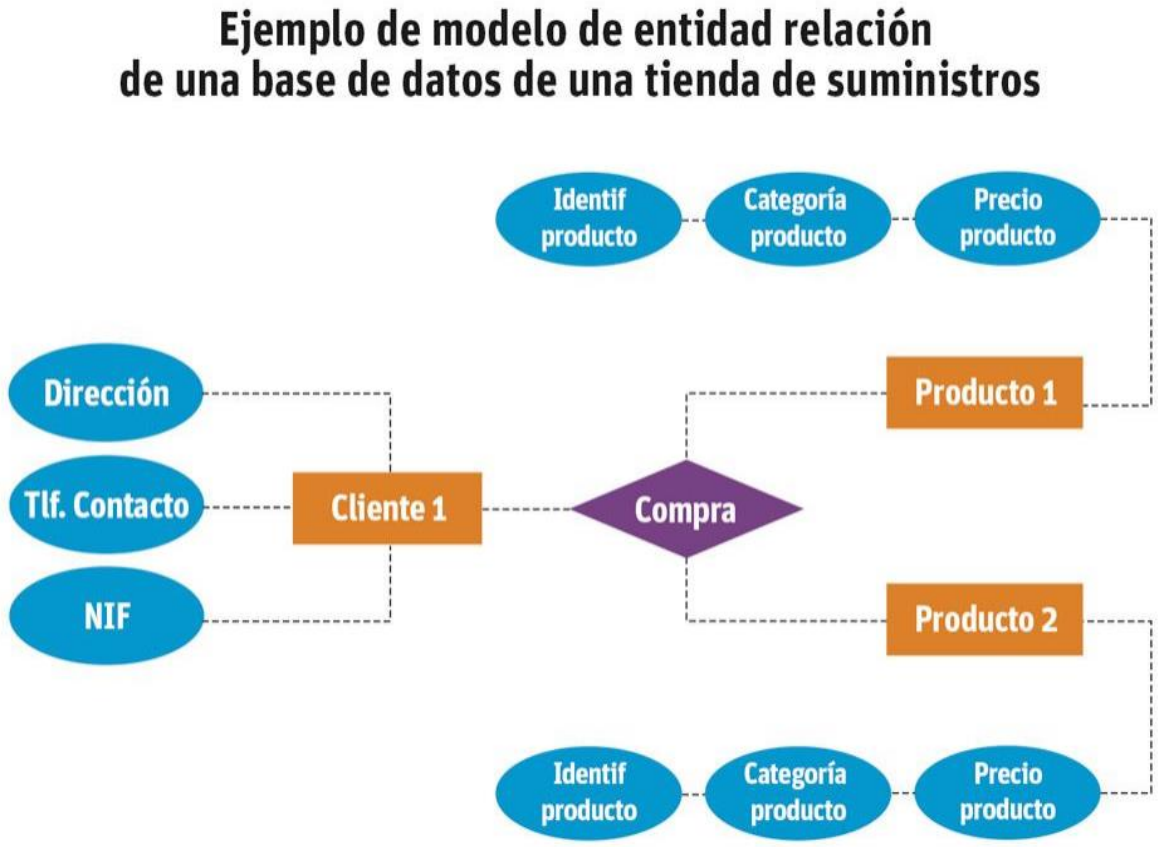
y herencia de atributos. Estas características permiten modelar relaciones más complejas y específicas entre las entidades.

- **Revisar y ajustar el modelo**: Una vez que hayas creado el modelo E-R inicial, es importante revisarlo y ajustarlo según sea necesario. Esto puede incluir la adición de entidades o relaciones faltantes, la corrección de errores, y la optimización del diseño para mejorar la eficiencia y la facilidad de uso.

2.2 Modelo Entidad Relación

Un modelo entidad-relación (ER) es un modelo conceptual utilizado en el diseño de bases de datos para representar las entidades relevantes dentro de un sistema y las relaciones entre ellas. En un diagrama entidad-relación, las entidades se representan como rectángulos, las relaciones como líneas conectadas entre las entidades y los atributos como óvalos dentro de las entidades.





2.3 Diseño con diagramas E-R

El diseño con diagramas entidad-relación (ER) es una técnica fundamental en el diseño de bases de datos y sistemas de información. Estos diagramas representan la estructura de una base de datos, mostrando las entidades (objetos o conceptos), sus atributos (características o propiedades), y las relaciones entre ellas. Los diagramas ER son esenciales para visualizar y planificar la estructura de una base de datos antes de su implementación, facilitando la comprensión y el análisis de los requisitos del sistema.

Componentes de un Diagrama ER

Entidades: Representadas por rectángulos, son los objetos o conceptos que existen en el sistema, como personas, productos, o eventos.

Relaciones: Representadas por diamantes, indican cómo las entidades se relacionan entre sí.

Atributos: Representados por óvalos, son las características o propiedades que describen las entidades.

***Cardinalidad**: Define las relaciones en términos de números, indicando cuántas instancias de una entidad están asociadas con instancias de otra entidad.

Ventajas de los Diagramas ER

- **Claridad**: Facilitan la comprensión de la estructura de la base de datos y las relaciones entre sus componentes.

- **Flexibilidad**: Permiten representar diferentes niveles de detalle, desde el modelo conceptual hasta el modelo físico, adaptándose a las necesidades específicas del proyecto.
- **Colaboración**: Herramientas como Miro y Lucidchart ofrecen funcionalidades para crear y compartir diagramas ER de manera colaborativa, facilitando el trabajo en equipo.

Creación de un Diagrama ER

- **Identificar las Entidades**: Comienza identificando las entidades principales del sistema.
- **Definir los Atributos**: Para cada entidad, define sus atributos.
- **Establecer las Relaciones**: Identifica cómo las entidades se relacionan entre sí y define las relaciones.
- **Determinar la Cardinalidad**: Especifica la cardinalidad de las relaciones para asegurar la integridad de los datos.
- **Revisar y Ajustar**: Revisa el diagrama para asegurar que representa correctamente el sistema y realiza ajustes según sea necesario.

Herramientas para Crear Diagramas ER

- **Miro**: Ofrece plantillas y herramientas intuitivas para crear diagramas ER de manera colaborativa.
- **Lucidchart**: Proporciona una plataforma para crear, compartir y colaborar en diagramas ER, con plantillas listas para usar y funcionalidades avanzadas.

Ejercicios:

Base de datos universitaria

Estudiante: Con atributos como ID, nombre, dirección, correo electrónico.

Profesor: Con atributos como ID, nombre, especialidad, departamento.

Curso: Con atributos como ID, nombre del curso, número de créditos.

Departamento Con atributos como ID, nombre del departamento, ubicación.

Una empresa de venta de productos electrónicos

Cliente: Con atributos como ID, nombre, dirección, correo electrónico, número de teléfono.

Producto: Con atributos como ID, nombre, descripción, precio, cantidad en stock.

Orden: Con atributos como ID, fecha de compra, estado de la orden, total.

Categoría: Con atributos como ID, nombre de la categoría, descripción.

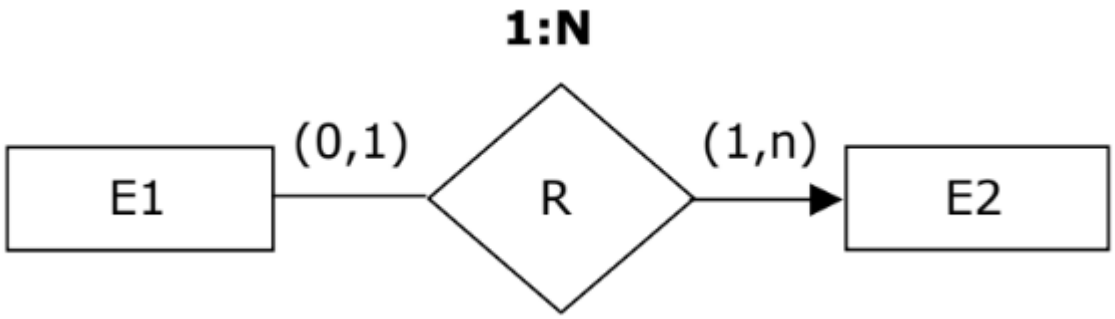
Proveedor: Con atributos como ID, nombre de la empresa, dirección, información de contacto.

Una clínica médica desea desarrollar un sistema de gestión de información para administrar sus operaciones diarias. Se te ha asignado la tarea de diseñar la base de datos para este sistema. La clínica tiene los siguientes requisitos:

Cada paciente tiene un número de identificación único, nombre, dirección, fecha de nacimiento y número de teléfono. Los médicos tienen un número de identificación único, nombre, especialidad y número de teléfono. Las citas tienen un número de identificación único, fecha y hora, médico asignado y paciente asociado. Las historias clínicas tienen un número de identificación único, fecha de creación, paciente asociado y detalles médicos (diagnósticos, tratamientos, prescripciones, etc.).

2.4 Modelo E-R Extendido

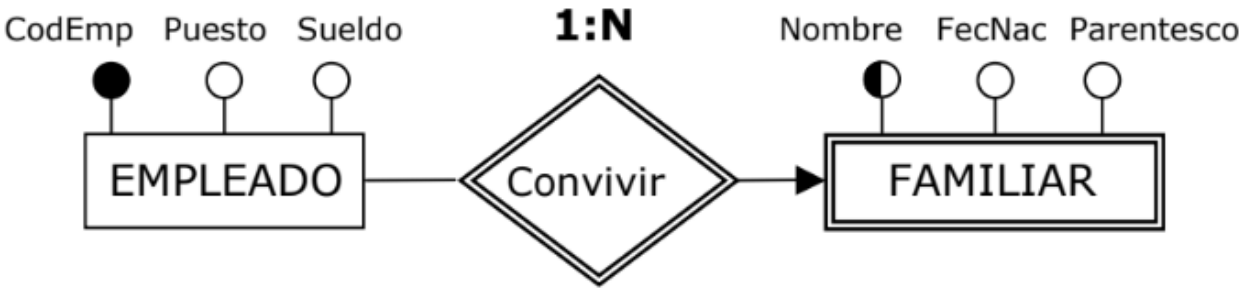
Los diagramas de entidad-relación extendida (EER, por sus siglas en inglés) son una versión avanzada de los diagramas de entidad-relación (ER) que se utilizan en el diseño de bases de datos. La extensión que los caracteriza introduce conceptos adicionales que permiten una representación más precisa y flexible de las relaciones y características de los datos en aplicaciones complejas. Aquí te explico los elementos básicos de los diagramas EER y cómo se comparan y contrastan con los diagramas ER estándar.



Elementos Básicos de los Diagramas EER

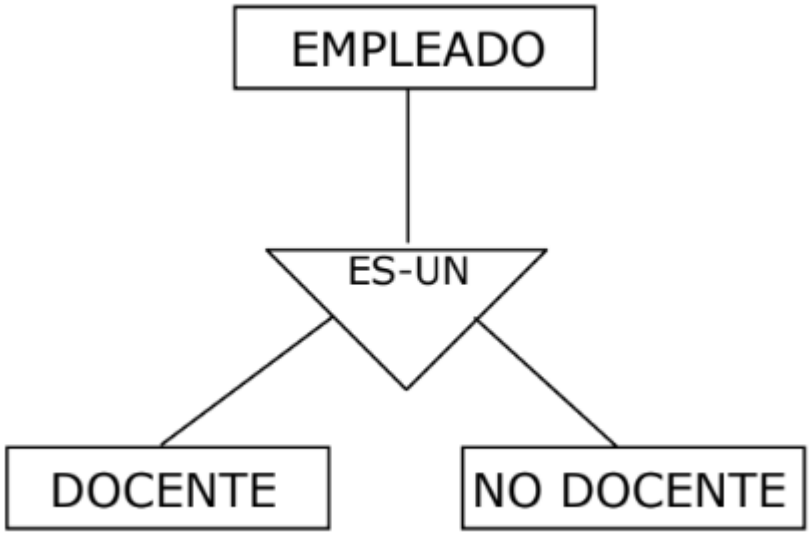
Entidades y Atributos

- **Entidades Fuertes**: Son similares a las entidades en los diagramas ER estándar. Representan objetos o conceptos con una existencia independiente.
- **Entidades Débiles**: Dependientes de otra entidad para su existencia. Se diferencian porque no tienen suficientes atributos para formar una clave primaria por sí solas.



Relaciones

- **Relaciones Regulares**: Similares a las relaciones en los diagramas ER, representan las interacciones entre dos o más entidades.
- **Relaciones de Especialización/Generalización**: Permiten modelar jerarquías de herencia entre entidades, donde una entidad puede ser subdividida en entidades más específicas que comparten algunos atributos con la entidad padre pero también tienen los suyos propios.
- **Categorización**: Es un caso especial de especialización/generalización donde una entidad puede ser una instancia de una entre varias clases padre, exclusivamente.



Atributos

Atributos Compuestos y Multivalorados: Similar a los ER diagramas, los atributos pueden ser tanto simples (ej. nombre, fecha de nacimiento) como compuestos (ej. dirección) o multivalorados (ej. teléfonos).

Entidad: Persona

- **Atributo Compuesto: Nombre**
- **Primer Nombre**
- **Segundo Nombre**
- **Apellido Paterno**
- **Apellido Materno**

En este ejemplo, "Nombre" es un atributo compuesto porque está formado por varios atributos más simples que son partes del nombre completo de una persona. Cada parte del nombre, como "Primer Nombre", "Segundo Nombre", "Apellido Paterno" y "Apellido Materno", son atributos simples que juntos forman el atributo compuesto "Nombre".

Atributos Derivados: Atributos cuyos valores se derivan de otros atributos o relaciones.

Entidad: Empleado

- **Fecha de Nacimiento**: 03/04/1985 (atributo simple)
- **Fecha Actual**: 21/04/2024 (atributo simple)
- **Edad**: calculado a partir de la diferencia entre la "Fecha Actual" y la "Fecha de Nacimiento".

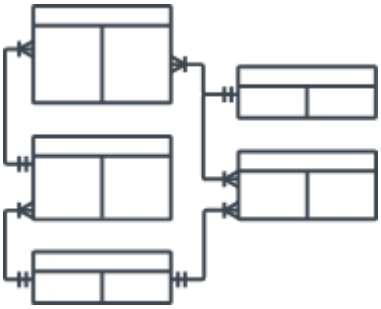
En este caso, "Edad" es un atributo derivado. Se calcula automáticamente usando la "Fecha de Nacimiento" del empleado y la "Fecha Actual". En una base de datos, esto podría implementarse a través de una función o un procedimiento almacenado que actualiza el valor de "Edad" cada vez que se consulta la entidad o cada vez que la fecha cambia.

Claves

- **Clave Primaria**: Identificador único para instancias de entidad.
  Clave Foránea: Atributo que crea un enlace entre dos entidades, apuntando a la clave primaria de otra entidad.

2.5 La notación E-R con UML

La notación E-R (Entidad-Relación) con UML (Lenguaje Unificado de Modelado) se utiliza para modelar sistemas, especialmente en el diseño de bases de datos y sistemas de software orientados a objetos. UML es un conjunto de notaciones y diagramas estándar que permite modelar distintos tipos de sistemas, incluyendo sistemas de software, hardware, y procesos de negocio. Aunque UML no preescribe un proceso o método estándar para desarrollar un sistema, ofrece una notación estándar y semánticas esenciales para el modelado de sistemas orientados a objetos.



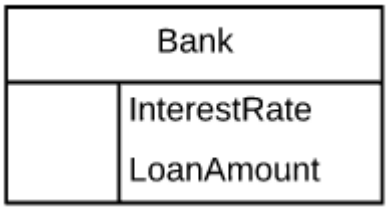
Símbolos físicos de diagramas ER

El modelo de datos físicos es el nivel más detallado de los diagramas entidad-relación y representa el proceso de agregar información a la base de datos. Los modelos ER físicos muestran todas las estructuras de tablas, incluidos nombre de columna, tipo de datos en la columna, restricciones de la columna, clave primaria, clave foránea y relaciones entre tablas.

Tal como se indica a continuación, las tablas son otra forma de representar entidades. Las partes clave de las tablas entidad-relación son las siguientes:

Campos

Los campos representan la parte de una tabla que establece los atributos de la entidad. Los atributos generalmente son vistos como columnas en la base de datos que el diagrama ER modela.



En la imagen anterior, TasadeInterés y CantidaddePréstamo son ambos atributos de la entidad, que están contenidos como campos.

Claves

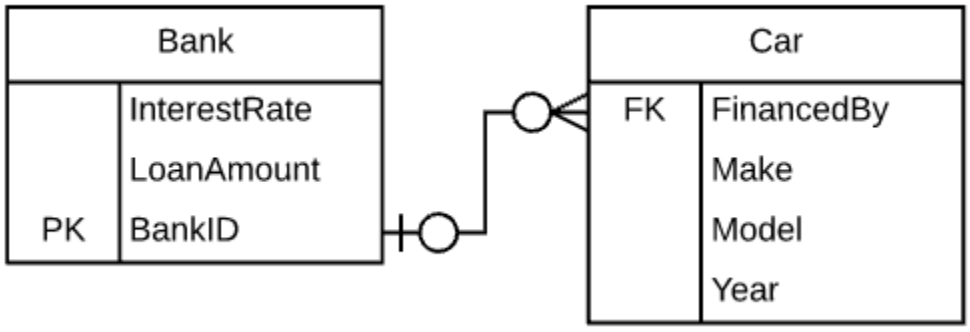
Las claves son una forma de categorizar atributos. Los diagramas ER ayudan a los usuarios a modelar sus bases de datos por medio de diversas tablas que aseguran que la base de datos esté organizada y sea eficiente y rápida. Las claves se usan para vincular diversas tablas en una base de datos entre sí de la manera más eficiente posible.

Claves primarias

Las claves primarias son un atributo o combinación de atributos que identifican de forma exclusiva una y solo una instancia de una entidad.

Claves extranjeras

Las claves extranjeras son creadas siempre que un atributo se relaciona con otra entidad en una relación de uno a uno o de uno a muchos.



Cada auto solo puede ser financiado por un banco, por lo tanto la clave primaria IdBanco de la tabla Banco se usa como la clave extranjera FinanciadoPor en la tabla Auto. Este IdBanco se puede usar como la clave extranjera para múltiples autos.

Tipos

Tipos se refiere al tipo de datos en el campo correspondiente en una tabla. Tipos también puede referirse a los tipos de entidades, los cuales describen la composición de una entidad; por ejemplo, los tipos de entidad de un libro son autor, título y fecha de publicación.

Entity
Field
Field
Field

Entity	
Key	Field
Key	Field
Key	Field

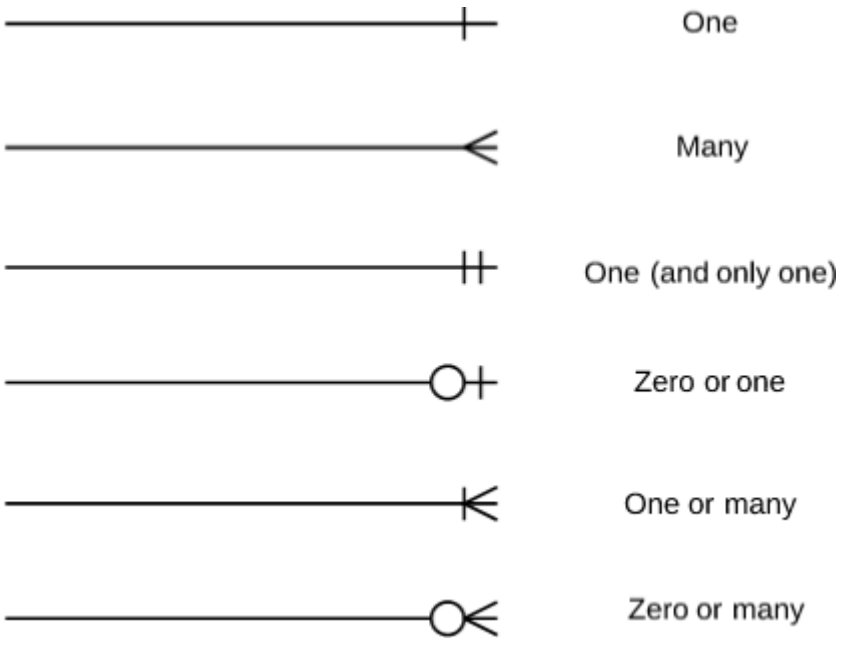
Entity	
Field	Type
Field	Type
Field	Type

Entity		
Key	Field	Type
Key	Field	Type
Key	Field	Type

Cardinalidad y ordinalidad

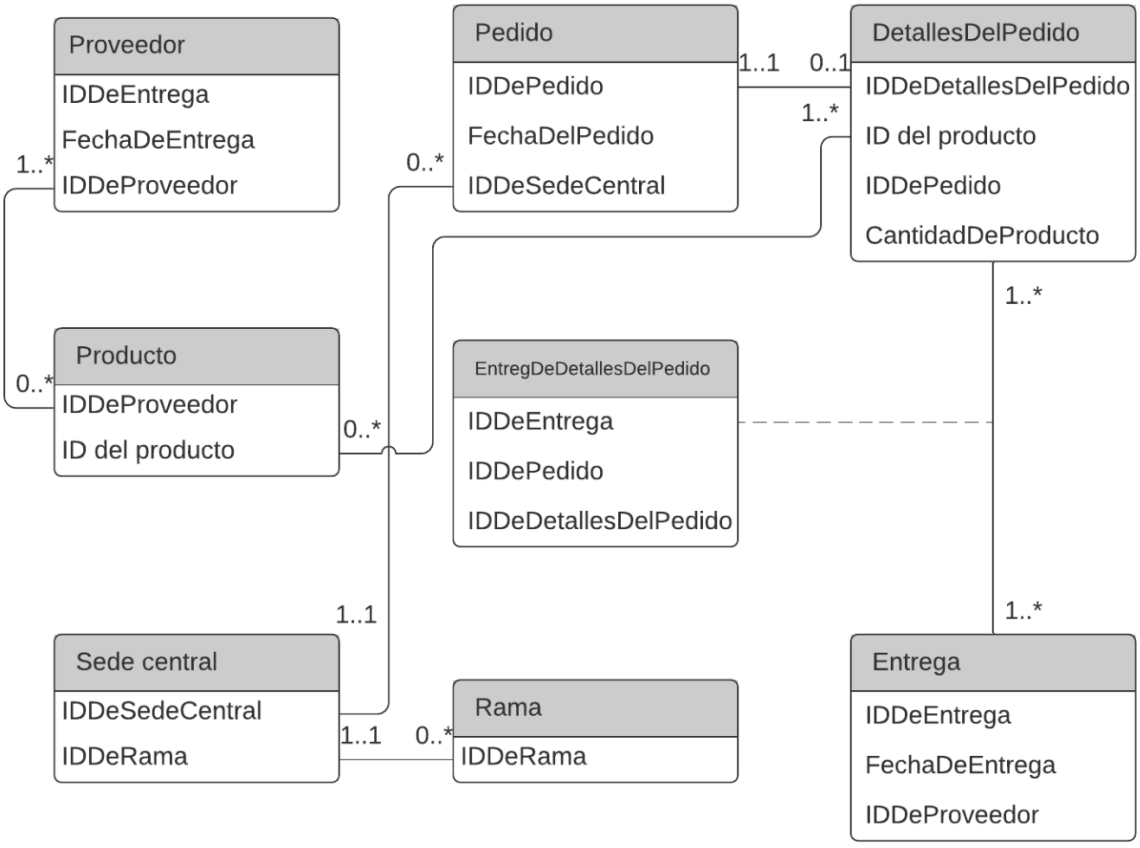
Cardinalidad se refiere al número máximo de veces que una instancia en una entidad se puede relacionar con instancias de otra entidad. Por otra parte, ordinalidad es el número mínimo de veces que una instancia en una entidad se puede asociar con una instancia en la entidad relacionada.

La cardinalidad y la ordinalidad se muestran a través del estilo de una línea y su extremo, según el estilo de notación seleccionado.



Ejemplo de diagrama entidad-relación (Notación UML)

Lindi H | April 23, 2024



3 Modelo relacional

3.1 Introducción al modelo relacional

3.2 conversión de modelo E-R a modelo relacional

3.3 Esquema de la base de deatos

Un esquema de base de datos proporciona una vista estructurada de cómo se organizan y relacionan las tablas dentro de una base de datos. Aquí tienes un ejemplo simple de un esquema de base de datos para una aplicación de gestión de biblioteca:

Esquema de la Base de Datos "Biblioteca"

Tabla: Libros

- **ID_Libro**: Identificador único del libro (entero, clave primaria)
- **Título**: Título del libro (cadena de caracteres)
- **Autor**: Autor del libro (cadena de caracteres)
- **Año_Publicación**: Año de publicación del libro (entero)
- **Editorial**: Editorial del libro (cadena de caracteres)

Tabla: Usuarios

- **ID_Usuario**: Identificador único del usuario (entero, clave primaria)
- **Nombre**: Nombre del usuario (cadena de caracteres)
- **Apellido**: Apellido del usuario (cadena de caracteres)
- **Email**: Dirección de correo electrónico del usuario (cadena de caracteres)

Tabla: Préstamos

- **ID_Préstamo**: Identificador único del préstamo (entero, clave primaria)
- **ID_Libro**: Identificador del libro prestado (entero, clave extranjera referenciando a Libros)
- **ID_Usuario**: Identificador del usuario que realizó el préstamo (entero, clave extranjera referenciando a Usuarios)
- **Fecha_Préstamo**: Fecha en que se realizó el préstamo (fecha)
- **Fecha_Devolución**: Fecha en que se debe devolver el libro (fecha)

En este esquema:

- La tabla Libros contiene información sobre los libros disponibles en la biblioteca.
- La tabla Usuarios almacena información sobre los usuarios que utilizan la biblioteca.
- La tabla Préstamos registra los préstamos de libros, relacionando los libros prestados con los usuarios que los tomaron prestados.

Relaciones entre Tablas

La tabla Préstamos tiene dos claves extranjeras que se relacionan con las tablas Libros y Usuarios. Esto significa que cada registro en la tabla Préstamos está asociado con un libro específico de la tabla Libros y con un usuario específico de la tabla Usuarios.

3.4 Restricciones

El modelo relacional es un modelo de datos basado en la lógica y la teoría de conjuntos, utilizado principalmente para gestionar bases de datos. Una característica crucial de este modelo es su conjunto de restricciones que aseguran la precisión y la integridad de los datos dentro de la base de datos relacional. A continuación, te detallo las principales restricciones en el modelo relacional:

1. **Restricciones de Dominio:**

Las restricciones de dominio especifican que los valores de una columna deben ser del tipo de dato declarado y deben cumplir con cualquier otra restricción definida para esa columna, como rangos de valores o formatos específicos. Por ejemplo, una columna de "edad" podría restringirse solo a valores numéricos positivos.

2. **Restricciones de Clave Primaria (Primary Key Constraints):**

Una clave primaria es un campo o conjunto de campos que identifica de manera única cada fila en una tabla. Las restricciones de clave primaria aseguran que no haya valores duplicados en la columna o conjunto de columnas que se han definido como clave primaria, y que ningún valor de clave primaria pueda ser null.

3. **Restricciones de Clave Extranjera (Foreign Key Constraints):**

Una clave extranjera en una tabla es un campo o conjunto de campos que se refiere a la clave primaria de otra tabla. Las restricciones de clave extranjera garantizan la integridad referencial

entre las tablas. Es decir, cualquier valor de la clave extranjera debe coincidir con un valor de la clave primaria en la tabla referenciada, o ser null si es permitido.

4. **Restricciones de Unicidad (Unique Constraints):**

Similar a la clave primaria, estas restricciones aseguran que todos los valores en una columna o conjunto de columnas sean únicos en toda la tabla. A diferencia de las claves primarias, las columnas con restricciones de unicidad pueden aceptar valores null.

5. **Restriciones de Integridad de Entidad:**

Esta restricción se asegura de que ningún campo que sea parte de la clave primaria pueda ser null, garantizando que cada fila y entidad pueda ser identificada de manera única.

6. **Restricciones de Integridad Referencial:**

Aparte de las claves extranjeras, esta categoría de restricciones puede incluir reglas adicionales que dicten cómo se deben manejar las operaciones de actualización y eliminación de las claves foráneas. Por ejemplo, la acción en cascada, que automáticamente elimina o actualiza las filas referenciadas cuando la fila a la que se hace referencia es eliminada o modificada.

7. **Restricciones de Comprobación (Check Constraints):**

Estas restricciones permiten especificar una condición lógica que los datos en una columna específica deben cumplir. Por ejemplo, que el valor en una columna de "porcentaje" debe estar entre 0 y 100.

3.4.1 Restricciones de Integridad de Entidad

Las restricciones de integridad de entidad son fundamentales en el modelo relacional para asegurar que cada fila en una tabla pueda ser identificada de manera única. Esta restricción se impone a través de la clave primaria de la tabla. Una clave primaria es un campo o conjunto de campos que tiene valores únicos para cada fila de la tabla, y que no puede contener valores nulos. Aquí te ofrezco un ejemplo detallado para ilustrar esta restricción:

Ejemplo:

Supongamos que tenemos una base de datos de un colegio. Dentro de esta base, existe una tabla llamada Estudiantes que guarda información relevante de los alumnos. Consideremos la siguiente estructura para la tabla Estudiantes:

- **ID_Estudiante** (entero, clave primaria)
- **Nombre** (cadena de caracteres)
- **Apellido** (cadena de caracteres)
- **Fecha_de_Nacimiento** (fecha)

En este caso, ID_Estudiante es la clave primaria de la tabla Estudiantes. Esta columna debe cumplir con las siguientes condiciones para respetar la restricción de integridad de entidad:

- **Unicidad:** Ningún ID_Estudiante puede repetirse en la tabla. Cada estudiante debe tener un ID_Estudiante único. Esto asegura que cada fila (es decir, cada estudiante) pueda ser identificada de manera única a través de su ID_Estudiante.
- **No nulidad:** El campo ID_Estudiante no puede estar vacío (es decir, no puede contener valores nulos). Cada estudiante debe tener un ID_Estudiante asignado.

Ejemplo de Operación que Violaría la Restricción:

- Imagina que intentamos insertar los siguientes registros en la tabla Estudiantes:
- Insertar (ID_Estudiante, Nombre, Apellido, Fecha_de_Nacimiento) VALUES (12345, 'Ana', 'Pérez', '2003-04-15')
- Insertar (ID_Estudiante, Nombre, Apellido, Fecha_de_Nacimiento) VALUES (12345, 'Luis', 'Martínez', '2004-09-22')

Este intento fallaría debido a la restricción de integridad de entidad. La razón es que estamos intentando usar el mismo ID_Estudiante (12345) para dos estudiantes diferentes, lo cual viola la unicidad requerida por la clave primaria.

Ejemplo de Operación Correcta:

- Insertar (ID_Estudiante, Nombre, Apellido, Fecha_de_Nacimiento) VALUES (12345, 'Ana', 'Pérez', '2003-04-15')
- Insertar (ID_Estudiante, Nombre, Apellido, Fecha_de_Nacimiento) VALUES (12346, 'Luis', 'Martínez', '2004-09-22')

3.4.2 Restricciones de Integridad Referencial

Las restricciones de integridad referencial son esenciales en un modelo relacional para mantener la coherencia y la validez de los datos entre tablas relacionadas. Estas restricciones aseguran que las relaciones entre tablas se mantengan correctas y coherentes al evitar la existencia de referencias inválidas o huérfanas entre las tablas. Aquí te proporciono un ejemplo detallado para ilustrar cómo funcionan estas restricciones:

Ejemplo:
Supongamos que tenemos dos tablas en una base de datos de una escuela: una tabla Estudiantes y otra tabla Calificaciones.

Tabla Estudiantes**:**

- **ID_Estudiante** (entero, clave primaria)
- **Nombre** (cadena de texto)
- **Apellido** (cadena de texto)

Tabla Calificaciones**:**

- **ID_Calificacion** (entero, clave primaria)
- **ID_Estudiante** (entero, clave extranjera)
- **Asignatura** (cadena de texto)
- **Nota** (entero)

La columna ID_Estudiante en la tabla Calificaciones es una clave extranjera que hace referencia a la columna ID_Estudiante en la tabla Estudiantes. Esto significa que cada entrada

en la columna ID_Estudiante de Calificaciones debe corresponder a un valor existente en la columna ID_Estudiante de Estudiantes, garantizando así que cada calificación está asociada con un estudiante existente.

Ejemplo de Operación que Violaría la Restricción:

Imagina que intentamos insertar el siguiente registro en la tabla Calificaciones:

- Insertar (ID_Calificacion, ID_Estudiante, Asignatura, Nota) VALUES (101, 9999, 'Matemáticas', 85)

Si en la tabla Estudiantes no existe un estudiante con ID_Estudiante igual a 9999, esta inserción fallará debido a la restricción de integridad referencial, ya que estamos intentando asignar una calificación a un estudiante que no existe.

Ejemplo de Operación Correcta:

- Insertar (ID_Estudiante, Nombre, Apellido) VALUES (12345, 'Ana', 'Pérez')
- Insertar (ID_Calificacion, ID_Estudiante, Asignatura, Nota) VALUES (101, 12345, 'Matemáticas', 85)

En este caso, el ID_Estudiante 12345 existe en la tabla Estudiantes antes de que se intente insertar en la tabla Calificaciones. Por lo tanto, la inserción en Calificaciones es exitosa y respeta la integridad referencial.

Consideraciones Adicionales:

- **Acciones en Cascada:** Si se elimina un estudiante, se puede configurar la base de datos para que automáticamente elimine todas sus calificaciones asociadas (ON DELETE CASCADE), o que impida la eliminación si existen calificaciones asociadas (ON DELETE RESTRICT).
- **Actualizaciones en Cascada:** Similar a las eliminaciones, si se actualiza el ID_Estudiante en la tabla Estudiantes, se pueden actualizar automáticamente todas las referencias correspondientes en Calificaciones (ON UPDATE CASCADE).

3.5 Integridad de dominio

La integridad de dominio es un aspecto fundamental del modelo relacional que se refiere a la definición de reglas válidas para los valores y tipos de datos permitidos en cada columna de una tabla. La integridad de dominio garantiza que todos los datos en una columna se adhieran a un conjunto definido de criterios, lo que ayuda a mantener la calidad y la coherencia de los datos en una base de datos.

Elementos Clave de la Integridad de Dominio

1. Tipo de Datos: Cada columna en una tabla está definida para contener un tipo específico de datos, como enteros, decimales, textos, fechas, etc. Esta definición previene que tipos de datos incorrectos sean almacenados en la columna.

2. Restricciones: Se pueden aplicar diversas restricciones en cada columna para limitar los valores que se pueden ingresar. Estas restricciones incluyen:

   - Valores Mínimos y Máximos: Definir un rango permitido para valores numéricos.
   - Longitud de Cadena: Establecer una longitud mínima y/o máxima para los datos de tipo texto.
   - Formatos Permitidos: Especificar un formato requerido, como el patrón de un correo electrónico o un número de teléfono.
   - Conjuntos de Valores Permitidos: Limitar los valores a un conjunto específico, por ejemplo, un estado civil que solo puede ser 'Soltero', 'Casado', 'Divorciado', etc.

3. Valores Predeterminados: Establecer valores automáticos para una columna cuando no se proporciona un valor explícito.

4. Restricciones de NULL: Definir si una columna puede o no aceptar valores nulos.

Ejemplo Práctico

Supongamos que tenemos una tabla llamada Empleados en una base de datos corporativa con las siguientes columnas:

- ID_Empleado (entero, clave primaria)
- Nombre (cadena de texto, no nulo)
- Edad (entero, valor entre 18 y 65)
- Email (cadena de texto, debe coincidir con un formato de email)

- Fecha_Ingreso (fecha, no nulo, por defecto es la fecha actual)
- Departamento (cadena de texto, valores permitidos: 'RRHH', 'Tecnología', 'Administración')

Definición de Integridad de Dominio

- ID_Empleado: Clave primaria, no puede ser nulo, debe ser un número único.
- Nombre: No puede ser nulo, lo que asegura que cada empleado tenga un nombre asignado.
- Edad: Debe ser un número entero entre 18 y 65. Esto garantiza que todos los empleados cumplan con la política de edad de la empresa.
- Email: Debe coincidir con un patrón de correo electrónico válido, como usuario@dominio.com.
- Fecha_Ingreso: No puede ser nula y su valor por defecto es la fecha en la que el registro es creado, asegurando que siempre tengamos registrado cuándo comenzó a trabajar un empleado.
- Departamento: Solo puede tener uno de los valores predefinidos ('RRHH', 'Tecnología', 'Administración'), lo cual facilita la gestión organizativa.

Beneficios de la Integridad de Dominio

Mantener la integridad de dominio ayuda a prevenir errores de entrada de datos, simplifica el análisis de datos y mejora la calidad general de los datos. Las bases de datos que aplican efectivamente la integridad de dominio son más robustas y confiables, facilitando operaciones como la generación de reportes, búsquedas y mantenimiento de datos de forma más eficiente y con menos errores.