

Unidad 2.

Diseño de Bases de Datos y el Modelo Entidad-Relación

I. Introducción al Diseño de Bases de Datos

El diseño de bases de datos constituye una fase fundamental en el ciclo de vida del desarrollo de sistemas de información. Su relevancia radica en que una estructura de datos bien concebida es la base para la eficiencia, la integridad y la escalabilidad de cualquier sistema que gestione información. Un diseño deficiente puede acarrear problemas significativos a largo plazo, como inconsistencia de datos, bajo rendimiento y dificultades para adaptarse a nuevos requisitos.

Importancia y Fases del Proceso de Diseño de Bases de Datos

El proceso de diseño de bases de datos es una actividad estructurada que se inicia con la comprensión de los requisitos de información de una organización y culmina con la implementación física de la base de datos. Este proceso se descompone típicamente en varias fases interconectadas, cada una con objetivos específicos y herramientas particulares.

Etapas del Proceso de Diseño

El desarrollo de una base de datos se articula a través de un proceso sistemático que comprende las siguientes etapas clave:

1. **Análisis de Requisitos:** Esta fase inicial se enfoca en la recopilación y el análisis exhaustivo de las necesidades de información de la organización y de los usuarios finales. El resultado de esta etapa es una especificación detallada que define la funcionalidad y los datos requeridos por el sistema.
2. **Diseño Conceptual:** Se elabora un esquema de alto nivel de la base de datos, independiente del sistema de gestión de bases de datos (SGBD) específico. En este contexto, el **Modelo Entidad-Relación (E-R)** es la herramienta principal utilizada para representar de manera abstracta la estructura de los datos, las entidades y sus interrelaciones.
3. **Diseño Lógico:** En esta etapa, el esquema conceptual se transforma en un modelo de datos compatible con el SGBD seleccionado, comúnmente el modelo relacional. Se definen las tablas, los atributos, las claves primarias y foráneas, y se formalizan las relaciones entre las entidades.
4. **Refinamiento de los Esquemas:** El esquema lógico resultante es sometido a un proceso de optimización mediante técnicas de **normalización**. El objetivo es minimizar la redundancia de datos y las anomalías de actualización, garantizando así la integridad y

consistencia del esquema.

5. **Diseño Físico de la BD:** Esta fase aborda la implementación a nivel físico. Se toman decisiones estratégicas sobre las estructuras de almacenamiento, como la creación de índices, la organización de archivos y las particiones, con el fin de optimizar el rendimiento y la eficiencia del acceso a los datos.
6. **Diseño de Aplicaciones y de la Seguridad:** Se diseñan las aplicaciones que interactuarán con la base de datos y se establecen las políticas de seguridad. Esto incluye la definición de los privilegios de acceso, los roles de usuario y los mecanismos de control para proteger la información contra accesos no autorizados.

Se destacan tres etapas, el **Diseño Conceptual**, es el proceso que se enfoca en describir los datos a un alto nivel de abstracción, de manera independiente de cualquier sistema de gestión de bases de datos (SGBD) específico. La herramienta predominante para esta fase es el Modelo Entidad-Relación (ER), que permite representar los objetos del mundo real y sus interconexiones.¹ Posteriormente, el **Diseño Lógico** transforma este modelo conceptual en un esquema de datos específico para un modelo de datos particular, como el relacional, definiendo tablas, atributos y restricciones de integridad adecuadas para el SGBD seleccionado. Finalmente, el **Diseño Físico** se encarga de la implementación concreta del diseño lógico en un SGBD real, abordando aspectos de almacenamiento, índices y optimización del rendimiento para asegurar una operación eficiente.¹

Diversas fuentes bibliográficas resaltan la importancia de estas etapas. Elmasri (2007) proporciona una sólida introducción a la metodología de bases de datos y su entorno en los Capítulos 1 y 2 de *Fundamentos de bases de datos*.¹ El Capítulo 12 de la misma obra profundiza en el proceso de diseño e implementación.¹ Begg y Connolly (2005), en su Capítulo 9 de *Sistemas de bases de datos*, abordan las etapas clave del desarrollo y diseño de sistemas de bases de datos, incluyendo la planificación y la definición del sistema.² Ramakrishnan (2007) dedica su Capítulo 2 de *Sistemas de gestión de bases de datos* a los principios de diseño, sentando una base esencial.³ Asimismo, Piattini Velthuis et al. (2007) en *Tecnología y diseño de bases de datos* y Nieto Bernal y Capacho Portilla (2017) en *Diseño de base de datos* ofrecen una perspectiva general y práctica del proceso de modelado, diseño e implementación.⁵

Visión General del Proceso de Diseño: Conceptual, Lógico y Físico

Profundizando en las fases del diseño, la **fase conceptual** tiene como objetivo principal crear un modelo de datos de alto nivel que represente la información de la organización de una manera que sea comprensible tanto para los técnicos como para los usuarios del negocio. Este modelo es independiente de la tecnología, lo que significa que no se preocupa por cómo se almacenarán los datos, sino por qué información se necesita y cómo se relaciona. El Modelo ER es la herramienta estándar para esta fase, permitiendo una representación clara de las entidades, sus atributos y las relaciones entre ellas.

La **fase lógica** toma el modelo conceptual y lo transforma en un esquema en un modelo de datos específico. Para el modelo relacional, esto implica definir tablas, sus columnas (atributos), las claves primarias que identifican únicamente cada fila, y las claves foráneas que establecen las relaciones entre tablas. También se definen las restricciones de integridad, asegurando la validez y consistencia de los datos.

Finalmente, la **fase física** se ocupa de la implementación real de la base de datos. Esto incluye la definición de estructuras de almacenamiento en disco, la elección de métodos de acceso (como índices) para optimizar la recuperación de datos, y otras consideraciones de rendimiento que son específicas del SGBD elegido.

Es importante comprender que, si bien los textos académicos presentan estas fases de manera secuencial, la interdependencia entre ellas es profunda. Un diseño conceptual robusto y bien definido es la base indispensable para las fases subsiguientes. Si el modelo conceptual contiene errores o es incompleto, estos defectos se propagarán y se amplificarán en las fases lógica y física, lo que podría resultar en una base de datos ineficiente, difícil de mantener y que no cumple adecuadamente con los requisitos del usuario.

Por lo tanto, la **inversión de tiempo y esfuerzo en la fase conceptual**, aunque pueda parecer abstracta, **es crítica** para evitar costosas reingenierías en etapas posteriores del desarrollo.

Además, el proceso de diseño de bases de datos no es estrictamente lineal, sino que a menudo se comporta como un proceso iterativo. Las lecciones aprendidas en una fase posterior, como la detección de problemas de rendimiento durante la fase física, pueden requerir ajustes y refinamientos en el diseño lógico o incluso en el conceptual. La mención de "*prototyping*" en Begg y Connolly (Capítulo 9)² sugiere esta naturaleza adaptable. Un diseñador de bases de datos competente no solo sigue una secuencia de pasos, sino que también es capaz de retroalimentar y refinar el diseño a medida que surgen nuevos requisitos o se descubren problemas, lo que garantiza que la base de datos evolucione para satisfacer las necesidades cambiantes del negocio.

Elmasri (2007) - Capítulo 12, es un recurso esencial para comprender el proceso completo de diseño e implementación.¹ Begg y Connolly (2005) en su Capítulo 9 también cubren el ciclo de vida del desarrollo de sistemas de bases de datos y las etapas de diseño.² De Miguel Castaño (2000) en Diseño de bases de datos relacionales aborda explícitamente el "Diseño conceptual: modelo E/R" y el "Diseño lógico: modelo relacional", lo que lo convierte en una fuente directa para entender la distinción y el flujo entre estas fases.⁹

II. El Modelo Entidad-Relación (ER)

El Modelo Entidad-Relación (ER) es una herramienta de modelado de datos de alto nivel que permite representar la estructura de la información de una organización de manera intuitiva y gráfica. Es el pilar del diseño conceptual de bases de datos, facilitando la comunicación entre analistas, diseñadores y usuarios finales.

Conceptos Fundamentales: Entidades, Atributos (Tipos y Claves)

En el corazón del Modelo ER se encuentran las **entidades**, que son objetos o conceptos del mundo real que se desean representar en la base de datos. Una entidad puede ser tangible (como un "Empleado" o un "Producto") o intangible (como un "Pedido" o un "Curso"). Un **conjunto de entidades** es una colección de entidades del mismo tipo, por ejemplo, todos los empleados de una empresa.¹

Las entidades son descritas por **atributos**, que son propiedades o características que las definen. Los atributos pueden clasificarse de varias maneras:

- **Simples vs. Compuestos:** Un atributo simple es indivisible (ej., Nombre), mientras que un atributo compuesto puede descomponerse en sub-atributos (ej., Dirección puede tener Calle, Número, Ciudad, País).
- **Monovalorados vs. Multivalorados:** Un atributo monovalorado tiene un único valor para cada entidad (ej., FechaNacimiento), mientras que un atributo multivalorado puede tener múltiples valores (ej., Teléfono de una persona).¹
- **Almacenados vs. Derivados:** Un atributo almacenado se guarda explícitamente en la base de datos (ej., FechaNacimiento), mientras que un atributo derivado se calcula a partir de otros atributos y no se almacena directamente (ej., Edad derivada de FechaNacimiento y la fecha actual).¹

Para identificar únicamente las entidades dentro de un conjunto de entidades, se utilizan las **claves**:

- **Superclave:** Es un conjunto de uno o más atributos que, tomados colectivamente, permiten identificar de forma única una entidad en un conjunto de entidades.
- **Clave Candidata:** Es una superclave mínima, es decir, una superclave de la que no se puede eliminar ningún atributo sin perder la propiedad de unicidad. Un conjunto de entidades puede tener varias claves candidatas.
- **Clave Primaria:** De todas las claves candidatas, se selecciona una para identificar de forma única las entidades en el mundo real. Esta es la clave primaria y es fundamental para la integridad de la base de datos.
- **Clave Alternativa:** Las claves candidatas que no son elegidas como clave primaria se denominan claves alternativas.¹

Elmasri (2007) en su Capítulo 3 de *Fundamentos de bases de datos* es un recurso principal para estos conceptos, cubriendo extensivamente los tipos de entidades, conjuntos de entidades, atributos y claves.¹ Silberschatz et al. (2014) en su Capítulo 2 de *Fundamentos de bases de datos* también son directamente relevantes para la definición de estos conceptos básicos.¹¹ Begg y Connolly (2005) dedican su Capítulo 11 a estos temas, incluyendo tipos de entidades/instancias y atributos.² Date (2001) introduce el Modelo Entidad/Relación en su Capítulo 13 de *Introducción a los sistemas de bases de datos*.¹²

Relaciones: Tipos, Grados, Roles y Restricciones (Cardinalidad y Participación)

Las **relaciones** describen cómo las entidades se asocian entre sí. Un **tipo de relación** define la asociación entre tipos de entidades, mientras que un **conjunto de relaciones** es la colección de instancias de esa asociación.

El **grado de una relación** se refiere al número de tipos de entidades que participan en ella. Las relaciones más comunes son binarias (entre dos tipos de entidades), pero también pueden ser ternarias (entre tres) o de mayor grado. Las **relaciones recursivas** ocurren cuando el mismo tipo de entidad participa múltiples veces en una relación, desempeñando diferentes **roles** (ej., un empleado "supervisa" a otro empleado; aquí, "supervisor" y "supervisado" son roles).²

Las **restricciones de cardinalidad** (también conocidas como multiplicidad) definen el número de instancias de una entidad que pueden asociarse con instancias de otra entidad en una relación. Las cardinalidades más comunes son:

- **Uno a uno (1:1):** Una instancia de la entidad A se relaciona con una única instancia de la entidad B, y viceversa.
- **Uno a muchos (1:N):** Una instancia de la entidad A se relaciona con múltiples instancias de la entidad B, pero una instancia de B se relaciona con una única instancia de A.
- **Muchos a muchos (M:N):** Múltiples instancias de la entidad A se relacionan con múltiples instancias de la entidad B, y viceversa.¹

Las **restricciones de participación** especifican si la participación de una entidad en una relación es obligatoria o no:

- **Participación Total (Obligatoria):** Cada instancia de una entidad debe participar en al menos una instancia de la relación (ej., cada "Empleado" debe trabajar en un "Departamento").
- **Participación Parcial (Opcional):** Una instancia de una entidad puede participar o no en la relación (ej., un "Empleado" puede "Dirigir" un "Proyecto", pero no todos los empleados dirigen proyectos).¹

Elmasri (2007) en el Capítulo 3 de *Fundamentos de bases de datos* detalla los tipos de relaciones, conjuntos de relaciones, roles y restricciones estructurales.¹ Silberschatz et al. (2014) en su Capítulo 2 también abordan estos aspectos.¹¹ Begg y Connolly (2005) en su Capítulo 11 cubren el grado de relación, relaciones recursivas, multiplicidad, cardinalidad y participación.²

Diagramas Entidad-Relación: Notaciones y Construcción

Los diagramas Entidad-Relación son representaciones gráficas del modelo conceptual. Existen varias notaciones, siendo las más comunes la notación de Chen, la notación Pata de Gallo (Crow's Foot) y la notación de Elmasri/Navathe. Aunque los símbolos pueden variar, los conceptos subyacentes son los mismos.

La **capacidad** de un diseñador para **traducir los requisitos del negocio** a un diagrama ER comprensible es **fundamental**.

El diagrama ER no es solo una herramienta técnica; es un lenguaje visual que facilita la comunicación entre los expertos en bases de datos y los usuarios finales que no tienen conocimientos técnicos. Una notación clara y consistente ayuda a validar el diseño conceptual con los interesados del negocio, asegurando que la base de datos resultante realmente satisfaga sus necesidades. Esto subraya la importancia de la fase conceptual como puente entre el mundo del negocio y el mundo técnico.

Elmasri (2007) incluye en su Capítulo 3 "ER diagrams, naming conventions, and design issues" y un "Example of another notation: UML class diagrams", lo que es excelente para cubrir múltiples notaciones y aspectos prácticos.¹ Begg y Connolly (2005) en su Capítulo 11 abordan el uso de UML para la diagramación y la identificación/corrección de "connection traps" (trampas de conexión), como las multiplicativas o de bucle.² Silberschatz et al. (2014) en su Capítulo 2 presentan la notación estándar utilizada en su texto, que es muy común en el ámbito académico.¹¹

Aspectos Clave del Diseño ER: Identificación y Modelado de Requisitos

El proceso de identificación de requisitos es el punto de partida para cualquier diseño de base de datos.

Se discute la importancia de técnicas de determinación de hechos (*fact-finding techniques*) para recolectar información precisa sobre las necesidades del usuario y del sistema. Estas técnicas incluyen entrevistas, observación, análisis de documentos existentes y cuestionarios. Begg y Connolly (2005) dedican un capítulo entero a estas técnicas, enfatizando cómo la recopilación y análisis de requisitos de los usuarios son cruciales para el diseño de la base de datos.²

Una vez recolectados, los requisitos funcionales y no funcionales se traducen sistemáticamente

en los componentes del modelo ER: entidades, atributos y relaciones. Por ejemplo, un requisito como "la empresa necesita almacenar información sobre sus empleados, incluyendo su nombre, dirección y número de teléfono" se traduce en una entidad Empleado con atributos Nombre, Dirección (compuesto) y Teléfono (multivalorado).

Un aspecto crucial a considerar es que las restricciones de cardinalidad y participación son más que simples símbolos en un diagrama; representan las reglas de negocio inherentes al modelo de datos.

Si estas restricciones no se identifican y modelan correctamente en la fase ER, la base de datos resultante no podrá garantizar la integridad y la consistencia de los datos. Por ejemplo, si una relación obligatoria se modela erróneamente como opcional, se podrían insertar datos inconsistentes, lo que llevaría a problemas de calidad de datos y a decisiones erróneas.

Esto significa que el modelado de restricciones en el ER es una formalización crítica de las reglas de negocio, y su correcta identificación es tan importante como la identificación de las entidades y relaciones mismas.

III. Elementos Avanzados del Modelo Entidad-Relación (EER)

El Modelo Entidad-Relación Extendido (EER) amplía las capacidades del Modelo ER básico para permitir un modelado conceptual más rico y preciso, especialmente para aplicaciones complejas que requieren representar jerarquías y tipos de relaciones más sofisticados.

Conjuntos de Entidades Débiles: Definición, Identificación y Representación

Un **conjunto de entidades débil** es un tipo de entidad cuya existencia depende de la existencia de otra entidad, conocida como **entidad fuerte** o **entidad identificadora**. A diferencia de las entidades fuertes, un conjunto de entidades débil no posee una clave primaria propia que lo identifique únicamente por sí mismo. En su lugar, se identifica mediante una combinación de su **clave parcial** (también llamada discriminador) y la clave primaria de la entidad fuerte de la que depende.¹

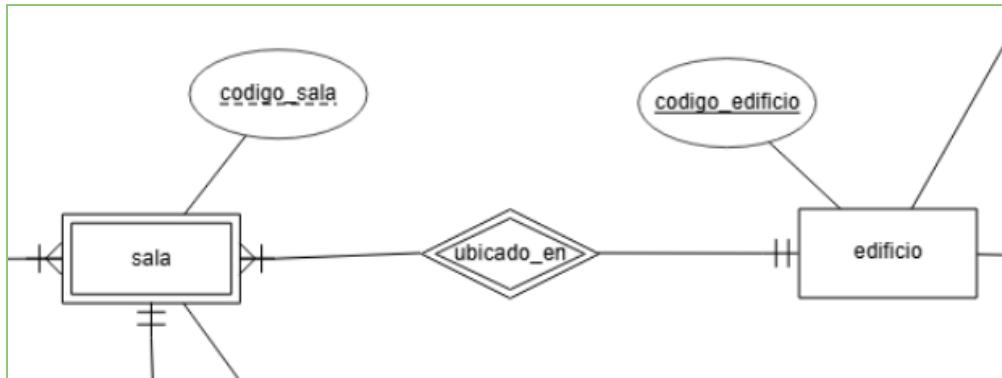
Para identificar entidades débiles en un escenario de requisitos, se busca la dependencia existencial.

Caso *Edificios y Salas de Conferencias*: Usando el ejemplo de edificios y salas de conferencias, el Edificio es una entidad fuerte, mientras que la Sala de Conferencias es una entidad débil.

Un **Edificio** es una entidad fuerte porque tiene un identificador único propio, como un *codigo_edificio* (ej., E001, E002). No necesita de ninguna otra entidad para ser identificado o existir.

Una **Sala de Conferencias** es una entidad débil por las siguientes razones:

- **Dependencia Existencial:** Una sala de conferencias no tiene sentido por sí sola; siempre pertenece a un edificio específico. Si el edificio se demuele o deja de existir en el sistema, las salas de conferencias asociadas a ese edificio también "desaparecen".
- **Identificación:** Una sala de conferencias a menudo se identifica por un *codigo_sala* (ej., 101, 205, 303). Sin embargo, este *codigo_sala* solo es único *dentro del contexto de un mismo edificio*. El edificio "Torre Central" puede tener una sala 101, y el edificio "Anexo Norte" también puede tener una sala 101. Para identificar de forma única una sala de conferencias en toda la base de datos, necesitamos combinar su *codigo_sala* (su **clave parcial**) con el *codigo_edificio* del edificio al que pertenece (la clave primaria de la entidad fuerte).



En los diagramas EER, las **entidades débiles** se representan con un **rectángulo doble**, y la **relación** que las conecta con su entidad fuerte identificadora se representa con un **rombo doble** y una línea de conexión gruesa.¹ La **clave parcial** se subraya con una **línea discontinua**.

La dependencia existencial de una entidad débil tiene una implicación directa y profunda en la integridad de los datos. Si la entidad fuerte identificadora se elimina, todas las entidades débiles asociadas deben ser eliminadas en cascada para mantener la consistencia.

Esto no es solo una regla de modelado conceptual, sino una restricción de integridad referencial crítica que debe ser implementada en el diseño lógico y físico de la base de datos (por ejemplo, mediante claves foráneas con acciones ON DELETE CASCADE). Esto significa que la correcta identificación y modelado de entidades débiles en la fase conceptual es un requisito previo para asegurar la integridad referencial y prevenir datos huérfanos o inconsistentes en el sistema implementado.

Características del Modelo E-R Extendido (EER)

El EER introduce conceptos avanzados para modelar relaciones más complejas:

- **Generalización y Especialización (Jerarquías IS-A):**
 - La **generalización** es un proceso ascendente que combina tipos de entidades con características comunes en una entidad de nivel superior llamada **superclase**. Por ejemplo, Empleado y Cliente pueden generalizarse a Persona.
 - La **especialización** es un proceso descendente que divide una superclase en subclases basadas en características distintivas o roles específicos. Por ejemplo, Empleado puede especializarse en Ingeniero y Gerente, cada una con atributos o relaciones adicionales.¹
 - Estas jerarquías se conocen como jerarquías "IS-A" (es-un), ya que una instancia de la subclase "es un" tipo de la superclase. Las subclases heredan todos los atributos y relaciones de sus superclases.¹
 - Se definen **restricciones de participación** (total/parcial) y **disyunción**

(disjunta/solapada) para especificar cómo las entidades de la superclase se relacionan con sus subclases. Una restricción disjunta significa que una instancia de la superclase puede pertenecer a una sola subclase, mientras que solapada permite pertenecer a varias. Una participación total significa que cada instancia de la superclase debe pertenecer a al menos una subclase.¹ Date (2001) aborda las jerarquías de tipos y la herencia en su Capítulo 19.¹²

- **Agregación y Composición:**

- La **agregación** permite modelar una relación como una entidad de orden superior para que pueda participar en otra relación. Es útil para representar relaciones "parte-todo" o "es-un-componente-de" donde el "todo" puede existir sin las "partes".²
- La **composición** es una forma más fuerte de agregación, donde las partes no pueden existir independientemente del todo. Si el todo se elimina, las partes también se eliminan.

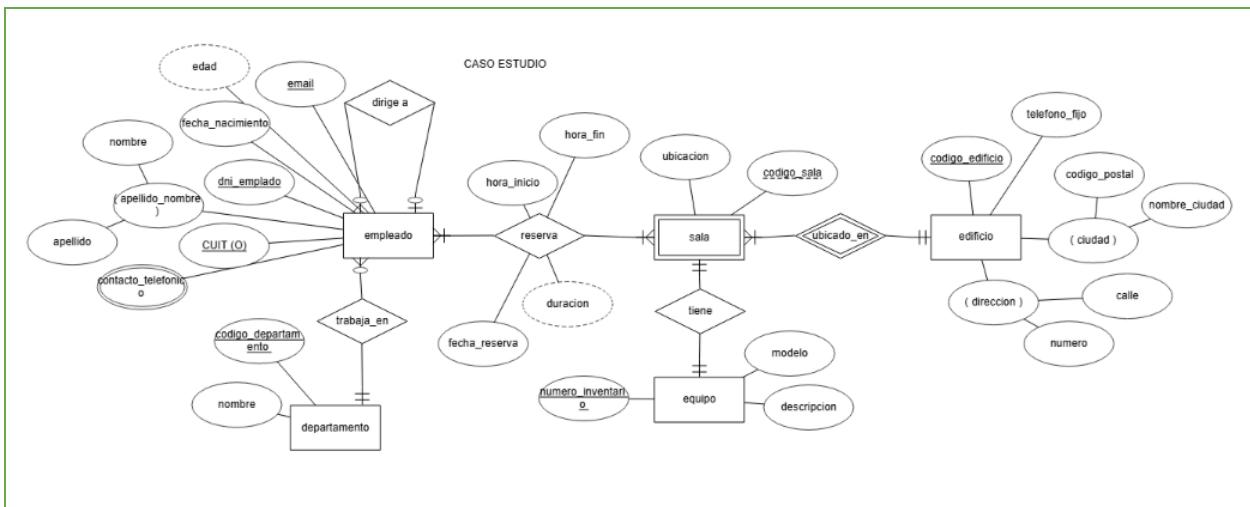
- **Categorías (Tipos Unión):**

- Las **categorías** permiten modelar una subclase que puede tener múltiples superclases, representando una "unión" de diferentes tipos de entidades. Por ejemplo, una entidad Propietario puede ser una Persona o una Compañía. Esto es útil cuando una entidad puede ser de varios tipos diferentes, pero se trata de manera uniforme en ciertos contextos.¹

La introducción de entidades débiles, generalización/especialización, agregación y categorías va más allá de la simple representación de objetos y sus conexiones. Estas características del EER permiten modelar matrices del mundo real que son difíciles o imposibles de capturar con el ER básico. Por ejemplo, una entidad débil no solo "existe" sino que "depende" de otra, lo que implica una restricción existencial. La especialización/generalización permite modelar jerarquías de conceptos ("es-un") que reflejan clasificaciones naturales. Esto significa que el EER no es solo una extensión técnica, sino una herramienta crucial para aumentar la *fidelidad semántica* del modelo conceptual, acercándolo más a la complejidad y las reglas de negocio del dominio real. Un modelo EER bien construido reduce la ambigüedad y mejora la comprensión del sistema de información. Elmasri (2007) en su Capítulo 4 es la fuente principal para todos estos conceptos de EER.¹

IV. Reducción del Modelo ER/EER a Esquemas Relacionales

Esta sección describe el proceso de transformación sistemática de un modelo conceptual (ER o EER) en un esquema lógico relacional, que es la base para la implementación de la base de datos en un SGBD relacional.



Principios de Mapeo entre Modelos Conceptual y Lógico

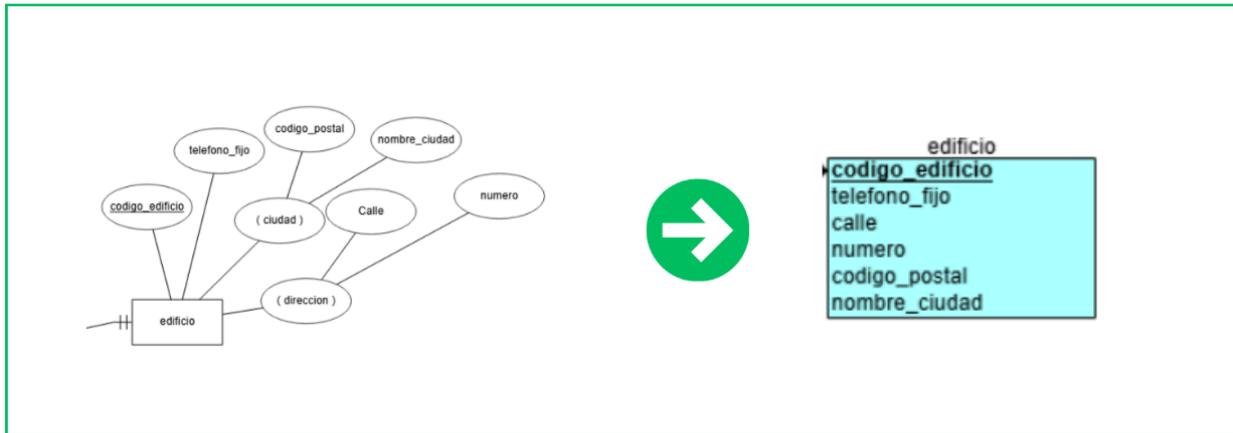
El objetivo fundamental del mapeo es traducir los constructos de alto nivel del modelo ER/EER (entidades, atributos, relaciones) a las estructuras del modelo relacional: tablas (o relaciones), columnas (atributos), claves primarias y claves foráneas, y otras restricciones de integridad. Este es un proceso sistemático que se basa en un conjunto de reglas bien definidas, lo que garantiza una transformación correcta y consistente del diseño conceptual al diseño lógico.¹

Elmasri (2007) dedica su Capítulo 7 a este proceso, cubriendo el mapeo de ER a relacional y de EER a relacional.¹ Silberschatz et al. (2014) en su Capítulo 7 también abordan este proceso de mapeo.¹¹ De Miguel Castaño (2000) en Diseño de bases de datos relacionales y Piattini Velthuis et al. (2007) en Tecnología y diseño de bases de datos también cubren aspectos de esta transformación.⁵

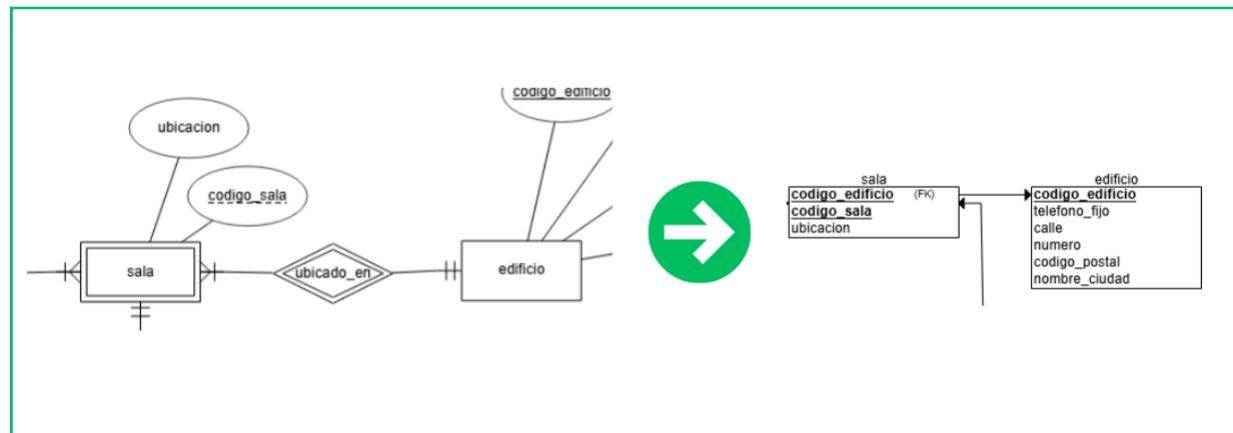
Reglas de Transformación para Entidades Fuertes y Débiles

Las reglas de transformación para entidades son las siguientes:

- **Entidades Fuertes:** Cada tipo de entidad fuerte en el diagrama ER/EER se mapea directamente a una tabla relacional. Los atributos simples de la entidad se convierten en columnas de la tabla. La clave primaria de la entidad se designa como la clave primaria de la tabla relacional.¹ Por ejemplo, una entidad EDIFICIO con clave codigo_edificio y atributo telefono se convierte en la tabla EDIFICIO (codigo_edificio, telefono).



- **Entidades Débiles:** Cada tipo de entidad débil se mapea a una tabla relacional. La clave primaria de esta nueva tabla se forma combinando la clave parcial (discriminador) de la entidad débil con la clave primaria de la entidad fuerte identificadora. La clave primaria de la entidad fuerte identificadora se convierte en una clave foránea en la tabla de la entidad débil, estableciendo la dependencia existencial.¹ Por ejemplo, si SALA es una entidad débil de EDIFICIO, la tabla resultante podría ser SALA (*codigo_edificio*, *codigo_sala*, ubicacion), donde *codigo_edificio* es clave foránea referenciando a EDIFICIO.

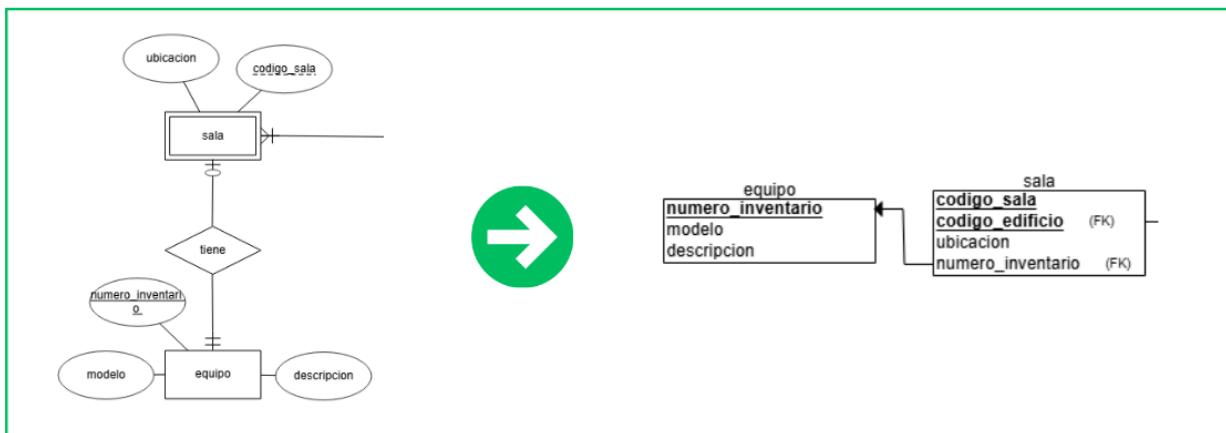


Reglas de Transformación para Relaciones (1:1, 1:N, M:N)

Las relaciones entre entidades también se transforman en estructuras relacionales siguiendo reglas específicas:

- **Relaciones 1:1:** Existen varias estrategias. La más común es añadir la clave primaria de una de las entidades participantes como una clave foránea en la tabla de la otra entidad. La elección de qué clave foránea añadir depende de la participación (total o parcial) de las entidades. Alternativamente, se puede crear una tabla separada para la relación, conteniendo las claves primarias de ambas entidades como clave primaria combinada y

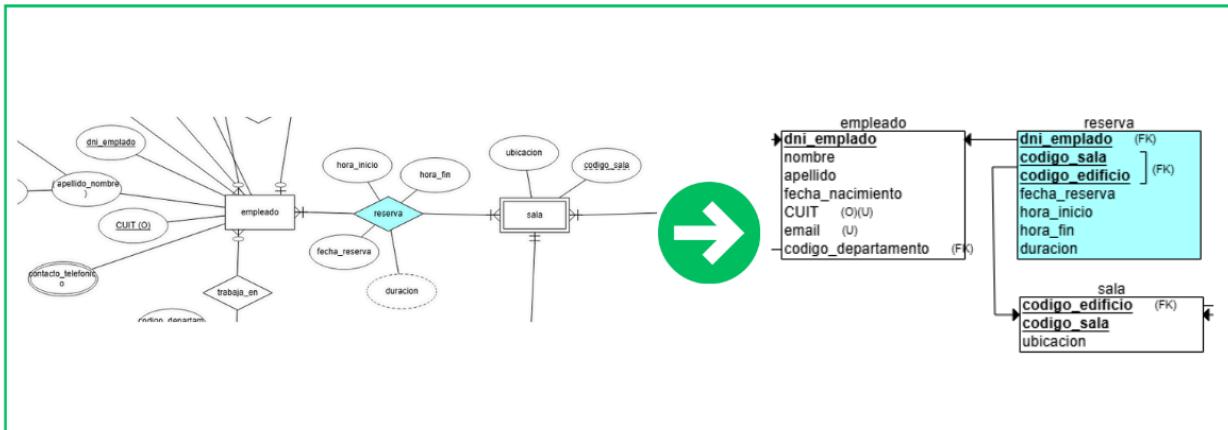
claves foráneas.¹



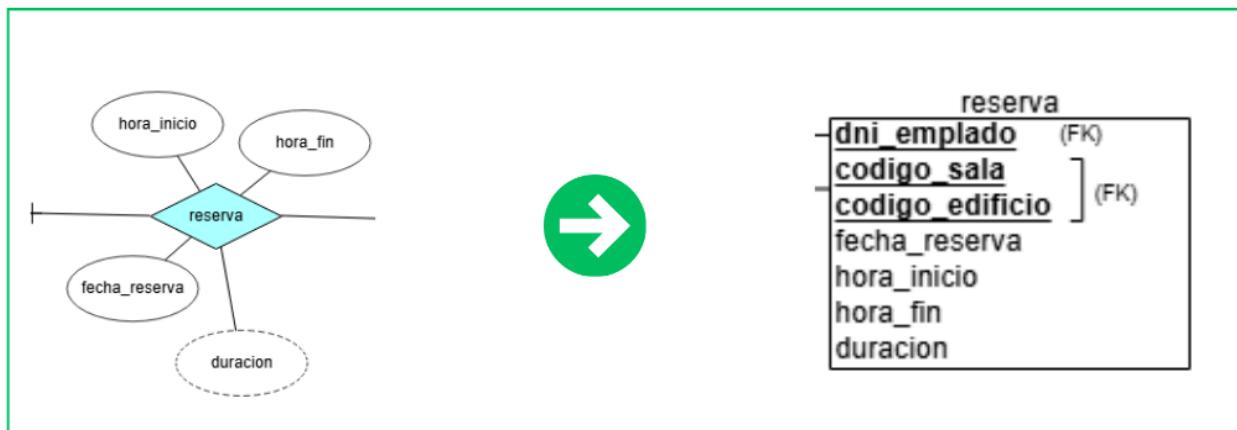
- **Relaciones 1:N:** La clave primaria de la entidad que está en el lado "1" de la relación se añade como una clave foránea en la tabla de la entidad que está en el lado "N".¹ Por ejemplo, en una relación TrabajaEn entre EMPLEADO (N) y DEPARTAMENTO (1), el idDept (clave primaria de DEPARTAMENTO) se añade como clave foránea en la tabla EMPLEADO.



- Relaciones M:N:** Este tipo de relaciones siempre se mapea a una nueva tabla relacional. Esta nueva tabla tendrá como clave primaria la combinación de las claves primarias de las entidades participantes, las cuales también actuarán como claves foráneas referenciando sus respectivas tablas.¹ Por ejemplo, una relación InscritoEn entre ESTUDIANTE y CURSO (M:N) se convierte en una tabla INSCRIPCION (*idEstudiante, idCurso*).



- Atributos de Relación:** Si una relación en el modelo ER/EER tiene atributos propios (es decir, atributos que describen la relación en sí y no a las entidades participantes), estos se convierten en columnas de la tabla que representa la relación. Esto es particularmente común en relaciones M:N.¹



Mapeo de Jerarquías de Generalización/Especialización y Otros Constructos EER

El mapeo de jerarquías de generalización/especialización es más complejo y ofrece varias estrategias, cada una con sus propias ventajas y desventajas en términos de redundancia, rendimiento y complejidad de consultas:

- Mapear cada clase a una tabla:** Se crea una tabla para la superclase y una tabla para cada subclase. Las tablas de las subclases incluyen la clave primaria de la superclase como clave

primaria y clave foránea.

- **Mapear solo las superclases:** (Menos común para jerarquías complejas, más para disjuntas y totales).
- **Mapear solo las subclases:** Se crean tablas solo para las subclases, y cada tabla de subclase incluye todos los atributos heredados de la superclase. Esto puede llevar a redundancia de atributos si una subclase tiene múltiples superclases.
- **Una tabla por jerarquía:** Se crea una única tabla para toda la jerarquía, que incluye columnas para todos los atributos de la superclase y todas las subclases, junto con un atributo discriminador para indicar a qué subclase pertenece cada fila. Esta estrategia puede introducir muchos valores nulos si los atributos de las subclases son exclusivos.

Elmasri (2007) en su Capítulo 7 es la fuente principal para el mapeo de constructos EER a relaciones.¹ Las diferentes estrategias para mapear jerarquías de generalización/especialización no son simplemente opciones arbitrarias; cada una presenta compensaciones significativas entre redundancia de datos, complejidad de consultas y rendimiento. Por ejemplo, mapear cada subclase a una tabla separada puede reducir la redundancia pero aumentar el número de operaciones JOIN necesarias para recuperar información completa, lo que podría afectar el rendimiento de las consultas. Mapear todo a una sola tabla (si es posible) puede simplificar las consultas pero introducir muchos valores nulos y redundancia. Esto significa que la elección de la estrategia de mapeo no es trivial y debe basarse en un análisis de los patrones de acceso a los datos y los requisitos de rendimiento específicos de la aplicación.

No existe una estrategia "mejor" universal, sino una "más adecuada" para cada contexto particular.

El mapeo de **Agregación** implica tratar la relación agregada como una entidad y mapearla como una tabla relacional, con claves foráneas a las entidades participantes. Las **Categorías** a menudo se mapean creando una tabla para la categoría, cuya clave primaria es una clave foránea a cada una de sus superclases, y puede tener una clave primaria propia.

Consideraciones para la Integridad Referencial y Restricciones

La transformación al modelo relacional debe preservar las restricciones de integridad definidas en el modelo conceptual.

- La **integridad de entidad** se asegura al definir claves primarias en cada tabla, garantizando que cada fila sea única y no tenga valores nulos en su clave primaria.
- La **integridad referencial** se implementa mediante claves foráneas, que aseguran que los valores en una columna (o conjunto de columnas) de una tabla (la tabla de referencia) coincidan con los valores de la clave primaria de otra tabla (la tabla referenciada), o sean

nulos. Esto refleja las restricciones de participación y cardinalidad del modelo ER/EER.¹

Otras restricciones, como las de atributos multivalorados (que requieren tablas adicionales), atributos derivados (que pueden manejarse con vistas o cálculos en la aplicación) y restricciones CHECK (para validar valores de columnas), también deben ser consideradas en la fase lógica y física.

A continuación, se presenta una tabla que resume las reglas de transformación para los principales constructos del Modelo ER/EER a sus equivalentes en el Modelo Relacional. Esta tabla es fundamental para los estudiantes porque sistematiza el conocimiento, facilita la aplicación práctica de las reglas de transformación y refuerza la conexión entre los modelos conceptual y lógico.

Tabla 2: Mapeo de Constructos ER/EER a Esquemas Relacionales

Constructo ER/EER	Transformación al Modelo Relacional	Ejemplo Breve
Entidad Fuerte	Se convierte en una tabla. Sus atributos simples son columnas. La clave primaria de la entidad es la clave primaria de la tabla.	EMPLEADO (<i>idEmpleado</i> , nombre, fechaNac)
Atributo Compuesto	Cada componente simple del atributo compuesto se convierte en una columna separada en la tabla de la entidad.	DIRECCION (calle, numero, ciudad, pais) en EMPLEADO (<i>idEmpleado</i> , nombre, calle, numero, ciudad, pais)
Atributo Multivalorado	Se crea una nueva tabla separada para el atributo. La clave primaria de esta nueva tabla es la combinación de la clave primaria de la entidad original y el atributo multivalorado.	TELEFONO de EMPLEADO: EMPLEADO_TELEFONO (<i>idEmpleado</i> , <i>telefono</i>)
Entidad Débil	Se convierte en una tabla. Su clave primaria es la combinación de su clave parcial y la clave primaria de la entidad fuerte identificadora (que se convierte en una clave foránea).	SALA (<i>codigo_edificio</i> , <i>codigo_sala</i> , ubicacion) donde <i>codigo_edificio</i> es FK a EDIFICIO.
Relación 1:1	Opción 1 (más común): La clave	EMPLEADO (<i>idEmpleado</i> ,...,

	primaria de una de las entidades se añade como clave foránea en la tabla de la otra entidad.	<i>idJefe</i>) donde idJefe es FK a EMPLEADO.
	Opción 2: Se crea una nueva tabla para la relación, con las claves primarias de ambas entidades como claves foráneas y combinadas como clave primaria.	CASAMIENTO (<i>idHombre</i> , <i>idMujer</i> , fechaMatrimonio)
Relación 1:N	La clave primaria de la entidad del lado "1" se añade como clave foránea en la tabla de la entidad del lado "N".	DEPARTAMENTO (<i>idDept</i> , nombre) y EMPLEADO (<i>idEmpleado</i> ,..., <i>idDept</i>) donde <i>idDept</i> es FK a DEPARTAMENTO.
Relación M:N	Se crea una nueva tabla para la relación. Su clave primaria es la combinación de las claves primarias de las entidades participantes (que también son claves foráneas).	ESTUDIANTE (<i>idEstudiante</i> ,...) y CURSO (<i>idCurso</i> ,...) -> INSCRIPCION (<i>idEstudiante</i> , <i>idCurso</i> , fechalInscripcion)
Atributo de Relación	Se convierte en una columna de la tabla que representa la relación (para M:N, o para 1:1 si se crea tabla separada).	En INSCRIPCION (<i>idEstudiante</i> , <i>idCurso</i> , fechalInscripcion), fechalInscripcion es un atributo de la relación.
Generalización/Especialización (EER)	Estrategia 1 (Una tabla por superclase y subclase): Cada superclase y subclase se mapea a una tabla separada. Las subclases heredan la clave primaria de la superclase como FK.	PERSONA (<i>idPersona</i> , nombre) EMPLEADO (<i>idPersona</i> , salario) FK a PERSONA CLIENTE (<i>idPersona</i> , limiteCredito) FK a PERSONA
	Estrategia 2 (Una tabla por jerarquía): Una sola tabla para toda la jerarquía, con columnas para todos los atributos y un atributo discriminador.	PERSONA (<i>idPersona</i> , nombre, salario, limiteCredito, tipoPersona)
	Estrategia 3 (Una tabla por subclase): Solo se crean tablas para las subclases, incluyendo atributos heredados de la	EMPLEADO (<i>idEmpleado</i> , nombre, salario) CLIENTE (<i>idCliente</i> , nombre, limiteCredito)

	superclase.	
Categoría (Tipo Unión)	Se crea una tabla para la categoría. Su clave primaria es una clave foránea a cada una de sus superclases, y puede tener una clave primaria propia.	PROPIETARIO (<i>idPropietario</i> , tipoPropietario) donde idPropietario es FK a PERSONA o COMPAÑIA.
Agregación	La relación agregada se trata como una entidad y se mapea como una tabla relacional, con claves foráneas a las entidades participantes.	Si (EMPLEADO, PROYECTO) es una relación agregada PARTICIPA_EN_PROYECTO, se mapea a una tabla PARTICIPA_EN_PROYECTO (<i>idEmpleado</i> , <i>idProyecto</i> , rol)

Conclusión sobre la Reducción a Esquemas Relacionales

Es importante destacar que la "Reducción a esquemas relacionales" es el paso que precede directamente a la normalización de la base de datos (tema de la Unidad 3). Los esquemas relacionales resultantes del mapeo ER/EER a menudo ya cumplen con ciertas formas normales (como 1FN, 2FN o incluso 3FN).¹ Esto significa que un buen diseño conceptual ER/EER es la base para un esquema relacional bien estructurado y menos propenso a anomalías de actualización, inserción y eliminación. La normalización no es un paso aislado, sino una refinación del esquema lógico que se inicia con el mapeo del modelo conceptual. La eficiencia y precisión del mapeo ER/EER influyen directamente en la facilidad y el grado de normalización posterior.

V. Conclusiones y Puntos Clave para el Estudio de la Unidad 2

La Unidad 2 del programa de la asignatura Bases de Datos I, aborda conceptos centrados en el diseño de bases de datos y el Modelo Entidad-Relación, cruciales para construir sistemas de gestión de información robustos y eficientes.

Conceptos Fundamentales

El diseño de bases de datos es un proceso estructurado que se inicia con una comprensión profunda de los requisitos del negocio. El Modelo Entidad-Relación (ER) y sus extensiones (EER) son herramientas poderosas para el modelado conceptual, permitiendo una representación fiel de la realidad del negocio y sus reglas. La transformación sistemática del modelo conceptual al relacional es un paso crítico que sienta las bases para la implementación de la base de datos. Finalmente, la correcta identificación y aplicación de restricciones, como la cardinalidad, la participación y las dependencias, es esencial para garantizar la integridad y consistencia de los datos almacenados.

Importancia del Enfoque Iterativo y las Compensaciones

Es fundamental reconocer que el diseño de bases de datos no es un proceso lineal y que las decisiones de diseño a menudo implican compensaciones entre diferentes factores. Por ejemplo, la elección de una estrategia de mapeo para jerarquías EER puede implicar equilibrar la redundancia de datos con la complejidad de las consultas y el rendimiento del sistema. Un diseñador experto comprende que no siempre existe una solución "perfecta", sino una "más adecuada" para cada contexto específico, basada en un análisis cuidadoso de los requisitos y las prioridades del sistema.

Preparación para Próximas Unidades

Los esquemas relacionales obtenidos a partir de la reducción del modelo ER/EER son el punto de partida para la siguiente fase del diseño: la normalización. La comprensión sólida de cómo un modelo conceptual se traduce en un esquema relacional es indispensable para aplicar correctamente las formas normales y refinar el diseño lógico, eliminando redundancias y anomalías.

Reflexiones Finales

Al revisar los contenidos de esta unidad, se hace evidente que el modelo ER/EER actúa como la verdadera "arquitectura" de la información en una organización.¹ Al igual que un arquitecto de edificios diseña los planos antes de la construcción, el diseño conceptual ER/EER define la estructura fundamental de los datos, sus relaciones y las reglas que los rigen, mucho antes de pensar en la tecnología específica (SQL, Oracle, etc.). Esto significa que la calidad de esta "arquitectura" conceptual determinará la robustez, flexibilidad y mantenibilidad de la base de datos a largo plazo, trascendiendo la tecnología subyacente. Un diseño conceptual deficiente, por muy bien implementado que esté, resultará en un sistema frágil y costoso de adaptar.

Por otra parte, la profunda atención a las restricciones (cardinalidad, participación, dependencias, etc.) en el modelo ER/EER revela que la base de datos no es solo un repositorio de datos, sino una implementación formal de las reglas de negocio de una organización.¹ Cada restricción en el diagrama ER/EER corresponde a una regla sobre cómo se relacionan los datos en el mundo real (por ejemplo, "un empleado trabaja en un solo departamento", "un departamento tiene muchos empleados"). Esto significa que el proceso de diseño de bases de datos es, en esencia, un ejercicio de ingeniería de requisitos y modelado de procesos de negocio, donde el modelo ER/EER se convierte en una representación formal y verificable de esas reglas. Los errores en el modelado de estas restricciones pueden llevar a que la base de datos permita estados inconsistentes que no son válidos en el negocio, afectando la toma de decisiones y la operación.

Referencias Bibliográficas por Tema

Diseño de Bases de Datos y Visión General del Proceso:

- **Elmasri, R. (2007). *Fundamentos de bases de datos*, 5a ed. Pearson.**
 - Capítulo 1: Bases de datos y usuarios de bases de datos (Introducción y características de la metodología).¹
 - Capítulo 2: Conceptos y arquitectura de los sistemas de bases de datos (Entorno del sistema).¹
 - Capítulo 12: Metodología práctica de diseño de bases de datos y uso de los diagramas UML (Proceso de diseño e implementación).¹
- **Begg, C. E., Connolly, T. M. (2005). *Sistemas de bases de datos: un enfoque práctico para diseño, implementación y gestión*. Pearson Educación.**
 - Capítulo 9: El Modelo Relacional y Los Lenguajes Relacionales (Proceso de desarrollo de sistemas de bases de datos, administración de datos/bases de datos).²
 - Capítulo 10: Técnicas de determinación de hechos (Identificación de requisitos).²
- **Ramakrishnan, R. (2007). *Sistemas de gestión de bases de datos*, Tercera Edición. McGraw-Hill/Interamericana de España, S.A.U.** Capítulo 2: Introducción al diseño de bases de datos.³
- **Nieto Bernal, W. y Capacho Portilla, J. R. (2017). *Diseño de base de datos*. Universidad del Norte.** (Disponible en abierto: <https://elibro.net/es/lc/unne/titulos/70030>) Este libro es una referencia general para el modelado, diseño e implementación, útil para la visión general del proceso.⁶

El Modelo Entidad-Relación (ER): Conceptos, Restricciones y Diagramas:

- **Elmasri, R. (2007). *Fundamentos de bases de datos*, 5a ed. Pearson.** Capítulo 3: Modelado de datos con el modelo Entidad-Relación (ER) (Entidades, atributos, relaciones, restricciones, diagramas ER, notaciones).¹
- **Silberschatz, A., Korth, H. F., Sudarshan, S. (2014). *Fundamentos de bases de datos*, 6th Edición. McGraw-Hill Interamericana de España S.L.** Capítulo 2: Modelo entidad/relación (Conceptos ER, notación, restricciones).¹¹
- **Begg, C. E., Connolly, T. M. (2005). *Sistemas de bases de datos: un enfoque práctico para diseño, implementación y gestión*. Pearson Educación.** Capítulo 11: Modelo Entidad - Relación (Conceptos ER, grado de relación, cardinalidad, participación, trampas de conexión).²
- **Date, C. J. (2001). *Introducción a los sistemas de bases de datos*. Pearson Educación.** Capítulo 13: Modelado semántico (Introducción al Modelo E/R).¹²

Elementos Avanzados del Modelo E-R Extendido (EER):

- **Elmasri, R. (2007). *Fundamentos de bases de datos*, 5a ed. Pearson.** Capítulo 4: El modelo Entidad-Relación mejorado (EER) (Subclases, superclases, herencia, especialización, generalización, categorías, agregación).¹
- **Begg, C. E., Connolly, T. M. (2005). *Sistemas de bases de datos: un enfoque práctico para diseño, implementación y gestión*.** Pearson Educación. Capítulo 12: Modelo Entidad - Relación Avanzada (Limitaciones del ER básico, especialización/generalización, agregación, composición).²
- **Date, C. J. (2001). *Introducción a los sistemas de bases de datos*.** Pearson Educación. Capítulo 19: Herencia de tipo (Jerarquías de tipos, polimorfismo, especialización).¹²

Reducción a Esquemas Relacionales:

- **Elmasri, R. (2007). *Fundamentos de bases de datos*, 5a ed. Pearson.** Capítulo 7: Diseño de bases de datos relacionales por mapeado ER- y EER-a-relacional (Reglas de transformación detalladas).¹
- **Silberschatz, A., Korth, H. F., Sudarshan, S. (2014). *Fundamentos de bases de datos*, 6th Edición. McGraw-Hill Interamericana de España S.L.** Capítulo 7: Diseño de bases de datos relacionales (Mapeo de ER a relacional).¹¹
- **de Miguel Castaño, A. (2000). *Diseño de bases de datos relacionales*.** RA-MA Editorial. Contiene secciones sobre "Diseño conceptual: modelo E/R" y "Diseño lógico: modelo relacional", que implican el proceso de mapeo.⁹
- **Piattini Velthuis, M. G. et al. (2007). *Tecnología y diseño de bases de datos*.** RA-MA EDITORIAL. Menciona "Diseño de bases de datos relacionales" en su contenido parcial, lo que sugiere cobertura de este proceso.⁵

Consideraciones Adicionales (Integridad y Restricciones en el Modelo Relacional):

- **Elmasri, R. (2007). *Fundamentos de bases de datos*, 5a ed. Pearson.**
 - Capítulo 5: El modelo de datos relacional y las restricciones de una base de datos relacional.¹
 - Capítulo 8: SQL-99: definición del esquema, restricciones, consultas y vistas (Especificación de restricciones en SQL).¹
- **Silberschatz, A., Korth, H. F., Sudarshan, S. (2014). *Fundamentos de bases de datos*, 6th Edición. McGraw-Hill Interamericana de España S.L.** Capítulo 6: Ligaduras de integridad.¹¹
- **Begg, C. E., Connolly, T. M. (2005). *Sistemas de bases de datos: un enfoque práctico para diseño, implementación y gestión*.** Pearson Educación. Capítulo 6: Definición de Datos

(Integrity enhancements in SQL).²

Obras citadas

1. Fundamentos de Sistemas de Bases de Datos [5 ed ... - dokumen.pub, fecha de acceso: julio 28, 2025, <https://dokumen.pub/fundamentos-de-sistemas-de-bases-de-datos-5nbsped-9788478290857.html>
2. Base de Datos Connolly | PDF | Base de datos relacional | Bases de ..., fecha de acceso: julio 28, 2025, <https://es.scribd.com/document/437679847/base-de-datos-Connolly-doc>
3. Sistemas De Gestión De Bases De Datos 3.º Ed. Ramakrishnan ..., fecha de acceso: julio 28, 2025, https://www.mercadolibre.com.pe/sistemas-de-gestion-de-bases-de-datos-3-ed-ramakris_hnan/up/MPEU2462248494
4. Book Sistemas de Gestión de Base de Datos Raghu Ramakrishnan, fecha de acceso: julio 28, 2025, <https://www.bookdelivery.com/il-en/book-sistemas-de-gestion-de-base-de-datos/9788448156381/p/1016889>
5. Tecnología y diseño de bases de datos / > Catálogo en línea Koha - UISEK, fecha de acceso: julio 28, 2025, <https://koha.uisek.edu.ec/cgi-bin/koha/opac-detail.pl?biblionumber=3587>
6. Diseño de base de datos - Uninorte, fecha de acceso: julio 28, 2025, <https://editorial.uninorte.edu.co/gpd-diseno-de-base-de-datos.html>
7. Descripción: Tecnología y diseño de bases de datos / - Catálogo SIIDCA - CSUCA, fecha de acceso: julio 28, 2025, <https://catalogosiidca.csuca.org/Record/UDELAS.089166/Description>
8. Diseño de bases de datos - Capacho, José Rafael, Nieto Bernal, Wilson - Google Libros, fecha de acceso: julio 28, 2025, <https://books.google.co.cr/books?id=TLBJDwAAQBAJ&printsec=frontcover>
9. Diseño de bases de datos relacionales - ResearchGate, fecha de acceso: julio 28, 2025, https://www.researchgate.net/publication/50839137_Diseno_de_bases_de_datos_relacionales
10. Diseño de bases de datos relacionales - Bibliotecas UNLaR - Universidad Nacional de La Rioja, fecha de acceso: julio 28, 2025, <https://biblioteca.unlar.edu.ar/cgi-bin/koha/opac-detail.pl?biblionumber=12138>
11. Tabla de Contenidos: Fundamentos de bases de datos /, fecha de acceso: julio 28, 2025, <https://catalogosiidca.csuca.org/Record/UES.12636/TOC>
12. introducion-a-los-sistemas-de-bases-de-datos-cj-date.pdf, fecha de acceso: julio 28, 2025, <https://unefazuliasistemas.files.wordpress.com/2011/04/introducion-a-los-sistemas-de-bases-de-datos-cj-date.pdf>

Última revisión: 21 ago 2025 