

# Unidad 3. El modelo relacional

## Introducción

El contenido de este documento se basa y se limita a la bibliografía sugerida para la asignatura, que incluye las obras destacadas para la disciplina. Principalmente, se hace referencia a Fundamentos de bases de datos, 6ta Edición, de Silberschatz, Korth y Sudarshan, y a Fundamentos de bases de datos, 5ta Edición, de Elmasri. La comunidad académica reconoce estos textos como recursos indispensables para quienes buscan aprender los fundamentos de las bases de datos, destacando la claridad de sus explicaciones y la calidad de sus ejercicios.<sup>3</sup>

## Tema 1: El Modelo Relacional - Fundamentos y Conceptos

### 2.1. Definiciones Clave para el Entendimiento del Modelo Relacional

La comprensión del modelo relacional comienza con la asimilación de sus componentes elementales, los cuales se organizan en una jerarquía lógica. Una **relación**, comúnmente conocida como tabla, es la estructura bidimensional fundamental que organiza los datos en filas y columnas. Cada relación representa un conjunto de entidades del mundo real, como "Estudiantes" o "Cursos". Dentro de esta estructura, un **atributo** es una columna que describe una propiedad o característica de las entidades, por ejemplo, nombre o identificador\_curso. Por su parte, una **tupla**, o fila, representa una instancia única de la entidad descrita por la relación, es decir, un registro individual completo que contiene un valor para cada uno de los atributos.

A un nivel más abstracto, la estructura y el contenido de una relación se definen por dos conceptos clave. El **esquema de relación** (también llamado intención) es la estructura estática de la relación, definida por su nombre y el conjunto de atributos. La notación formal para un esquema de relación R con atributos A1,A2,...,An es R(A1,A2,...,An).<sup>5</sup> A diferencia del esquema, la

**instancia de relación** (o extensión) es el conjunto dinámico y actual de tuplas que se encuentra en la relación en un momento dado.<sup>5</sup> Un aspecto fundamental del modelo relacional es que, si bien el esquema es relativamente estático y rara vez cambia, la instancia es dinámica y se transforma constantemente con cada operación de inserción, actualización o eliminación de datos.<sup>5</sup>

### 2.2. Métricas de una Relación: Grado y Cardinalidad

Para caracterizar el tamaño y la estructura de una relación, se utilizan dos métricas principales: el grado y la cardinalidad. El **grado** de una relación se define como el número de atributos que componen su esquema.<sup>6</sup> Esta es una medida de la estructura de la relación y, al igual que el esquema, es estática. Por ejemplo, una relación con los atributos (nombre, apellido, fecha\_nacimiento) tiene un grado de 3. Por otro lado, la **cardinalidad** se refiere al número de tuplas en una instancia de relación.<sup>7</sup> Esta medida es dinámica y refleja el estado actual de los datos; por lo tanto, la cardinalidad de una relación puede

cambiar de manera constante a medida que se agregan o eliminan tuplas.

Un concepto adicional crucial es el **dominio** de un atributo. El dominio define el conjunto de valores permitidos para un atributo en particular, como el tipo de dato INTEGER para una edad o un conjunto de valores discretos como ('F', 'M') para un género.<sup>7</sup> Comprender este concepto es esencial para asegurar la consistencia y la integridad de los datos, ya que garantiza que los valores ingresados cumplan con las reglas predefinidas.

### 2.3. Tabla de Referencia: Conceptos Fundamentales

Para clarificar y diferenciar los conceptos, la siguiente tabla resume las definiciones clave del modelo relacional.

Concepto	Sinónimo	Descripción
<b>Relación</b>	Tabla	Estructura bidimensional con filas y columnas que almacena los datos.
<b>Atributo</b>	Columna	Una propiedad o característica de las entidades en una relación.
<b>Tupla</b>	Fila, Registro	Un registro individual que representa una instancia única de una entidad.
<b>Esquema de Relación</b>	Intensión	La estructura estática de la relación, definida por sus atributos.
<b>Instancia de Relación</b>	Extensión	El conjunto dinámico de tuplas en un momento dado.
<b>Grado</b>		Número de atributos en el esquema de la relación.
<b>Cardinalidad</b>		Número de tuplas en una instancia de la relación.
<b>Dominio</b>		Conjunto de valores permitidos para un atributo.

### 2.4. La Lógica Subyacente del Modelo

La distinción entre un esquema estático y una instancia dinámica es la piedra angular del modelo relacional y la base de la arquitectura de un SGBD. La descripción del esquema, con todos sus metadatos (nombres de tablas, atributos, tipos de datos, etc.), se almacena de manera permanente en una parte

especial del sistema llamada catálogo de datos.<sup>5</sup> En contraste, la instancia de la base de datos reside en los archivos de datos que se manipulan activamente. La mayoría de las operaciones de consulta y manipulación de datos, como SELECT, INSERT y UPDATE, actúan sobre la instancia sin alterar el esquema. Esta separación es fundamental y permite distinguir claramente entre el Lenguaje de Definición de Datos (DDL), que se utiliza para crear y modificar el esquema, y el Lenguaje de Manipulación de Datos (DML), que se usa para operar sobre la instancia. Esta comprensión de la dicotomía entre la estructura estática y el contenido dinámico es vital para cualquier desarrollador o diseñador de bases de datos.

## Tema 2: Restricciones de Integridad - Coherencia y Consistencia

### 3.1. Claves en el Modelo Relacional

Las restricciones de integridad son condiciones que se establecen para garantizar que las modificaciones en una base de datos no comprometan la coherencia y la consistencia de los datos, protegiéndolos contra daños accidentales.<sup>8</sup> Dentro de estas, las restricciones de clave son esenciales para la identificación única de las tuplas.

El proceso de identificación de una clave es un ejercicio de refinamiento. Se comienza con una **superclave**, que es cualquier conjunto de uno o más atributos que, tomados colectivamente, permiten identificar de forma única una tupla en una relación.<sup>8</sup> Es importante notar que una superclave puede contener atributos innecesarios o redundantes. A continuación, se refina este concepto para llegar a una **clave candidata**, que es una superclave *mínima*. Esto significa que no se puede eliminar ningún atributo de una clave candidata sin que se pierda la propiedad de unicidad.<sup>8</sup> Una tupla puede ser distinguida de las demás por un número determinado de atributos, que son estas claves candidatas.<sup>8</sup> Finalmente, la **clave primaria** es la clave candidata que el diseñador de la base de datos ha seleccionado para identificar de forma exclusiva las tuplas de una relación. La clave primaria debe cumplir con dos propiedades cruciales: **unicidad** (no puede haber valores duplicados) y **no nulidad** (ninguno de sus atributos puede ser nulo).<sup>7</sup>

### 3.2. Restricciones de Clave Externa (Foreign Key)

Las claves primarias, por sí solas, garantizan la unicidad dentro de una relación, pero las **claves externas** (o foráneas) son el mecanismo que establece las conexiones lógicas entre relaciones.<sup>7</sup> Una clave externa es un conjunto de atributos en una tabla "hija" que hace referencia a una clave única (usualmente la clave primaria) en una tabla "padre".<sup>10</sup> Su función principal es implementar la **integridad referencial**, una regla que asegura que cada valor de la clave externa debe coincidir con un valor de la clave primaria en la tabla referenciada, o bien, ser nulo.<sup>1</sup>

Cuando se modifica o elimina una tupla en la tabla padre, el SGBD debe actuar para mantener la integridad referencial. Esto se define a través de las acciones en cascada:

- ON DELETE CASCADE: Elimina automáticamente las tuplas hijas cuando se elimina la tupla padre a la que hacen referencia. Es útil para dependencias fuertes.<sup>10</sup>
- ON DELETE SET NULL: Establece los valores de la clave externa en las tuplas hijas a NULL cuando se

elimina la tupla padre. Esto es posible solo si la clave externa no tiene la restricción NOT NULL.<sup>11</sup>

- ON DELETE RESTRICT y ON DELETE NO ACTION: Ambas opciones previenen la eliminación de una tupla padre si existen tuplas hijas que la referencian. La diferencia técnica radica en el momento de la comprobación: RESTRICT verifica la restricción antes de la operación, mientras que NO ACTION la evalúa al final.<sup>11</sup> Un análisis similar aplica para las acciones ON UPDATE.

### 3.3. Restricciones Generales

Además de las claves, existen otras restricciones que contribuyen a la integridad de los datos. Las **restricciones de dominio** aseguran que los valores de un atributo pertenezcan a un dominio predefinido, como un tipo de dato o un rango específico.<sup>7</sup> Por ejemplo, un atributo edad podría tener un dominio de valores enteros entre 0 y 120. Por otro lado, las **restricciones de integridad general** (o Assertions) son reglas más complejas que deben ser satisfechas por la base de datos en su conjunto, como la condición de que el salario de un empleado no puede ser negativo.<sup>12</sup> A diferencia de las restricciones de dominio o de clave, que a menudo se aplican a una sola tabla, estas reglas pueden involucrar múltiples atributos en diferentes relaciones.

### 3.4. Tabla de Referencia: Tipos de Claves y Restricciones

Concepto	Descripción	Propiedades
<b>Superclave</b>	Conjunto de atributos que identifica una tupla de forma única.	Puede contener atributos redundantes.
<b>Clave Candidata</b>	Superclave mínima. No se puede eliminar ningún atributo sin perder la unicidad.	Unicidad. Puede haber varias en una tabla.
<b>Clave Primaria</b>	Clave candidata elegida por el diseñador para identificar tuplas.	Unicidad y No Nulidad. Solo puede haber una por tabla.
<b>Clave Externa</b>	Conjunto de atributos en una tabla que referencia a una clave primaria en otra.	Establece relaciones entre tablas. Implementa la integridad referencial.

### Acciones de Integridad Referencial

Acción	ON DELETE	ON UPDATE
<b>CASCADE</b>	Elimina tuplas hijas.	Actualiza tuplas hijas.
<b>SET NULL</b>	Establece la clave externa a NULL.	Establece la clave externa a NULL.
<b>RESTRICT</b>	Rechaza la operación.	Rechaza la operación.
<b>NO ACTION</b>	Rechaza la operación al final de la instrucción.	Rechaza la operación al final de la instrucción.

### 3.5. La Lógica Detrás de la Coherencia

La jerarquía de las claves (superclave, clave candidata, clave primaria) no es una simple lista de términos, sino un proceso de refinamiento conceptual que el diseñador de la base de datos debe seguir. Una superclave es el punto de partida; una clave candidata es el refinamiento de la unicidad, eliminando la redundancia; y la clave primaria es la elección final del diseñador, que define el identificador canónico de la tabla.

Además, las claves externas son la materialización de las relaciones entre entidades que se definen en la fase de diseño conceptual.<sup>1</sup> La correcta elección de las acciones en cascada para la integridad referencial es un aspecto crítico que distingue un diseño robusto de uno defectuoso. Un manejo inadecuado puede conducir a la aparición de datos "huérfanos"—registros en una tabla hija que referencian un registro padre que ya no existe—, lo cual compromete la consistencia de la base de datos. Por lo tanto, el entendimiento de las implicaciones de cada decisión de diseño es vital para construir un sistema confiable.

## Tema 3: Álgebra Relacional - El Corazón Lógico de las Consultas

### 4.1. Introducción al Álgebra Relacional

El álgebra relacional es un lenguaje de consulta formal y procedimental, y es la base teórica sobre la que se construyen la mayoría de los lenguajes de consulta comerciales, como SQL.<sup>1</sup> Aunque SQL es un lenguaje no procedimental (ya que el usuario simplemente especifica "qué" datos quiere sin detallar "cómo" obtenerlos), un SGBD lo traduce internamente a un plan de ejecución basado en operaciones del álgebra relacional. Una analogía útil es que el álgebra relacional es a SQL lo que la aritmética es a una calculadora: la notación matemática subyacente que describe las operaciones.<sup>13</sup> Tanto Silberschatz como

Elmasri dedican capítulos completos a este tema, subrayando su importancia como un marco formal para la manipulación de datos.<sup>14</sup>

## 4.2. Operadores Fundamentales y Ejemplos

El álgebra relacional se compone de un conjunto de operadores que toman una o dos relaciones como entrada y producen una nueva relación como resultado.

- **Selección ( $\sigma$ ):** Este operador filtra las tuplas horizontalmente de una relación basándose en un predicado o condición lógica. Por ejemplo,  $\sigma_{\text{salario} > 50000}(\text{Empleados})$  devuelve una nueva relación con todas las tuplas de los empleados que cumplen con esa condición.<sup>16</sup>
- **Proyección ( $\pi$ ):** Este operador selecciona atributos verticalmente. El resultado es una nueva relación que contiene solo las columnas especificadas, y que, por la naturaleza del modelo relacional como conjunto, elimina automáticamente las tuplas duplicadas si las hubiera.<sup>16</sup> Por ejemplo,  $\pi_{\text{Nombre}, \text{Direccion}}(\text{Proveedores})$  devolvería solo esas dos columnas de la relación.
- **Unión ( $\cup$ ), Intersección ( $\cap$ ), y Diferencia ( $-$ ):** Estos operadores se basan en la teoría de conjuntos. La unión combina todas las tuplas de dos relaciones compatibles<sup>16</sup>; la intersección encuentra las tuplas comunes a ambas; y la diferencia obtiene las tuplas de una relación que no están en la otra.<sup>16</sup>
- **Producto Cartesiano ( $\times$ ):** Este operador combina cada tupla de la primera relación con cada tupla de la segunda, resultando en una nueva relación que es la concatenación de ambas. Si la primera relación tiene N tuplas y la segunda tiene M, el resultado tendrá  $N \times M$  tuplas.<sup>16</sup> Es la base para la operación JOIN.
- **Renombrado ( $\rho$ ):** Este operador permite cambiar el nombre de una relación o de sus atributos. Es especialmente útil en operaciones complejas que requieren combinar una relación consigo misma, como el autounión.<sup>18</sup>
- **Join ( $\bowtie$ ):** Es un operador derivado que combina tuplas de dos relaciones basándose en una condición sobre sus atributos. El join natural es una forma común que combina tuplas con valores iguales en atributos de nombres comunes y luego elimina las columnas duplicadas.<sup>16</sup> Un join general, conocido como Theta-Join, puede ser expresado como una selección sobre un producto cartesiano:  $R \bowtie \theta S \equiv \sigma_{\theta}(R \times S)$ .<sup>16</sup>

## 4.3. Tabla de Referencia: Operadores del Álgebra Relacional

Símbolo	Nombre	Descripción	Analogía en SQL
$\sigma$	Selección	Filtrar tuplas horizontalmente según una condición.	WHERE
$\pi$	Proyección	Seleccionar atributos (columnas) verticalmente.	SELECT

U	Unión	Combina las tuplas de dos relaciones compatibles.	UNION
∩	Intersección	Obtiene las tuplas comunes a dos relaciones compatibles.	INTERSECT
-	Diferencia	Obtiene las tuplas en la primera relación que no están en la segunda.	EXCEPT
×	Producto Cartesiano	Combina todas las tuplas de dos relaciones.	CROSS JOIN
ρ	Renombrado	Cambia el nombre de una relación o atributo.	AS
⋈	Join	Combina tuplas basándose en atributos o una condición.	JOIN

#### 4.4. La Ecuación Detrás de la Práctica

La naturaleza derivada de un operador como JOIN, que se puede expresar como una selección sobre un producto cartesiano, demuestra que incluso las operaciones más complejas en las bases de datos tienen sus raíces en un conjunto de operadores más primitivos.<sup>16</sup> Esta comprensión es fundamental para apreciar cómo funcionan los optimizadores de consultas de un SGBD, ya que su objetivo principal es encontrar la forma más eficiente de ejecutar una consulta compleja, descomponiéndola en una secuencia óptima de operaciones más simples.

Además, existe una distinción crucial entre la teoría y la práctica. Si bien el álgebra relacional se basa en la teoría de conjuntos (donde no se permiten duplicados), la mayoría de los motores de bases de datos modernos trabajan con un concepto llamado "álgebra de bolsas" (bags), que sí permite tuplas repetidas.<sup>19</sup> Esto explica por qué en SQL, una proyección de un subconjunto de columnas (SELECT columna1, columna2) a menudo conserva las filas duplicadas, y el usuario debe especificar explícitamente la palabra clave DISTINCT para eliminarlas, a diferencia de la proyección teórica del álgebra relacional.

## Tema 4: Valores Nulos - La Ambivalencia de los Datos

### 5.1. Significado y Consecuencias de un Valor Nulo

Un valor **nulo** (NULL) no es un valor en sí mismo, sino un marcador que indica la ausencia de un valor. Su significado puede variar: puede representar un valor que es desconocido, que no es aplicable a un registro en particular, o que simplemente no existe.<sup>7</sup> Es fundamental no confundir NULL con un valor numérico cero o con una cadena de texto vacía, ya que su semántica es completamente diferente.<sup>20</sup>

Los valores nulos tienen un impacto directo en las restricciones de integridad. Por definición, la restricción de **clave primaria** prohíbe los valores nulos, ya que cada tupla debe ser identificable de forma única y completa. Sin embargo, en el caso de las **claves externas**, los valores nulos están permitidos a menos que se especifique una restricción NOT NULL.<sup>7</sup> La decisión de permitir nulos en una clave externa es una consideración de diseño crítica, ya que puede relajar la integridad referencial y permitir que una tupla hija no tenga una referencia a una tupla padre.

### 5.2. Efecto en Comparaciones y Funciones de Agregación

La presencia de valores nulos introduce una complicación en la lógica de las consultas. En el modelo relacional, las comparaciones que involucran un valor nulo no resultan en TRUE o FALSE, sino en un tercer estado de verdad: **UNKNOWN**.<sup>20</sup> Por ejemplo, la expresión  $10 = \text{NULL}$  no es ni verdadera ni falsa, sino desconocida. Este comportamiento afecta directamente a las cláusulas WHERE, ya que solo se devuelven las tuplas para las que el predicado de la condición se evalúa como TRUE, excluyendo aquellas que resultan en FALSE o UNKNOWN.<sup>21</sup>

En cuanto a las funciones de agregación (SUM, AVG, MIN, MAX), la mayoría de ellas ignoran los valores nulos en el cálculo del resultado.<sup>22</sup> Por ejemplo, la función AVG para un conjunto de valores que incluye NULL solo promedia los valores que no son nulos. La excepción más notable es la función COUNT. COUNT(\*) cuenta el número total de filas en un conjunto de resultados, incluyendo aquellas con valores nulos.<sup>23</sup> En contraste, COUNT(columna) solo cuenta las tuplas en las que el valor de la columna especificada no es nulo.<sup>23</sup>

### 5.3. Manejo de Valores Nulos en SQL

Debido a la lógica de tres valores, no se puede utilizar el operador de igualdad (=) para buscar valores nulos en SQL. En su lugar, se utilizan los operadores especiales IS NULL e IS NOT NULL.<sup>21</sup> Por ejemplo, para encontrar a todos los empleados con un correo electrónico no registrado, se usaría una cláusula WHERE email IS NULL.<sup>24</sup> De manera similar, para encontrar a los empleados con un correo electrónico registrado, se utilizaría WHERE email IS NOT NULL.<sup>24</sup> El manejo correcto de los valores nulos es crucial para garantizar que las consultas devuelvan los resultados esperados y para evitar errores lógicos al actualizar o eliminar datos.

## Tema 5: Del Modelo E-R al Relacional - El Mapeo Lógico

### 6.1. La Fase de Diseño Lógico

El diseño de bases de datos es un proceso en el que se traduce un modelo conceptual del mundo real a una estructura formal que puede ser implementada. La traducción del **Modelo Entidad-Relación (E-R)** al **modelo relacional** es la fase de diseño lógico que marca la transición del nivel conceptual (independiente del SGBD) al nivel lógico (específico del modelo relacional).<sup>7</sup> Este proceso es algorítmico y se rige por un conjunto de reglas de mapeo.

### 6.2. Reglas de Mapeo: Paso a Paso

- **Entidades Fuertes:** Cada entidad fuerte se transforma en una tabla en el modelo relacional. Sus atributos se convierten en columnas, y la clave de la entidad se convierte en la clave primaria de la tabla.<sup>26</sup>
- **Atributos:**
  - Los atributos simples se mapean directamente a una columna.
  - Los atributos compuestos se "aplanan" al crear una columna para cada uno de sus componentes.<sup>27</sup>
  - Los atributos **multivaluados**, que pueden tener múltiples valores para una misma entidad, requieren la creación de una **nueva tabla** para almacenar sus valores. La clave primaria de esta nueva tabla es una clave compuesta por la clave primaria de la entidad original y el atributo multivaluado. Esto se debe a que el modelo relacional no permite celdas que contengan múltiples valores.<sup>27</sup>

### 6.3. Mapeo de Relaciones

- **Relaciones 1:1:** Se pueden manejar de dos maneras: la primera opción es propagar la clave primaria de una de las entidades a la otra tabla como clave externa. Para evitar valores nulos, es preferible propagar la clave de la entidad con participación parcial a la entidad con participación total. La segunda opción es fusionar las dos tablas en una sola, manteniendo la clave primaria de una de ellas.<sup>26</sup>
- **Relaciones 1:N:** La clave primaria de la entidad del lado "uno" se propaga como una clave externa a la tabla de la entidad del lado "muchos".<sup>26</sup>
- **Relaciones N:M:** Se crea una **nueva tabla** para la relación. La clave primaria de esta nueva tabla será una clave compuesta por las claves primarias de las tablas de las entidades participantes, las cuales también actúan como claves externas que refieren a las tablas originales.<sup>10</sup> Esta es la única forma de representar la cardinalidad múltiple en el modelo relacional.

### 6.4. Mapeo de Entidades Débiles

Una entidad débil se transforma en una nueva tabla que incluye sus propios atributos, además de la clave primaria de la entidad fuerte de la cual depende. La clave primaria de esta nueva tabla será una **clave compuesta** formada por la clave primaria de la entidad fuerte y el identificador parcial de la

entidad débil.<sup>27</sup>

## 6.5. Tabla de Referencia: Reglas de Mapeo E-R a Relacional

Concepto E-R	Regla de Mapeo al Modelo Relacional
<b>Entidad Fuerte</b>	Crear una tabla con los atributos de la entidad. Su clave de entidad será la clave primaria de la tabla.
<b>Atributo Multivaluado</b>	Crear una nueva tabla que contenga el atributo multivaluado y la clave primaria de la entidad a la que pertenece. La clave de esta nueva tabla será compuesta.
<b>Entidad Débil</b>	Crear una tabla que incluya la clave primaria de la entidad fuerte de la que depende y su propio identificador parcial. La clave de la tabla será la composición de ambos.
<b>Relación 1:1</b>	Propagar la clave de la entidad con participación total como clave externa en la tabla de la entidad con participación parcial. O bien, fusionar ambas tablas.
<b>Relación 1:N</b>	Propagar la clave primaria de la entidad del lado "1" como clave externa en la tabla de la entidad del lado "N".
<b>Relación N:M</b>	Crear una nueva tabla para la relación. La clave primaria de esta tabla será una clave compuesta por las claves primarias de las entidades que participan en ella.

## 6.6. La Razón de Ser de las Reglas

Las reglas de mapeo no son arbitrarias; son una consecuencia directa de las limitaciones y fortalezas del modelo relacional. La necesidad de crear una nueva tabla para las relaciones N:M, por ejemplo, surge de la restricción fundamental del modelo relacional de no permitir atributos multivaluados en una única celda. Si se intentara propagar la clave primaria de una entidad a la otra, no se podría representar la multiplicidad de la relación. De esta manera, el algoritmo de mapeo proporciona una solución estructurada que preserva la lógica del modelo conceptual en la estructura de las tablas relacionales.

## Conclusiones y Referencias

La Unidad 3 ha proporcionado una inmersión completa en el modelo relacional, desde sus conceptos más abstractos hasta su aplicación práctica en el diseño. La comprensión de los fundamentos—relación, atributo, tupla—y la distinción entre un esquema estático y una instancia dinámica, es la base para entender cómo los datos se organizan de manera lógica. Las restricciones de integridad, en particular la jerarquía de las claves y el manejo de las claves externas, son esenciales para garantizar la coherencia y consistencia de los datos.

El álgebra relacional es el lenguaje matemático subyacente que valida la capacidad de consulta del modelo, proporcionando un marco procedimental que, si bien es teórico, ilumina cómo los SGBD optimizan las consultas. Finalmente, la fase de diseño lógico, que traduce el modelo E-R al relacional, demuestra cómo se aplican todas estas teorías para crear un esquema de base de datos robusto y funcional.

La integración de estos conceptos—desde la teoría de conjuntos hasta la pragmática de las reglas de mapeo—es lo que convierte el modelo relacional en un pilar fundamental de la informática moderna.

### Bibliografía Sugerida

- Silberschatz, A., Korth, H. F., Sudarshan, S. (2014). Fundamentos de bases de datos , 6ta Edición.
- Elmasri, R. (2007). Fundamentos de bases de datos , 5ta Edición.
- Begg, C. E., Connolly, T. M. (2005). Sistemas de bases de datos: un enfoque práctico para diseño, implementación y gestión.

### Obras citadas

1. Fundamentos de Bases de Datos Silberschatz Korth Sudarshan | PDF - Scribd, fecha de acceso: agosto 26, 2025,  
<https://es.scribd.com/document/393877858/Fundamentos-de-Bases-de-Datos-Silberschatz-Korth-Sudarshan.pdf>
2. Silberschatz - Cliqueapostilas.com, fecha de acceso: agosto 26, 2025,  
<https://www.cliqueapostilas.com/Content/apostilas/20c688d1c1828d8462417d8044728ee9.pdf>
3. Libro Fundamentos de Bases de Datos De Abraham Silberschatz; Henry Korth; S. Sudarshan - Buscalibre, fecha de acceso: agosto 26, 2025,  
<https://www.buscalibre.us/libro-fundamentos-de-bases-de-datos/9788448190330/p/42112475>
4. BASES DE DATOS - UNED, fecha de acceso: agosto 26, 2025,  
[http://portal.uned.es/FadmonGuiasWeb/htdocs/abrir\\_fichero/abrir\\_fichero.jsp?idGuia=74252](http://portal.uned.es/FadmonGuiasWeb/htdocs/abrir_fichero/abrir_fichero.jsp?idGuia=74252)
5. 4. DISEÑO DE BASES DE DATOS RELACIONALES - cursos o no. AIU, fecha de acceso: agosto 26, 2025,  
<https://cursos.aiu.edu/base%20de%20datos/pdf/tema%204.pdf>
6. 2.1 Definición de modelo de datos - LIC Alejandra López - ITSLR, fecha de acceso: agosto 26, 2025,  
[http://itslr-alelopj.weebly.com/uploads/9/3/6/4/936494/fbd-tutorial\\_u2.pdf](http://itslr-alelopj.weebly.com/uploads/9/3/6/4/936494/fbd-tutorial_u2.pdf)
7. u03\_el\_modelo\_relacional\_guia\_bibliográfica
8. Manejadores de Bases de Datos, fecha de acceso: agosto 26, 2025,  
[https://www.uv.mx/personal/ermeneses/files/2020/09/Clase6-ModeloDeDatos\\_parteIII\\_ModeloRelacional.pdf](https://www.uv.mx/personal/ermeneses/files/2020/09/Clase6-ModeloDeDatos_parteIII_ModeloRelacional.pdf)
9. Tesis\_Dolores Cuadra 2003 - HULAT, fecha de acceso: agosto 26, 2025,  
[https://hulat.inf.uc3m.es/sites/default/files/tesis/tesis\\_dolores\\_cuadra\\_2003.pdf](https://hulat.inf.uc3m.es/sites/default/files/tesis/tesis_dolores_cuadra_2003.pdf)

10. Fundamentos de bases de datos - IHMC Public Cmaps (2), fecha de acceso: agosto 26, 2025,  
<https://cursa.ihmc.us/rid=1TBDTPB8W-ZHQ89-3LKP/2%20base%20de%20datos.pdf>
11. Sentencia CREATE TABLE - IBM, fecha de acceso: agosto 26, 2025,  
<https://www.ibm.com/docs/es/db2/11.5.x?topic=statements-create-table>
12. Bases de datos - UOC, fecha de acceso: agosto 26, 2025,  
<https://www.uoc.edu/pdf/masters/oficiales/img/913.pdf>
13. Uma Proposta para Mapeamento de Documentos JSON para, fecha de acceso: agosto 26, 2025,  
<https://repositorio.ufsc.br/bitstream/handle/123456789/255794/TCC.pdf?sequence=1&isAllowed=y>
14. Fundamentos de Sistemas de Bases de Datos 5th Edition Ramez A ..., fecha de acceso: agosto 26, 2025,  
<https://es.scribd.com/document/904685645/Fundamentos-de-Sistemas-de-Bases-de-Datos-5th-Edition-Ramez-A-Elmasri-full>
15. Fundamentos bd | PDF - SlideShare, fecha de acceso: agosto 26, 2025,  
<https://es.slideshare.net/slideshow/fundamentos-bd-27420795/27420795>
16. Teórico 1 - Algebra Relacional - Bases de Datos 2, fecha de acceso: agosto 26, 2025,  
<https://www.fing.edu.uy/tecnoinf/maldonado/cursos/bd2/materiales/teo/bd2-teorico01.pdf>
17. Álgebra Relacional - Lienzos UV, fecha de acceso: agosto 26, 2025,  
[https://lienzos.uv.mx/Uploads/resources/LenguajeAlgebraRelacional\\_18e0.pdf](https://lienzos.uv.mx/Uploads/resources/LenguajeAlgebraRelacional_18e0.pdf)
18. Tema III: El enfoque relacional - Departamento de Informática e Ingeniería de Sistemas, fecha de acceso: agosto 26, 2025, [http://webdiis.unizar.es/asignaturas/BD/transparenciasBD/PDFs\\_1x1/leccion\\_5.pdf](http://webdiis.unizar.es/asignaturas/BD/transparenciasBD/PDFs_1x1/leccion_5.pdf)
19. Álgebra relacional y de bolsas, fecha de acceso: agosto 26, 2025,  
<https://users.exa.unicen.edu.ar/catedras/dbases1/clases/c4-2008-s3.pdf>
20. Fundamentos de Bases de Datos 9789587650020, 9789587654875 - DOKUMEN.PUB, fecha de acceso: agosto 26, 2025,  
<https://dokumen.pub/fundamentos-de-bases-de-datos-9789587650020-9789587654875.html>
21. IS NULL and IS NOT NULL Conditions - IBM, fecha de acceso: agosto 26, 2025,  
<https://www.ibm.com/docs/en/informix-servers/14.10.0?topic=condition-is-null-is-not-null-conditions>
22. Db2 13 - SQL de Db2 : funciones de agregación - IBM, fecha de acceso: agosto 26, 2025,  
<https://www.ibm.com/docs/es/db2-for-zos/13.0.0?topic=functions-aggregate>
23. Funciones de agregación | Referencia SQL de Claris FileMaker, fecha de acceso: agosto 26, 2025,  
<https://help.claris.com/es/sql-reference/content/aggregate-functions.html>
24. SQL IS NOT NULL Operator - GeeksforGeeks, fecha de acceso: agosto 26, 2025,  
<https://www.geeksforgeeks.org/sql/sql-is-not-null-operator/>
25. FUNDAMENTOS DE BASES DE DATOS 6ED - Ingebook, fecha de acceso: agosto 26, 2025,  
[https://www.ingebook.com/ib/NPcd/IB\\_BooksVis?cod\\_primaria=1000187&codigo\\_libro=5935](https://www.ingebook.com/ib/NPcd/IB_BooksVis?cod_primaria=1000187&codigo_libro=5935)
26. Reglas de Transformación Del Modelo ER A Modelo Relacional | PDF - Scribd, fecha de acceso: agosto 26, 2025,  
<https://es.scribd.com/document/178828719/Reglas-de-transformacion-del-modelo-ER-a-Modelo-Relacional>
27. Pasaje de Modelo E-R a Modelo Relacional - FING, fecha de acceso: agosto 26, 2025,  
<https://www.fing.edu.uy/tecnoinf/paysandu/cursos/2do/bd1/material/bd1-8.pdf>
28. Teórico 10 – Del MER al MR (entidades débiles y relaciones) - FING, fecha de acceso: agosto 26, 2025,  
<https://www.fing.edu.uy/tecnoinf/maldonado/cursos/bd1/materiales/teo/teorico10.pdf>
29. Mer A MR | PDF | Base de datos relacional | Modelo de datos - Scribd, fecha de acceso: agosto 26, 2025,  
<https://es.scribd.com/document/444522694/mer-a-mr>