

Matemáticas Discretas

Bases de datos relacionales

Clase 9



INTRODUCCIÓN DE LA CLASE

En el mundo digital actual, gran parte de la información que utilizamos a diario se organiza y gestiona a través de bases de datos. Cuando consultas tus calificaciones en línea, realizas una compra en una tienda virtual o revisas los registros de acceso a un sistema, detrás de todo ello existen estructuras matemáticas que permiten organizar los datos de forma clara y eficiente. Estas estructuras se apoyan en el concepto de relación, estudiado previamente, que se convierte ahora en el fundamento para entender cómo funcionan las bases de datos relacionales.

En esta clase exploraremos cómo las relaciones matemáticas se generalizan para modelar tablas de datos con múltiples atributos, cómo operamos con ellas mediante herramientas inspiradas en la teoría de conjuntos y, finalmente, cómo se definen las claves que garantizan la identificación única de la información. De este modo, podrás comprender no solo la lógica detrás de las bases de datos relacionales, sino también su utilidad práctica para organizar y consultar datos en diferentes contextos de la ciberseguridad y otras áreas afines.

Clase 9: Bases de datos relacionales

RDA 2: Resolver problemas prácticos y teóricos mediante la aplicación de técnicas y herramientas de la Matemática Discreta, como lógica matemática, análisis de conjuntos y teoría de grafos en el manejo de estructuras discretas.

9. Bases de datos relacionales

Hasta ahora hemos trabajado con relaciones binarias, es decir, aquellas que vinculan elementos de dos conjuntos y que, representadas en forma de tabla, poseen únicamente dos columnas. El prefijo «bi» en una relación binaria R hace referencia precisamente a este hecho: cada par ordenado (a, b) corresponde a una fila en una tabla de dos atributos.

Sin embargo, en muchos contextos resulta más útil y natural extender esta idea y permitir que una tabla contenga un número arbitrario de columnas. De esta manera, una relación con n atributos se denomina **relación n -aria**, y cada fila de la tabla representa una n -upla. Este enfoque constituye la base de las bases de datos

relacionales, donde la información se organiza en tablas que reflejan de manera directa las generalizaciones de las relaciones matemáticas estudiadas.

El modelo de bases de datos relacionales, introducido por Edgar Codd en 1970, se fundamenta justamente en esta visión matemática de las relaciones. En dicho modelo, cada tabla es una relación n -aria, cada fila corresponde a un registro (o tupla) y cada columna se interpreta como un atributo que proviene de un dominio específico. Este marco formal ofrece varias ventajas:

- Permite describir y manipular la información con un alto grado de precisión.
- Facilita el diseño de sistemas en los que la integridad de los datos es prioritaria.
- Hace posible expresar consultas complejas mediante un lenguaje algebraico, que luego puede traducirse en implementaciones prácticas (como SQL).

9.1. Generalización de relaciones

Recordemos que una relación puede entenderse como un subconjunto de un producto cartesiano. Hasta el momento hemos trabajado principalmente con relaciones binarias, es decir, relaciones entre dos conjuntos. Sin embargo, es posible ampliar esta noción a más de dos conjuntos, permitiendo así representar situaciones más complejas y realistas.

Recordemos que, dados los conjuntos $A_1, A_2, \dots, A_n$, se define el producto cartesiano de $A_1, A_2, \dots, A_n$ como

$$A_1 \times A_2 \times \dots \times A_n = \{(x_1, x_2, \dots, x_n) : x_1 \in A_1, x_2 \in A_2, \dots, x_n \in A_n\}.$$

Así, una relación n -aria puede verse como una colección de n -uplas que describen cómo interactúan elementos de distintos conjuntos (Epp 2020).

Definición 1: Relación n -aria

Dados los conjuntos $A_1, \dots, A_n$, se dice que R es una *relación n -aria en $A_1, \dots, A_n$* si $R \subseteq A_1 \times A_2 \times \dots \times A_n$. A los conjuntos A_i se los denomina *dominios de la relación* y n es el *grado de la relación*.

En otras palabras, una relación n -aria está formada por n -uplas que vinculan simultáneamente elementos de varios conjuntos. Las relaciones binarias son un caso particular en el que $n = 2$.

Ejemplo 1. Sea $A = \{\text{Guillermo}, \text{María}, \text{Beatriz}\}$ el conjunto de estudiantes y $B = \{\text{Álgebra}, \text{Cálculo}\}$ el conjunto de cursos. La relación $R \subseteq A \times B$ dada por

$$S = \{(\text{Guillermo}, \text{Álgebra}), (\text{María}, \text{Cálculo}), (\text{Beatriz}, \text{Álgebra})\}$$

representa qué estudiante está inscrito en qué curso. Aquí tenemos una relación binaria ($n = 2$), cuyos dominios son A y B.

Visto como base de datos, cada par ordenado corresponde a un registro, con dos campos: el estudiante y el curso. Los atributos de la relación son precisamente los conjuntos A (Estudiantes) y B (Cursos).

Tabla 1. Relación binaria Estudiantes–Cursos

<i>Estudiante</i>	<i>Curso</i>
<i>Guillermo</i>	<i>Álgebra</i>
<i>María</i>	<i>Cálculo</i>
<i>Beatriz</i>	<i>Álgebra</i>

Nota: creación propia (Merino, A., 2025).

En el contexto de las bases de datos, a una relación n -aria R en $A_1, \dots, A_n$ se denomina una tabla de datos, a cada elemento de R se lo llama *registro*; a cada entrada de un registro se la denomina *campo*; y a cada conjunto A_i se lo llama *atributo*.

Así, en el ejemplo anterior, tenemos que la tabla tiene dos atributos (Estudiantes y Curso), además tiene tres registros (tres elementos o tres filas), por ejemplo $(\text{Guillermo}, \text{Álgebra})$ es un registro; finalmente, cada registro tiene dos campos, así, el campo Estudiante del segundo registro es María.

Ejemplo 2. Ahora supongamos que además queremos incluir la calificación obtenida en cada curso. Introducimos un tercer conjunto $C = \{0, 10, 20, 30, 40, 50\}$ que representa las notas posibles. La relación $R \subseteq A \times B \times C$ puede expresarse como

$$R = \{(\text{Guillermo}, \text{Álgebra}, 40), (\text{María}, \text{Cálculo}, 50), (\text{Beatriz}, \text{Álgebra}, 20)\}.$$

Se trata de una relación ternaria ($n = 3$), cuyos dominios son A, B y C. En este caso, cada 3-upla constituye un registro, y cada valor dentro de la tupla es un campo. Los atributos de la relación son los conjuntos: Estudiantes (A), Cursos (B) y Calificaciones (C).

Tabla 2. Relación ternaria Estudiantes–Cursos–Calificaciones

<i>Estudiante</i>	<i>Curso</i>	<i>Calificación</i>
<i>Guillermo</i>	<i>Álgebra</i>	<i>40</i>
<i>María</i>	<i>Cálculo</i>	<i>50</i>
<i>Beatriz</i>	<i>Álgebra</i>	<i>20</i>

Nota: creación propia (Merino, A., 2025).

Estos ejemplos muestran cómo, al aumentar el número de dominios, crecen también el grado y la complejidad de la relación, dando lugar a estructuras tabulares que forman la base de las bases de datos relacionales.

9.2. Operaciones básicas

En el contexto de las bases de datos relacionales, las operaciones que se realizan sobre las tablas están inspiradas en las operaciones de conjuntos. Estas operaciones permiten extraer, combinar o transformar información, y forman parte del llamado **álgebra relacional** (Johnsonbaugh 2018).

El álgebra relacional proporciona un conjunto de operadores bien definidos que reciben como entrada una o varias relaciones y producen como salida otra relación. De esta manera, se garantiza que el resultado de cualquier consulta pueda seguir siendo tratado con las mismas reglas, manteniendo la consistencia dentro del modelo relacional. Entre las operaciones más comunes se encuentran la *selección*, que permite filtrar registros según condiciones específicas; la *proyección*, que se utiliza para escoger columnas determinadas; y la *reunión* (join), que combina información de varias tablas en función de atributos compartidos.

A continuación, estudiaremos con detalle algunas de estas operaciones, ilustrando su definición formal y su aplicación práctica mediante ejemplos en tablas.

Definición 2: Operador selección.

Sean R una relación n -aria y C una condición sobre los elementos de $A_1 \times \dots \times A_n$. Se define

$$\sigma_C(R) = \{(x_1, \dots, x_n) \in R : (x_1, \dots, x_n) \text{ cumple } C\}.$$

El operador *selección* permite obtener únicamente los registros que cumplen cierta condición. Por ejemplo, si R es la relación ternaria Estudiantes–Cursos–Calificaciones de la sección anterior, podemos seleccionar únicamente los estudiantes que aprobaron con nota mayor o igual a 30.

$$\sigma_{\text{Calificación} \geq 30}(R) = \{(\text{Guillermo}, \text{Álgebra}, 40), (\text{María}, \text{Cálculo}, 50)\}.$$

Tabla 3. Selección de estudiantes con calificación mayor o igual a 9

Estudiante	Curso	Calificación
Guillermo	Álgebra	40
María	Cálculo	50

Nota: creación propia (Merino, A., 2025).

Definición 3: Operador proyección.

Sea R una relación n -aria y $\{i_1, \dots, i_m\} \subseteq \{1 \dots n\}$. Para $(x_1, \dots, x_n) \in R$, se define

$$\pi_{i_1, \dots, i_m}(R) = \{(x_{i_1}, \dots, x_{i_m}) : (x_1, \dots, x_n) \in R\}.$$

La *proyección* permite elegir solo algunos atributos de una relación, descartando los demás. Por ejemplo, si proyectamos la relación R de Estudiantes–Cursos–Calificaciones sobre los atributos Estudiante y Curso, se obtiene:

$$\pi_{\text{Estudiante}, \text{Curso}}(R) = \{(\text{Guillermo}, \text{Álgebra}), (\text{María}, \text{Cálculo}), (\text{Beatriz}, \text{Álgebra})\}.$$

Tabla 4

Proyección de la relación sobre Estudiantes y Cursos

Estudiante	Curso
Guillermo	Álgebra
María	Cálculo
Beatriz	Álgebra

Nota: creación propia (Merino, A., 2025).

Definición 4: Operador producto cartesiano.

Sean $R \subseteq A_1 \times \dots \times A_m$ y $S \subseteq B_1 \times \dots \times B_n$ dos relaciones. El *producto cartesiano* de R y S , denotado $R \times S$, es la relación

$$R \times S = \{(x_1, \dots, x_m, y_1, \dots, y_n) : (x_1, \dots, x_m) \in R, (y_1, \dots, y_n) \in S\}.$$

El operador *producto cartesiano* permite combinar todos los registros de una relación con todos los registros de otra. Si R tiene r registros y S tiene s , entonces $R \times S$ tendrá $r \cdot s$ registros.

Consideremos la relación S de Cursos–Docentes:

$$S = \{(\text{Álgebra}, \text{Prof. Merino}), (\text{Cálculo}, \text{Prof. Cueva})\}.$$

Tabla 5. *Relación Cursos–Docentes*

Curso	Docente
Álgebra	Prof. Merino
Cálculo	Prof. Cueva

Nota: creación propia (Merino, A., 2025).

Entonces el producto cartesiano es:

$$R \times S = \{ (\text{Guillermo}, \text{Álgebra}, 40, \text{Álgebra}, \text{Prof. Merino}), \\ (\text{Guillermo}, \text{Álgebra}, 40, \text{Cálculo}, \text{Prof. Cueva}), \\ (\text{María}, \text{Cálculo}, 50, \text{Álgebra}, \text{Prof. Merino}), \\ (\text{María}, \text{Cálculo}, 50, \text{Cálculo}, \text{Prof. Cueva}), \\ (\text{Beatriz}, \text{Álgebra}, 20, \text{Álgebra}, \text{Prof. Merino}), \\ (\text{Beatriz}, \text{Álgebra}, 20, \text{Cálculo}, \text{Prof. Cueva}) \}.$$

Tabla 6. *Producto cartesiano de Estudiantes–Cursos–Calificaciones y Cursos–Docentes*

Estudiante	Curso (R)	Calificación	Curso (S)	Docente
Guillermo	Álgebra	40	Álgebra	Prof. Merino
Guillermo	Álgebra	40	Cálculo	Prof. Cueva
María	Cálculo	50	Álgebra	Prof. Merino
María	Cálculo	50	Cálculo	Prof. Cueva
Beatriz	Álgebra	20	Álgebra	Prof. Merino
Beatriz	Álgebra	20	Cálculo	Prof. Cueva

Nota: creación propia (Merino, A., 2025).

El producto cartesiano por sí mismo genera todas las posibles combinaciones entre los registros de dos relaciones, lo cual puede resultar de utilidad para ciertos análisis. Sin embargo, en la práctica, rara vez se requiere toda esta información, ya que el número de combinaciones puede crecer rápidamente y producir resultados redundantes.

Lo interesante del producto cartesiano es que sirve como base para definir una de las operaciones más importantes del álgebra relacional: la *reunión* (*join*). De hecho, la reunión no es más que una **combinación de dos operaciones**: primero se aplica el producto cartesiano entre dos relaciones y luego una selección que filtra las combinaciones que cumplen cierta condición.

Definición 5: Operador reunión (Theta-join).

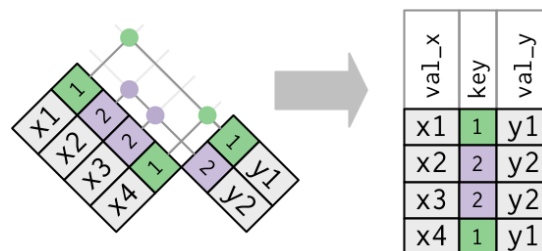
Sean $R \subseteq A_1 \times \dots \times A_m$ y $S \subseteq B_1 \times \dots \times B_n$ dos relaciones, y sea θ una condición sobre los atributos de R y S . La *reunión* de R y S bajo la condición θ , denotada

$$R \bowtie_{\theta} S,$$

se define como

$$R \bowtie_{\theta} S = \sigma_{\theta}(R \times S).$$

Figura 1. Ejemplificación del operador reunión



Nota: tomado de Grolemond (n.d.)

La reunión (o join) permite combinar dos relaciones a partir de una condición sobre atributos comunes, eliminando las combinaciones que no son relevantes.

Siguiendo el ejemplo anterior, el producto cartesiano $R \times S$ genera todas las combinaciones posibles (como se mostró antes). Si ahora aplicamos la condición $R.\text{Curso} = S.\text{Curso}$ (es decir, el atributo Curso en la relación R coincida con el atributo Curso de la relación S), obtenemos la reunión:

$$R \bowtie_{R.\text{Curso}=S.\text{Curso}} S = \{ (\text{Guillermo}, \text{Álgebra}, 40, \text{Prof. Merino}), \\ (\text{María}, \text{Cálculo}, 50, \text{Prof. Cueva}), \\ (\text{Beatriz}, \text{Álgebra}, 20, \text{Prof. Merino}) \}.$$

Tabla 7. Reunión entre Estudiantes–Cursos–Calificaciones y Cursos–Docentes

Estudiante	Curso	Calificación	Docente
Guillermo	Álgebra	40	Prof. Merino
María	Cálculo	50	Prof. Cueva
Beatriz	Álgebra	20	Prof. Merino

Nota: creación propia (Merino, A., 2025).

La utilidad de la operación de reunión radica en que permite combinar información proveniente de distintas relaciones en una sola estructura. En el ejemplo anterior, a partir de la relación Estudiantes–Cursos–Calificaciones y de la relación Cursos–Docentes, hemos obtenido una nueva relación que incluye no solo el curso y la calificación de cada estudiante, sino también el docente responsable de dicho curso. De esta manera, la reunión enriquece los datos disponibles y facilita responder preguntas más complejas, como: ¿qué docente colocó calificaciones inferiores a 30?

Para esto, deberíamos hacer un par de operaciones más, primero seleccionar los registros que tengas calificaciones menores a 30:

$$\sigma_{\text{Calificación} < 30}(R \bowtie_{R.\text{Curso}=S.\text{Curso}} S).$$

Tabla 8. Registros con calificaciones menores a 30

Estudiante	Curso	Calificación	Docente
Beatriz	Álgebra	20	Prof. Merino

Nota: creación propia (Merino, A., 2025).

Finalmente, tomemos una proyección para tener solo el nombre de los docentes:

$$\pi_{\text{Docente}}(\sigma_{\text{Calificación} < 30}(R \bowtie_{R.\text{Curso}=S.\text{Curso}} S)).$$

Tabla 9. Docentes que colocaron calificaciones inferiores a 30

Docente
Prof. Merino

Nota: creación propia (Merino, A., 2025).

En álgebra relacional, una consulta (en inglés, query) es una expresión que combina operadores como selección, proyección y reunión para responder una pregunta sobre los datos. El resultado de una consulta es siempre una nueva relación. En el ejemplo anterior, se tenía la pregunta «¿qué docente colocó calificaciones inferiores a 30?», la consulta (query) necesaria para responder fue:

$$\pi_{\text{Docente}}(\sigma_{\text{Calificación} < 30}(R \bowtie_{R.\text{Curso} = S.\text{Curso}} S))$$

y obtuvimos que el docente Merino es el único que colocó calificaciones inferiores a 30.

En el siguiente recurso se presenta una visión formal del álgebra relacional, incluyendo otros operadores adicionales que no alcanzamos a cubrir en esta clase:

- **Título del enlace:** Relational Algebra Overview
- **Descripción del enlace:** Explicación detallada de los operadores fundamentales del álgebra relacional.
- **Enlace:** https://data101.org/notes/2-relational_algebra/primitives.html

En el siguiente video permite ver ejemplos concretos de cómo se aplican los operadores básicos del álgebra relacional:

- **Título del enlace:** Álgebra Relacional - operadores básicos
- **Descripción del enlace:** Video introductorio sobre los operadores básicos del álgebra relacional, con ejemplos.
- **Enlace:** <https://youtu.be/hsNZnJjYXGM?si=0QRViySrxHJDxagZ>

Consideremos el siguiente ejemplo del uso del álgebra relacional.

Ejemplo 3. En un Campeonato Universitario, cada equipo pertenece a un grupo (A o B) y se registran los partidos disputados. Trabajaremos con dos relaciones:

- **Equipos:** relación 3-aria. Dominios: nombres de equipos (Eq), Grupos, Ciudades.
- **Partidos:** relación 5-aria. Dominios: Jornadas, Equipos, Equipos, Goles locales (GolesL), Goles visitantes (GolesV).

Tabla 10. Relación Equipos (Eq, Grupo, Ciudad)

<i>Eq</i>	<i>Grupo</i>	<i>Ciudad</i>
<i>Alfa</i>	<i>A</i>	<i>Quito</i>
<i>Beta</i>	<i>A</i>	<i>Cuenca</i>
<i>Gamma</i>	<i>B</i>	<i>Loja</i>
<i>Delta</i>	<i>A</i>	<i>Ambato</i>

Nota: creación propia (Merino, A., 2025).

Tabla 11. *Relación Partidos (Jornada, Local, Visitante, GolesL, GolesV)*

<i>Jornada</i>	<i>Local</i>	<i>Visitante</i>	<i>GolesL</i>	<i>GolesV</i>
<i>1</i>	<i>Alfa</i>	<i>Beta</i>	<i>1</i>	<i>2</i>
<i>1</i>	<i>Gamma</i>	<i>Delta</i>	<i>0</i>	<i>0</i>
<i>2</i>	<i>Delta</i>	<i>Alfa</i>	<i>0</i>	<i>1</i>
<i>2</i>	<i>Beta</i>	<i>Gamma</i>	<i>3</i>	<i>1</i>
<i>3</i>	<i>Beta</i>	<i>Delta</i>	<i>0</i>	<i>0</i>
<i>3</i>	<i>Gamma</i>	<i>Alfa</i>	<i>0</i>	<i>2</i>

Nota: creación propia (Merino, A., 2025).

Pregunta: ¿Qué equipos del Grupo A ganaron al menos un partido como visitantes?

Para dar respuesta, construyamos la consulta (query) como composición de operaciones.

1. Filtrar partidos ganados por el visitante (selección): Seleccionamos de Partidos solo aquellas tuplas donde el visitante anotó más que el local:

$$P_1 = \sigma_{\text{GolesV} > \text{GolesL}}(\text{Partidos}).$$

Con esto, la relación P_1 contiene únicamente victorias de visitante.

2. Anexar el grupo del equipo visitante (reunión / theta-join): Unimos P_1 con Equipos haciendo coincidir el nombre del visitante con el atributo Eq:

$$P_2 = P_1 \bowtie_{\text{Visitante}=\text{Eq}} \text{Equipos}.$$

Ahora cada fila lleva, además de los datos del partido, el Grupo y la Ciudad del equipo visitante.

3. Quedarnos solo con visitantes del Grupo A (selección).

$$P_3 = \sigma_{\text{Grupo}='A'}(P_2).$$

4. Obtener la lista de equipos (proyección, sin duplicados).

$$\text{Resultado} = \pi_{\text{Eq}}(P_3).$$

Resumiendo, la consulta es:

$$\pi_{\text{Eq}} \left(\sigma_{\text{Grupo}='A'} \left(\sigma_{\text{GolesV} > \text{GolesL}} (\text{Partidos}) \bowtie_{\text{Visitante}=\text{Eq}} \text{Equipos} \right) \right).$$

Tabla 12

Equipos del Grupo A con al menos una victoria como visitantes

<i>Eq</i>
<i>Alfa</i>
<i>Beta</i>

Nota: creación propia (Merino, A., 2025).

Con este resultado, los equipos del Grupo A que ganaron al menos un partido como visitantes son Alfa (jornadas 2 y 3) y Beta (jornada 1).

Te invitamos a escribir la tabla resultante en cada paso para comprobar las operaciones realizadas.

9.3. Claves primarias y compuestas

En las bases de datos relacionales es fundamental poder identificar de manera única cada registro de una tabla. Para ello se utilizan las *claves*, que pueden ser primarias o compuestas.

Definición 6: Clave primaria.

Sean R una relación n -aria en $A_1, \dots, A_n$. Se dice que el atributo A_i es una *clave primaria* para R si para todo $x, y \in R$ se cumple que

$$x_i = y_i \quad \text{implica} \quad x = y.$$

Es decir, no existen dos registros diferentes con el mismo valor en el atributo A_i .

Ejemplo 4. Consideremos la relación de Estudiantes:

Tabla 13*Relación Estudiantes (ID, Nombre, Ciudad)*

<i>ID</i>	<i>Nombre</i>	<i>Ciudad</i>
101	Guillermo	Quito
102	María	Cuenca
103	Beatriz	Loja

Nota: creación propia (Merino, A., 2025).

Aquí el atributo ID es una clave primaria, pues cada valor de ID identifica de forma única a un estudiante.

Definición 7: Clave Compuesta

Sean R una relación n -aria en $A_1, \dots, A_n$. Se dice que los atributos $A_{i_1}, \dots, A_{i_m}$ forman una *clave compuesta* para R si para todo $x, y \in R$ se cumple que

$$(x_{i_1}, \dots, x_{i_m}) = (y_{i_1}, \dots, y_{i_m}) \quad \text{implica} \quad x = y.$$

En este caso, la combinación de varios atributos permite distinguir de manera única cada registro.

Ejemplo 5. Consideremos la relación de Matrículas:

Tabla 14*Relación Matrículas (Estudiante, Curso, Semestre)*

<i>Estudiante</i>	<i>Curso</i>	<i>Semestre</i>
Guillermo	Álgebra	2025A
María	Cálculo	2025A
Guillermo	Cálculo	2025A
Beatriz	Álgebra	2025B

Nota: creación propia (Merino, A., 2025).

En esta tabla no hay un único atributo que identifique a los registros: un mismo estudiante puede estar en varios cursos, y un curso puede tener varios estudiantes. La

clave compuesta está formada por la pareja (Estudiante, Curso), que garantiza que no se repita un mismo registro.

De esta manera, vemos que las claves permiten garantizar la unicidad de los registros en una base de datos relacional. Según el contexto, puede bastar con un solo atributo que actúe como clave primaria, o puede ser necesario recurrir a la combinación de varios atributos formando una clave compuesta. Estos mecanismos resultan fundamentales para mantener la coherencia y la organización de la información dentro de las tablas.

Además, las claves no solo cumplen un papel de identificación, sino que también facilitan la creación de vínculos entre distintas relaciones. En sistemas más complejos, un registro de una tabla puede estar relacionado con registros de otra a través de atributos comunes, y son precisamente las claves las que hacen posible establecer estas conexiones de manera confiable. Así, las claves primarias y compuestas se convierten en la base para que las bases de datos no sean simplemente colecciones de tablas aisladas, sino sistemas estructurados en los que la información puede integrarse y consultarse de forma eficiente.

Referencias

Epp, S. S. (2020). Discrete Mathematics with Applications (5.a ed.). Cengage Learning.

Johnsonbaugh, R. (2018). Discrete Mathematics (8.a ed.). Pearson Educación.

Wickham, H. y Grolemond, G. (s.f.). 13 Relational data | R for Data Science [Recuperado el 17 de agosto de 2025]. <https://r4ds.had.co.nz/relational-data.html#join-matches>

Términos

- **Álgebra relacional:** conjunto de operaciones formales que se aplican sobre relaciones en bases de datos, permitiendo consultar, combinar y transformar información de manera sistemática.
- **Relación n -aria:** subconjunto del producto cartesiano $A_1 \times A_2 \times \dots \times A_n$, donde cada registro es una n -upla formada por elementos de los dominios $A_1, \dots, A_n$.

Profundización

Recurso 1

- **Título del recurso:** Bases de datos relacionales con Pandas
- **Descripción del enlace:** Este recurso presenta una guía de Python para escribir operaciones del álgebra relacional.
- **Enlace:** <https://alephsub0.org/material-nuevo/andres-merino/bases-de-datos-relacionales-con-pandas/>

Recurso 2

- **Título del recurso:** Bases de datos relacionales con Pandas
- **Descripción del enlace:** Este recurso presenta una guía de Python y ejercicios sobre2023 operaciones del álgebra relacional.
- **Enlace:** <https://colab.research.google.com/github/FCENA-PUCE/Python-Notebooks-Educativos/blob/main/Base-datos-relacionales-Pandas.ipynb>



La excelencia no se improvisa

síguenos

