

GESTIONANDO LOS DATOS RELACIONES Y ALGEBRA RELACIONAL

**AUTOR: MAX JOSÉ BERMÚDEZ LEÓN
DICIEMBRE: 2020**



Introducción

En el entorno del mercado actual, la competitividad y la rapidez de maniobra de una empresa son imprescindibles para su éxito. Para conseguirlo existe cada vez una mayor demanda de datos y, por tanto, más necesidad de gestionarlos. Esta demanda siempre ha estado patente en empresas y sociedades, pero en estos años se ha disparado debido al acceso multitudinario a las redes integradas en Internet y a la aparición de los dispositivos móviles que también requieren esa información.

En informática se conoce como dato a cualquier elemento informativo que tenga relevancia para un usuario. Desde su nacimiento, la informática se ha encargado de proporcionar herramientas que faciliten la manipulación de los datos. Antes de la aparición de las aplicaciones informáticas, las empresas tenían como únicas herramientas de gestión de datos los ficheros con cajones, carpetas y fichas de cartón. En este proceso manual, el tiempo requerido para manipular estos datos era enorme. Pero la propia informática ha adaptado sus herramientas para que los elementos que el usuario utiliza en cuanto a manejo de datos se parezcan a los manuales.

La clientela fundamental del profesional informático es la empresa. La empresa se puede entender como un sistema de información formado por diversos objetos: el capital, los recursos humanos, los inmuebles, los servicios que presta, etc.

Los sistemas de información actuales se basan en bases de datos (BD) y sistemas de bases de datos (SGBD) que se han convertido en elementos imprescindibles de la vida cotidiana de la sociedad moderna.



Contenido

Introducción.....	1
Gestionando los datos.....	3
Concepto de Relaciones	3
Cardinalidad	3
Concepto de Tabla.....	13
Tipos de Tablas	14
Concepto de Dominio	14
Dominio y atributo	15
Concepto de Tupla o Registro	16
Concepto de Esquemas.....	17
Algebra Relacional.....	18
Tuplas	18
Operaciones Básicas.....	19
Conclusiones y recomendaciones.....	26
Referencias bibliográficas.....	27

La integridad de datos se refiere a la exactitud, fiabilidad, precisión, integralidad y coherencia general de los datos. La integridad de datos se mantiene gracias a un conjunto de procesos, reglas y normas que se ponen en práctica durante la fase de diseño.

Gestionando los datos

Concepto de Relaciones

Los diseños de la base de datos están estrechamente relacionados con las relaciones de la base de datos, la asociación entre dos columnas en una o más tablas. Los tipos de relaciones no deben confundirse con los tipos de datos en SQL, esto se analizó en el módulo 2, siendo

conceptos distintos.

Las relaciones se definen sobre la base de columnas de clave coincidentes.

En SQL, estas relaciones se definen utilizando restricciones de clave principal a clave externa. Se crea un enlace entre dos tablas donde la clave principal de una tabla se asocia con la clave externa de otra tabla utilizando las relaciones de la base de datos.

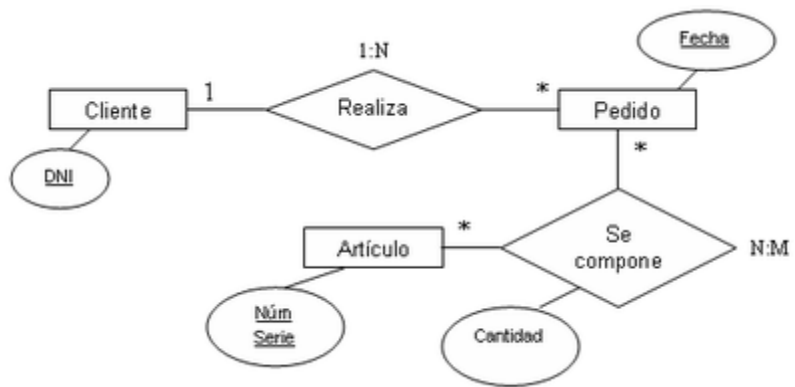
Cardinalidad

La cardinalidad es una restricción en una relación que especifica el número de instancias de entidad que una entidad específica puede estar relacionada a través de la relación.

Cuando preguntamos ¿Cuántos empleados pueden trabajar en un solo departamento? ¿en qué departamentos puede trabajar un empleado? estamos haciendo preguntas con respecto a la cardinalidad de la relación.

Las tres clasificaciones son: uno a uno, uno a muchos y muchos a muchos.

A continuación, el diagrama muestra una relación entre las líneas de pedidos y los clientes.



La "n" representa un "número arbitrario de instancias", y el "1" representa "como máximo uno".

Interpretamos las especificaciones de cardinalidad con la siguiente regla comercial:

La "n" indica que la misma entidad de pedidos se puede especificar a cualquier número de pedidos.

El "1" indica que una entidad de cliente especifica a lo más un Pedido.

Mapeo de Cardinalidades

La cardinalidad define la cantidad de entidades en un conjunto de entidades, que pueden asociarse con el número de entidades de otro conjunto a través del conjunto de relaciones.

Uno a uno

Una entidad del conjunto de entidades A puede asociarse con a lo sumo una entidad del conjunto de entidades B y viceversa.

Se identifica de la siguiente forma:

- 1:1

Ejemplo dos entidades, profesor y departamento, con llaves primarias, código_profesor y jefe_depto respectivamente, un profesor solo puede ser jefe de un departamento y un departamento solo puede tener un jefe.

Las relaciones uno-a-uno tiene 1 especificado para ambas cardinalidades, y no parecen surgir muy a menudo. Para ilustrar un uno-a-uno, requerimos reglas comerciales muy específicas.

Supongamos que tenemos personas y vehículos. Supongamos que solo nos preocupamos por el conductor actual de un vehículo, y que solo nos preocupa el vehículo actual que el conductor está conduciendo. Entonces, tenemos una relación uno-a-uno entre Vehículo y Persona.

Uno a muchos

Una entidad del conjunto de entidades A puede asociarse con más de una entidad del conjunto de entidades B, sin embargo, una entidad del conjunto de entidades B puede



asociarse con, como máximo, a una entidad.

Se identifica:

- 1:N

Por ejemplo: dos entidades, vendedor y ventas, con llaves primarias, código_vendedor y venta, respectivamente, un vendedor puede tener muchas ventas, pero una venta solo puede tener un vendedor

Muchos a uno

Más de una entidad del conjunto de entidades A puede asociarse como máximo con una entidad del conjunto de entidades B, sin embargo, una entidad del conjunto de entidades B puede asociarse con más de una entidad del conjunto de entidades A.

Se identifica:

- N:1

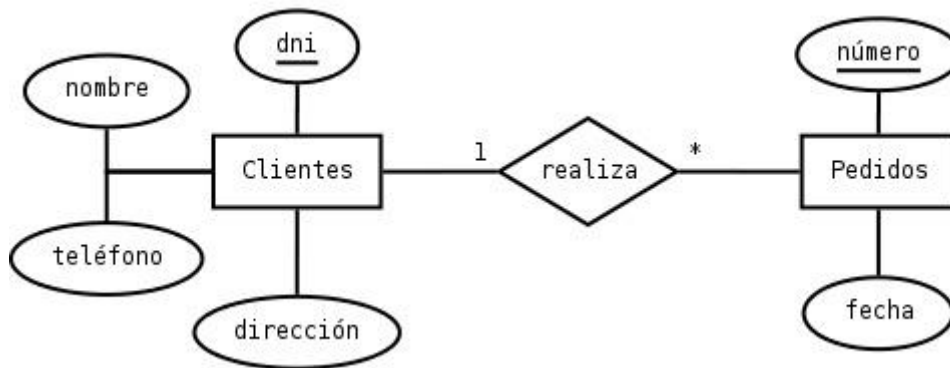
Por ejemplo, empleado-centro de trabajo.

Una entidad de A puede asociarse con más de una entidad desde B y viceversa.

Este tipo de relación tiene 1 y n especificado para sus cardinalidades, y es muy común en diseños de bases de datos. Supongamos que tenemos clientes y pedidos y las reglas comerciales:

- una orden está relacionada con un cliente
- un cliente puede tener cualquier número (cero o más) de pedidos.

Decimos que hay una relación uno a muchos entre el cliente y el pedido, y dibujamos esto como:



Relación de muchos a muchos

Una entidad en A se puede relacionar con 0 o con muchas entidades en B y viceversa.

Se representa

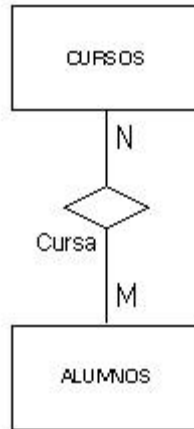
- N:M

Por ejemplo, asociaciones-ciudadanos, donde muchos ciudadanos pueden pertenecer a una misma asociación, y cada ciudadano puede pertenecer a muchas asociaciones distintas.

Otro ejemplo es, supongamos que estamos interesados en cursos y estudiantes y el hecho de que los estudiantes se inscriben en los cursos: cualquier estudiante puede tomar varios cursos, un curso puede ser tomado por varios estudiantes.



Esta situación se representa con una relación de muchos a muchos entre curso y estudiante:



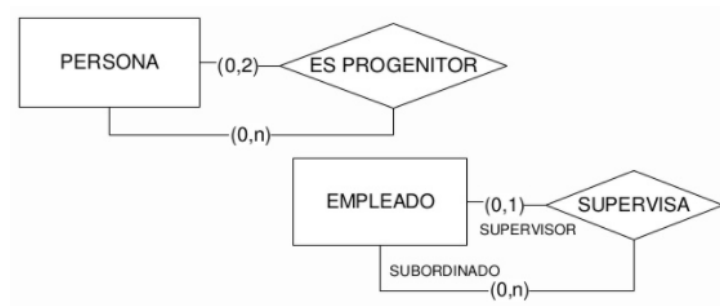
Conjunto de relación

Un conjunto de relaciones de tipo similar se denomina conjunto de relaciones. Al igual que las entidades, una relación también puede tener atributos. Estos atributos se llaman atributos descriptivos.

Grado de relación – Tipos de Relaciones

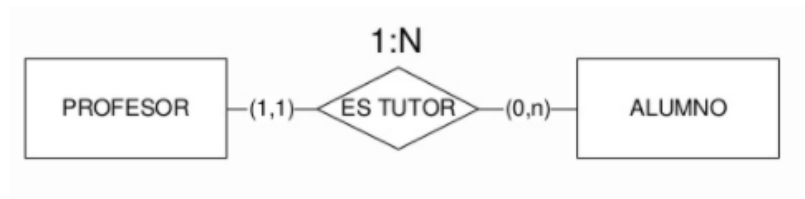
El número de entidades participantes en una relación define el grado de la relación.

- Reflexivas = Grado 1
 - Son relaciones donde participa solo 1 entidad. Se relacionan ocurrencias de la entidad con otras ocurrencias de la propia entidad. Se les denomina anillo o de grado uno.
 - Relaciona una entidad consigo misma por lo que también se les llaman relaciones reflexivas.



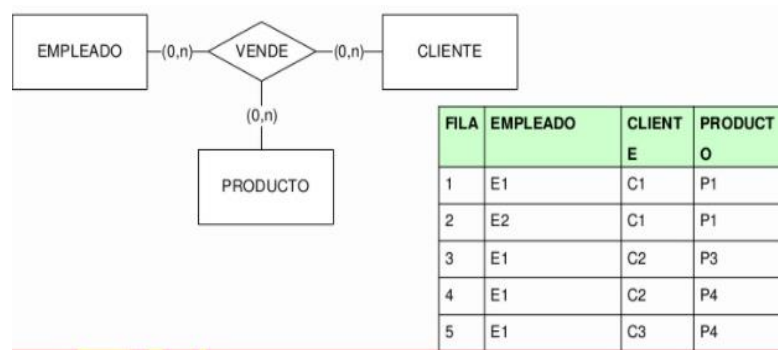
- Binario = Grado 2

- Las relaciones en las que participan dos entidades son binarias o de grado dos.
- Este tipo de relación une a dos entidades.
- Este grado de relación es la más popular en cualquier Diagrama ER.



- Ternario = Grado 3

- Son relaciones donde participan 3 entidades.
- (A,B) -> ; (A,C) -> B ; (B,C) -> A



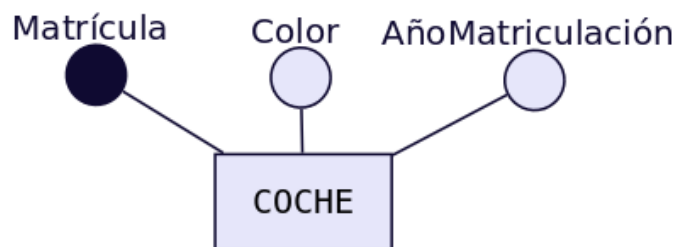
Relaciones de identificación

Antes de analizar las relaciones débiles, debemos recordar que son entidades fuertes o regulares y entidades débiles.

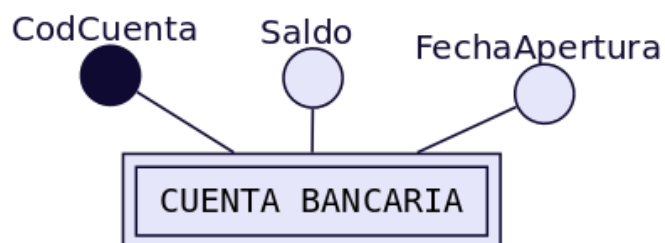
- Una entidad es fuerte si puede existir por sí misma sin que dependa de la existencia de otra entidad.
- Una entidad débil es aquella que sí que necesita de otra para existir. Por ejemplo, en un video-club lo que realmente se alquila a los clientes no son películas, sino las copias de películas que tiene el video-club adquiridas.

En el modelo Entidad-Relación se representa de la siguiente manera

Entidad Fuerte



Entidad Débil



Relación débil

Las relaciones débiles son conexiones entre una entidad débil y su propietario.

Una relación de identificación es una relación entre un tipo de entidad fuerte y una débil, donde se requiere la clave del tipo de entidad fuerte para identificar de manera única las instancias del tipo de entidad débil. El tipo de entidad débil tendrá un atributo de clave parcial (discriminador) que, junto con la clave del tipo de entidad fuerte, identifica de forma única a la entidad débil.



Relaciones no identificativas

Una relación no identificativa es una relación entre tipos de entidades fuertes. Cada entidad tipo tiene una clave especificada, cada entidad tiene un atributo que lo identifica de manera única y lo distingue de cualquier otra instancia en el conjunto de entidad o extensión correspondiente.

Relaciones recursivas

La recursión en un modelo de datos es un tema especialmente difícil. Decimos que tenemos una relación recursiva si el mismo tipo de entidad aparece más de una vez en una relación. Estas relaciones parecen ocurrir siempre en algún lugar en un modelo de datos de tamaño regular a grande. Un típico ejemplo de negocios surge cuando tenemos una regla comercial como "un empleado supervisa otros empleados".



Concepto de Tabla

Tabla en las bases de datos, se refiere al tipo de modelado de datos donde se guardan los datos recogidos por un programa. Su estructura general se asemeja a la vista general de un programa de tablas

Las tablas se componen de dos estructuras:

- Campo: Corresponde al nombre de la columna. Debe ser único y además de tener un tipo de dato asociado.
- Registro: Corresponde a cada fila que compone la tabla. Ahí se componen los datos y los registros. Eventualmente pueden ser nulos en su almacenamiento.

En la definición de cada campo, debe existir un nombre único, con su tipo de dato correspondiente. Esto es útil a la hora de manejar varios campos en la tabla, ya que cada nombre de campo debe ser distinto entre sí.

A los campos se les puede asignar, además, propiedades especiales que afectan a los registros insertados. El campo puede ser definido como índice o autoincrementable, lo cual permite que los datos de ese campo cambien solos o sean el principal indicador a la hora de ordenar los datos contenidos.

Cada tabla creada debe tener un nombre único en la cada base de datos, haciéndola accesible mediante su nombre o su seudónimo (alias) (dependiendo del tipo de base de datos elegida).

La estructura de las tablas viene dada por la forma de un archivo plano, los cuales en un inicio se componían de un modo similar.

Tipos de Tablas

Tablas persistentes

Son aquellas que permiten que los registros sean eliminados o borrados manualmente y tenemos de tres tipos:

- Base: Es donde se encuentra toda la información de todos los registros sin que se haga ninguna validación adicional.
- Vista: Es una relación que se hace en referencia a una fila o columna específica.
- Instantáneo: Es todo registro que se puede ver de manera inmediata con solo una referencia.

Tablas temporales

Son aquellas tablas que son eliminadas automáticamente por el motor de base de datos.

Concepto de Dominio

Un dominio describe un conjunto de posibles valores para cierto atributo. Como un dominio restringe los valores del atributo, puede ser considerado como una restricción. Matemáticamente, atribuir un dominio a un atributo significa "todos los valores de este atributo deben de ser elementos del conjunto especificado".

Distintos tipos de dominios son: enteros, cadenas de texto, fecha, no procedurales etc.

El concepto de dominio es el mismo en el modelo E-R y en el modelo relacional. Pero en este modelo tiene mayor importancia, ya que será un dato importante a la hora de dimensionar la relación.

De nuevo estamos ante un concepto muy flexible. Por ejemplo, si definimos un atributo del tipo entero, el dominio más amplio sería, lógicamente, el de los números enteros. Pero este dominio es infinito, y sabemos que los ordenadores no pueden manejar infinitos números enteros. Al definir un atributo de una relación dispondremos de distintas opciones para guardar datos enteros. Si en nuestro caso usamos la variante de "entero pequeño", el dominio estará entre -128 y 127. Pero, además, el atributo corresponderá a una característica concreta de una entidad; si se tratase, por ejemplo, de una calificación sobre 100, el dominio estaría restringido a los valores entre 0 y 100.

Dominio y atributo

- Dominio: conjunto finito de valores homogéneos y atómicos, caracterizados por un nombre y un tipo de datos; además de ciertas restricciones en algunos casos.
- Los dominios se pueden definir por extensión (= España, Costa Rica, Etc.) o por intensión ($Edad > 10 < 15$).
- Atributo: un atributo A, es el papel que juega un dominio D en una relación; se dice que D es el dominio de A y se denota $Dom(A)$; así el atributo Nacionalidad de la tabla Autor, definidos sobre el dominio Nacionalidades, nos indica que dicho dominio tiene el papel de nacionalidad del autor en la tabla Autor.



Concepto de Tupla o Registro

En el contexto de una base de datos relacional, un registro (también llamado fila o tupla) representa un objeto único de datos implícitamente estructurados en una tabla.

En términos simples, una tabla de una base de datos puede imaginarse formada de filas y columnas (campos o atributos).

Cada fila de una tabla representa un conjunto de datos relacionados, y todas las filas de la misma tabla tienen la misma estructura. No puede haber un registro duplicado, los datos deben ser diferentes en al menos uno de los campos.

Un registro es un conjunto de campos que contienen los datos que pertenecen a una misma entidad. Se le asigna automáticamente un número consecutivo (número de registro) que en ocasiones es usado como índice, aunque lo normal y práctico es asignarle a cada registro un campo clave para su búsqueda.

Concepto de Esquemas

El esquema de una base de datos (en inglés, database schema) describe la estructura de una base de datos, en un lenguaje formal soportado por un sistema de gestión de base de datos (DBMS). En una base de datos relacional, el esquema define sus tablas, sus campos en cada tabla y las relaciones entre cada campo y cada tabla.

El esquema es generalmente almacenado en un diccionario de datos. Aunque generalmente el esquema es definido en un lenguaje de base de datos, el término se usa a menudo para referirse a una representación gráfica de la estructura de base de datos.

Podemos decir que un esquema es, la estructura u organización formal de las tablas de una base de datos.

Típicamente, un diseñador de bases de datos crea un esquema de base de datos para ayudar a los programadores cuyo software interactuará con la base. Al proceso de crear un esquema de base de datos se le llama modelado de datos. Al seguir el enfoque de tres esquemas para el diseño de bases de datos, este paso seguiría la creación de un esquema conceptual. Los esquemas conceptuales se enfocan en las necesidades informativas de una organización, más que en la estructura de una base de datos.

Hay dos tipos principales de esquemas de bases de datos:

- Un esquema lógico de base de datos expresa las restricciones lógicas que se aplican a los datos almacenados. Puede definir las restricciones de integridad, las vistas y las tablas.
- Un esquema físico de base de datos dispone cómo se almacenan los datos físicamente en un sistema de almacenamiento en términos de archivos e índices.

En el nivel más básico, un esquema de base de datos indica qué tablas o relaciones



componen la base de datos, así como los campos incluidos en cada tabla. Por lo tanto, los términos diagrama de esquema y diagrama de relaciones de entidades con frecuencia son intercambiables

Algebra Relacional

El álgebra relacional es un conjunto de operaciones que describen paso a paso cómo computar una respuesta sobre las relaciones, tal y como éstas son definidas en el modelo relacional. Denominada de tipo procedimental, a diferencia del Cálculo relacional que es de tipo declarativo.

El "Álgebra Relacional" es un lenguaje, que define una serie de operaciones, que se realizan utilizando "operadores", cada uno de los cuales puede trabajar sobre uno o varios conjuntos de datos produciendo como resultado un nuevo conjunto de datos.

El conjunto de datos resultante de una operación puede a su vez ser utilizado en una nueva operación, en forma anidada, tal como se hace con las operaciones aritméticas. Esta propiedad es conocida como "clausura".

Tuplas

Una tupla se define como una función finita que asocia unívocamente los nombres de los campos de una relación con los valores de una instancia de la misma. En términos simples, es una fila de una tabla relacional.

Operaciones Básicas

Cada operador del álgebra acepta una o dos relaciones y retorna una relación como resultado. σ y Π son operadores unarios, el resto de los operadores son binarios. Las operaciones básicas del álgebra relacional son:

Selección - restricción (σ)

Permite seleccionar un subconjunto de tuplas de una relación (R), todas aquellas que cumplan la(s) condición(es) P , esto es:

$$\sigma_P(R)$$

Ejemplo:

$$\sigma_{\text{Apellido=Gomez}}(\text{Alumnos})$$

Selecciona todas las tuplas que contengan Gómez como apellido en la relación Alumnos.

Una condición puede ser una combinación booleana, donde se pueden usar operadores como: \wedge , \vee , combinándolos con operadores $<$, $>$, \leq , \geq , $=$, \neq .

Opera sobre una o más tablas, no siendo necesario que éstas posean la misma estructura, y devolviendo una nueva tabla cuyo contenido es todas las filas de las tablas indicadas que satisfacen una cierta condición, tal como puede verse en el siguiente ejemplo:

TABLA A	
X	Y
1	23
78	32
67	5
15	320
7	5

SELECCIÓN CON $X < 50$	
X	Y
1	23
15	320
7	5

SELECT * FROM A WHERE X<50

Proyección (Π)

Permite extraer columnas (atributos) de una relación, dando como resultado un subconjunto vertical de atributos de la relación, esto es:

$$\Pi_{A_1, A_2, \dots, A_n}$$

donde A_1, A_2, \dots, A_n son atributos de la relación R .

Ejemplo:

$$\Pi_{Apellido, Semestre, NumeroControl} (Alumnos)$$

Selecciona los atributos Apellido, Semestre y NumeroControl de la relación Alumnos, mostrados como un subconjunto de la relación Alumnos.

Opera sobre una o más tablas, no siendo necesario que éstas posean la misma estructura, y devolviendo una nueva tabla cuyo contenido es todas las filas de las tablas indicadas que satisfacen una cierta condición, tal como sucede con la selección, sólo que la proyección permite indicar cuáles columnas se desea obtener en el resultado, tal como puede verse en el siguiente ejemplo:

TABLA A		
X	Y	Z
1	23	11
78	32	321
67	5	33
15	320	5
7	5	212

PROYECCIÓN DE X y Z CON X<50	
X	Z
1	11
15	5
7	212

SELECT X, Z FROM A WHERE X<50

Producto Cartesiano (\times)

El producto cartesiano de dos relaciones se escribe como:

$$R \times S$$

y entrega una relación, cuyo esquema corresponde a una combinación de todas las tuplas de R con cada una de las tuplas de S, y sus atributos corresponden a los de R seguidos por los de S.

Ejemplo:

$$\textit{Alumnos} \times \textit{Maestros}$$

Muestra una nueva relación, cuyo esquema contiene cada una de las tuplas de la relación Alumnos junto con las tuplas de la relación Maestros, mostrando primero los atributos de la relación Alumnos seguidos por las tuplas de la relación Maestros.

Unión (\cup)

La operación

$$R \cup S$$

retorna el conjunto de tuplas que están en R, o en S, o en ambas. R y S deben ser uniones compatibles.

En otras palabras, opera sobre dos o más tablas, siendo necesario que todas posean la misma estructura, devolviendo una nueva tabla cuyo contenido es la combinación de los contenidos de todas y cada una de las tablas originales, descartando las filas repetidas, tal como puede verse en el siguiente ejemplo:

TABLA A	
CODIGO	LOCALIDAD
1425	Buenos Aires
2000	Rosario
3000	Santa Fe



TABLA B	
CODIGO	LOCALIDAD
2000	Rosario
3000	Santa Fe
2128	Arroyo Seco
2121	Perez

A UNION B	
CODIGO	LOCALIDAD
1425	Buenos Aires
2000	Rosario
3000	Santa Fe
2128	Arroyo Seco
2121	Perez

SELECT * FROM A UNION SELECT * FROM B

Diferencia (-)

La diferencia de dos relaciones, R y S denotada por:

$$R - S$$

entrega todas aquellas tuplas que están en R, pero no en S. R y S deben ser uniones compatibles.

Estas operaciones son fundamentales en el sentido en que

- Todas las demás operaciones pueden ser expresadas como una combinación de éstas
- Ninguna de estas operaciones puede ser omitidas sin que con ello se pierda información.

Intersección (\cap)

La intersección de dos relaciones se puede especificar en función de otros operadores básicos:

$$R \cap S = R - (R - S)$$

La intersección, como en Teoría de conjuntos, corresponde al conjunto de todas las tuplas que están en R y en S, siendo R y S uniones compatibles.

Unión natural (\bowtie) (Natural Join) - Reunión

La operación unión natural en el álgebra relacional es la que permite reconstruir las tablas originales previas al proceso de normalización. Consiste en combinar las proyección, selección y producto cartesiano en una sola operación, donde la condición θ es la igualdad Clave Primaria = Clave Externa (o Foránea), y la proyección elimina la columna duplicada (clave externa).

Expresada en las operaciones básicas, queda

$$R \bowtie S = \Pi_{A_1, A_2 \dots A_n}(\sigma_{\theta}(R \times S))$$

Una reunión theta (θ -Join) de dos relaciones es equivalente a:

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

donde la condición θ es libre.

Si la condición θ es una igualdad se denomina EquiJoin.

Opera sobre dos o más tablas, que poseen estructuras diferentes, y devolviendo una nueva



tabla cuyo contenido es un conjunto de filas con las columnas deseadas provenientes de las diferentes tablas, en el que las filas de las diferentes tablas en juego son relacionadas mediante alguna condición, tal como puede verse en el siguiente ejemplo:

TABLA A	
V	W
1	23
78	32
67	5

TABLA B		
X	Y	Z
5	15	320
78	37	5
1	33	3
78	5	404

REUNIÓN DE A y B TOMANDO W e Y CON X=V y Z>10	
W	Y
32	5

SELECT A.W, B.Y FROM A JOIN B ON (A.V=B.X AND Z>10)

División (/) (Cociente)

Supongamos que tenemos dos relaciones $A(x, y)$ y $B(y)$ donde el dominio de y en A y B , es el mismo.

El operador división A / B retorna todos los valores de x tales que para todo valor y en B existe una tupla $\langle x, y \rangle$ en A .

Opera sobre dos tablas. Si se divide una tabla B por una tabla A , se obtiene una nueva tabla cuyas columnas serán aquellas de la tabla B que no existen en la tabla A , y cuyas filas serán tales que cumplan con estar relacionadas con todas y cada una de las filas de la tabla A . Este caso es más difícil de visualizar, pero se puede entender analizando en detalle el siguiente ejemplo:

TABLA A	
CODIGO	
1425	
2000	
3000	

TABLA B	
CODIGO	INDICE
1425	15%
2000	27%
3000	33%
2128	45%
2121	13%
2000	15%
3000	15%

B DIVIDIDA POR A	
INDICE	15%

La columna INDICE es la única de la tabla B que no existe en la tabla A, y el valor 15% es el único valor de la misma que aparece en filas que se relacionan con todas las filas de la tabla A, es decir, las filas en las que CODIGO toma los valores 1425, 2000 y 3000, que son todos los que aparecen en la tabla A.

Agrupación (G) (Unión)

Permite agrupar conjuntos de valores en función de un campo determinado y hacer operaciones con otros campos.

Conclusiones y recomendaciones

Para especificar una consulta en álgebra relacional, es preciso definir uno o más pasos que sirven para ir construyendo, mediante operaciones de álgebra relacional, una nueva relación que contenga los datos que responden a la consulta a partir de las relaciones almacenadas.

Los lenguajes basados en el álgebra relacional son procedimentales, dado que los pasos que forman la consulta describen un procedimiento.

Las operaciones del álgebra relacional sirven para hacer consultas a una base de datos. Es preciso conocer estas operaciones porque nos permiten saber qué servicios de consulta debe proporcionar un lenguaje relacional. Además, constituye la base para el estudio del tratamiento de las consultas que efectúan los SGBD internamente (especialmente en lo que respecta a la optimización de consultas).

Referencias bibliográficas

Esta lectura constituyen un apoyo en el proceso de aprendizaje, así mismo para apoyarse aun mas en el proceso, se debe utilizar los libros,

- Bertone, N. (2017). *Introducción a las bases de datos: fundamentos y diseño*. Pearson Educación
- Silberschatz, A. & Sudarshan, S. (2018). *Fundamentos de bases de datos* (6a. ed.). McGraw-Hill Interamericana.
- Bernal, Nieto. (2017). *Diseño de base de datos*. Universidad del Norte
- Pulido Romero, E. Escobar Domínguez, Ó. y Núñez Pérez, J. Á. (2019). *Base de datos*. Grupo Editorial Patria.

Tecnologías-Información. (2020). Tipos de Relaciones. Sitio de descarga de la información: <https://www.tecnologias-informacion.com/tiposrelaciones.html>

Wikipedia. (2016). Esquema de una base de datos. Sitio de descarga de la información: https://es.wikipedia.org/wiki/Esquema_de_una_base_de_datos

Bases de datos distribuidasTICS. Sitio de descarga de la información: <https://sites.google.com/site/basededatosdistribuidastics/algebra-relacional>

DorvinEduardo. (2016). Algebra Relacional. Sitio de descarga de la información: <https://es.slideshare.net/DorvinEduardo/algebra-relacional-62265241#:~:text=Conclusiones%20Las%20operaciones%20del%20%20C3%A1lgebra,debe%20proporcionar%20un%20lenguaje%20relacional.>



www.usanmarcos.ac.cr

San José, Costa Rica