Elizabeth León Guzmán, Ph.D.
Jonatan Gómez Perdomo, Ph. D.
Arles Rodríguez, Ph.D.
Camilo Cubides, Ph.D. (c)
Carlos Andres Sierra, M.Sc.

Bases de Datos Relacionales

Consultas y Operadores de Algebra Relacional

Para cualquier aclaración o información adicional puede escribir al correo soportemtic22_bog@unal.edu.co
o radicar solicitud en la mesa de ayuda https://educacioncontinuavirtual.unal.edu.co/soporte

Research Group on Data Mining
Grupo de Investigación en Minería de Datos – (Midas)
Research Group on Artificial Life
Grupo de Investigación en Vida Artificial – (Alife)
Computer and System Department
Engineering School
Universidad Nacional de Colombia

Agenda

Introducción AR I

Un modelo de datos relacional
necesita un conjunto de **operaciones** para manejar los datos

El conjunto básico de operaciones del modelo relacional constituyen el **álgebra relacional**.

Introducción AR II

Las operaciones permiten especificar una **solicitud de recuperación** de datos de una relación. Se pueden aplicar operaciones sobre una o más relaciones dando resultado a una nueva relación que a su vez se puede manipular también usando operaciones de la misma álgebra.

Agenda

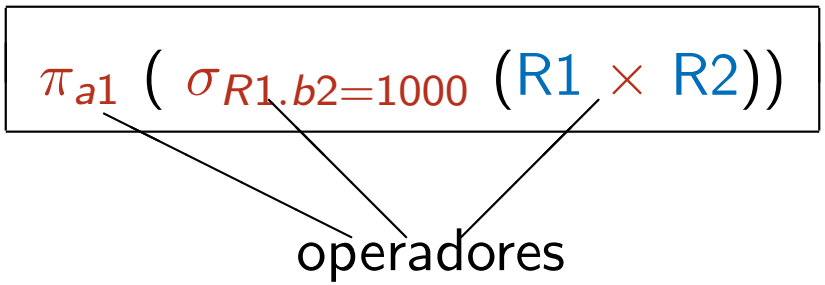
Operadores Básicos de Álgebra Relacional

Los siguientes son los operadores de algebra relacional básicos para realizar consultas:

- Proyección (π)
- Selección (σ)
- Renombrar (ρ)
- Eliminar Duplicados (δ)
- Producto cartesiano ($\times$)
- *Join* o enlace o reunión ($\bowtie$)

Expresión de Álgebra Relacional

Una secuencia de operaciones de álgebra relacional forma una expresión de álgebra relacional, cuyo resultado también representa una relación que representa el resultado una consulta de base de datos (solicitud de recuperación de datos).



Ejemplos

Las siguientes son expresiones en Algebra Relacional para realizar consultas:

- $\pi_{a1,b1}$ (R1 $\bowtie$ R2)
- $\pi_{a1,b1,c1}$ (R1 $\bowtie$ R2 $\bowtie$ R3)
- $\pi_{R1.a1,R2.b2}$ ($\sigma_{R2.b2>15 R1.a1=''Ana''}$ (R1 $\times$ R2))

Proyección I

- Selecciona el valor de ciertos atributos de todas las tuplas de una relación

$$\pi_{a1,a2,a3,...,an}(R) = t[a1, a2, a3, ..., an] : t \in R$$

atributos — relación

- Selecciona columnas completas

Ejemplo

LIBRO

libId	libNombre	libPub
1001	Cien años de soledad	1967
1002	La vorágine	1924
1003	María	1867
1004	Cóndores no se entierran todos los días	1971

$$\pi_{libNombre}(LIBRO) = \left\{ \begin{array}{c} Cien\ años\ de\ soledad \\ La\ vorágine \\ María \\ Cóndores\ no\ se\ entierran\ todos\ los\ días \end{array} \right\}$$

$$\pi_{libNombre,libPub}(LIBRO) = ?$$

Proyección II

Ejemplo (continuación)

LIBRO

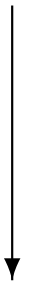
libId	libNombre	libPub
1001	Cien años de soledad	1967
1002	La vorágine	1924
1003	María	1867
1004	Cóndores no se entierran todos los días	1971

$$\pi_{libNombre, libPub}(LIBRO) = \left\{ \begin{array}{c} \textit{Cien años de soledad}, 1967 \\ \textit{La vorágine}, 1924 \\ \textit{María}, 1867 \\ \textit{Cóndores no se entierran todos los días}, 1971 \end{array} \right\}$$

Proyección en SQL I

Sintaxis

$$\pi_{a1,a2,a3,\ldots,an}(R)$$



```
SELECT a1,a2,a3,...,an FROM R
```

Proyección en SQL II

Ejemplos

Ejemplos

```sql
-- Selecciona el título de los libros
SELECT libNombre FROM libro;

-- Selecciona el título y año de publicación de los libros
SELECT libNombre, libPub FROM libro;

-- Selecciona los nombres y apellidos de los autores
SELECT autNombre, autApellido FROM autor;

-- Selecciona todos los atributos de los autores
SELECT * FROM autor;
```

Para los ejemplos cargar el script de creación e inserción de datos de la Librería libreria.sql

Selección I

- Selecciona ciertas tuplas que cumplen una condición

$$\sigma_{condición}(R) = \{t \in R : \text{condición es verdadera } \}$$

- Selecciona filas completas

Ejemplo

LIBRO

libId	libNombre	libPub
1001	Cien años de soledad	1967
1002	La vorágine	1924
1003	María	1867
1004	Cóndores no se entierran todos los días	1971

$$\sigma_{libId=1003}(LIBRO) = \{\ 1003, María, 1867\ \}$$

$$\sigma_{libPub>1960}(LIBRO) = ?$$

Selección en SQL I

Sintaxis

$$\sigma_{condición}(R)$$

```
SELECT * FROM R WHERE condición
```

Selección en SQL II

Ejemplos

Ejemplos

```
-- CONDICIONES NUMERICAS

-- Selecciona todos los atributos del libro con identificador 1003
SELECT * FROM libro WHERE libId=1003;

-- Selecciona todos los atributos de los libros que fueron publicados despues de 1960
SELECT * FROM libro WHERE libPub > 1960;
```

```
-- CONDICIONES CON CADENAS DE CARACTERES

-- El operador más usado es LIKE
-- Selecciona todos los atributos de los libros que continen la cadena "los".
SELECT * FROM libro WHERE libNombre LIKE "%los%";

-- Selecciona todos los atributos de los libros que inicien con la letra "C" y no tienen
    la cadena "los".
SELECT * FROM libro WHERE libNombre LIKE "C%" and libNombre NOT LIKE "%los%" ;
```

Selección en SQL III

Ejemplos

En búsqueda de cadena de caracteres la sentencia más usada en MySQL es LIKE:

- Consulta Exacta
  LIKE 'La Vorágine'
- Consulta omitiendo una parte
  LIKE La Vor
- Consulta omitiendo un caracter
  LIKE La V_ra__ne'

Ejemplos

```
SELECT * FROM libro WHERE libNombre LIKE "%sol%";

SELECT * FROM libro WHERE libNombre LIKE "%a_os%";
```

Selección en SQL IV

Ejemplos

En búsqueda de cadena de caracteres se pueden usar funciones que se aplican a este tipo de dato:

- LENGHT('hola') → 4
- CONCAT('hola',' todos') → hola todos
- LOWER('HoLa') → hola
- UPPER('HoLa') → HOLA
- RTRIM('HoLa ') → HoLa //quitar espacios

Ejemplos

```
SELECT * FROM libro WHERE LENGHT(libNombre) > 8;

SELECT * FROM libro WHERE LOWER(libNombre) LIKE "maría";
```

Composición (π y σ)

En una consulta se pueden combinar los operadores de selección (σ) y proyección (π). Se seleccionan las tuplas que cumplen una condición y luego se proyectan solo las atributos deseados.

Ejemplo

En este ejemplo se seleccionan las tuplas que cumplen la condición $libPub > 1960$ y luego se proyectan las columnas de nombre y año de publicación.

LIBRO

libId	libNombre	libPub
1001	Cien años de soledad	1967
1002	La vorágine	1924
1003	María	1867
1004	Cóndores no se entierran todos los días	1971

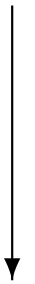
$$\pi_{libNombre,libPub}(\ \sigma_{libPub>1960}(LIBRO)\ ) = \left\{ \begin{array}{c} Cien\ años\ de\ soledad, 1967 \\ Cóndores\ no\ se\ entierran\ todos\ los\ días, 1971 \end{array} \right\}$$

MIDAS

Hechos QUE CONECTAN

Composición (π y σ) en SQL I

$$\pi_{a1,a2,\ldots,an}(\sigma_{condición}(R))$$



```
SELECT a1,a2,...,an FROM R WHERE condición
```

MIDAS

Hechos QUE CONECTAN

UNIVERSIDAD NACIONAL DE COLOMBIA un

Composición (π y σ) en SQL II

Ejemplos

Ejemplos

```
-- Selecciona el nombre del libro con identificador 1003
SELECT libNombre FROM libro WHERE libId=1003;

-- Selecciona el identificador, nombre y año de publicación de los libros que fueron
    publicados despues de 1960
SELECT libId, libNombre, libPub FROM libro WHERE libPub > 1960;

-- Selecciona el nombre de los libros que continen la cadena "los".
SELECT libNombre FROM libro WHERE libNombre LIKE "%los%";

-- Selecciona el nombre y años de publicación de los libros que continen la cadena "C" y
    no tienen la cadena "los".
SELECT libNombre, libPub FROM libro WHERE libNombre LIKE "C%" and libNombre NOT LIKE "%
    los%";
```

Otras consultas

MySQL

Las columnas numéricas se pueden usar para ordenar usando ORDER BY. Adicionalmente se puede usar LIMIT para restringir el máximo de elementos a retornar.

Ejemplos

```
-- Selecciona los tres primeros nombres de los libros ordenados por año de publicación
    descendentemente
SELECT libNombre FROM libro ORDER BY libPub DESC LIMIT 3;
```

Consultas con agregaciones I

MySQL

Existe la posibilidad de usar operadores de agregación, que usan los valores de una columna completa. Las funciones de agregación más usadas son: AVG: promedio, MIN y MAX : mínimo y máximo SUM: suma Count: contar

Ejemplos

```
-- Selecciona el año más reciente de publicación
SELECT MAX(libPub) FROM libro;

-- Selecciona el precio promedio de los libros
SELECT AVG(libPrecio) FROM libro;

-- Cuenta los libros
SELECT count(libId) from libro;

-- Selecciona el nombre del libro más costoso

SELECT libNombre FROM libro WHERE libPrecio = (SELECT MAX(libPrecio) FROM libro)
```

Consultas con agregaciones II

MySQL

Cuando se realizan consultas con agregaciones se pueden agrupar por los valores de alguna de las columnas. De esa manera se tienen resultados de la agregación por cada valor.

Ejemplos

```
-- Cuenta las ventas por cada libId
SELECT libId, COUNT(vtaId) FROM venta GROUP BY (libId);

-- Calcula el promedio de los libros por cada autor
SELECT autId, AVG(libPrecio) FROM libro GROUP BY (autId)
```

Si se desea tener una condición después de realizar una agrupación se usa la sentencia HAVING (funciona igual que un WHERE).

```
-- Cuenta las ventas por cada libId y selecciona los que tienen ventas mayores a 3
SELECT libId, COUNT(vtaId) FROM venta GROUP BY (libId) HAVING COUNT(vtaId) > 3;
```

Renombrar (ρ)

Se puede renombrar atributos y relaciones (tablas) por medio del operador ρ en el momento de realizar consultas. Sin embargo, este renombramiento se hace en memoria, es decir los nombres de las tablas y de los atributos se mantienen (no alteran el esquema de la tabla original).

Ejemplo

Renombra el atributo IdNombre de la relación LIBRO por titulo, y luego lo proyecta.

$$\pi_{titulo}(\ \rho_{titulo \leftarrow IdNombre}(LIBRO)\ )$$

Renombrar (ρ) en SQL I

$$\rho_{nuevoNombre \leftarrow anteriorNombre}(R)$$

```
-- Renombra el atributo a1 como b1
   SELECT a1 AS b1 FROM R

-- Renombra la relación (tabla) R como R2
   SELECT R2.a1 FROM R AS R2
```

MIDAS

Hechos QUE CONECTAN

UNIVERSIDAD NACIONAL DE COLOMBIA un

Renombrar (ρ) en SQL II

Ejemplos

Ejemplos

```
-- Selecciona el nombre del libro con identificador 1003 y lo renombra por título
SELECT libNombre AS titulo FROM libro WHERE libId=1003;

-- Selecciona el identificador y el nombre de los libros que fueron publicados despues de
    1960, y los renombra.
SELECT libId AS "Código del libro", libNombre AS "Título" FROM libro WHERE libPub > 1960;

-- Renombra el nombre de la tabla LIBRO por LIBRO2 y luego proyecta los nombres
-- de los libros
SELECT libro2.libNombre AS "Título" FROM libro AS libro2;

-- Selecciona nombres y apellidos del autor, los concatena y los proyecta con otro nombre
SELECT concat(autNombre, concat(" ",autApellido)) AS "Nombre Autor" FROM autor WHERE
    autPais='Colombia';
```

Producto Cartesiano ($\times$) I

Producto cartesiano de dos conjuntos A y B:

$$A \times B = \{(a, b) : a \in A \wedge b \in B\}$$

Ejemplo

$A = \{s, t\}$
$B = \{u, v, w\}$
$A \times B = \{s, t\} \times \{u, v, w\}$
$A \times B = \{(s, u), (s, v), (s, w), (t, u), (t, v), (t, w)\}$

La cardinalidad es:
$|A \times B| = |A| * |B| = 2*3 = 6$

Producto Cartesiano ($\times$) II

Operador que se aplica a dos relaciones devolviendo el producto cartesiano de estas. Asume que cada una de las relaciones son conjuntos y cada una de las tuplas son los elementos del conjunto.

Ejemplo

LIBRO

libId	libNombre	libPub	autId
1001	Cien años de soledad	1967	1
1002	La vorágine	1924	3
1003	María	1867	4
1004	Cóndores no se entierran todos los días	1971	2

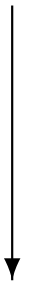
AUTOR

autId	autNombre
1	Gabriel García Márquez
2	Gustavo Alvarez Gardeazábal
3	José Eustasio Rivera
4	Jorge Isaacs

$$LIBRO \times AUTOR = \left\{ \begin{array}{c} 1001, \textit{Cien años de soledad}, 1967, 1, 1, \textit{Gabriel García Márquez} \\ 1001, \textit{Cien años de soledad}, 1967, 1, 2, \textit{Gustavo Alvarez Gardeazábal} \\ 1001, \textit{Cien años de soledad}, 1967, 1, 3, \textit{José Eustasio Rivera} \\ 1001, \textit{Cien años de soledad}, 1967, 1, 4, \textit{Jorge Isaacs} \\ 1002, \textit{La vorágine}, 1924, 3, 1, \textit{Gabriel García Márquez} \\ 1002, \textit{La vorágine}, 1924, 3, 2, \textit{Gustavo Alvarez Gardeazábal} \\ 1002, \textit{La vorágine}, 1924, 3, 3, \textit{José Eustasio Rivera} \\ 1002, \textit{La vorágine}, 1924, 3, 4, \textit{Jorge Isaacs} \\ 1003, \textit{María}, 1867, 4, 1, \textit{Gabriel García Márquez} \\ 1003, \textit{María}, 1867, 4, 2, \textit{Gustavo Alvarez Gardeazábal} \\ 1003, \textit{María}, 1867, 4, 3, \textit{José Eustasio Rivera} \\ 1003, \textit{María}, 1867, 4, 4, \textit{Jorge Isaacs} \\ 1004, \textit{Cóndores no se entierran todos los días}, 1971, 2, 1, \textit{Gabriel García Márquez} \\ 1004, \textit{Cóndores no se entierran todos los días}, 1971, 2, 2, \textit{Gustavo Alvarez Gardeazábal} \\ 1004, \textit{Cóndores no se entierran todos los días}, 1971, 2, 3, \textit{José Eustasio Rivera} \\ 1004, \textit{Cóndores no se entierran todos los días}, 1971, 2, 4, \textit{Jorge Isaacs} \end{array} \right\}$$

Producto Cartesiano ($\times$) en SQL I

$$R1 \times R2$$



```
SELECT * FROM R1,R2;
```

Seleccion combinaciones correctas

del Producto Cartesiano

Selección cuando el atributo en común es igual (llave foránea o externa).

$$\sigma_{R1.a_k=R2.a_k}(\ R1 \times R2\ )$$

Ejemplo

LIBRO

libId	libNombre	libPub	autId
1001	Cien años de soledad	1967	1
1002	La vorágine	1924	3
1003	María	1867	4
1004	Cóndores no se entierran todos los días	1971	2

AUTOR

autId	autNombre
1	Gabriel García Márquez
2	Gustavo Alvarez Gardeazábal
3	José Eustasio Rivera
4	Jorge Isaacs

$\sigma_{LIBRO.autId=AUTOR.autId}(\ LIBRO \times AUTOR\ ) =$

$$\left\{ \begin{array}{c} 1001, \textit{Cien años de soledad}, 1967, 1, 1, \textit{Gabriel García Márquez} \\ 1002, \textit{La vorágine}, 1924, 3, 3, \textit{José Eustasio Rivera} \\ 1003, \textit{María}, 1867, 4, 4, \textit{Jorge Isaacs} \\ 1004, \textit{Cóndores no se entierran todos los días}, 1971, 2, 2, \textit{Gustavo Alvarez Gardeazábal} \end{array} \right\}$$

MIDAS

Hechos QUE CONECTAN

UNIVERSIDAD NACIONAL DE COLOMBIA un

Seleccion combinaciones correctas en SQL I

$$\sigma_{R1.a_k = R2.a_k}(R1 \times R2)$$

```
SELECT * FROM R1,R2 WHERE R1.ak = R2.ak;
```

Seleccion combinaciones correctas en SQL II

Ejemplos

Ejemplos

```
SELECT * FROM libro,autor;

SELECT libNombre, autNombre, autApellido FROM libro,autor WHERE libro.autId=autor.autId;

SELECT vtaFecha,libNombre FROM libro,venta WHERE libro.libId = venta.libId;

SELECT libNombre, ediNombre FROM libro,editorial WHERE libro.ediId = editorial.ediId;
```

JOIN ($\bowtie$)

El operador join $\bowtie$ permite combinar registros de una o más relaciones.

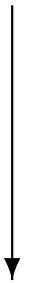
$$\sigma_{R1.a_k=R2.a_k}(\ R1 \times R2\ )$$

$$R1 \bowtie_k R2$$

JOIN ($\bowtie$) en SQL I

MySQL

$$R1 \bowtie_k R2$$



```
SELECT * FROM R1 JOIN R2 USING K;
```

Si el atributo por el que se quiere combinar no se llama igual en las dos relaciones, se usa:

```
SELECT * FROM R1 JOIN R2 ON (R1.a = R2.b);
```

MIDAS

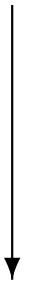
Hechos QUE CONECTAN

UNIVERSIDAD NACIONAL DE COLOMBIA un

JOIN Natural ($\bowtie$)

Combinar según los atributos que tengan el mismo nombre en las dos tablas(todos los atributos que tengan el mismo nombre). Omitir el subíndice del $\bowtie$.

$$R1 \bowtie R2$$



```
SELECT * FROM R1 NATURAL JOIN R2;
```

JOIN en SQL II

Ejemplos - MySQL

Ejemplos

```
SELECT * FROM libro JOIN autor using(autId);

SELECT libNombre, autNombre, autApellido FROM libro JOIN autor using(autId);

SELECT libNombre, autNombre, autApellido FROM libro NATURAL JOIN autor;

SELECT vtaFecha,libNombre, vtaCantidad FROM libro NATURAL JOIN venta;

SELECT libNombre, ediNombre FROM libro JOIN editorial USING (ediId);

-- JOIN de tres relaciones. nombre del libro, de la editorial y del autor de todos los
    libros.
SELECT libNombre, ediNombre, autNombre, autApellido  FROM libro NATURAL JOIN editorial
    NATURAL JOIN autor;
```

Otras Consultas

Ejemplos - MySQL

Ejemplos

```
-- Paises de la nacionalidad de los autores
SELECT DISTINCT autPais AS "Paises" FROM autor;

-- Total de libros vendidos en el año 2021
SELECT sum(vtaCantidad) AS "Cantidad" FROM venta WHERE year(vtaFecha) > 2020;

-- Total de libros que tiene cada autor
SELECT concat(autNombre, concat(" ",autApellido)) AS "Nombre Autor", count(libId) AS "Nú
    mero" FROM libro NATURAL JOIN autor GROUP BY autId;

-- Total de unidades vendidas por libro. Listarla en orden descendente por la cantidad
    vendida.
SELECT libNombre, sum(vtaCantidad) AS "Total" FROM libro NATURAL JOIN venta GROUP BY
    libId ORDER BY Total DESC;
```

Ejemplo Completo SQL I

Museo - MySQL

Ejemplo

Cargar el script museo.sql y verificar con SELECT que todas las tablas esten cargadas.

```
-- Modificar la columna de costo de obra dejando por defecto el valor 500
ALTER TABLE obra ALTER obr_costo SET DEFAULT 500;

-- Insertar una nueva obra de Da Vinci sin costo (verificar que tenga el valor por
    defecto después de la inserción)
INSERT INTO obra (obr_id, obr_nombre, obr_tipo, exp_id) values (115, 'Salvator Mundi', '
    pintura', 1003);

-- Incrementar el precio de las esculturas en 10%
UPDATE obra SET obr_costo=obr_costo*1.1 WHERE obr_tipo = 'escultura';

--Consultar los nombres de las obras con su correspondiente tipo
SELECT obr_nombre, obr_tipo FROM obra;

--Consultar los diferentes tipos de obras sin repetidos y renombrando el atributo por
    TIPO
SELECT DISTINCT obr_tipo AS TIPO FROM obra;
```

Ejemplo Completo SQL II

Museo - MySQL

Ejemplo (continuación)

```
--Explique lo que realiza la siguiente expresión:
SELECT obr_nombre FROM obra WHERE obr_nombre LIKE '%ma%' AND obr_nombre NOT LIKE '%cena%'
    ;

--Listar todas los nombres de las obras en mayúscula
SELECT UPPER(obr_nombre) FROM obra;

-- Selecciona las obras cuyo costo es mayor al promedio de los costos
SELECT obr_costo FROM obra WHERE obr_costo >(SELECT AVG(obr_costo) FROM obra);

--Nombre de las 5 obras más costosas
SELECT obr_nombre FROM obra ORDER BY obr_costo DESC LIMIT 5

--Contar el número de obras por cada uno de los tipos de obras
SELECT obr_tipo, count(obr_id) FROM obra GROUP BY obr_tipo;

-- Listar los nombres de las obras y  exposiciones de las obras tipo pintura
SELECT obr_nombre,exp_nombre FROM obra JOIN exposicion USING (exp_id) WHERE obr_tipo = '
    pintura';
```

Agenda

Problemas I

Problema

1. Usando el script museo.sql del ejercicio anterior, realizar las siguientes consultas
   - 1. Nombre de las obras que tienen un costo menor a 1/3 del promedio.
   - 2. Listar los nombres de las obras ordenadas alfabéticamente
   - 3. Sumar el costo de las obras por tipo de obra
   - 4. Contar las obras que tiene cada una de las exposiciones
   - 5. Listar los nombres, tipo y costo de las obras de la exposición “Da Vinci”
   - 6. Proyectar nombre del museo, nombre de la obra de las presentaciones de las obras de “Da Vinci”

Problemas II

Problema

1. Usar la base de datos Veterinaria creada en los problemas de la sesión anterior, para realizar las siguientes consultas:
   - 1. Nombre y descripción de las mascotas que son gatos.
   - 2. Tipo y nombre de las mascotas que son juguetones.
   - 3. Nombres de los productos que contienen carne.
   - 4. Nombre de las mascotas que han tenido servicio de peluqueria.
   - 5. Nombre de los dueños y las mascotas a los que se les ha vendido productos.
   - 6. Nombre de los empleados que han realizado servicio de Guardería.

MIDAS

Hechos QUE CONECTAN

UNIVERSIDAD NACIONAL DE COLOMBIA un

Problemas III

Problema (continuación)

- 7. Contar cuantas mascotas hay por cada uno de los tipos de mascotas. Proyectar el tipo y el conteo.
- 8. Calcular el promedio de los precios de los productos para los perros.
- 9. Calcular el promedio de las ventas realizadas a los gatos
- 10. Calcular cuantos servicios ha tenido la mascota llamada "Max".
- 11. Calcular el promedio de las ventas realizadas por cada uno de los tipos de servicios.
- 12. Calcular cuantas mascotas tienen cada uno de las personas que tienen mascotas.

Referencias

1. Gillenson, M. Administración de Bases de Datos. LIMUSA WILEY
2. Coronel, Morris, Rob. Bases de Datos: Diseño, Implementación y Administración. CENGAGE Learning
3. Elmasri, R.; Navathe, S.B. Fundamentos de Sistemas de Bases de Datos. 3ª ed. Addison-Wesley
4. Silberschatz, A;Korth, H; Sudarshan, S. Fundamentos de Bases de Datos. 3ª edición. Madrid: McGraw-Hill.
5. León, E. Notas curso Bases de Datos. Universidad Nacional de Colombia