

FUNDAMENTOS DE BASES DE DATOS

Modelo Relacional

Contenido

Modelo
Relacional

→ Estructura de bases de datos relacionales

→ Algebra relacional

→ Algebra relacional extendido

→ Modificaciones de la base de datos

→ Vistas

Ejemplo de un conjunto de relaciones

Estudiantes		
id	nombre	dirección
1111111	Pedro Pérez	Calle 1 1-1
2222222	Juan Sánchez	Calle 2 2-2
3333333	Maria Gómez	
4444444	Rosa Martínez	Calle 4 4-4

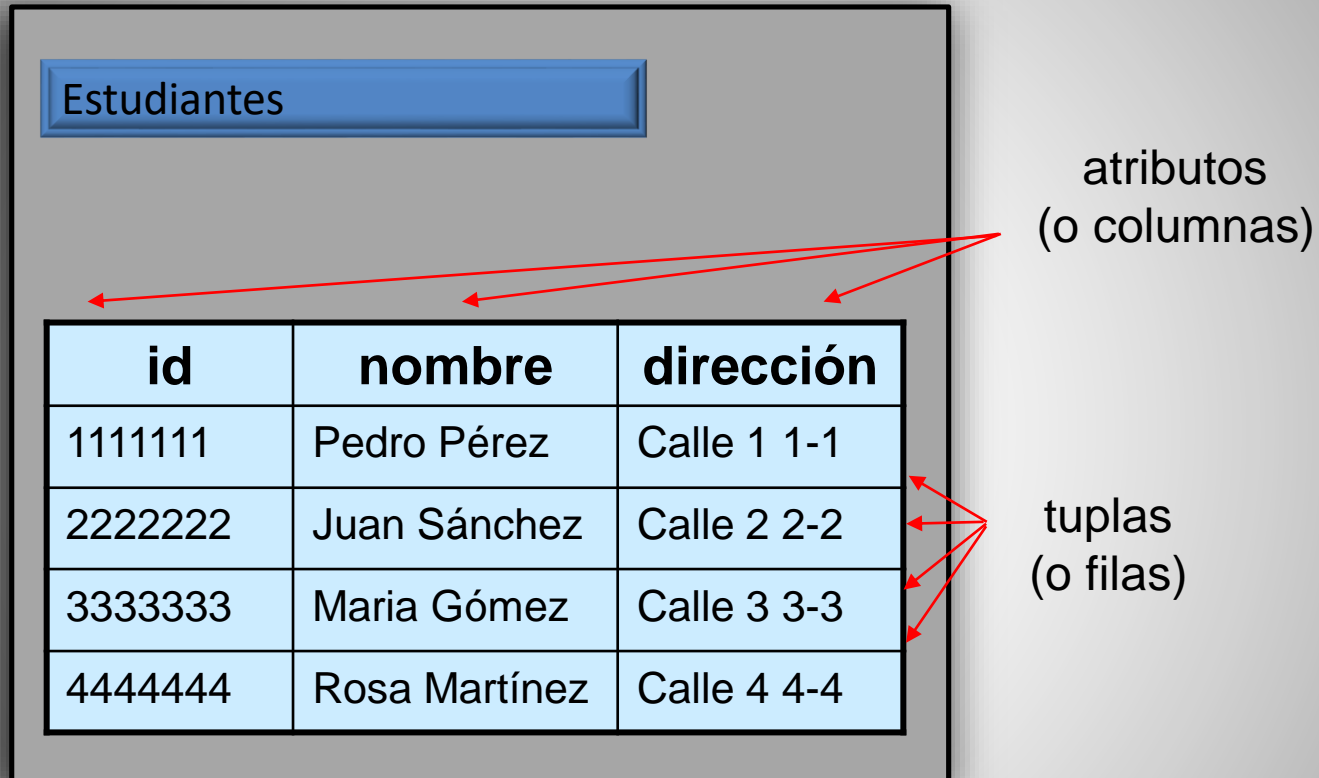
Cada atributo
Tiene un nombre

NULL es parte
de todos los
dominios

El conjunto
de valores
es **Dominio**

Cada valor
de atributo
es **Atómico**

Instancia de las Relaciones



Un elemento t de r es una tupla, representado por una fila en la tabla

Base de Datos

Estudiante

id	nombre	dirección
1111111	Pedro Pérez	Calle 1 1-1
2222222	Juan Sánchez	Calle 2 2-2
3333333	Maria Gómez	Calle 3 3-3
4444444	Rosa Martínez	Calle 4 4-4

Asignatura

id	nombre	Int-hora
IS-1	Bases de Datos	4
IS-2	Ing de Software	3
IS-3	Programación	3
IS-4	Redes	4

RELACIÓN

Inscribe

Id_e	Id_a	horario
1111111	IS-2	Lun-10
1111111	IS-1	Mar-13
4444444	IS-3	Vie - 18
2222222	IS-2	Lun - 10

Algebra Relacional

El álgebra relacional es un lenguaje procedimental que consta de 6 operadores básicos:

Selección

Proyección

Unión

Diferencia de conjunto

Producto cartesiano

Renombramiento

Los operadores toman una o dos relaciones como entrada, y su resultado es una nueva relación

Selección

Notación: $\sigma_p(r)$

p es llamado predicado de la selección

Definido como:

$$\sigma_p(\mathbf{r}) = \{t \mid t \in r \textbf{ and } p(t)\}$$

Donde **p** es una formula consistente con el cálculo proposicional de terminos conectados por : \wedge (**y**), \vee (**o**), \neg (**no**)

Cada termino es uno de:

<atributo> **op** <atributo> o <constante>

donde **op** es uno de: $=$, \neq , $>$, \geq , $<$, \leq

Ejemplo de selección:

$\sigma_{\text{nombre}=\text{"Pedro Pérez"}}(\text{Estudiante})$

Selección

Ejemplo de selección:

$\sigma_{\text{nombre}=\text{"Pedro Pérez"}}(\text{Estudiante})$

Equivalente en SQL:

select * ***from*** Estudiantes ***where*** nombre = 'Pedro Pérez'

$\sigma_{\text{nombre}=\text{"Pedro Pérez"}}(\text{Estudiante})$

Algebra Relacional

Selección

Relación R

A	B	C	D
a	a	1	7
a	b	5	7
b	b	12	3
b	b	23	10

$\sigma_{A=B \wedge D > 5} (R)$

A	B	C	D
a	a	1	7
b	b	23	10

Proyección

Notación: $\Pi_{A1, A2, \dots, Ak}(r)$

donde $A1, A2$ son nombres de atributos y r es el nombre de la relación.

El resultado es definido como una relación de k columnas obtenidas de borrar las columnas que no están listadas

Las filas duplicadas son removidas del resultado

Ej. Para mostrar solo los atributos de id y nombre de Estudiante

$\Pi_{id, nombre}(Estudiante)$

Proyección

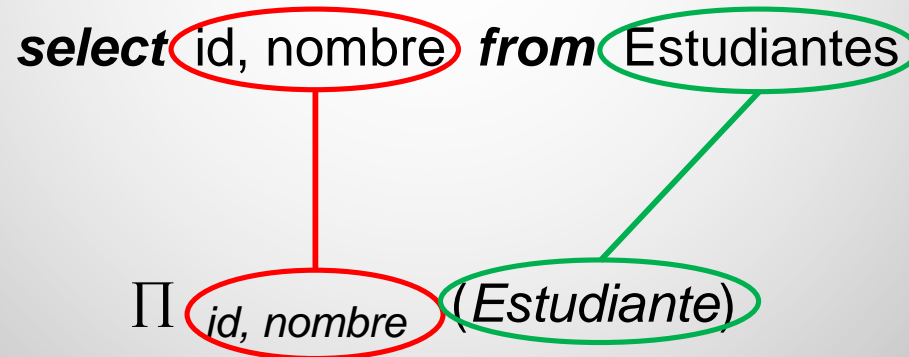
Ej. Para mostrar solo los atributos de id y nombre de Estudiante

$\Pi_{id, nombre} (Estudiante)$

Equivalente en SQL:

select id, nombre **from** Estudiantes

$\Pi_{id, nombre} (Estudiante)$



Algebra Relacional

Proyección

Relación R

A	B	C	D
a	a	1	7
a	b	5	7
b	b	12	3
b	b	23	10

$\Pi_{A,D}(R)$

A	D
a	7
a	7
b	3
b	10



A	D
a	7
b	3
b	10

Unión

Notación: $r \cup s$

Definido como:

$$r \cup s = \{t \mid t \in r \text{ or } t \in s\}$$

Para que $r \cup s$ sea válida:

- r, s deben ser de la misma *aridad* (mismo número de atributos)
- Los dominios de las tributos deben ser compatibles
- r y s pueden ser relaciones temporales que sean resultado de expresiones del álgebra relacional.

Ej.

$$\Pi_{\text{id, nombre}} (\text{Estudiante}) \cup \Pi_{\text{id, nombre}} (\text{Profesor})$$

Algebra Relacional

Unión

Relaciones R y S

R

A	D
a	1
a	2
b	1

S

A	D
a	2
b	3

$R \cup S$

A	D
a	1
a	2
b	1
b	3

Unión

Ej. Para unir los identificadores y nombres de Estudiantes y Profesores

$$\Pi_{id, nombre} (Estudiante) \cup \Pi_{id, nombre} (Profesores)$$

Equivalente en SQL:

select id,nombre **from** Estudiantes

union

select id,nombre **from** Profesores

$$\Pi_{id, nombre} (Estudiante) \cup \Pi_{id, nombre} (Profesores)$$

Diferencia de conjunto

Notación: $r - s$

Definida como:

$$r - s = \{t \mid t \in r \textbf{ and } t \notin s\}$$

Se debe asegurar que la diferencia de conjuntos se realice entre relaciones compatibles:

- r y s deben ser de la misma aridad
- los dominios de los atributos de r y s sean compatibles

Algebra Relacional

Diferencia de conjunto

Relaciones **R** y **S**

R

A	D
a	1
a	2
b	1

S

A	D
a	2
b	3

R - S

A	D
a	1
b	1

Diferencia de conjunto

Ej. Códigos de los estudiantes que no inscribieron asignaturas

$$\Pi_{cod_e}(Estudiantes) - \Pi_{cod_e}(Inscribe)$$

Equivalente en SQL:

```
select cod_e from Estudiantes  
except  
select cod_e from Inscribe
```

$$\Pi_{cod_e}(Estudiante) - \Pi_{cod_e}(Inscribe)$$

Producto cartesiano

Notación: $r \times s$

Definido como:

$$r \times s = \{t \ q \mid t \in r \textbf{ and } q \in s\}$$

Asume que los atributos de $r(R)$ y $s(S)$ son disyuntos.
(Esto esto, $R \cap S = \emptyset$).

Si los atributos de $r(R)$ y $s(S)$ no son disyuntos,
entonces se debe usar renombramiento.

Algebra Relacional

Producto cartesiano

Relaciones **R** y **S**

R

A	B
a	1
b	2

S

C	D
c	1
d	4

R X S

A	B	C	D
a	1	c	1
a	1	d	4
b	2	c	1
b	2	d	4

Producto cartesiano

Ej. Producto cartesiano entre Estudiantes e Inscribe
Estudiantes X Inscribe

Equivalente en SQL:

select * from Estudiantes cross join Inscribe

Estudiantes X Inscribe

select * from Estudiantes, Inscribe

Renombramiento

Nos permite nombrar, y por lo tanto referirse al resultado de relaciones de expresiones del álgebra relacional, también permite referirse a una relación por más de un nombre.

Ejemplo: $\rho_X(E)$

Devuelve la expresión E bajo el nombre X

Si la expresión del álgebra relacional E tiene aridad n , entonces

$$\rho_{X(A_1, A_2, \dots, A_n)}(E)$$

devuelve el resultado de la expresión E bajo el nombre X , y con los atributos renombrados a A_1, A_2, \dots, A_n .

Renombramiento

Ej. Renombramiento de las relaciones Estudiantes e Inscribe

$$\sigma_{E.cod_e=l.cod_e} (\rho_E(Estudiantes) \times \rho_l(Inscribe))$$

Equivalente en SQL:

select * from Estudiantes ***as E*** ***cross join*** Inscribe ***as l***
where E.cod_e=l.cod_e

$$\sigma_{E.cod_e=l.cod_e} (\rho_E(Estudiantes) \times \rho_l(Inscribe))$$

select * from Estudiantes ***E*** , Inscribe ***l***
where E.cod_e=l.cod_e

Algebra Relacional

Composición de operaciones

Se pueden construir expresiones usando múltiples operaciones

Ejemplo:

$$\sigma_{D=1}(R \times S)$$

R

A	B
a	1
b	2

S

C	D
c	1
d	4

A	B	C	D
a	1	c	1
b	2	c	1

Álgebra Relacional

Con base a los operadores básicos, se pueden definir otros operadores adicionales:

Intersección

Reunión natural

División

Asignación

Estas operaciones no adicionan poder al álgebra relacional, pero ayudan a simplificar las consultas

Intersección

Notación: $r \cap s$

Definida como:

$$r \cap s = \{ t \mid t \in r \textbf{ and } t \in s \}$$

Asume:

- r, s tienen la misma aridad
- Atributos de r y s son compatibles

Note: $r \cap s = r - (r - s)$

Algebra Relacional

Intersección

Relaciones **R** y **S**

R

A	D
a	1
a	2
b	1

S

A	D
a	2
b	3

$R \cap S$

A	D
a	2

Intersección

Ej. Estudiantes y profesores que se llamen igual

$$\Pi_{nom_e}(Estudiantes) \cap \Pi_{nom_p}(Profesores)$$

Equivalente en SQL:

select nom_e ***from*** Estudiantes

intersect

select nom_p ***from*** Profesores

$$\Pi_{nom_e}(Estudiante) \cap \Pi_{nom_p}(Profesores)$$

Reunión natural

Notación: $r \bowtie s$

La reunión natural es una operación binaria que permite combinar ciertas selecciones (atributos con nombres comunes) y un producto cartesiano en una sola operación y elimina los atributos repetidos.

Ejemplo: $R = (A, B, C, D), S = (E, B, D)$
 \bowtie Esquema resultado = (A, B, C, D, E)

r s es definido como:

$$\Pi_{r.A, r.B, r.C, r.D, s.E} (\sigma_{r.B=s.B \wedge r.D=s.D} (r \times s))$$

Algebra Relacional

Reunión natural

Relaciones **R** y **S**

R

A	B
a	1
a	2
b	1

S

A	D
a	2
b	3

R ⋈ **S**

A	B	D
a	1	2
a	2	2
b	1	3

Reunión natural

Ej. Lista de estudiantes y asignaturas con sus notas

$\Pi_{nom_e, nom_a, n1, n2, n3} (Estudiantes \bowtie Inscribe \bowtie Asignaturas)$

Equivalente en SQL:

select nom_e, nom_a, n1, n2, n3 ***from*** Estudiantes
natural join Inscribe ***natural join*** Asignaturas

$\Pi_{nom_e, nom_a, n1, n2, n3} (Estudiantes \bowtie Inscribe \bowtie Asignaturas)$



Reunión natural - Izquierda

Notación: $r \bowtie s$

La reunión natural es una operación binaria que permite combinar ciertas selecciones (atributos con nombres comunes) y un producto cartesiano en una sola operación y elimina los atributos repetidos, pero se asegura de que la relación de la izquierda contenga todos sus registros y completa los campos que le correspondería con la relación de la derecha con valores NULL.

Algebra Relacional

Reunión natural - Izquierda

Relaciones R y S

R

A	B
a	1
a	2
b	1

S

A	D
a	2
c	3

$R \bowtie S$

A	B	D
a	1	2
a	2	2
b	1	null

Reunión natural - Izquierda

Ej. Lista de estudiantes y asignaturas con sus notas

$$\Pi_{nom_e, nom_a, n1, n2, n3} (\rho_E(Estudiantes) \Rightarrow \bowtie \rho_I(Inscribe) \Rightarrow \bowtie \rho_A(Asignaturas))$$

Equivalente en SQL:

```
select nom_e, nom_a, n1, n2, n3 from Estudiantes E  
  left join Inscribe I on E.cod_e = I.cod_e  
  left join Asignaturas A on A.cod_a = I.cod_a
```

$$\Pi_{nom_e, nom_a, n1, n2, n3} (\rho_E(Estudiantes) \Rightarrow \bowtie \rho_I(Inscribe) \Rightarrow \bowtie \rho_A(Asignaturas))$$

Reunión natural - Derecha

Notación: $r \bowtie_r s$

La reunión natural hacia la derecha es una operación binaria que permite combinar ciertas selecciones (atributos con nombres comunes) y un producto cartesiano en una sola operación y elimina los atributos repetidos, pero se asegura de que la relación de la derecha contenga todos sus registros y completa los campos que le correspondería con la relación de la izquierda con valores NULL.

Algebra Relacional

Reunión natural - derecha

Relaciones R y S

R

A	B
a	1
a	2
b	1

S

A	D
a	2
c	3

$R \bowtie S$

A	B	D
a	1	2
a	2	2
c	null	3

Reunión natural - derecha

Ej. Lista de estudiantes y asignaturas con sus notas

$$\Pi_{nom_e, nom_a, n1, n2, n3} (\rho_E(Estudiantes) \bowtie \rho_I(Inscribe) \bowtie \rho_A(Asignaturas))$$

Equivalente en SQL:

```
select nom_e, nom_a, n1, n2, n3 from Estudiantes E  
  right join Inscribe I on E.cod_e = I.cod_e  
  right join Asignaturas A on A.cod_a = I.cod_a
```

$$\Pi_{nom_e, nom_a, n1, n2, n3} (\rho_E(Estudiantes) \bowtie \rho_I(Inscribe) \bowtie \rho_A(Asignaturas))$$

Reunión natural - Completa

Notación: $r \bowtie s$

La reunión natural completa es una operación binaria que permite combinar ciertas selecciones (atributos con nombres comunes) y un producto cartesiano en una sola operación y elimina los atributos repetidos, pero se asegura de que la relación de la derecha y de la izquierda contengan todos sus registros y completa los campos que no tienen correspondencia con la otra relación relación de la izquierda o de la derecha con valores NULL.

Algebra Relacional

Reunión natural - completa

Relaciones R y S

R

A	B
a	1
a	2
b	1

S

A	D
a	2
c	3

$$R \bowtie S$$

A	B	D
a	1	2
a	2	2
b	1	null
c	null	3

Reunión natural – completa

Ej. Lista de estudiantes y asignaturas con sus notas

$$\Pi_{nom_e, nom_a, n1, n2, n3} (\rho_E(Estudiantes) \bowtie \rho_I(Inscribe) \bowtie \rho_A(Asignaturas))$$

Equivalente en SQL:

```
select nom_e, nom_a, n1, n2, n3 from Estudiantes E  
      full join Inscribe I on E.cod_e=I.cod_e  
      full join Asignaturas A on A.cod_a=I.cod_a
```

$$\Pi_{nom_e, nom_a, n1, n2, n3} (\rho_E(Estudiantes) \bowtie \rho_I(Inscribe) \bowtie \rho_A(Asignaturas))$$

División

Adecuada para consultas que incluyen la expresión “para todos”.

Si r y s son relaciones de los esquemas R y S respectivamente donde

$$R = (A1, \dots, Am, B1, \dots, Bn)$$

$$S = (B1, \dots, Bn)$$

El resultado de $r \div s$ es una relación sobre el esquema

$$R - S = (A1, \dots, Am)$$

$$r \div s = \{ t \mid t \in \Pi_{R-S}(r) \wedge \forall u \in s (tu \in r) \}$$

$$r \div s = \Pi_{R-S}(R) - \Pi_{R-S}((\Pi_{R-S}(R) \times S) - \Pi_{R-S, S}(R))$$

Algebra Relacional

División

Relaciones R y S

R

A	D
a	1
a	2
a	3
b	1
b	2

S

D
2
3

$R \div S$

A
a

Asignación

La operación de asignación (\leftarrow) prevee una conveniente forma de expresar queries complejos.

Ejemplo: Escribir $r \div s$ como s

$$temp1 \leftarrow \Pi_{R-S}(r)$$

$$temp2 \leftarrow \Pi_{R-S}((temp1 \times s) - \Pi_{R-S,S}(r))$$

$$result \leftarrow temp1 - temp2$$

El resultado a la derecha de \leftarrow es asignado a la relación de la izquierda de \leftarrow .

Se pueden usar variables en las subsiguientes expresiones.

Proyección Generalizada

Extensión de la operación de proyección que permite funciones aritméticas para ser usadas en la lista de proyección.

Notación: $\Pi_{F_1, F_2, \dots, F_n}(E)$

E es cualquier expresión del álgebra relacional
cada uno de los F_1, F_2, \dots, F_n es una expresión aritmética que involucra constantes y atributos en el esquema de E .

Ej.:

$\Pi_{cod_e, cod_a, n1*0.35+n2*0.35+n3*.3} (Inscripción)$

Proyección Generalizada

Ej. Lista de estudiantes y asignaturas con su nota definitiva

$\Pi_{nom_e, nom_a, n1*+.35+n2*+.35, n3*+.3} (Estudiantes \bowtie Inscribe \bowtie Asignaturas)$

Equivalente en SQL:

select nom_e, nom_a, n1*.35+n2*.35+n3*.3 ***from*** Estudiantes
natural join Inscribe ***natural join*** Asignaturas

$\Pi_{nom_e, nom_a, n1*+.35+n2*+.35, n3*+.3} (Estudiantes \bowtie Inscribe \bowtie Asignaturas)$

Funciones de agregación

Las funciones de agregación toman un conjunto de valores y devuelve como resultado un único valor.

Las siguientes son las funciones de agregación definidas:

- SUM Toma un conjunto de valores y devuelve su suma
- COUNT Devuelve el número de elementos del conjunto
- AVG Devuelve la media de los valores del conjunto
- MAX Devuelve el valor máximo del conjunto
- MIN Devuelve el valor mínimo del conjunto

Funciones de agregación

Select max(A) from T1 --> 56
Select min(C) from T1 --> 3
Select avg(B) from T1 --> 48.16
Select sum(B) from T1 --> 289
Select count(B) from t1 --> 6
Select count(*) from t1 --> 7
Select sum(B)/count(B) from t1 --> 48.16
Select sum(B)/count(*)from t1 --> 41.29

T1

A	B	C
12	10	6
42		5
32	89	3
56	42	48
7	25	
9	23	19
56	100	18

Agrupamiento

Notación: $A_1, \dots, A_n \text{ } \rho(r)$

Donde A_1, \dots, A_n son atributos que aparecen en la relación r .

Las funciones de agrupamiento (ρ), agrupa los atributos por los valores diferentes que puede tener un atributo o conjunto de atributos.

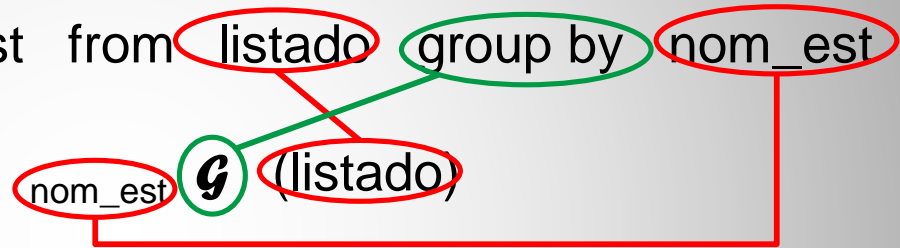
El resultado muestra solamente un valor por cada valor diferente del atributo, o por cada valor diferente de la combinación de atributos.

Ejemplo: $\text{nom_est } \rho(\text{listado})$

Algebra Relacional

Agrupamiento

Ejemplo: `select nom_est from listado group by nom_est`



Listado

Nom_est	Nom_asg	Nota
Pedro Pérez	Base de Datos	4.3
Juan Gómez	Ingeniería de Software	2.5
Rosa Martínez	Base de Datos	3.8
Pedro Pérez	Programación Objetos	4.7
Pablo Calderón	Base de Datos	2.7
Pedro Pérez	Ingeniería de Software	3.0
Rosa Martínez	Programación Objetos	4.1



Nom_est
Pedro Pérez
Juan Gómez
Rosa Martínez
Pablo Calderón

Agrupamiento

Combinando el agrupamiento con las funciones de agregación podemos tener consultas como:

nom_est \mathcal{G} avg(nota) (listado)

```
select nom_est, avg(nota)
from listado
group by nom_est;
```

Nom_est	Avg(nota)
Pedro Pérez	4.0
Juan Gómez	2.5
Rosa Martínez	4.0
Pablo Calderón	2.7

Agrupamiento

Combinando el agrupamiento con las funciones de agregación podemos tener consultas como:

nom_asg \mathcal{G} count(*) (listado)

```
select nom_asg, count(*)  
from listado  
group by nom_asg;
```

Nom_est	Count(*)
Base de Datos	3
Ingeniería de Software	2
Programación Objetos	2

BIBLIOGRAFÍA

C. J. Date, *An Introduction to Database Systems*. Boston: Pearson.

A. Silberschatz, H. F. Korth, y S. Sudarshan, *Database System Concepts*. Mc. Graw Hill.

T. M. Connolly y C. E. Begg, *Database systems a practical approach to design, implementation, and management*. Pearson.