

ÁLGEBRA RELACIONAL

Bases de datos (GIITIN)

Grado en Ingeniería Informática en Tecnologías de la Información

Elena Montañés Roces

Índice

- Lenguajes de consulta
- Álgebra relacional
- Operadores fundamentales
- Otros operadores
- Álgebra relacional extendida

Lenguajes de consulta

- Un **lenguaje de consulta** es un lenguaje en el que los usuarios solicitan información de la base de datos
- Dos tipos:
 - **Procedimentales:** describe las operaciones necesarias para obtener la información (Álgebra relacional, en el que se basa SQL)
 - **No procedimentales:** describe cómo es la información requerida (Cálculo relacional de dominios y de tuplas)

Álgebra relacional

- Procedimental: formada por un conjunto de **operadores fundamentales** (necesarios para obtener cualquier información de la base de datos)
- Cada operador se aplica sobre una o varias relaciones y produce una relación.
- Incorpora otros operadores derivados de los fundamentales para facilitar las consultas
- Se ha ampliado formando el **Álgebra relacional extendida**

Ejemplo

■ Universidad

- alumno=(@nmat, nombre)
- titulacion=(@codigo, nombre, créditos, escuela)
- asignatura=(@nombre, créditos, codigoTitulación)
- matricula=(@nmatAlumno, @nombreAsignatura)
- empresa=(@CIF, siglas, numempleados, numdirectivos)
- practica=(@nmatAlumno, @CIFEmpresa)

O. fundamentales: Selección

- Necesita: una relación r
- Produce: una relación formada por las tuplas de r que cumplen un predicado booleano
 - Sintaxis: $\sigma_{\text{predicado}}(r)$ (**WHERE**)
donde predicado es (por preferencia)
 $\text{Predicado} ::= (\text{Predicado})$
 - | $\neg \text{Predicado}$
 - | $\text{Predicado} \wedge \text{Predicado}$
 - | $\text{Predicado} \vee \text{Predicado}$
 - | ExpresiónBool (Depende del tipo de dato)
 - SQL: WHERE

O. fundamentales: Selección

- Titulaciones impartidas en la Facultad de Ciencias y con más de 120 créditos.

$\sigma_{créditos > 120 \wedge escuela = 'FC'}(titulación)$.

SELECT * FROM titulación WHERE escuela='FC' and créditos>120

Cod.	cred.	escuela
1	130	FC
2	100	FC
3	125	FC
4	130	FD

titulación

Cod.	cred.	escuela
1	130	FC
3	125	FC

O. fundamentales: Proyección

- Necesita: una relación r
- Produce: una relación formada por las mismas tuplas que r pero proyectada sobre los atributos indicados
 - Sintaxis: $\Pi_{\text{Atributo1}, \text{Atributo2}, \dots}(r)$
donde atributo1, atributo2, ... son atributos de r
 - SQL: SELECT

O. fundamentales: Proyección

- El código y la escuela donde se imparten las titulaciones con más de 125 créditos.

$\Pi_{\text{codigo, escuela}} (\sigma_{\text{créditos} > 125} (\text{titulación}))$

SELECT código, escuela FROM titulación WHERE créditos > 125

Cod.	cred.	escuela
1	130	FC
2	100	FC
3	125	FC
4	130	FD

titulación

Cod.	escuela
1	FC
4	FD

O. fundamentales: Unión

- Necesita: dos relaciones r_1 y r_2
- Produce: una relación formada por la unión de las tuplas de las relaciones r_1 y r_2
 - No se muestran tuplas repetidas
 - Es necesario que los atributos de las relaciones que se indiquen tengan el mismo dominio
- Sintaxis: $r_1 \cup r_2$
- SQL: UNION (o OR en WHERE)

O. fundamentales: Unión

- El número de matrícula de los alumnos que hagan prácticas en la empresa cuyo CIF sea '111' o matriculados en BD

$\Pi_{nmatAlumno} (\sigma_{CIFEmpresas='111'} (\text{practica})) \cup \Pi_{nmatAlumno} (\sigma_{nombreAsignatura='BD'} (\text{matricula}))$

(SELECT nmatAlumno FROM practica WHERE CIFEmpresa='111') UNION (SELECT nmatAlumno FROM matricula WHERE nomAsig='BD');

Nmat Alum	CIF Emp
1	'222'
2	'111'
2	'222'
3	'111'

Nmat Alum	Nom Asig
1	ED
2	BD
1	SO
3	SO

NmatAlumno
2
3

O. fundamentales: Diferencia

- Necesita: dos relaciones r_1 y r_2
- Produce: una relación formada por las tuplas de r_1 que no estén en r_2
 - Es necesario que los atributos de las relaciones que se indiquen tengan el mismo dominio
- Sintaxis: $r_1 - r_2$
- SQL: EXCEPT (MINUS o NOT IN)

O. fundamentales: Diferencia

- El número de matrícula de los alumnos que hagan prácticas en la empresa cuyo CIF sea '111' pero que no estén matriculados en BD

$\Pi_{nmatAlumno} (\sigma_{CIFEmpresa='111'} (practica)) - \Pi_{nmatAlumno} (\sigma_{nombreAsignatura='BD'} (matricula))$

(SELECT nmatAlumno FROM practica WHERE CIFEmpresa='111') MINUS (SELECT nmatAlumno FROM matricula WHERE nomAsig='BD')

Nmat Alum	CIF Emp
1	'222'
2	'111'
2	'222'
3	'111'

practica

Nmat Alum	Nom Asig
1	ED
2	BD
1	SO
3	SO

matricula

NmatAlumno
3

O. fundam.: Prod. Cartesiano

- Necesita: dos relaciones r_1 y r_2
- Produce: una relación formada por todas las combinaciones de las tuplas de r_1 y r_2
 - El número de tuplas obtenido será el producto del n° de tuplas de r_1 por el n° de tuplas de r_2
- Sintaxis: $r_1 \times r_2$
- SQL: INNER JOIN (sin condición ON)

O. fundam.: Prod. cartesiano

- El número de matrícula y el nombre de los alumnos matriculados en alguna asignatura

$\Pi_{nmat, nombre} (\sigma_{nmatAlumno=nmat}(\text{matricula} \times \text{alumno}))$

SELECT nmat, nombre FROM alumno INNER JOIN matricula ON
nmat=nmatAlumno;

Nmat	nombre
1	P
2	J
3	A

alumno

Nmat Alum	Nom Asig
1	ED
1	BD
3	BD

matricula

Nmat	nombre
1	P
1	P
3	A

O. fundam.: Renombramiento

- Necesita: una relación r
- Produce: una relación idéntica a r pero con un nombre diferente
 - Sintaxis: **pnombre(r)** ó **pnombre(Atributo1,Atributo2,...)(r)**
donde nombre es el nuevo nombre otorgado a la relación
y atributo1, atributo2,... son atributos de r
 - SQL: CREATE VIEW (o en el FROM)

O. fundam.: Renombramiento

- Denominar mediante AlumnosBD los números de matrícula de los alumnos de BD

ρ AlumnosBD(Π_{nmatAlumno}(σ_{nombreAsignatura='BD'}(matricula)))

CREATE VIEW AlumnosBD IS SELECT nmatAlumno FROM matricula WHERE nombreAsignatura='BD';

NmatAlum	NomAsig
1	ED
1	BD
2	BD
3	ED

matricula

NmatAlum
1
2

AlumnosBD

Otros operadores: Intersección

- Necesita: dos relaciones r_1 y r_2
- Produce: una relación formada por las tuplas que estén a la vez en r_1 y en r_2
 - Es necesario que los atributos de las relaciones que se indiquen tengan el mismo dominio
- Sintaxis: $r_1 \cap r_2$ (equiv. $r_1 - (r_1 - r_2)$)
- SQL: INTERSECT (o dos IN en el WHERE)

Otros operadores: Intersección

- El número de matrícula de los alumnos que hagan prácticas en la empresa cuyo CIF sea '111' y que estén matriculados en BD

$\Pi_{nmatAlumno} (\sigma_{CIFEmpresa='111'} (\text{practica})) \cap \Pi_{nmatAlumno} (\sigma_{nombreasignatura='BD'} (\text{matricula}))$

(SELECT nmatAlumno FROM practica WHERE CIFEmpresa='111') INTERSECT (SELECT nmatAlumno FROM matricula WHERE nomAsig='BD')

Nmat Alum	CIF Emp
1	'222'
2	'111'
2	'222'
3	'111'

practica

Nmat Alum	Nom Asig
1	ED
2	BD
1	SO
3	SO

matricula

NmatAlumno
2

Otros operad.: Reunión natural

- Necesita: dos relaciones r_1 y r_2
- Produce: una relación formada por todas las combinaciones de las tuplas de r_1 y r_2 y realiza una selección forzando la igualdad de los atributos comunes a ambas relaciones
 - Sintaxis: $r_1 \bowtie r_2$
(equiv. $\Pi R_1 \cup R_2 (\sigma_p(r_1 \times r_2))$ donde p predicado de igualdad entre atributos comunes de r_1 y r_2).
 - Si $R_1 \cap R_2 = \emptyset$ entonces $r_1 \bowtie r_2 = r_1 \times r_2$
 - SQL: INNER JOIN (con condición en ON)

Otros operad.: Reunión natural

- El número de matrícula y el nombre de los alumnos matriculados en alguna asignatura

$\Pi_{\text{nmat, nombre}} (\text{matricula} \bowtie \text{alumno})$

SELECT nmat, nombre FROM alumno INNER JOIN matricula ON nmat=nmatalumno;

Nmat	nombre
1	P
2	J
3	A

alumno

Nmat Alum	Nom Asig
1	ED
1	BD
3	BD

matricula

Nmat	nombre
1	P
1	P
3	A

Otros operad.: División

- Necesita: dos relaciones r_1 y r_2
- Produce: Siendo $r_1(R_1)$ y $r_2(R_2)$ una relación formada por todas las tuplas t :
 1. $t \in \Pi R_1 - R_2$
 2. Para cada tupla $t_2 \in r_2$, $\exists t_1 \in r_1$ que cumple:
 1. $t_1[R_2] = t_2[R_2]$
 2. $t_1[R_1 - R_2] = t$
- Sintaxis: $r_1 \div r_2$
(equiv. $\Pi R_1 - R_2(r_1) - \Pi R_1 - R_2((\Pi R_1 - R_2(r_1)) \times r_2) - \Pi R_1 - R_2, R_2(r_1))$
- SQL: NOT EXISTS (también con NOT IN)

Otros operad.: División

- El nombre de las asignaturas en las que estén matriculados todos los alumnos

$\Pi_{\text{NombreAsignatura}} (\text{matricula} \div \text{alumno})$

```
SELECT nombreasignatura FROM matricula m1 WHERE EXISTS (SELECT * FROM alumno) AND  
NOT EXISTS (SELECT * FROM alumno al WHERE NOT EXISTS (SELECT * FROM matricula m2  
WHERE m2.nombreAsignatura=m1.nombreAsignatura AND m2.nmatAlumno=al.nmat))
```

Nmat	nombre
1	P
2	J
3	A

alumno

Nmat Alum	Nom Asig
1	ED
3	ED
1	BD
2	BD
3	BD

matricula

NombreAsig
BD

Otros operad.: Asignación

- Necesita: una relación r
- Produce: Describe una expresión por partes
 - Sintaxis: **nombre ← r**
 - SQL: CREATE VIEW

Otros operad.: Asignación

- Expresar por partes los números de matrícula de los alumnos de BD

$\Pi_{nmatAlumno}(\sigma_{nombreAsignatura='BD'}(matricula))$

$p \leftarrow \sigma_{nombreAsignatura='BD'}(matricula)$

$\Pi_{nmatAlumno}(p)$

```
CREATE VIEW p SELECT * FROM matricula WHERE nombreAsignatura='BD';
SELECT nmatAlumno FROM p;
```

Álg. Rel. Extendida: Proyección generalizada

- Necesita: una relación r
- Produce: una relación formada por las mismas tuplas que r pero proyectada sobre los atributos o expresiones indicados
 - Sintaxis: $\Pi_{\text{exp1}, \text{exp2}, \dots}(r)$.
donde $\text{exp1}, \text{exp2}, \dots$ pueden ser atributos de r o expresiones aritméticas de los atributos de r
 - SQL: SELECT

Álg. Rel. Extendida: Proyección generalizada

- El CIF y las siglas de las empresas junto con el ratio empleados-directivos
 $\Pi_{CIF, \text{siglas}, \text{numempleados}/\text{numdirectivos}} (\text{empresa})$
SELECT CIF, siglas, numempleados/numdirectivos) FROM empresa;

CIF	Siglas	Numempl eados	Num Direct ivos
111	UNO	100	4
222	DOS	200	2
333	TRES	300	6
444	CUATRO	400	2

CIF	Siglas	
111	UNO	25
222	DOS	100
333	TRES	50
444	CUATRO	200

empresa

Álg. Rel. Extendida: Funciones de agregación

- Necesita: una relación r
- Produce: un valor agregado
 - Sintaxis: $Gf1(at1), \dots, fn(atn)(r)$ ó
 $g1, \dots, gm Gf1(at1), \dots, fn(atn)(r)$
donde $t1, \dots, atn$ son atributos de r , $f1, \dots fn$ son funciones de agregación (min, max, avg, sum, count) y $g1, \dots, gm$ son atributos por los que agrupar para realizar la agregación
 - SQL: SELECT, GROUP BY, HAVING

Álg. Rel. Extendida: Funciones de agregación

- El número medio de créditos de las titulaciones

Gavg(creditos)(titulación)

SELECT **avg**(créditos) FROM titulación;

Cod.	cred.	escuela
1	175	FC
2	100	FC
3	125	FC
4	100	FD

titulación

125

Álg. Rel. Extendida: Funciones de agregación

- El número medio de créditos de las titulaciones por cada escuela

`escuelaGavg(creditos)(titulación)`

`SELECT escuela, avg(creditos) FROM titulación GROUP BY escuela;`

Cod.	cred.	escuela
1	175	FC
2	100	FC
3	125	FC
4	100	FD

titulación

escuela	
FC	133,4
FD	100

Álg. Rel. Extendida: Reunión natural externa

- Necesita: dos relaciones r_1 y r_2
- Produce: una relación formada por las mismas tuplas que la reunión natural pero añade información de:
 - aquellos elementos de r_1 que no se relacionan con ningún elemento de r_2
 - aquellos elementos de r_2 que no se relacionan con ningún elemento de r_1
- Sintaxis Izquierda: $r_1 \Rightarrow\Leftarrow r_2$ (r_1 no con r_2)
 - SQL: LEFT JOIN
- Sintaxis Derecha: $r_1 \Rightarrow\Leftarrow r_2$ (r_2 no con r_1)
 - SQL: RIGHT JOIN
- Sintaxis Completa: $r_1 \Rightarrow\Leftarrow r_2$ (r_1 no con r_2 y r_2 no con r_1)
 - SQL: FULL JOIN

Álg. Rel. Extendida: Reunión natural externa

- El número de matrícula de todos los alumnos y para aquellos que estén matriculados el nombre de las asignaturas de las que están matriculados

$\Pi_{\text{Inmat}, \text{nombreAsignatura}}(\text{alumno} \Rightarrow \Delta \text{matricula})$

SELECT nmat, nomAsig FROM alumno LEFT JOIN matricula ON nmat=nmatalumno

Nmat	nombre
1	P
2	J
3	A

alumno

Nmat Alum	Nom Asig
1	ED
1	BD
3	BD

matricula

Nmat	NomAsig
1	ED
1	BD
2	null
3	BD

Álg. Rel. Extendida: Reunión natural externa

- El número de matrícula de todos los alumnos y para aquellos que estén matriculados el nombre de las asignaturas de las que están matriculados

$\Pi_{\text{mat}, \text{nombreAsignatura}}(\text{matricula} \triangleright \triangleleft \text{alumno})$

SELECT nomAsig, nmat FROM matricula RIGHT JOIN alumno ON nmat=nmatalumno

Nmat	nombre
1	P
2	J
3	A

alumno

Nmat Alum	Nom Asig
1	ED
1	BD
3	BD

matricula

NomAsig	Nmat
ED	1
BD	1
null	2
BD	3