

Ejemplos Consultas

• Select

Ejemplo 1: Selecciona todos los elementos de la tabla alumnos y todos sus atributos y los devuelve ordenados por apellido y nombre.

```
select * from alumnos order by ape1, ape2, nombre;
```

Ejemplo 2: Selecciona el nombre y los apellidos de los alumnos menores de 25 años y los devuelve ordenados por apellidos y nombres

```
Select nombre, ape1, ape2 from alumnos where edad <= 25  
order by ape1, ape2, nombre.
```

Ejemplo 3: Selecciona el nombre y los apellidos de aquellos alumnos entre 20 y 30 años que son de Andalucía Oriental. Los devuelve sin ordenar

```
Select nombre, ape1, ape2 from alumnos where (edad  
between 20 and 30) and provincia in ('Jaen', 'Granada')
```

• Producto cartesiano

Ejemplo 1: Seleccionar el nombre del departamento, dirigido por Perico.

```
Select departamento.nombre from departamento,  
profesores where profesores.nombre = 'Perico' and departamento.  
director = profesor.NRP
```

Ejemplo 2 Selecciona el nombre y los apellidos de los alumnos menores de 25 años matriculados de la asignatura 'bd1s'

```
Select nombre, ape1, ape2 from alumnos, matricula  
where (edad > 25) and (alumnos.dni = matricula.dni and  
matricula.c_asi = 'bd1s') order by ape1, ape2, nombre.
```

Ejemplo 3: Selecciona los nombres de asignaturas optativas de 4.5 o más créditos de las que está matriculado 'José López'.

```
Select nombres from alumnos, asignatura, matricula
where (carácter = 'op' and credit + creditpt >= 4.5 and
ape1 = 'Jose' and ape2 = 'Lopez' and alumnos.dni = matricula.dni
and matricula.cod-assign = asignatura.cod-assign)
```

• Booleans

Ejemplo 1: Selecciona los alumnos matriculados de alguna asignatura optativa.

```
Select alumnos.dni, ape1, ape2, nombre from matricula, alumnos
where c-asi in (select asigna.c-assign from asigna where carácter = 'op')
and alumno.dni = matricula.dni order by ape1, ape2, nombre
```

Ejemplo 2: Selecciona aquellos alumnos que han obtenido la máxima la: máxima calificación en fbd en el curso 2017-2018

```
Select dni from matricula where c-asi = 'fbd' and
curso = '2017-2018' and calificacion >= all (select calificacion
from matricula)
```

Ejemplo 3: Selecciona los alumnos matriculados en FBD

```
Select dni, ape1, ape2 from alumnos where exists (
select * from matricula where alumno.dni = matricula.dni
and c-assign = 'FBD')
```

Ejemplo 4: Selecciona asignaturas de las que no este matriculado ningún alumno

```
Select c-assign, nombres from asigna where not exists (
select * from matricula where asignatura.c-assign =
matricula.c-assign);
```


Ejemplo 5: Selecciona asignaturas de último curso de las que no está matriculado ningún alumno

```
Select c.asig, nombres, from asigna where curso >= all  
( Select asAluc.curso from asigna as Aluc ) and not exists  
( Select * from matricula where asignatura.c_asig = matricula.  
c_asig );
```

• Operadores conjuntistas

Ejemplo 1: Selecciona los alumnos más jóvenes: Coge todos los alumnos y le quita aquellos para los que hay otro alumno con mayor edad

```
(Select dni from alumnos)
```

minus

```
(Select alumnos.dni from alumnos, alumnos al where (al.edad <  
alumno.edad))
```

Ejemplo 2 Selecciona aquellas asignaturas de más de 6 créditos vigentes en el curso 2017-2018. // Cogemos asig de más de 6 créditos y las del curso 17/18 y hacemos una intersección

```
(Select c.asig from asignatura where credit + creditpr > 6)
```

intersect

```
(Select codas from matricula where curso = '2017-2018')
```

• Division // Consultas SQL según álgebra relacional

Ejemplo 1: Asignaturas en que están matriculados todos los alumnos de Almería. // Selección de asignaturas donde el conjunto de todos los alumnos de Almería, al quitarle todos los alumnos de la asignatura, no tenga algún alumno

```
Select c.asi, nombres from asigna where not exists (
```

```
(select dni from alumnos where provincia = 'Almería')
```

minus

```
(Select matricula.dni from matricula where matricula.c_asig =  
asignatura.c.asi))
```

Ejemplo 2: Alumnos matriculados en todas las asignaturas optativas.
// Selección de los alumnos donde el conjunto de las asignaturas optativas, al quitarte las que está matriculado está vacío

```
Select ap1, ap2, nombre from alumnos where not exists(  
(select c-asi from asigna where caracter = 'op')  
minus  
(select c-asi from matricula where matricula.dni = alumnos.dni))
```

Ejemplo 3: Alumnos que han aprobado todas las asignaturas de segundo de GII. // Selección de los alumnos para que al coger todas las asignaturas de 2º de GII y quitarte las asignaturas de las que estaba matriculado y calificado como aprobado, el conjunto queda vacío

```
Select ap1, ap2, nombre from alumnos where not exists (  
(select c-asi from asigna where curso = 2 and grado = 'GII')  
minus  
(select c-asi from matricula where matricula.dni = alumnos.dni  
and calificacion = 'ay'))
```

Ejemplo 4: Asignaturas que se han impartido todos los años. // Selección de asignaturas para las que, al coger el curso académico de todas las asignaturas posibles y quitarse los cursos donde la asignatura ha tenido matrículas, el conjunto que queda es vacío.

```
Select c-asi, nombres from asigna where not exists (  
(select matricula.curso from matricula)  
minus  
(select ma.curso from matricula ma where asigna.c-asi =  
ma.c-asi))
```


1 Consultas SQL según cálculo relacional

Ej2. Selección de alumnos para los que no existe una asignatura optativa para la que no exista una matrícula del alumno.

```
Select ap1, ap2, nombre from alumno where not exists (  
  select c_asig from asigna where caracte = 'op' and not exists (  
    select * from matricula where matricula.dni = alumno.dni and  
    matricula.c_asig = asigna.c_asig));
```

Ej1. Selección de asignatura para la que no existe un alumno de Almería para el que no exista una matrícula del alumno en dicha asignatura

```
Select c_asig, nombres from asigna where not exists (  
  select dni from alumnos where provincia = 'Almería' and  
  not exists (select * from matricula where matricula.casi =  
    asigna.c_asig and matricula.dni = alumno.dni));
```

Ej3. Alumno para los que no exista una asignatura de 2º GII para los que no exista una matrícula del alumno en dicha asignatura con calificación de aprobado.

```
Select ap1, ap2, nombre from alumno where not exists (  
  select asi from asigna where curso = 2 and grado = 'GII'  
  and not exists (select * from matricula where alumnos.dni =  
    matricula.dni and matricula.casi = asigna.casi and  
    calificacion = 'ap'))
```

• Grup BY

Ejemplo 1: Para cada sexo distinto, muestra la media de su edad, ordenada por sexo.

```
Select sexo, avg(edad) from alumnos group by sexo order by  
sexo
```

Ejemplo 2: Para cada curso y asignatura, cuenta el n° de matrículas existentes.

```
Select curso, c-asi, count(*) from matricula group by curso,  
c-asi order by curso, codas
```

Ejemplo 3: Para cada sexo distinto de alumnos, muestra la media de edad, sólo cuando sexo es 'V'

```
Select sexo, avg(edad) from alumno group by sexo order by  
sexo having sexo = 'V'
```


Ejemplos Álgebra Relacional

1. Encontrar los nombres de los profesores que no tienen categoría AS y pertenecen a las áreas de conocimiento TSEÑAL o ARQUIT

$$\pi_{\text{nom-prof}} \left(\sigma_{(\text{categoria} \neq \text{AS}) \wedge (\text{area} = \text{TSEÑAL} \vee \text{area} = \text{ARQUIT})} (\text{profesores}) \right)$$

2. Encontrar el DNI y el nombre de aquellos alumnos que nacieron antes del 1-1-80

$$\pi_{\text{dni, nom-alumnos}} \left(\sigma_{\text{fecha-nac} \leq 01-01-80} (\text{alumnos}) \right)$$

3. Obtener el DNI y el nombre de aquellos alumnos matriculados de la asignatura de código BDI que son becarios.

$$\pi_{\text{alumnos, DNI, nom-alumno}} \left(\sigma_{(\text{alumnos.DNI} = \text{matricula.DNI}) \wedge (\text{beca} = \text{SI}) \wedge (\text{cod-asis} = \text{SI})} (\text{alumnos} \times \text{matricula}) \right)$$

4. Encontrar la lista de los profesores (NRP y nombre) que imparten la asignatura BDI

$$\pi_{\text{grupo, NRP, nombre}} \left(\sigma_{\text{profesores.NRP} = \text{grupo.NRP}} (\text{profesores} \times \sigma_{\text{cod-asis} = \text{BDI}} (\text{grupos})) \right)$$

5. Encontrar el nombre de los profesores con categoría CU o TU que pertenecen al departamento de nombre electrónica.

$$\pi_{\text{nom-prof}} \left(\sigma_{\text{profesor.cod-dep} = \text{departamento.cod-dep}} \left(\sigma_{\text{categoria} = \text{CU} \vee \text{categoria} = \text{TU}} (\text{profesores}) \times \sigma_{\text{nom-dep} = \text{Electrónica}} (\text{departamentos}) \right) \right)$$

6. Encontrar el nombre de las asignaturas de las que está matriculado el alumno 'Luis'

$$\pi_{\text{nom-asis}} \left(\sigma_{\text{alumno.DNI} = \text{matricula.DNI}} \left(\text{matricula} \times \sigma_{\text{nom-alu} = \text{Luis}} (\text{alumnos}) \right) \right)$$

7.- Encontrar el nombre y el DNI de aquellos alumnos cuya provincia es almería y que están matriculados de alguna asignatura de 1º curso

$$\pi_{\text{nombre}, \text{DNI}} \left(\sigma_{\text{alumnos.DNI} = \text{matricula.DNI} \wedge \text{matricula.codasig} = \text{asign.codasig}} \right. \\ \left. (\text{matricula} \times \sigma_{\text{provincia} = \text{Almería}} (\text{alumnos}) \times \sigma_{\text{curso} = 1} (\text{asignaturas})) \right)$$

8. Encontrar aquellas asignaturas optativas que están en cursos superiores a la asignatura 'Base de Datos'

$$p(\text{asignatura}) = \text{asig}$$

$$\pi_{\text{asig nombre}} \left(\sigma_{\text{asignaturas.curso} < \text{asig.curso} \wedge (\sigma_{\text{asignaturas.nom_asig} = \text{'Base Datos'}}} \right. \\ \left. (\text{asignatura}) \times \sigma_{\text{asig.caracter} = \text{op}} (\text{asig})) \right)$$

π_{asig}

9.- Encontrar las asignaturas que sólo tiene un profesor

$$\pi_{\text{cod_asig}} (\text{grupos}) - \pi_{\text{grupos.cod_asig}} \left(\sigma_{\text{grupo.cod_asig} = \text{gru.cod_asig}} \right. \\ \left. \wedge \text{grupos.NRP} < > \text{gru.NRP} (\text{grupos} \times \text{gru}) \right)$$

10.- Encontrar los códigos de aquella asignatura que o bien son de 2º año o bien no tiene matriculado ningún alumno

$$\pi_{\text{cod_asig}} \left(\sigma_{\text{curso} = 2} (\text{asignatura}) \right) \cup \left(\pi_{\text{cod_asig}} (\text{asignatura}) - \pi_{\text{cod_asig}} (\text{matricula}) \right)$$

11. Obtener para cada profesor, su NRP, su nombre y el nombre del departamento

$$\pi_{NRP, nombre_prof, nom_dep} (departamento \bowtie profesor)$$

12. Obtener el DNI y el nombre de aquellos alumnos matriculados de la asignatura de código BD1 que son becarios

$$\pi_{alumno.DNI, nom_alumno} (\sigma_{beca=s} (alumnos) \bowtie \sigma_{cod_asig=BD1} (matriculados))$$

13. Encontrar alumnos becarios que vienen de almería

$$\sigma_{beca=s \wedge provincia=Almeria} (alumnos) \quad \begin{array}{l} \wedge \rightarrow \wedge \rightarrow y \\ \vee \rightarrow \vee \rightarrow o \end{array}$$

$$\sigma_{beca=s} (alumnos) \wedge \sigma_{provincia=almeria} (alumnos)$$

14.- Encontrar los profesores que tienen categoría 'TU' o 'CU' y dan clases en asignaturas de 2 ciclo.

$$\pi_{nombre_prof, NRP} (\sigma_{categoria='TU' \vee categoria='CU'} (profesores) \wedge$$

$$\pi_{profesores.NRP, profesores.nombre} ((profesores) \bowtie (grupos) \bowtie \sigma_{curso=2} (asignaturas))$$

15.- Encontrar la asignatura en la que dan clase todos los profs de área 'COMPUT' que son de categoría 'CU'

Dividendo: Divisor

Divisor

$$\pi_{NRP} (\sigma_{area=COMPUT \wedge categoria='CU'} (profesores))$$

Dividendo

$$\pi_{cod_asig, NRP} (grupos)$$

16. Encontrar aquellas aulas que estau ocupadas todos los días
y todas las horas en las que se uniate alguna clase

Π cada día, día, hora (clase): Π día, hora (clase)

Ejemplos Cálculo Relacional

1. Encontrar los datos de aquellos profesores que son asociados

$\{P \mid \text{profesor}(P) \text{ and } P.\text{categoria} = 'AS'\}$

2. Encontrar el nombre de aquellos profesores que son asociados

$\{P.\text{nom-prof} \mid \text{profesor}(P) \text{ and } P.\text{categoria} = 'AS'\}$

3. Para cada profesor encontrar el nombre del departamento en el que trabaja

$\{P.NRP, D.\text{nom-dep} \mid \text{profesor}(P) \text{ and } \text{Departamento}(D) \text{ and } P.\text{cod-dep} = D.\text{cod-dep}\}$

4. Encontrar los nombres de los prof

$\{P.\text{nombre-prof} \mid \text{profesor}(P) \text{ and } \text{grupos}(Ag) \text{ and } Ag.\text{cod-orig} = 'FP' \text{ and } Ag.\text{cod-orig} = 'TA' \text{ and } Ag.NRP =$

5. Encontrar el nombre de aquellos profesores prácticos.

$\{P.\text{nombre-prof} \mid \text{profesor}(P) \text{ and } A.\text{tipo} = 'Practica' \text{ and } A.NRP = P.NRP\}$

6. Encontrar parejas de profesores que sean del mismo departamento

$\{Px.\text{nom-prof}, Py.\text{nom-prof} \mid \text{profesor}(Px) \text{ and } \text{profesor}(Py) \text{ and } Px.\text{cod-dep} = Py.\text{cod-dep} \text{ and } Px.NRP < Py.NRP\}$

7. Encontrar las asignaturas en las que dan clase todos los profesores del área 'COMPUT'

$\{A.\text{cod-orig} \mid \text{asignatura}(A) \text{ and } \text{not}(\text{exists } P) (\text{Profesores}(P) \text{ and } Pareja = 'COMPUT' \text{ and } \text{not}(\text{exists } G) \text{ and } G.\text{cod-orig} = A.\text{cod-orig} \text{ and } G.NRP = P.NRP))\}$

1. Asignaturas (Cod-Orig, Nom-orig, créditos, carácter, curso) 'FBD' 'TA'

2. Profesores (NRP, Nom-Prof, Categoría, Área, Cod-Dep) and

3. Departamentos (Cod-Dep, Nom-dep, P and Directo)

4. Grupos (Cod-Orig, Cod-Grupo, Tipo, NRP, Max-Al)

5. (grupo LA)

1. Encontrar el nº de horas que trabaja cada trabajador

{ T.nombre, O.horas-por-semana | Trabajadores (T) and Oficinas (O) and
T.tipo-de-oficio = O.tipo-de-oficio };

2. Encontrar el nombre de trabajadores que han trabajado en el edificio 312 o 460

{ T.nombre | Trabajadores (T) and (exists A) (Asignaciones (A)
and A.id-trabajador = T.id-Trabajador and A.id-edificio = 312 or
A.id-edificio = 460) };

3. Encontrar parejas de trabajadores que tengan el mismo oficio

{ Tx.nombre, Ty.nombre | Trabajadores (Tx) and Trabajadores (Ty)
and Tx.tipo-de-oficio = Ty.tipo-de-oficio and Tx.nombre < Ty.nombre };

- ☐ Trabajadores (id-Trabajador
- ☐ nombre, tel-hr, tipo-de-oficio,
- ☐ id-sopv)
- ☐ Edificios (id-edificio, dir-edificio,
- ☐ tipo, nivel-calidad, categoria)
- ☐ Asignaciones (id-trabajador,
- ☐ id-edificio, fecha-inicio,
- ☐ num-dias)
- ☐ Oficinas (tipo-de-edificio,
- ☐ planta, horas-por-semana)
- ☐
- ☐
- ☐