

Parte A: Consultas de una tabla

Para familiarizarse, muestre por pantalla las cinco primeras filas de cada tabla y observe sus estructuras. Cómo lo haría?

1 Devuelva todos los datos de las notas que no sean de la materia 75.1.

```
select *  
from notas  
where codigo <> 75 or numero <> 1
```

```
select * from notas  
where codigo != 75 OR numero <> 1
```

```
select * from notas  
where not ((codigo = 75) AND (numero = 1))
```

2 Devuelva para cada materia dos columnas: una llamada “codigo” que contenga una concatenación del código de departamento, un punto y el número de materia, con el formato “XX.YY” (ambos valores con dos dígitos, agregando ceros a la izquierda en caso de ser necesario) y otra con el nombre de la materia.

```
to_char()
```

```
select concat(codigo,'.',to_char(numero, 'FM00')) as codigo, nombre from materias
```

```
select to_char(codigo,'fm00') || '.' || to_char(numero,'fm00') AS codigo  
from materias m
```

```
select concat(LPAD(m.codigo, 2, '0') , "." , LPAD(m.numero, 2, '0')) AS codigo, m.nombre  
from materias m;
```

3 Para cada nota registrada, devuelva el padrón, código de departamento, número de materia, fecha y nota expresada como un valor entre 1 y 100.

```
select padron,  
       codigo as codigo_departamento,  
       numero as numero_materia,  
       fecha,  
       (nota * 10) as nota from notas  
Order by nota
```

4 Ídem al anterior pero mostrando los resultados paginados en páginas de 5 resultados cada una, devolviendo la segunda página.

```
select padron,  
       codigo as codigo_departamento,  
       numero as numero_materia,  
       fecha,  
       (nota * 10) as nota from notas  
Order by padron, codigo, numero, fecha  
limit 5 offset 5
```

5 Ejecute una consulta SQL que devuelva el padrón y nombre de los alumnos cuyo apellido es “Molina”.

```
select padron, nombre from alumnos  
where apellido = 'Molina'
```

```
SELECT padron, nombre FROM alumnos  
WHERE UPPER(apellido) = 'MOLINA'  
-- LOWER
```

— Con ILIKE (no usa los índices)

```
SELECT padron, nombre FROM alumnos  
WHERE apellido ILIKE 'molina';
```

```
SELECT padron, nombre, apellido FROM alumnos
WHERE apellido ILIKE 'm%';
```

6 Obtener el padrón de los alumnos que ingresaron a la facultad en el año 2010.

```
select padron
from alumnos
where fecha_ingreso >= '2010-01-01' and fecha_ingreso <= '2010-12-31'
```

```
-- Usando BETWEEN
SELECT padron, nombre
FROM alumnos
WHERE fecha_ingreso BETWEEN '2010-01-01' AND '2010-12-31'
```

```
SELECT padron
FROM alumnos
WHERE TO_CHAR(fecha_ingreso, 'YYYY') = '2010'
```

```
SELECT a.padron
FROM alumnos a
```

```
SELECT padron
FROM alumnos
WHERE EXTRACT(YEAR FROM fecha_ingreso) = '2010';
```

Parte B: Funciones de agregación

7 Obtener la mejor nota registrada en la materia 75.15.

```
select MAX(nota)
from notas
where codigo = 75 and numero = 15
```

8 Obtener el promedio de notas de las materias del departamento de código 75.

```
SELECT AVG(nota)
FROM notas
WHERE codigo = 75
```

9 Obtener el promedio de nota de aprobación de las materias del departamento de código 75.

```
SELECT AVG(nota)
FROM notas
WHERE codigo = 75 AND nota > 4
```

10 Obtener la cantidad de alumnos que tienen al menos una nota.

```
select count(distinct padron)
from notas
```

Parte C: Operadores de conjunto

11 Devolver los padrones de los alumnos que no registran nota en materias.

```
SELECT padron FROM alumnos  
EXCEPT  
SELECT padron FROM notas;
```

```
SELECT padron  
FROM alumnos  
WHERE padron NOT IN (SELECT padron FROM notas)
```

```
select a.padron  
from alumnos a left outer join notas n on a.padron = n.padron  
where n.padron is null
```

```
SELECT n.padron  
FROM alumnos a  
LEFT JOIN notas n ON n.padron = a.padron  
WHERE n.nota IS NULL
```

12 Con el objetivo de traducir a otro idioma los nombres de materias y departamentos, devolver en una única consulta los nombres de todas las materias y de todos los departamentos.

```
SELECT nombre FROM materias  
UNION  
SELECT nombre FROM departamentos
```

Parte D: Joins

13 Devolver para cada materia su nombre y el nombre del departamento.

```
select m.nombre as nombre_de_materia, d.nombre as nombre_de_departamento
from materias m join departamentos d ON m.codigo = d.codigo
```

```
SELECT mat.nombre as materia, dept.nombre as departamento
FROM materias mat INNER JOIN departamentos dept ON (mat.codigo = dept.codigo)
```

```
SELECT m.nombre, d.nombre
FROM materias m
INNER JOIN departamentos d USING(codigo)
```

```
SELECT materias.nombre, departamentos.nombre
FROM materias, departamentos
WHERE materias.codigo = departamentos.codigo
```

14 Para cada 10 registrado, devuelva el padrón y nombre del alumno y el nombre de la materia correspondiente a dicha nota.

```
select a.padron, a.nombre, m.nombre
from notas n
join alumnos a on n.padron = a.padron
join materias m on n.codigo = m.codigo and n.numero = m.numero
where n.nota = 10
```

-- Con todas las condiciones en el WHERE

```
SELECT al.padron as padron_alumno, (al.nombre || ' ' || al.apellido) as nombre_alumno,
ma.nombre as materia
FROM (notas n INNER JOIN alumnos al USING (padron))
      INNER JOIN materias ma ON (ma.codigo = n.codigo AND ma.numero =
n.numero)
WHERE n.nota = 10
```

VOLVEMOS del intervalo 20 45

15 Listar para cada carrera su nombre y el padrón de los alumnos que estén anotados en ella. Incluir también las carreras sin alumnos inscriptos.

```
select c.nombre, a.padron
from carreras c
Left join inscripto_en ie on c.codigo = ie.codigo
```

```
left join alumnos a on ie.padron = a.padron
order by c.nombre
```

16 Listar para cada carrera su nombre y el padrón de los alumnos con padrón mayor a 75000 que estén anotados en ella. Incluir también las carreras sin alumnos inscriptos con padrón mayor a 75000.

```
select c.nombre, a.padron
from carreras as c
left outer join inscripto_en as ie on c.codigo = ie.codigo
left outer join alumnos as a on ie.padron = a.padron and a.padron > 75000
order by c.nombre
```

17 Listar el padrón de aquellos alumnos que tengan más de una nota en la materia 75.15.

Me parece muy feo esto pero no se me ocurre otra

```
select distinct n1.padron
from notas n1
join notas n2 on n1.padron = n2.padron
and n1.codigo = n2.codigo
and n1.numero = n2.numero
and n1.fecha > n2.fecha
where n1.codigo = 75 and n2.numero = 15
```

```
SELECT DISTINCT n1.padron
FROM notas n1, notas n2
WHERE n1.padron = n2.padron
      AND n1.codigo = n2.codigo
      AND n1.numero = n2.numero
      AND n1.fecha > n2.fecha
      AND n1.codigo = 75
      AND n1.numero = 15
```

```
select n.padron
from notas n
where codigo = 75 and numero = 15
group by padron
having count(*) > 1
```

18 Obtenga el padrón y nombre de los alumnos que aprobaron la materia 71.14 y no aprobaron la materia 71.15.

```
select a.padron, a.nombre, a.apellido
from alumnos a join
(select distinct padron from notas where nota > 4 and codigo = 71 and numero =
14
except
select distinct padron from notas where nota > 4 and codigo = 71 and numero =
15) sc
on a.padron =sc.padron
```

```
SELECT a.padron, a.nombre
FROM alumnos a JOIN notas n USING(padron)
WHERE n.codigo = 71 AND n.numero = 14 AND n.nota >= 4
EXCEPT
SELECT a.padron, a.nombre
FROM alumnos a JOIN notas n USING(padron)
WHERE n.codigo = 71 AND n.numero = 15 AND n.nota >= 4
```

```
SELECT a.padron, a.nombre
FROM alumnos a
JOIN notas n USING(padron)
WHERE n.codigo = 71 AND n.numero = 14 AND n.nota >= 4
and not exists (
    SELECT n1.padron -- 1
    FROM notas n1
    WHERE n1.codigo = 71 AND n1.numero = 15 AND n1.nota >= 4
    and n1.padron = a.padron
)
```

```
SELECT a.padron, a.nombre
```

```
FROM alumnos a
JOIN notas n USING(padron)
WHERE n.codigo = 71 AND n.numero = 14 AND n.nota >= 4
and a.padron <> ALL (
    SELECT a.padron
    FROM notas n1
    WHERE n1.codigo = 71 AND n1.numero = 15 AND n1.nota >= 4
    and n1.padron = a.padron
)
```

19 Obtener, sin repeticiones, todos los pares de padrones de alumnos tales que ambos alumnos rindieron la misma materia el mismo día. Devuelva también la fecha y el código y número de la materia.

```
select n1.padron, n2.padron, n1.fecha, n1.codigo, n1.numero
from notas n1
join notas n2 on n1.padron > n2.padron and n1.codigo = n2.codigo and n1.numero =
n2.numero and n1.fecha = n2.fecha
```

Parte E: Agrupamiento

20 Para cada departamento, devuelva su código, nombre, la cantidad de materias que tiene y la cantidad total de notas registradas en materias del departamento. Ordene por la cantidad de materias descendente.

```
select d.codigo, d.nombre, count(distinct m.numero) as
cantidad_de_materias, count(n.nota) as cantidad_de_notas
from departamentos d
join materias m
on d.codigo = m.codigo
join notas n
on d.codigo = n.codigo
and m.numero = n.numero
group by d.codigo, d.nombre
order by cantidad_de_materias desc
```

```
select d.codigo, d.nombre, Count(distinct m.numero) as cant_materias, count(1)
from departamentos d
left join materias m on m.codigo = d.codigo
left join notas n on n.codigo = d.codigo and n.numero = m.numero
group by d.codigo
order by Count(distinct m.numero) desc
```

```
select d.nombre, d.codigo, count(n.nota) as cant_notas, count(
distinct m.numero) as cant_materias
from departamentos d join notas n using(codigo) join materias m
using(codigo, numero)
group by d.codigo, d.nombre
order by cant_materias desc
```

21 Para cada carrera devuelva su nombre y la cantidad de alumnos inscriptos. Incluya las carreras sin alumnos.

```
select c.codigo, c.nombre, count(ie.padron)
from carreras c
left outer join inscripto_en ie
on c.codigo = ie.codigo
left outer join alumnos a
on ie.padron = a.padron
group by c.codigo, c.nombre
```

```

SELECT c.nombre, COUNT(DISTINCT ie.padron)
FROM carreras c
      LEFT JOIN inscripto_en ie ON (c.codigo = ie.codigo)
GROUP BY c.nombre

```

```

select c.codigo, c.nombre, count(ie.padron)
from carreras c
left join inscripto_en ie
      on c.codigo = ie.codigo
group by c.codigo

```

```

select c.nombre, count(ie.padron) as cant_inscriptos
from   carreras c
      left outer join inscripto_en ie using(codigo)
group by c.nombre

```

22 Para cada alumno con al menos tres notas, devuelva su padrón, nombre, promedio de notas y mejor nota registrada.

```

SELECT a.padron
      ,a.nombre
      ,avg(n.nota)
      ,max(n.nota)
FROM alumnos a
LEFT JOIN notas n ON n.padron = a.padron
GROUP BY a.padron
HAVING count(*) >= 3

```

```

SELECT al.padron, al.nombre, AVG(n.nota), MAX(n.nota)
FROM alumnos al
      INNER JOIN notas n ON (al.padron = n.padron) —use inner xq necesito q sea mayor 3
GROUP BY al.padron, al.nombre
HAVING COUNT(*) >= 3

```

```

select sc.padron, sc.nombre, sc.apellido, sc.promedio, nota_maxima
from (select n.padron, a.nombre, a.apellido, count(n.nota) as
cantidad_de_notas, avg(n.nota) as promedio, max(n.nota) as nota_maxima
from notas n join alumnos a on n.padron = a.padron
group by n.padron, a.nombre, a.apellido ) as sc

```

```
where sc.cantidad_de_notas > 3
```

Parte F: Consultas avanzadas

23 Obtener el código y número de la o las materias con mayor cantidad de notas registradas.

```
select n.codigo, n.numero, count(1)
from notas n
group by n.codigo, n.numero
having count(1) >= all (
    select count(1) from notas n
    group by n.codigo, n.numero
)
```

```
select n.codigo, n.numero, count(1)
from notas n
group by n.codigo, n.numero
having count(1) = (
    select count(1) from notas n
    group by n.codigo, n.numero
    order by count(1) desc
    limit 1
)
```

```
select  codigo, numero
from    (
        select  codigo,
                numero,
                rank () over(order by count(notas))
                as ranking_cant_notas
        from    notas
        group   by codigo, numero
    ) x
where   x.ranking_cant_notas = 1;
```

```
select n.codigo, n.numero
from   notas n
group  by n.codigo, n.numero
having count(n.nota) =      (
```

```

        select max(materias_y_notas.cant_notas)
        from (
            select n_aux.codigo, n_aux.numero,
count(n_aux.nota) as cant_notas
            from notas n_aux
            group by n_aux.codigo, n_aux.numero
        ) materias_y_notas
    )

```

24 Obtener el padrón de los alumnos que tienen nota en todas las materias.

```

select padron
from notas
group by padron
having COUNT(DISTINCT codigo || '.' || numero) = (select count(1) from materias)
order by padron;

```

--Obtener el padrón de los alumnos que
--tienen nota en todas las materias.

```

select padron from alumnos a
where not exists (
    select 1 from materias m
    where not exists (
        select 1 from notas n
        where n.padron = a.padron
        and n.codigo = m.codigo
        and n.numero = m.numero
    )
)

```

```

select padron from alumnos a
where not exists (
    select m.codigo, m.numero from materias m
    except
    select n.codigo, n.numero from notas n
    where n.padron = a.padron
)

```

```

select a.padron
from alumnos a,
    (
        select count(*) cant_materias
        from materias
    ) x,
    (
        select oportunidades_en_misma_materia.padron,
        count(oportunidades_en_misma_materia.*) as cant_materias_alumno
        from (
            select n.padron, n.codigo, n.numero, count(*)
            from notas n
            group by n.padron, n.codigo, n.numero
        ) oportunidades_en_misma_materia
        group by oportunidades_en_misma_materia.padron
    ) y
where x.cant_materias = y.cant_materias_alumno
and y.padron = a.padron

```

#Solucion con join y consultas anidadas

```

select tabla_notas_materias.padron
from (select n.padron , count(distinct(m.nombre)) as
cant_de_materias_no_repetidas_con_nota
from notas n inner join materias m on n.codigo = m.codigo and n.numero =
m.numero
group by n.padron) tabla_notas_materias, (select count(m.nombre) as
cant_de_materias_distintas
from materias m) tabla_con_cant_de_materias_distintas
where
tabla_notas_materias.cant_de_materias_no_repetidas_con_nota =
tabla_con_cant_de_materias_distintas.cant_de_materias_distintas;

```

25 Obtener el promedio general de notas por alumno (cuantas notas tiene en promedio un alumno), considerando únicamente alumnos con al menos una nota.


```

select avg(alumnos_y_cant_notas.cant_notas)
from alumnos
inner join (
    select n.padron,
           count(n.nota) as cant_notas
    from notas n
    group by n.padron
) alumnos_y_cant_notas using(padron)

```

```

select avg(x.cant_notas)
from (
    select count(notas) as cant_notas
    from notas
    group by padron
) x;

select avg(x.cant_notas)
from (
    select count(notas) as cant_notas
    from notas
    group by padron
) x;

```

```

select
cast (count(1) as decimal) / count(distinct n.padron)
from notas n

```