

Parte A: Consultas de una tabla

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

```
SELECT * FROM notas WHERE codigo != 75 OR numero != 1
```

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

```
SELECT CONCAT(LPAD(codigo::text, 2, '0'), '.', LPAD(numero::text, 2, '0')) AS codigo, nombre  
FROM materias
```

```
SELECT TO_CHAR(codigo,'fm00') || '.' || TO_CHAR(numero,'fm00') AS codigo, nombre FROM  
materias
```

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.

```
SELECT padron, codigo, numero, fecha, (nota * 10) AS nota FROM notas  
SELECT *, (nota * 10) AS nota10 FROM notas
```

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, numero, fecha, (nota * 10) AS nota FROM notas OFFSET 5 FETCH  
FIRST 5 ROWS ONLY
```

```
SELECT padron, codigo, numero, fecha, (nota * 10) AS nota FROM notas ORDER BY padron,  
codigo, numero, fecha LIMIT 5 OFFSET 5
```

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

```
select padron, nombre  
from alumnos  
where apellido like 'Molina'  
– ilike para case insensitive
```

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

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

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

```
SELECT padron
FROM alumnos
WHERE SUBSTR(fecha_ingreso::text,1,4) = '2010'
```

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

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

```
select padron from alumnos
where to_char(fecha_ingreso, 'YYYY') = '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
```

```
Select padron , nota
From notas
Where nota = (Select MAX(nota) As nota From notas Where codigo = 75 and numero = 15)
```

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

```
select (sum(nota)/count(nota)) as promedio_notas
from notas
where codigo = 75
```

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

```
Select codigo , numero , ROUND(AVG(nota),3)
From notas
Where codigo = 75
Group By codigo , numero
```

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

```
select (sum(nota)/count(nota)) as promedio_notas_aprobadas
from notas
```

where codigo = 75 and nota > 4

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

Select codigo , numero , ROUND(AVG(nota),3)

From notas

Where codigo = 75 and nota >= 4

Group By codigo , numero

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

select count(distinct padron) as alumnos_con_nota

from notas

Select padron

From notas

Group By padron

Having COUNT(*) >

1

Parte C: Operadores de conjunto

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

SELECT padron FROM alumnos EXCEPT SELECT DISTINCT padron FROM notas

SELECT *

FROM alumnos

WHERE padron not in (

 SELECT padron from notas

);

Select

 alumnos.padron ,

 alumnos.nombre ,

 alumnos.apellido

From

 notas

Right Join

 alumnos

On

 notas.padron = alumnos.padron and

 notas.padron = NULL

SELECT a.padron

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

```
select padron from alumnos a where not exists (select 1 from notas n where n.padron =
a.padron )
```

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
```

```
Select nombre From materias UNION ALL
Select nombre from departamentos
```

Parte D: Joins

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

```
select s.nombre nombre_materia, d.nombre nombre_depto
from materias as s inner join departamentos as d using(codigo)
```

```
select m.nombre , d.nombre
  from materias m , departamentos d
 where m.codigo = d.codigo
```

```
SELECT m.nombre materia, d.nombre departamento
FROM materias m
INNER JOIN departamentos d ON m.codigo = d.codigo
```

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

```
select al.padron, al.nombre, ma.nombre nombre_materia
from notas as nt inner join materias as ma on nt.codigo = ma.codigo and nt.numero =
ma.numero
inner join alumnos as al on nt.padron = al.padron
where nt.nota = 10
```

```
WITH mate_con_10 AS
(SELECT padron, codigo, numero
FROM public.notas
```

```
WHERE nota = 10)
SELECT al.padron, al.nombre, al.apellido, mat.nombre
FROM (mate_con_10 ma INNER JOIN public.alumnos al
ON (ma.padron = al.padron)) INNER JOIN public.materias mat ON
(ma.codigo = mat.codigo AND ma.numero = mat.numero)
```

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 ca.nombre nombre_carrera, al.padron
from ((carreras as ca left outer join inscripto_en as ie on ca.codigo = ie.codigo)
left outer join alumnos as al on ie.padron = al.padron)
```

```
SELECT c.nombre, i.padron
FROM carreras c
LEFT JOIN inscripto_en i on i.codigo = c.codigo
```

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 alu.nombre, ins.codigo, car.nombre nombre_carrera FROM
(((SELECT * FROM alumnos WHERE padron > 75000)) alu INNER JOIN inscripto_en ins ON
alu.padron=ins.padron
RIGHT JOIN carreras car ON car.codigo=ins.codigo)
```

```
SELECT c.nombre, i.padron
FROM carreras c
LEFT JOIN inscripto_en i on i.codigo = c.codigo and i.padron > 75000
```

```
SELECT C.NOMBRE, I.PADRON FROM CARRERAS C, INSCRIPTO_EN I WHERE
C.CODIGO = I.CODIGO
AND I.PADRON > 75000
UNION
SELECT C.NOMBRE,0 FROM CARRERAS C WHERE NOT EXISTS (SELECT 1 FROM
INSCRIPTO_EN I WHERE C.CODIGO = I.CODIGO
AND I.PADRON > 75000)
```

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

```
SELECT n.padron
FROM notas n
WHERE n.codigo = 75 AND n.numero = 15
GROUP BY n.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 alu.padron, alu.nombre
FROM notas n1 inner join notas n2 on(n1.padron = n2.padron) inner join alumnos alu
on(n1.padron = alu.padron)
WHERE n1.nota >= 4 AND n1.codigo = 71 AND n1.numero = 14 AND
n2.nota < 4 AND n2.codigo = 71 AND n2.numero = 15;
```

```
SELECT N.PADRON,A.NOMBRE FROM NOTAS N, ALUMNOS A WHERE N.CODIGO=71
AND N.NUMERO=14 AND N.NOTA >= 4 AND A.PADRON = N.PADRON
EXCEPT
SELECT N.PADRON,A.NOMBRE FROM NOTAS N, ALUMNOS A WHERE N.CODIGO=71
AND N.NUMERO=15 AND N.NOTA >= 4 AND A.PADRON = N.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 as padron1, n2.padron as padron2, n1.codigo, n1.numero, n1.fecha
    FROM notas n1 inner join notas n2 on(n1.fecha = n2.fecha AND n1.codigo = n2.codigo
AND n1.numero = n2.numero)
    WHERE n1.padron > n2.padron
)
EXCEPT
(
    SELECT n1.padron as padron1, n2.padron as padron2, n1.codigo, n1.numero, n1.fecha
    FROM notas n1 inner join notas n2 on(n1.fecha = n2.fecha AND n1.codigo = n2.codigo
AND n1.numero = n2.numero)
    WHERE n1.padron < n2.padron
)
```

```
SELECT n1.padron, n2.padron, codigo, numero, fecha FROM notas n1 INNER JOIN notas n2
USING(fecha, codigo, numero) WHERE n1.padron > n2.padron
```

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 cant_materias, COUNT(*) as cant_notas
FROM departamentos d inner join materias m on(d.codigo = m.codigo) inner join notas n on
(n.codigo = m.codigo AND n.numero = m.numero)
GROUP BY d.codigo, d.nombre
```

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

```
SELECT ca.nombre, COUNT(*)
FROM carreras ca inner join inscripto_en i on(ca.codigo = i.codigo)
GROUP BY ca.codigo
```

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

```
SELECT alu.padron, alu.nombre, MAX(n.nota), AVG(n.nota)
FROM notas n inner join alumnos alu on(n.padron = alu.padron)
GROUP BY alu.padron, alu.nombre
```

Parte F: Consultas avanzadas

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

```
SELECT codigo, numero
FROM notas
GROUP BY codigo, numero
HAVING COUNT(*) = (
    SELECT COUNT(*) as cant_notas
    FROM notas
    GROUP BY codigo, numero
    ORDER BY cant_notas DESC
    LIMIT 1
);
```

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

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