

## **CONSULTAS EN POSTGRE**

GIL CARDENAS OLIVER JESUS

Análisis Y Desarrollo De Sistemas De Información, Servicio Nacional De Aprendizaje

2451627: Construir El Sistema Que Cumpla Con Los Requisitos De La Solución Informática

Instructora: Heidy Lisbeth Adarme Romero

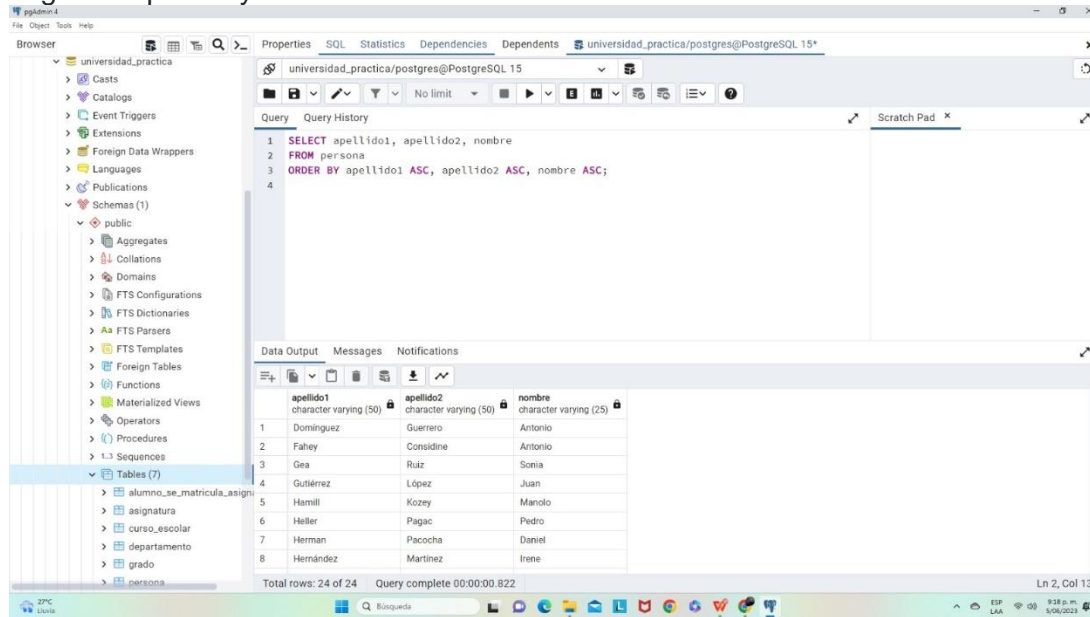
23 DE JUNIO DEL 2023

## SINTAXIS EN SQL A DESARROLLAR

Realice las respectivas consultas sobre las tablas utilizando la herramienta POSTGRESQL, a partir de los siguientes enunciados:

### Consultas sobre una tabla

1. Devuelve un listado con el primer apellido, segundo apellido y el nombre de todos los alumnos. El listado deberá estar ordenado alfabéticamente de menor a mayor por el primer apellido, segundo apellido y nombre.



The screenshot shows the pgAdmin 4 interface. The left sidebar displays the database structure for 'universidad\_practica/postgres@PostgreSQL 15'. The main pane shows a SQL query:

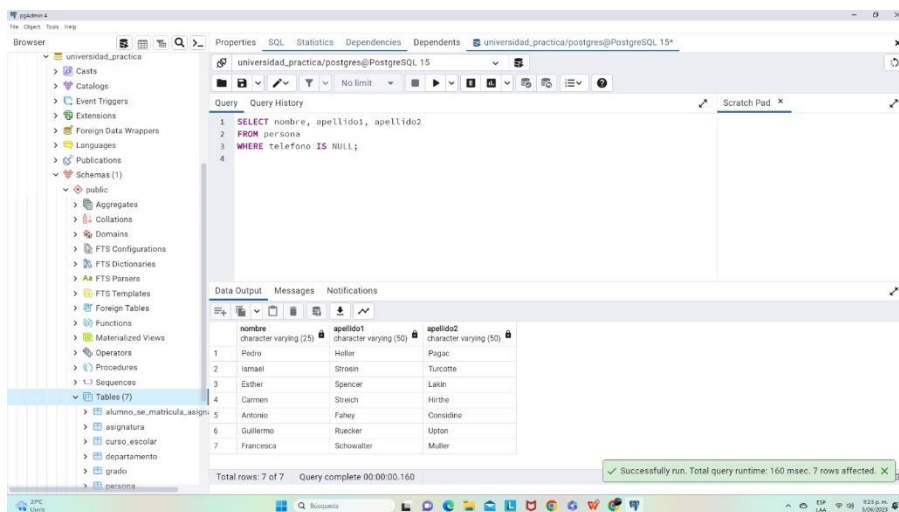
```
1 SELECT apellido1, apellido2, nombre
2 FROM persona
3 ORDER BY apellido1 ASC, apellido2 ASC, nombre ASC;
4
```

The 'Data Output' tab shows the results of the query:

	apellido1 character varying (50)	apellido2 character varying (50)	nombre character varying (25)
1	Dominguez	Guerrero	Antonio
2	Fahey	Considine	Antonio
3	Gea	Ruiz	Sonia
4	Guilérrez	López	Juan
5	Hamill	Kozey	Manolo
6	Heller	Pagac	Pedro
7	Herman	Pacocha	Daniel
8	Hernández	Martinez	Irene

Total rows: 24 of 24. Query complete 00:00:00.822. Ln 2, Col 13.

2. Averigua el nombre y los dos apellidos de los alumnos que **no** han dado de alta su número de teléfono en la base de datos.



The screenshot shows the pgAdmin 4 interface. The left sidebar displays the database structure for 'universidad\_practica/postgres@PostgreSQL 15'. The main pane shows a SQL query:

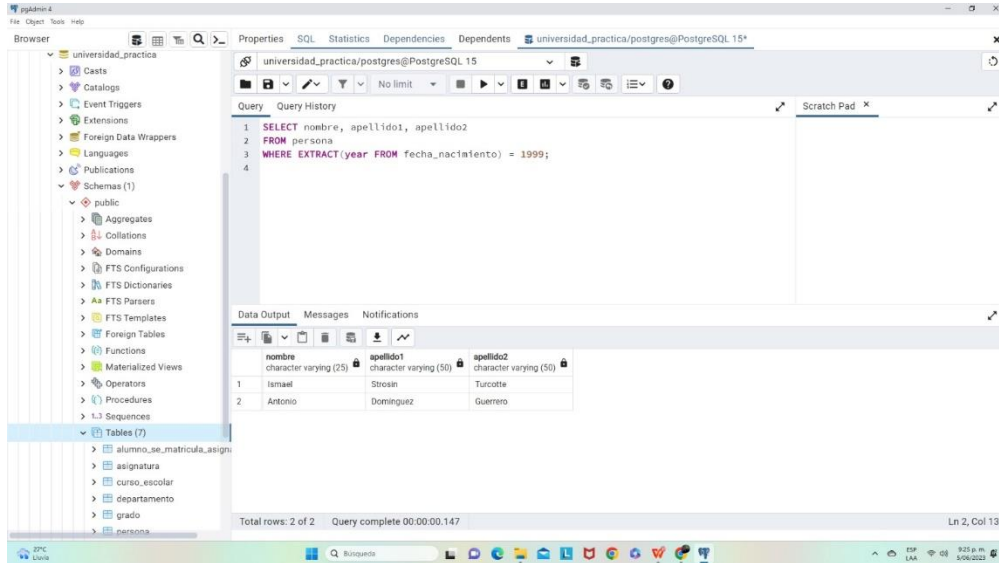
```
1 SELECT nombre, apellido1, apellido2
2 FROM persona
3 WHERE telefono IS NULL;
4
```

The 'Data Output' tab shows the results of the query:

	nombre character varying (25)	apellido1 character varying (50)	apellido2 character varying (50)
1	Pedra	Heller	Pagac
2	lamari	Strosin	Turcotte
3	Esther	Spencer	Lake
4	Carmen	Streich	Hirthe
5	Antonio	Fahey	Considine
6	Guillermo	Ruecker	Upton
7	Francesca	Schowalter	Muller

Total rows: 7 of 7. Query complete 00:00:00.160. Successfully run. Total query runtime: 160 msec. 7 rows affected.

3. Devuelve el listado de los alumnos que nacieron en 1999.



The screenshot shows the pgAdmin 4 interface. The left sidebar displays the database structure, including the 'public' schema and a table named 'personas'. The central pane shows a SQL query:

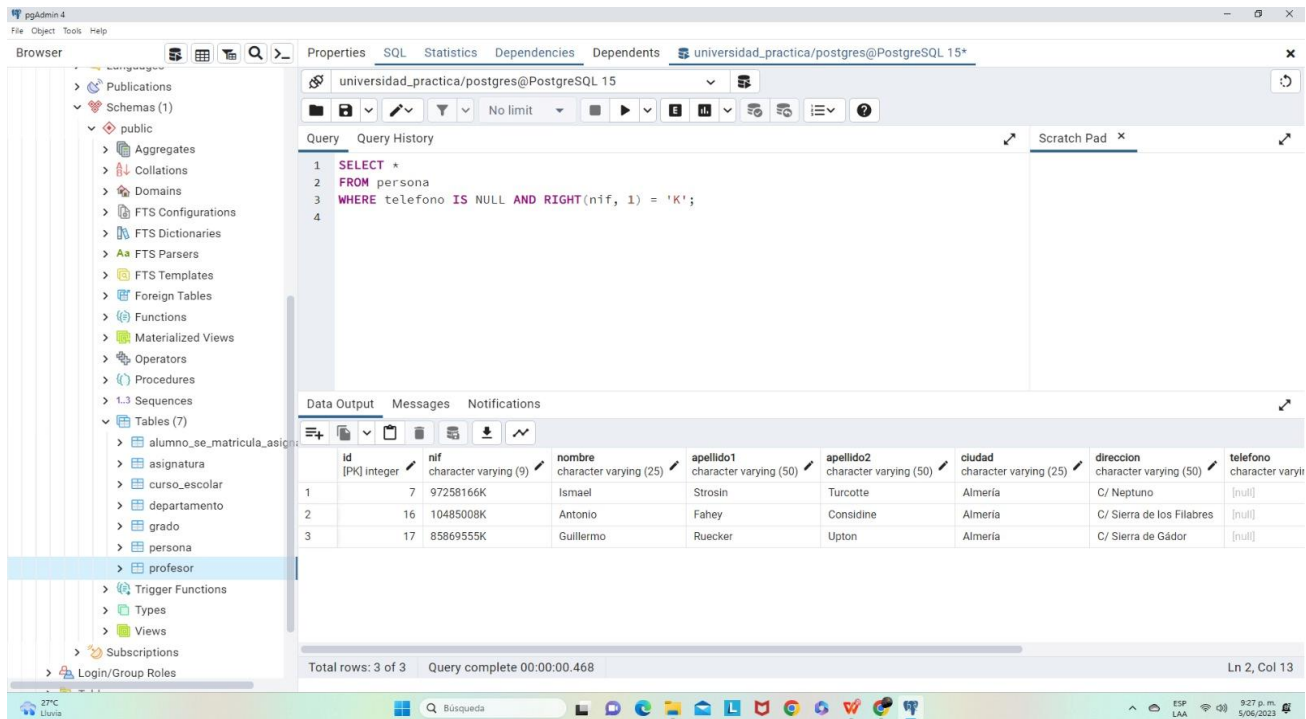
```
1 SELECT nombre, apellido1, apellido2
2 FROM persona
3 WHERE EXTRACT(year FROM fecha_nacimiento) = 1999;
4
```

The 'Data Output' tab at the bottom shows the results of the query:

	nombre character varying (25)	apellido1 character varying (50)	apellido2 character varying (50)
1	Ismael	Strosin	Turcotte
2	Antonio	Dominquez	Guerrero

Total rows: 2 of 2 Query complete 00:00:00.147 Ln 2, Col 13

4. Devuelve el listado de profesores que **no** han dado de alta su número de teléfono en la base de datos y además su nif termina en K.



The screenshot shows the pgAdmin 4 interface. The left sidebar displays the database structure, including the 'public' schema and a table named 'profesor'. The central pane shows a SQL query:

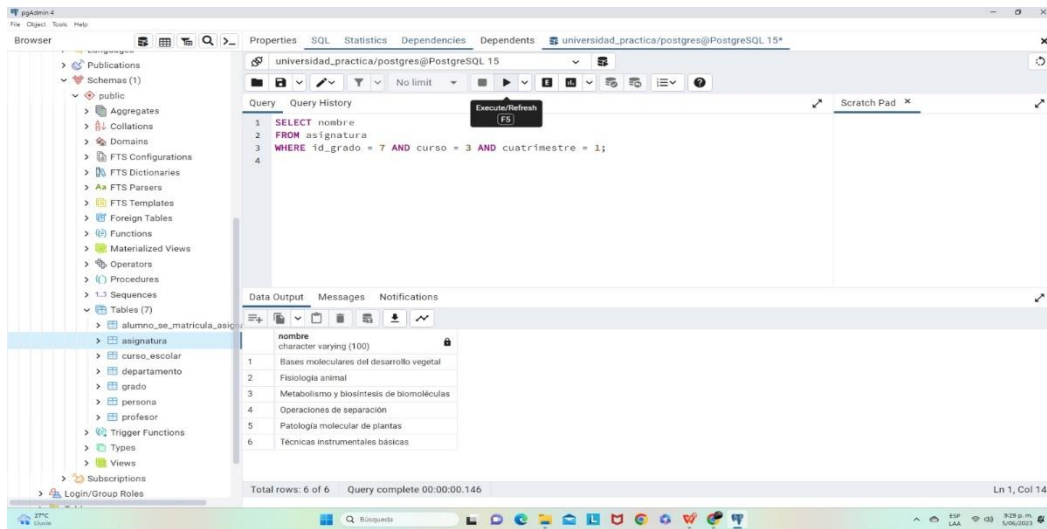
```
1 SELECT *
2 FROM persona
3 WHERE telefono IS NULL AND RIGHT(nif, 1) = 'K';
4
```

The 'Data Output' tab at the bottom shows the results of the query:

	id [PK] integer	nif character varying (9)	nombre character varying (25)	apellido1 character varying (50)	apellido2 character varying (50)	ciudad character varying (25)	direccion character varying (50)	telefono character varyi
1	7	97258166K	Ismael	Strosin	Turcotte	Almeria	C/ Neptuno	[null]
2	16	10485008K	Antonio	Fahey	Considine	Almeria	C/ Sierra de los Filabres	[null]
3	17	85869555K	Guillermo	Ruecker	Upton	Almeria	C/ Sierra de Gádor	[null]

Total rows: 3 of 3 Query complete 00:00:00.468 Ln 2, Col 13

5. Devuelve el listado de las asignaturas que se imparten en el primer cuatrimestre, en el tercer curso del grado que tiene el identificador 7.



## Consultas multitabla (Composición interna)

1. Devuelve un listado con los datos de todas las **alumnas** que se han matriculado alguna vez en el Grado en Ingeniería Informática (Plan 2015).

```
1 SELECT al.nombre, al.apellido1, al.apellido2
2 FROM persona al
3 INNER JOIN alumno_se_matricula asignatura asma ON al.id = asma.id_alumno
4 INNER JOIN asignatura asig ON asma.id_asignatura = asig.id
5 INNER JOIN grado gr ON asig.id_grado = gr.id
6 WHERE gr.nombre = 'Grado en Ingeniería Informática (Plan 2015)' AND al.sexo = 'M';
```

2. Devuelve un listado con todas las asignaturas ofertadas en el Grado en Ingeniería Informática (Plan 2015)

```
1 SELECT asig.nombre as, gr.nombre as
2 FROM asignatura asig INNER JOIN grado gr ON asig.id_grado = gr.id
3 WHERE gr.nombre = 'Grado en Ingeniería Informática (Plan 2015)';
```

3. Devuelve un listado de los profesores junto con el nombre del departamento al que están vinculados. El listado debe devolver cuatro columnas, primer apellido, segundo apellido, nombre y nombre del departamento. El resultado estará ordenado alfabéticamente de menor a mayor por los apellidos y el nombre.

```

1 SELECT pe.apellido1, pe.apellido2, pe.nombre, dep.nombre
2 FROM persona pe
3 INNER JOIN profesor pr ON pr.id_profesor = pe.id
4 INNER JOIN departamento dep ON pr.id_departamento = dep.id
5 ORDER BY pe.apellido1, pe.nombre

```

	apellido1 character varying (50)	apellido2 character varying (50)	nombre character varying (25)	nombre character varying (50)
1	Foley	Considine	Antonio	Economía y Empresa
2	Harnail	Krizey	Manolo	Informática
3	Kohler	Schoon	Alejandro	Matemáticas
4	Leinke	Rutherford	Cristina	Economía y Empresa
5	Monahan	Murray	Micaela	Agronomía
6	Ramirez	Gee	Zoe	Informática
7	Ruecker	Upton	Guillermo	Educación
8	Schmidt	Fisher	David	Matemáticas

4. Devuelve un listado con el nombre de las asignaturas, año de inicio y año de fin del curso escolar del alumno con nif 26902806M.

```

1 SELECT asig.nombre, c.anno_inicio, c.anno_fin, al.nif
2 FROM alumno_se_matricula_asignatura am
3 INNER JOIN persona al ON al.id = am.id_alumno
4 INNER JOIN curso_escolar c ON al.id = c.id
5 INNER JOIN asignatura asig ON am.id_asignatura = asig.id AND al.nif = '26902806M';

```

	nombre character varying (100)	anno_inicio double precision	anno_fin double precision	nif character varying (9)
1	Álgebra lineal y matemática discreta	2014	2015	26902806M
2	Cálculo	2014	2015	26902806M
3	Física para informática	2014	2015	26902806M

5. Devuelve un listado con el nombre de todos los departamentos que tienen profesores que imparten alguna asignatura en el Grado en Ingeniería Informática (Plan 2015)

```

1 SELECT d.nombre, asig.nombre
2 FROM departamento d
3 INNER JOIN profesor pr ON d.id = pr.id_departamento
4 INNER JOIN asignatura asig ON pr.id_profesor = asig.id_profesor
5 INNER JOIN grado gr ON asig.id_grado = gr.id
6 WHERE gr.nombre = 'Grado en Ingeniería Informática (Plan 2015)';

```

	nombre character varying	nombre character varying
1	Informática	Álgebra lineal y matemática discreta
2	Informática	Cálculo
3	Informática	Física para informática
4	Informática	Introducción a la programación
5	Informática	Organización y gestión de empresas
6	Informática	Estadística
7	Informática	Estructura y tecnología de computadores
8	Informática	Fundamentos de electrónica
9	Informática	Álgebra y estadística

6. Devuelve un listado con todos los alumnos que se han matriculado en alguna asignatura durante el curso escolar 2018/2019

```

1 SELECT al.nombre, al.apellido1, al.apellido2, asig.nombre
2 FROM persona al
3 INNER JOIN alumno_se_matricula_asignatura asma ON al.id = asma.id_alumno
4 INNER JOIN curso_escolar ce ON asma.id_curso_escolar = ce.id
5 INNER JOIN asignatura asig ON asma.id_asignatura = asig.id
6 WHERE ce.anyo_inicio = 2018 AND ce.anyo_fin = 2019;

```

## Consultas multitabla (Composición externa)

1. Devuelve un listado con los nombres de **todos** los profesores y los departamentos que tienen vinculados. El listado también debe mostrar aquellos profesores que no tienen ningún departamento asociado. El listado debe devolver cuatro columnas, nombre del departamento, primer apellido, segundo apellido y nombre del profesor. El resultado estará ordenado alfabéticamente de menor a mayor por el nombre del departamento, apellidos y el nombre.

```

1 SELECT d.nombre, p.apellido1, p.apellido2, p.nombre
2 FROM departamento d
3 JOIN profesor pr ON d.id = pr.id_departamento
4 RIGHT JOIN persona p ON p.id = pr.id_profesor
5 ORDER BY d.nombre ASC, p.apellido1 ASC, p.apellido2 ASC, p.nombre ASC;

```

nombre	apellido1	apellido2	nombre
character varying	character varying	character varying	character varying
6 Educación	Streich	Hirthe	Carmen
7 Informática	Hamill	Kozey	Manolo
8 Informática	Ramirez	Gea	Zoe
9 Matemáticas	Kohler	Schoen	Alejandro
10 Matemáticas	Schmidt	Fisher	David
11 Química y Física	Schwalter	Muller	Francesca
12 Química y Física	Stiedemann	Morissette	Alfredo
13 [null]	Dominguez	Guerrero	Antonio
14 [null]	Gea	Ruiz	Sonia
15 [null]	Gutiérrez	López	Juan
16 [null]	Heller	Pagac	Pedro

2 devuelve un listado con los profesores que no están asociados a un departamento

```

1 SELECT p.nombre, p.apellido1, pr.id_departamento
2 FROM persona p
3 LEFT JOIN profesor pr ON p.id = pr.id_profesor
4 WHERE pr.id_departamento IS NULL;

```

nombre	apellido1	id_departamento
character varying	character varying	integer
1 Daniel	Herman	[null]
2 Juan	Saez	[null]
3 Juan	Gutiérrez	[null]
4 Inma	Lakin	[null]
5 Sonia	Gea	[null]
6 José	Koss	[null]
7 Pedro	Heller	[null]
8 Irene	Hernández	[null]
9 Salvador	Sánchez	[null]
10 Antonio	Dominguez	[null]
11 Ramón	Herzog	[null]

3. Devuelve un listado con los departamentos que no tienen profesores asociados.

```

1 SELECT d.id, d.nombre
2 FROM departamento d
3 LEFT JOIN profesor p ON d.id = p.id_departamento
4 WHERE p.id_departamento IS NULL;
5

```

	id [PK] integer	nombre character varying
1	7	Filología
2	8	Derecho
3	9	Biología y Geología

4 Devuelve un listado con los profesores que no imparten ninguna asignatura

```

1 SELECT pr.id_profesor, p.nombre, p.apellido1
2 FROM profesor pr
3 LEFT JOIN asignatura a ON pr.id_profesor = a.id_profesor
4 JOIN persona p ON p.id = pr.id_profesor
5 WHERE a.id_profesor IS NULL;
6

```

	id_profesor integer	nombre character varying	apellido1 character varying
1	5	David	Schmidt
2	8	Cristina	Lemke
3	10	Esther	Spencer
4	12	Carmen	Streich
5	13	Alfredo	Stiedemann
6	15	Alejandro	Kohler
7	16	Antonio	Fahy
8	17	Guillermo	Ruecker
9	18	Micaela	Monahan
10	20	Francesca	Schowalter

5 devuelve un listado con las asignaturas que no tienen un profesor asignado.

```

1 SELECT a.nombre FROM asignatura a
2 LEFT JOIN profesor p ON p.id_profesor = a.id_profesor
3 WHERE a.id_profesor IS NULL;

```

	nombre character varying
1	Ingeniería de Requisitos
2	Integración de las Tecnologías de la Información en las Organizaciones
3	Modelado y Diseño del Software 1
4	Multiprocesadores
5	Seguridad y cumplimiento normativo
6	Sistema de Información para las Organizaciones
7	Tecnologías web
8	Teoría de códigos y criptografía
9	Administración de bases de datos
10	Herramientas y Métodos de Ingeniería del Software
11	Informática industrial y robótica
12	Ingeniería de Sistemas de Información
13	Modelado y Diseño del Software 2
14	Negocio Electrónico



6 devuelve un listado con todos los departamentos que tienen alguna asignatura que no se haya impartido en ningún curso escolar. El resultado debe mostrar el nombre del departamento y el nombre de la asignatura que no se haya impartido nunca.

The screenshot shows a SQL IDE interface with a query editor and a results pane. The query editor contains the following SQL code:

```

1 Select d.nombre, a.nombre FROM departamento d
2 JOIN profesor p ON d.id = p.id_departamento
3 JOIN asignatura a ON p.id_profesor = a.id_profesor
4 LEFT JOIN alumno_se_matricula_asignatura m ON a.id = m.id_asignatura
5 WHERE m.id_curso_escolar IS NULL;
6

```

The results pane displays a table with two columns: **nombre** (character varying) and **nombre** (character varying). The table contains 11 rows of data:

	nombre	nombre
1	Informática	Programación de Servicios Software
2	Informática	Arquitectura de Computadores
3	Informática	Estructura de Datos y Algoritmos II
4	Informática	Estructura de Datos y Algoritmos I
5	Informática	Fundamentos de Redes de Computadores
6	Informática	Sistemas Operativos
7	Informática	Ingeniería del Software
8	Informática	Desarrollo de interfaces de usuario
9	Informática	Planificación y Gestión de Proyectos Informátic...
10	Informática	Bases de Datos
11	Informática	Sistemas Inteligentes

The status bar at the bottom indicates: Total rows: 11 of 11, Query complete 00:00:00.121, Ln 1, Col 27.

## Consultas resumen

1Devuelve el número total de **alumnas** que hay.

The screenshot shows a SQL IDE interface with a query editor. The query editor contains the following SQL code:

```

1 SELECT COUNT(*) AS alumnas FROM persona
2 WHERE tipo = 'alumno' AND sexo = 'M';

```

2Calcula cuántos alumnos nacieron en 1999

The screenshot shows a SQL IDE interface with a query editor. The query editor contains the following SQL code:

```

1 SELECT COUNT(*) AS alumnos FROM persona
2 WHERE tipo = 'alumno' AND EXTRACT(YEAR FROM fecha_nacimiento) = 1999;

```



3Calcula cuántos profesores hay en cada departamento. El resultado sólo debe mostrar dos columnas, una con el nombre del departamento y otra con el número de profesores que hay en ese departamento. El resultado sólo debe incluir los departamentos que tienen profesores asociados y deberá estar ordenado de mayor a menor por el número de profesores.

```
Query  Query History
1 SELECT dep.nombre AS nombre_departamento,
2 COUNT(profesor.id_profesor) AS numero_profesores
3 FROM departamento AS dep
4 INNER JOIN profesor ON dep.id = profesor.id_departamento
5 GROUP BY dep.nombre
6 ORDER BY numero_profesores DESC;
```

Data Output Messages Notifications

4 devuelve un listado con todos los departamentos y el número de profesores que hay en cada uno de ellos. Tenga en cuenta que pueden existir departamentos que no tienen profesores asociados. Estos departamentos también tienen que aparecer en el listado.

```
Query  Query History
1 SELECT departamento.nombre, COUNT(profesor.id_departamento) AS "Numero de Profesores"
2 FROM departamento
3 LEFT JOIN profesor ON departamento.id = profesor.id_departamento
4 GROUP BY departamento.nombre
5 ORDER BY departamento.nombre;
```

5 devuelve un listado con el nombre de todos los grados existentes en la base de datos y el número de asignaturas que tiene cada uno. Tenga en cuenta que pueden existir grados que no tienen asignaturas asociadas. Estos grados también tienen que aparecer en el listado. El resultado deberá estar ordenado de mayor a menor por el número de asignaturas.

```
Query  Query History
1 SELECT grado.nombre, COUNT(asignatura.id_grado) AS "Numero de Asignaturas"
2 FROM grado
3 LEFT JOIN asignatura ON grado.id = asignatura.id_grado
4 GROUP BY grado.nombre
5 ORDER BY "Numero de Asignaturas" DESC;
```

6 devuelve un listado con el nombre de todos los grados existentes en la base de datos y el número de asignaturas que tiene cada uno, de los grados que tengan más de 40 asignaturas asociadas.

```
Query  Query History
1 SELECT grado.nombre, COUNT(asignatura.id_grado) AS "Numero de Asignaturas"
2 FROM grado
3 INNER JOIN asignatura ON grado.id = asignatura.id_grado
4 GROUP BY grado.nombre
5 HAVING COUNT(asignatura.id_grado) > 40
6 ORDER BY "Numero de Asignaturas" DESC;
```

7. Devuelve un listado que muestre el nombre de los grados y la suma del número total de créditos que hay para cada tipo de asignatura. El resultado debe tener tres columnas: nombre del grado, tipo de asignatura y la suma de los créditos de todas las asignaturas que hay de ese tipo. Ordene el resultado de mayor a menor por el número total de créditos

```

Query  Query History
1 SELECT grado.nombre, asignatura.tipo, SUM(asignatura.creditos) AS "Total de Créditos"
2 FROM grado
3 INNER JOIN asignatura ON grado.id = asignatura.id_grado
4 GROUP BY grado.nombre, asignatura.tipo
5 ORDER BY "Total de Créditos" DESC;
6

```

8. Devuelve un listado que muestre cuántos alumnos se han matriculado de alguna asignatura en cada uno de los cursos escolares. El resultado deberá mostrar dos columnas, una columna con el año de inicio del curso escolar y otra con el número de alumnos matriculados.

```

Query  Query History
1 SELECT anyo_inicio, COUNT(DISTINCT id_alumno) AS "Número de Alumnos Matriculados"
2 FROM curso_escolar
3 INNER JOIN alumno_se_matricula_asignatura
4 ON curso_escolar.id = alumno_se_matricula_asignatura.id_curso_escolar
5 GROUP BY anyo_inicio;
6

```

## Subconsultas

- Devuelve todos los datos del alumno más joven.

```

Query  Query History
1 SELECT * FROM persona
2 WHERE tipo = 'alumno'
3 ORDER BY fecha_nacimiento ASC LIMIT 1;
4
5

```

- Devuelve un listado con los profesores que no están asociados a un departamento.

```

Query  Query History
1 SELECT *
2 FROM profesor
3 WHERE id_departamento IS NULL;
4

```

- Devuelve un listado con los departamentos que no tienen profesores asociados.

```

1 SELECT *
2 FROM departamento
3 WHERE id NOT IN (
4     SELECT DISTINCT id
5     FROM profesor
6     WHERE id IS NOT NULL
7 );
8
9

```

- Devuelve un listado con las asignaturas que no tienen un profesor asignado

```

Query  Query History
1 select profesor, asignatura.nombre from asignatura left join profesor
2 on asignatura.id_profesor= profesor.id_profesor where profesor is null;
3

```