

Practica SQL Parcial 2

🕒 Creado	@13 de marzo de 2024 17:47
🏷️ Etiquetas	

Ejercicios de teoria

- - Averiguar todos los marineros con categoria superior a 7;

```
select M.nombrem
from marineros M
where M.categoria > 7;
```

- - Averiguar el nombre de los marineros que han reservado el barco 103;

```
select M.nombrem
from marineros M inner join reservas R on M.idm = R.idm
where R.idb = 103;
```

- - Averiguar el nombre de los marineros que han reservado barcos rojos;

```
select M.nombrem, R.idb, B.color
from marineros M inner join reservas R on M.idm = R.idm inner
join barcos B on R.idb = B.idb
where B.color like 'rojo';
```

- - Averiguar el color de los barcos reservados por Lorca;

```
select B.nombreb, B.color
from marineros M inner join reservas R on M.idm = R.idm inner
join barcos B on R.idb = B.idb
where M.nombrem like 'dominguez';
```

- - Averiguar el nombre de los marineros que han reservado como minimo un barco;

```
select distinct M.nombrem
from marineros M inner join reservas R on M.idm = R.idm;
```

- - Averiguar la edad de los marineros cuyo nombre comienza con B, o acaba con O y tiene como minimo 6 caracteres;

```
select M.nombrem,M.edad
from marineros M
where M.nombrem like "B_%__O";
```

- - Una consulta que liste los idb de los barcos que fueron reservados por marineros todos mayores de 18 años;

```
select R.idb
from reservas R
where R.idm = ANY (select M.idm
from marineros M
where M.edad >=18);
```

- - Averiguar el nombre de los marineros que han reservado barcos rojos o verdes

```
select distinct M.nombrem
from marineros M inner join reservas R on M.idm = R.idm inner join barcos B on R.idb = B.idb
where B.color like 'rojo' or B.color like 'verde';
```

- - Averiguar el nombre de los marineros que han reservado barcos rojos y verdes

```
(select distinct M.nombrem, M.idm
from marineros M inner join reservas R on M.idm = R.idm inner join barcos B on R.idb = B.idb
where B.color like 'rojo'
```

INTERSECT

```
select distinct M.nombrem, M.idm
```

```
from marineros M inner join reservas R on M.idm = R.idm inner join barcos B on R.idb = B.idb
where B.color like 'verde');
```

- - Averiguar el idm de todos los marineros que han reservado barcos rojos pero no verdes

```
select distinct M.idm
from marineros M inner join reservas R on M.idm = R.idm inner join barcos B on R.idb = B.idb
where B.color like 'rojo'
except
select distinct M2.idm
from marineros M2 inner join reservas R2 on M2.idm = R2.idm inner join barcos B2 on R2.idb = B2.idb
where B2.color like 'verde';
```

- - Averiguar el idm de los marineros que tengan una categoria de 10 o hayan reservado el barco 104

```
select M.idm,M.nombrem,M.categoria
from marineros M
where M.categoria = 10
UNION
select M.idm,M.nombrem,M.categoria
from marineros M inner join reservas R on M.idm = R.idm
where R.idb = 104;
```

- - Averiguar el nombre de los marineros que han reservado todos los barcos

```
select M.nombrem
from marineros M
where not exists ( select B.idb
from barcos B
except
(select R.idb
```

```
from reservas R
where R.idm = M.idm));
```

- - Averguar el promedio de edad de los marineros

```
select avg (M.edad)
from marineros M;
```

- - Averguar el promedio de edad de los marineros cuya categoria es 10

```
select avg (M.edad) as promedioEdad
from marineros M
where M.categoria = 10;
```

- - Averguar la edad del marinero mas joven

```
select M.nombrem, M.edad
from marineros M
where M.edad = (select MIN(M2.edad)
from marineros M2);
```

-- Contar el numero de marineros

```
select count(M.idm) as Cantidad_Marineros
from marineros M;
```

- - Contar el nombre de marineros diferentes

```
select count( distinct M.nombrem) as Cantidad_Marineros
from marineros M;
```

- - Averiguar el nombre de los marineros de mas edad que el marinero mas viejo de categoria 10

```
select M.nombrem,M.edad
from marineros M
where M.edad > (select MAX(M2.edad)
```

```
from marineros M2
where M2.categoria = 10);
```

- - Averiguar la edad del marinero mas joven de cada categoria

```
select M.categoria, MIN(M.edad)
from marineros M
group by M.categoria
order by M.categoria;
```

- - Averiguar la edad del marinero mas joven que tiene derecho a voto para cada categoria que tenga como minimo dos marineros con derecho a voto

```
select M.categoria, MIN(M.edad) as edadMinima
from marineros M
where M.edad >= 18
group by M.categoria
having count(*) > 1;
```

- - Para cada barco rojo, averiguar el numero de reserva realizadas

```
select B.idb, count(*) as reservas
from barcos B inner join reservas R on B.idb = R.idb
where B.color like 'rojo'
group by B.idb;
```

- - averiguar la edad media de los marineros de cada categoria que tenga, como minimo dos marineros

```
select M.categoria, avg(M.edad) as edadMedia
from marineros M
group by M.categoria
having count(*) > 1;
```

- - Averiguar las categorias para las que la edad media de los marineros es minima

```
select Temp.categoria,Temp.edadmedia
from (select M.categoria, AVG(M.edad) as edadmedia
from marineros M
group by M.categoria) as Temp
where Temp.edadmedia = (select MIN(Temp.edadmedia) from Temp);
```

Hospital

- - 1)Obtener todos los pacientes cuyos apellidos empiecen con la letra "S"

```
select [P.id](<http://p.id/>),P.apellido,P.nombre
from pacientes P
where P.apellido like 's%';
```

- - 2)Obtener los pacientes cuyo teléfono de contacto contiene el número "555".

```
select [P.id](<http://p.id/>),P.apellido
from pacientes P
where P.telefono like "%555%";
```

- - 3)Obtener la cantidad de pacientes por año de nacimiento.

```
select YEAR(P.fechaNacimiento), count(*) as cantidadPaciente
sEnAñoCorrespondiente
from pacientes P
group by YEAR(P.fechaNacimiento);
```

- - 4)Obtener los pacientes que tienen más de una dirección registrada.

```
select P.nombre,count(P.direccion) as cantidadDomicilios
from pacientes P
Group by P.nombre
having Count(P.direccion) > 1;
```

- - 5) Obtener los pacientes que no tienen asignada una obra social.

```
select *  
from pacientes P  
where obraSocialId is null;
```

- - 6) Obtener los pacientes cuyo correo electrónico no contiene el dominio "gmail.com".

```
select *  
from pacientes P  
where P.email NOT LIKE "%@gmail.com";
```

- - 7) Obtener los pacientes ordenados alfabéticamente por apellido y nombre.

```
Select *  
from pacientes P  
ORDER by P.apellido, P.nombre DESC;
```

- - 8) Obtener los pacientes cuyo nombre y apellido contienen la misma letra.

```
select *  
from pacientes P  
where LEFT(P.apellido,1) = LEFT(P.nombre,1);
```

- - 9) Obtener los pacientes que nacieron en un mes específico (por ejemplo, enero).

```
select *  
from pacientes P  
where month(P.fechaNacimiento) = 2;
```

- - 10) Obtener los pacientes cuya dirección contiene la palabra "avenida".

```
select *  
from pacientes P
```

```
where P.direccion like "%avenida%";
```

- - 11) Obtener los pacientes que tienen una fecha de nacimiento duplicada en la base de datos.

```
select [P.id](<http://p.id/>), P.dni, P.nombre, P.apellido,  
P.fechaNacimiento  
from pacientes P, pacientes P2  
where p.fechaNacimiento = P2.fechaNacimiento and [p.id](<ht  
tp://p.id/>) <> [p2.id](<http://p2.id/>);
```

- - 12) Obtener los pacientes cuyo DNI termina en un número par.

```
select *  
from pacientes P  
where right(P.dni,1) in ('0','2','4','6','8');
```

- - 13) Obtener los pacientes que tienen historias clínicas con diagnóstico "Hipertensión".

```
select *  
from pacientes P inner join historiasclinicas H on [P.id](<  
http://p.id/>) = H.pacienteId  
where H.diagnostico like 'Hipertension';
```

- - 14) Obtener los pacientes que han tenido visitas con motivo "Control de rutina".

```
select *  
from pacientes P inner join historiasclinicas H on [P.id](<  
http://p.id/>) = H.pacienteId  
where H.motivoVisita LIKE 'control de rutina';
```

- - 15) Obtener los pacientes cuyo nombre o apellido comienza con la letra "A"

```
select *  
from pacientes P
```



```
where left(P.nombre,1) like 'a' or left(P.apellido,1) like 'a';
```

Ejercicio de proveedores de la guía

- - 1. Encuentre los nombres de proveedores que provean alguna parte roja.

```
select Distinct S.sname  
from suppliers S inner join catalog C on S.sid = C.sid inner  
join parts P on P.pid = C.pid  
where P.color like 'red';
```

- - 2. Encuentre los sids de proveedores que provean alguna parte roja o alguna parte verde.

```
select Distinct S.sname, P.color  
from suppliers S inner join catalog C on S.sid = C.sid inner  
join parts P on P.pid = C.pid  
where P.color like 'red' or P.color like 'green';
```

- - 3. Encuentre los sids de proveedores que provean alguna parte roja o vivan en "Doderó".

```
select Distinct S.sname, S.sid  
from suppliers S inner join catalog C on S.sid = C.sid inner  
join parts P on P.pid = C.pid  
where P.color like 'red' or S.address like '%Doderó';
```

- - 4. Encuentre los sids de proveedores que provean alguna parte roja y alguna parte verde.

```
select distinct S.sname, S.sid  
from suppliers S inner join catalog C on S.sid = C.sid inner  
join parts P on P.pid = C.pid  
where P.color like 'red'
```

INTERSECT

```
select distinct S.sname, S.sid
from suppliers S inner join catalog C on S.sid = C.sid inner
join parts P on P.pid = C.pid
where P.color like 'green';
```

- - 5. Encuentre los sids de proveedores que provean cada parte.

```
select distinct C.sid
from catalog C
where NOT EXISTS(select P.pid
                  from parts P

                  except

                  select C2.pid
                  from catalog C2
                  where C2.sid = C.sid);
```

- - 6. Encuentre los sids de proveedores que provean cada parte roja.

```
select distinct C.sid
from catalog C
where NOT EXISTS(select P.pid
                  from parts P
                  where P.color like 'red'

                  except

                  select C2.pid
                  from catalog C2
                  where C2.sid = C.sid);
```

- - 7. Encuentre los sids de proveedores que provean cada parte verde o alguna roja.

```

select C.sid
from catalog C
where NOT EXISTS ( select P.pid
                    from parts P
                    where P.color like 'green'

                    except

                    select C2.pid
                    from catalog C2
                    where C.sid = C2.sid)

UNION

select C3.sid
from catalog C3 inner join parts P on C3.pid = P.pid
where P.color like 'red';

```

- - 8. Encuentre los sids de proveedores que provean cada parte roja o provean cada parte verde.

```

select c.sid
from catalog c
where not exists ( select p.pid
                   from parts p
                   where p.color like 'red'

                   except

                   select c1.pid
                   from catalog c1
                   where c.sid = c1.sid)

UNION

select c.sid
from catalog c
where not exists ( select p.pid
                   from parts p

```

```

        where p.color like 'green'

        except

        select c1.pid
        from catalog c1
        where c.sid = c1.sid);

```

- - 9. Encuentre los pares de sids tal que el proveedor con el primer sid tenga la misma pieza pero más costosa que el proveedor del segundo sid.

```

select C.sid, C.cost, C1.sid, C1.cost
from catalog C, catalog C1
where C.sid <> c1.sid and C.pid = C1.pid and C.cost > C1.co
st;

```

- - 10. Encuentre los pids de partes provistas por al menos dos proveedores diferentes.

```

select distinct C.pid
from catalog C, catalog C1
where C.pid = C1.pid and C.sid <> C1.sid;

```

- - 11. Encuentre los pids de las partes mas caras provistas por el proveedor llamado "WinnyPoppy"

```

Select S.sname , C.pid, C.cost
from suppliers S inner join catalog C on S.sid = C.sid
where S.sname like 'WinnyPoppy' and C.cost = ( select MAX(C
1.cost)
                                from catalog
C1);

```

- - 12. Encuentre los pids de partes provistas por cada proveedor a menos que \$200
 -- (si algún proveedor no provee la parte o cuesta igual o más de \$200, dicha parte no deberá ser listada).

```

SELECT p.pid
FROM parts AS p
WHERE NOT EXISTS (
  SELECT 1
  FROM `catalog` AS c
  WHERE c.cost < 200 AND NOT EXISTS (
    SELECT 1
    FROM parts AS p2 INNER JOIN `catalog` AS c2 ON p2.pid = c2.
    pid
    WHERE p.pid = p2.pid AND c.sid = c2.sid)
  );

```

Ejercicio de los aviones

- - 1. Encuentre los ids de pilotos certificados para algún avión Boeing.

```

select distinct E.eid
from employees E inner join certified C on E.eid = C.eid in
ner join aircraft A on C.aid = A.aid
where A.aname like '%Boeing%';

```

- - 2. Encuentre los nombres de pilotos certificados para algún avión Boeing.

```

select distinct E.eid, E.ename, A.aname
from employees E inner join certified C on E.eid = C.eid in
ner join aircraft A on C.aid = A.aid
where A.aname like '%Boeing%';

```

- - 3. Encuentre los aids de todos los aviones que pueden ser usados para vuelos sin escalas desde Bonn hasta Madras.

```

select A.aid
from aircraft A, flights F
where F.origin like 'Los Angeles' and F.destination like 'S
ydney' and A.cruisingrange > F.distance;

```

- - 4. Identifique los vuelos que pueden ser pilotados por cada piloto cuyo salario sea mayor a \$100,000.

```
select distinct E.ename, F.flno
from flights F,employees E INNER JOIN certified C on E.eid
= C.eid INNER JOIN aircraft A on C.aid = A.aid
where E.salary > 200000 and A.cruisingrange > F.distance;
```

- - 5. Encuentre los nombres de pilotos que pueden operar aviones con un rango mayor a 3,000 millas pero que no estén certificados para los aviones Boeing.

```
select distinct E.ename, A.aname, A.cruisingrange
from employees E inner join certified C on E.eid = C.eid in
ner join aircraft A on C.aid = A.aid
where A.cruisingrange > 3000 and A.aname NOT LIKE '%Boein
g%';
```

- - 6. Encuentre los eids de empleados que ganan el mayor salario.

```
select E.eid, E.ename
from employees E
where E.salary = (select MAX(E1.salary)
                  from employees E1);
```

- - 7. Encuentre los eids de empleados que ganen el segundo mayor salario.

```
select E.eid, E.ename
from employees E
where E.salary = (select MAX(E.salary)
                  from employees E
                  where E.salary < (select MAX(E.salary)
                                     From employees E));
```

- - 8. Encuentre los eids de empleados que están certificados para el mayor número de aviones.

```

WITH newt AS (
  SELECT c.eid, COUNT(c.aid) AS contador
  FROM certified AS c
  GROUP BY c.eid
)

select N.eid
from newt N
where N.contador = (select max(contador)
                    from newt);

```

- - 9. Encuentre los eids de los empleados que están certificados para exactamente 3 (tres) aviones.

```

select C.eid
from certified C
group by C.eid
having count(c.aid) = 3;

```

- - 10. Encuentre la cantidad total de dinero pagado en concepto de salario.

```

select sum(E.salary) as CantidadTotal
from employees E

```

Ejercicio Facultad

- - 1. Encuentre los nombres de todos los estudiantes Juniors (level = JR) que estén enrolados en una clase dictada por I. Teach.

```

select S.sname
from student S
join enrolled E on S.snum = E.snum
join class C on C.name = E.cname
join faculty F on C.fid = F.fid
where S.standing Like 'JR' and F.fname like 'I. Teach';

```

- - 2. Encuentre los nombres de todas las clases que cumplan estar en el aula R128 o tenga más de 5 estudiantes inscriptos.

```
select C.name
from class C
where C.room like 'R128' OR C.name in(
                                select E.cname
                                from enrolled E
                                group by E.cname
                                having count(E.snum)
                                >5);
```

- - 3. Encuentre los nombres de todos estudiantes que están inscriptos en dos clases que comiencen a la misma hora.
-- No se como se hace y este resultado saque de la guia hecha y esta mal

```
SELECT * FROM enrolled, class
WHERE cname = name
ORDER BY snum;
```

- - 4. Encuentre los nombres de miembros de la facultad que hay en cada aula en la cual se dicta alguna clase.

```
SELECT room, sname as miembro FROM class c JOIN enrolled e
JOIN student s
ON c.name = e.cname AND e.snum = s.snum
UNION
SELECT room, fname as miembro FROM class c JOIN faculty f
ON c.fid = f.fid
ORDER BY room;
```

Parcial 2024

- - 1. Mostrar nombre de los vendedores que han vendido productos con 20% de descuento en el segundo cuatrimestre del 2020


```
select V.nombre
from vendedores V inner join facturas F on V.codven = F.cod
ven inner join lineas_fac LF on F.codfac = LF.codfac
where year(F.fecha) = 2020 and LF.dto = 20 and (month(F.fecha)
between 5 and 8);
```

- - 2. Código y nombre de aquellas provincias en la que se realizaron mas de 4500 ventas durante el 2019, exceptuando Chaco y Corrientes

```
select P.codpro, P.nombre , count(*)
from facturas F inner join clientes C on F.codcli = C.codcli
inner join pueblos P on C.codpue = P.codpue
inner join provincias PR on PR.codpro = P.codpro
where P.nombre not in ('CHACO','CORRIENTES')
group by P.nombre, P.codpro
having count(*) > 4;
```

- - 3. Mostrar el nombre de aquellos clientes que han realizado compras todos los meses del 2019

```
select C.nombre
from clientes C
where not exists( select distinct month(F.fecha)
                  from facturas F
                  except
                  select distinct month(F.fecha)
                  from facturas F
                  where year(F.fecha) = 2024 and C.codcli=
F.codcli);
```

- - 4. Mostrar el nombre de aquellos vendedores con el máximo numero de ventas durante el 2020

```
select F.codven, count(*) as cant
from facturas F
where year(F.fecha) = 2020
group by codven
```

```
having count(*) = (select max(cant)
                    from (select F.codven, count(*) as cant
                          from factura F
                          where year(F.fecha) = 2020
                          group by F.codven));
```

- - 5. Mostrar el código y descripción de aquellos artículos cuyo descuento supera la media

```
select A.codart, A.descrip
from articulos A inner join lineas_fac F on A.codart = F.codart
where dto > ( select avg(F.dto)
              from lineas_fac F);
```

- - 6. Mostrar el numero máximo de facturas realizadas para un articulo durante el primer cuatrimestre 2024

Parcial aviones

- - 1. Listar los id_avion de todos los aviones que hacen actualmente vuelos directos de Córdoba a Mendoza.

```
select A.id_avion
from avion A inner join viajes V on A.id_avion = V.id_avion
inner join vuelos VU on V.id_vuelo = VU.id_vuelo
where A.autonomia_vuelo > VU.distancia and VU.origen LIKE
'Cordoba' and VU.destino LIKE 'Mendoza';
```

- - 2. Listar los id_avion de todos los aviones que podrían ser usados en vuelos directos desde Buenos Aires a Salta, cuya distancia de vuelo es de 1286 km.

```
select A.id_avion
from avion A inner join viajes V on A.id_avion = V.id_avion
inner join vuelos VU on V.id_vuelo = VU.id_vuelo
```

```
where A.autonomia_vuelo > 1286 and VU.origen LIKE 'Buenos A  
ires' and VU.destino LIKE 'Salta';
```

- - 3. Mostrar los nombres de los pilotos que pueden operar aviones con autonomia de vuelo superior a 4000 km, pero que no sean pilotos de aviones Boeing.

```
select E.nombre  
from empleados E inner join pilotos P on E.id_empleado = P.  
id_empleado inner join avion A on P.id_avion = A.id_avion  
where A.autonomia_vuelo > 4000 and A.marca NOT LIKE 'Boein  
g%';
```

- - 4. Mostrar el nombre y el sueldo de todos los empleados que no son pilotos.

```
SELECT E.nombre, E.sueldo  
FROM EMPLEADOS E  
WHERE E.id_empleado NOT IN (SELECT P.id_empleado FROM pilot  
os P);
```

```
SELECT E.nombre, E.sueldo  
FROM EMPLEADOS E  
WHERE NOT EXISTS (  
    SELECT 1  
    FROM PILOTOS P  
    WHERE P.id_empleado = E.id_empleado  
);
```

- - 5. Listar los id_vuelo que pueden ser piloteados por algún piloto cuyo salario sea mayor a \$50.000.

```
select V.id_vuelo  
from viajes V inner join pilotos P on V.id_avion = P.id_avi  
on inner join empleados E on E.id_empleado = P.id_empleado  
where E.sueldo > 50000;
```

- - 6. Mostrar el id del empleado que percibe el sueldo más alto.

```
select E.id_empleado
from empleados E
where E.sueldo = (select MAX(E.sueldo)
                  from empleados E);
```

- - 7.Nombre del o los pilotos con mayor cantidad de vuelos asignados.

```
select E.nombre, count(*) as vuelos_Asignados
from empleados E inner join pilotos P on E.id_empleado = P.
id_empleado inner join viajes V on P.id_avion = V.id_avion
group by E.nombre
having vuelos_Asignados = (select MAX(vuelos_Asignados)
                           from ( select count(V.id_vuelo)
                                as vuelos_Asignados
                                from viajes V inner join
pilotos P on P.id_avion = V.id_avion
                                group by P.id_empleado)as
temp);
```

- - 8) Nombre del piloto con mayor cantidad de horas de vuelo asignadas.

```
select E.nombre, SUM(TIMESTAMPDIFF(HOUR, VU.hora_salida, V
U.hora_llegada)) AS total_horas
from empleados E inner join pilotos P on E.id_empleado = P.
id_empleado inner join viajes V on P.id_avion = V.id_avion
inner join vuelos VU on V.id_vuelo = VU.id_vuelo
group by E.nombre
order by total_horas desc
limit 1;
```

- - 1.Listar los id_avion de todos los aviones que hacen actualmente vuelos directos de Córdoba a Mendoza.

```
select A.id_avion
from avion A inner join viajes V on A.id_avion = V.id_avion
```

```
inner join vuelos VU on V.id_vuelo = VU.id_vuelo
where A.autonomia_vuelo > VU.distancia and VU.origen LIKE
'Cordoba' and VU.destino LIKE 'Mendoza';
```

- - 2. Listar los id_avion de todos los aviones que podrían ser usados en vuelos directos desde Buenos Aires a Salta, cuya distancia de vuelo es de 1286 km.

```
select A.id_avion
from avion A inner join viajes V on A.id_avion = V.id_avion
inner join vuelos VU on V.id_vuelo = VU.id_vuelo
where A.autonomia_vuelo > 1286 and VU.origen LIKE 'Buenos A
ires' and VU.destino LIKE 'Salta';
```

- - 3. Mostrar los nombres de los pilotos que pueden operar aviones con autonomía de vuelo superior a 4000 km, pero que no sean pilotos de aviones Boeing.

```
select E.nombre
from empleados E inner join pilotos P on E.id_empleado = P.
id_empleado inner join avion A on P.id_avion = A.id_avion
where A.autonomia_vuelo > 4000 and A.marca NOT LIKE 'Boein
g%';
```

- - 4. Mostrar el nombre y el sueldo de todos los empleados que no son pilotos.

```
SELECT E.nombre, E.sueldo
FROM EMPLEADOS E
WHERE E.id_empleado NOT IN (SELECT P.id_empleado FROM pilot
os P);
```

```
SELECT E.nombre, E.sueldo
FROM EMPLEADOS E
WHERE NOT EXISTS (
    SELECT 1
    FROM PILOTOS P
```

```
WHERE P.id_empleado = E.id_empleado
);
```

- - 5. Listar los id_vuelo que pueden ser piloteados por algún piloto cuyo salario sea mayor a \$50.000.

```
select V.id_vuelo
from viajes V inner join pilotos P on V.id_avion = P.id_avion
inner join empleados E on E.id_empleado = P.id_empleado
where E.sueldo > 50000;
```

- - 6. Mostrar el id del empleado que percibe el sueldo más alto.

```
select E.id_empleado
from empleados E
where E.sueldo = (select MAX(E.sueldo)
                  from empleados E);
```

- - 7. Nombre del o los pilotos con mayor cantidad de vuelos asignados.

```
select E.nombre, count(*) as vuelos_Asignados
from empleados E inner join pilotos P on E.id_empleado = P.
id_empleado inner join viajes V on P.id_avion = V.id_avion
group by E.nombre
having vuelos_Asignados = (select MAX(vuelos_Asignados)
                           from ( select count(V.id_vuelo)
                                as vuelos_Asignados
                                from viajes V inner join
pilotos P on P.id_avion = V.id_avion
                                group by P.id_empleado)as
temp);
```

```
with aux as(
    select P.id_empleado, count(V.id_vuelo) as CantVuelo,
    E.nombre
    from viajes V inner join pilotos P on P.id_avion = V.id
_avion inner join empleados E on E.id_empleado = P.id_emple
ado
```

```

        group by P.id_empleado
    ), aux2 as (
        select max(T.CantVuelo) as R
        from aux T)
select A.nombre, aux2.R
from aux A,aux2
where A.CantVuelo = R;

```

- - 8) Nombre del piloto con mayor cantidad de horas de vuelo asignadas.

```

select E.nombre, SUM(TIMESTAMPDIFF(HOUR, VU.hora_salida, V
U.hora_llegada)) AS total_horas
from empleados E inner join pilotos P on E.id_empleado = P.
id_empleado inner join viajes V on P.id_avion = V.id_avion
inner join vuelos VU on V.id_vuelo = VU.id_vuelo
group by E.nombre
order by total_horas desc
limit 1;

```

```

SELECT E.nombre, SUM(HOUR(V.hora_llegada) - HOUR(V.hora_sal
ida)) AS total_horas
FROM EMPLEADOS E NATURAL JOIN PILOTOS P NATURAL JOIN VIAJES
J NATURAL JOIN VUELOS V
GROUP BY E.nombre
ORDER BY total_horas DESC
LIMIT 1;

```

- - 9. Liste los id_avion que no pueden realizar ningún vuelo que parta de Tierra del Fuego.

```

select distinct A.id_avion
from avion A inner join viajes V on A.id_avion= V.id_avion
inner join vuelos VU on V.id_vuelo = VU.id_vuelo
where VU.origen LIKE 'Tierra del Fuego%' and A.autonomia_vu
elo < VU.distancia;

```

- - 10. Nombre todos los pilotos que pilotearon todos los vuelos que parten de un mismo destino.

```

SELECT E.nombre
FROM PILOTOS P
JOIN EMPLEADOS E ON P.id_empleado = E.id_empleado
JOIN VIAJES VJ ON P.id_avion = VJ.id_avion
JOIN VUELOS V ON VJ.id_vuelo = V.id_vuelo
GROUP BY E.nombre, V.origen
HAVING COUNT(DISTINCT V.id_vuelo) = (
    SELECT COUNT(DISTINCT V1.id_vuelo)
    FROM VUELOS V1
    WHERE V1.origen = V.origen
);

WITH VuelosPorOrigenYPilotaje AS (
    SELECT
        v.origen,
        p.id_empleado AS piloto
    FROM
        VUELOS v
    INNER JOIN VIAJES vi ON v.id_vuelo = vi.id_vuelo
    INNER JOIN PILOTOS p ON vi.id_avion = p.id_avion
)
SELECT
    e.nombre AS piloto_nombre
FROM
    VuelosPorOrigenYPilotaje vpop
INNER JOIN EMPLEADOS e ON vpop.piloto = e.id_empleado
GROUP BY
    vpop.origen, e.nombre
HAVING
    COUNT(*) = (SELECT COUNT(*) FROM VUELOS WHERE origen =
vpop.origen)
;

```