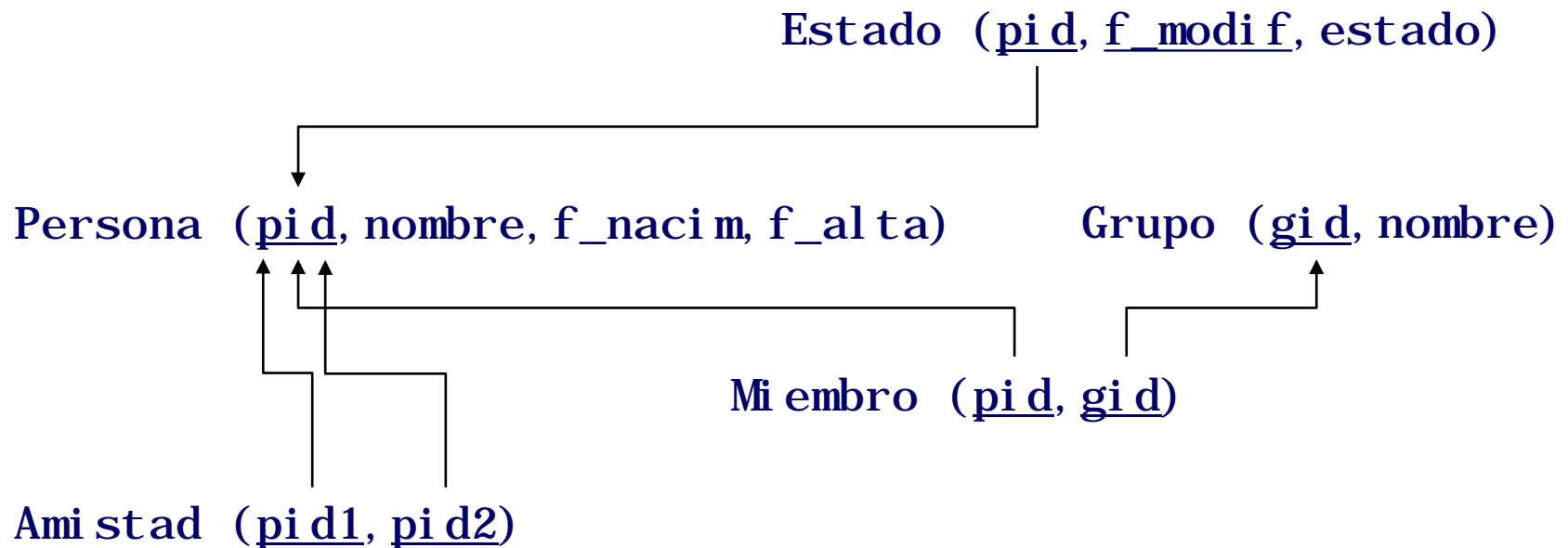


Ejercicio SQL: La red social

– Esquema



--por cada establecimiento de amistad, en esta tabla se incluye el registro (x,y), y el (y,x)
en general, esto reduce la complejidad de las consultas

...

- Devolver el número de amigos de cada persona (nombre)

```
select P.nombre, count(*)  
from Persona P, Amistad A  
where P.pid=A.pid1  
group by P.pid;
```

...

- Seleccionar a la(s) persona(s) con más número de amigos

```
with NAmi gos as  
  (select A. pi d1 as pi d, count(*) as nam  
   from Ami stad A  
   group by A. pi d1)
```

```
select P. nombre, NA. nam  
from Persona P, NAmi gos NA  
where P. pi d = NA. pi d and  
      NA. nam = select max(NA2. nam)  
                  from NAmi gos NA2;
```

...

- Obtener los nombres de las personas que no han realizado ninguna entrada de estado

```
select P.nombre  
from Persona P  
where P.pid not in (select E.pid  
                    from Estado E);
```

...

- Para cada persona que haya actualizado al menos dos veces el estado, determinar el tiempo en que han ocurrido esas actualizaciones

```
select T. pid, (T. fult-T. fpri) as tiempo
from ( select E. pid, max(E. f_modif) as fult,
              min(E. f_modif) as fpri
      from Estado E
      group by E. pid
      having count(*)>1
    ) T;
```

...

- Contar el número de amigos de 'nivel dos' (amigos, y amigos de amigos) de la persona 'X' (cuidado con no contar a la misma persona dos veces)

```
select count(*) - 1
from (
    (select P. pid, A2. pid2
     from Persona P, Amistad A1, Amistad A2
     where P. pid=A1. pid1 and A1. pid2=A2. pid1 and
     P. nombre like 'X%')
 union
    (select P. pid, A1. pid2
     from Persona P, Amistad A1
     where P. pid=A1. pid1 and
     P. nombre like 'X%')
) red1y2;
```

...

- Encontrar a las 5 personas con mayor numero de amigos

```
with NAmigos as (  
    select P.nombre, count(*) as nam  
    from Persona P, Amistad A  
    where P.pid=A.pid1  
    group by P.pid )
```

```
select NA.nombre, NA.nam  
from NAmigos NA  
where 5 > ( select count(*)  
            from NAmigos NA2  
            where NA2.nam > NA.nam  
            );
```

...

- Obtener el ratio de cohesión de cada grupo (nro de enlaces entre miembros / nro máximo de enlaces posibles entre los miembros)

```
select GE. gi d, GE. ne/(GM. nme*(GM. nme- 1)/2)
from ( select M1. gi d, count(*)/2 as ne
      from Mi embro M1, Mi embro M2
      where M1. gi d=M2. gi d and
            (M1. pi d, M2. pi d) in ( select * from Ami stad)
      group by M1. gi d
    ) GE,
    ( select M. gi d, count(*) as nme
      from Mi embro M
      group by M. gi d
    ) GM
where GE. gi d=GM. gi d
order by GE. gi d;
```


...

- Encontrar a los dos personas que hayan nacido con fechas más cercanas (sin contar las que han nacido en la misma fecha)

```
with Difs as (  
    select P1.nombre as nom1, P2.nombre as nom2,  
           abs(P1.f_nac - P2.f_nac) as dif  
    from Persona P1, Persona P2  
    where P1.pid<>P2.pid and abs(P1.f_nac - P2.f_nac)>0 )  
  
select DF.nom1, DF.nom2  
from Difs DF  
where DF.dif = ( select min(DF2.dif)  
                 from Difs DF2  
               );
```