



Solución propuesta

1. Nombre y correo de todos los usuarios.

```
1 | SELECT nombre, email FROM Usuarios;
```

2. Contenido de las publicaciones realizadas por el usuario con id = 1.

```
1 | SELECT contenido
2 |
3 | FROM Publicaciones
4 |
5 | WHERE id_usuario = 1;
```

3. Nombre y correo de los usuarios con más de 1 relación.

```
1 | SELECT U.nombre, U.email
2 |
3 | FROM Usuarios U
4 |
5 | JOIN Relaciones R ON U.id_usuario = R.id_usuario1 OR U.id_usuario = R.id_usuario2
6 |
7 | GROUP BY U.nombre, U.email
8 |
9 | HAVING COUNT(*) > 1;
```

4. Nombre de los usuarios ordenados alfabéticamente.



```
1 | SELECT nombre
2 |
3 | FROM Usuarios
4 |
5 | ORDER BY nombre ASC;
```

5. Nombre de los usuarios que han realizado publicaciones con más de 50 caracteres.

```
1 | SELECT DISTINCT U.nombre
2 |
3 | FROM Usuarios U
4 |
5 | JOIN Publicaciones P ON U.id_usuario = P.id_usuario
6 |
7 | WHERE LENGTH(P.contenido) > 50;
```

6. Nombre y correo de los usuarios cuyos nombres comienzan con 'A' o 'J'.

```
1 | SELECT nombre, email
2 |
3 | FROM Usuarios
4 |
5 | WHERE nombre LIKE 'A%' OR nombre LIKE 'J%';
```

7. Número total de usuarios registrados.

```
1 | SELECT COUNT(*) AS total_usuarios
2 |
3 | FROM Usuarios;
```

8. Nombre del primer y último usuario en orden alfabético.



```
1 | SELECT MIN(nombre) AS primer_usuario, MAX(nombre) AS ultimo_usuario
2 |
3 | FROM Usuarios;
```

9. Nombre y número de publicaciones de los usuarios que han realizado más de 3 publicaciones.

```
1 | SELECT U.nombre, COUNT(P.id_publicacion) AS num_publicaciones
2 |
3 | FROM Usuarios U
4 |
5 | JOIN Publicaciones P ON U.id_usuario = P.id_usuario
6 |
7 | GROUP BY U.nombre
8 |
9 | HAVING COUNT(P.id_publicacion) > 3;
```

10. Número de usuarios agrupados por cantidad de publicaciones realizadas.

```
1 | SELECT COUNT(*) AS num_usuarios, num_publicaciones
2 |
3 | FROM (
4 |
5 |     SELECT U.id_usuario, COUNT(P.id_publicacion) AS num_publicaciones
6 |
7 |     FROM Usuarios U
8 |
9 |     LEFT JOIN Publicaciones P ON U.id_usuario = P.id_usuario
10 |
11 |     GROUP BY U.id_usuario
12 |
13 | ) Subquery
14 |
15 | GROUP BY num_publicaciones;
```



11. Número de usuarios, publicaciones mínimas, máximas y promedio por número de relaciones.

```
1 SELECT COUNT(U.id_usuario) AS num_usuarios,
2
3     MIN(num_publicaciones) AS min_publicaciones,
4
5     MAX(num_publicaciones) AS max_publicaciones,
6
7     AVG(num_publicaciones) AS avg_publicaciones
8
9 FROM (
10
11     SELECT U.id_usuario, COUNT(P.id_publicacion) AS num_publicaciones
12
13     FROM Usuarios U
14
15     LEFT JOIN Publicaciones P ON U.id_usuario = P.id_usuario
16
17     GROUP BY U.id_usuario
18
19 ) Subquery;
```

12. Nombre de los usuarios que han comentado publicaciones del usuario con id = 1.

```
1 SELECT DISTINCT U.nombre
2
3 FROM Usuarios U
4
5 JOIN Comentarios C ON U.id_usuario = C.id_usuario
6
7 JOIN Publicaciones P ON C.id_publicacion = P.id_publicacion
8
9 WHERE P.id_usuario = 1;
```

13. Nombre del usuario, contenido de la publicación y texto de los comentarios recibidos.



```
1 | SELECT U.nombre, P.contenido, C.comentario
2 |
3 | FROM Usuarios U
4 |
5 | JOIN Publicaciones P ON U.id_usuario = P.id_usuario
6 |
7 | JOIN Comentarios C ON P.id_publicacion = C.id_publicacion;
```

14. Usuarios que han enviado mensajes al usuario con id = 1.

```
1 | SELECT DISTINCT U.nombre
2 |
3 | FROM Usuarios U
4 |
5 | JOIN Mensajes M ON U.id_usuario = M.id_usuario_emisor
6 |
7 | WHERE M.id_usuario_receptor = 1;
```

15. Nombre y total de publicaciones realizadas por cada usuario.

```
1 | SELECT U.nombre, COUNT(P.id_publicacion) AS total_publicaciones
2 |
3 | FROM Usuarios U
4 |
5 | LEFT JOIN Publicaciones P ON U.id_usuario = P.id_usuario
6 |
7 | GROUP BY U.nombre;
```

16. Usuarios que han enviado mensajes con más de 100 caracteres.

```
1 | SELECT DISTINCT U.nombre
2 |
3 | FROM Usuarios U
```



```
4  
5 JOIN Mensajes M ON U.id_usuario = M.id_usuario_emisor  
6  
7 WHERE LENGTH(M.mensaje) > 100;
```

17. Usuario con la mayor cantidad de publicaciones.

```
1 SELECT U.nombre, COUNT(P.id_publicacion) AS total_publicaciones  
2  
3 FROM Usuarios U  
4  
5 JOIN Publicaciones P ON U.id_usuario = P.id_usuario  
6  
7 GROUP BY U.nombre  
8  
9 ORDER BY COUNT(P.id_publicacion) DESC  
10  
11 FETCH FIRST 1 ROW ONLY;
```

18. Número promedio de comentarios por publicación.

```
1 SELECT AVG(num_comentarios) AS promedio_comentarios  
2  
3 FROM (  
4  
5     SELECT P.id_publicacion, COUNT(C.id_comentario) AS num_comentarios  
6  
7     FROM Publicaciones P  
8  
9     LEFT JOIN Comentarios C ON P.id_publicacion = C.id_publicacion  
10  
11    GROUP BY P.id_publicacion  
12  
13 );
```



19. Usuarios que no tienen publicaciones.

```
1 | SELECT nombre
2 |
3 | FROM Usuarios
4 |
5 | WHERE id_usuario NOT IN (SELECT id_usuario FROM Publicaciones);
6 |
```

20. Nombre y total de amigos (relaciones) de cada usuario.

```
1 | SELECT U.nombre, COUNT(R.id_usuario2) AS total_amigos
2 |
3 | FROM Usuarios U
4 |
5 | LEFT JOIN Relaciones R ON U.id_usuario = R.id_usuario1
6 |
7 | GROUP BY U.nombre;
```

21. Usuarios que tienen exactamente 3 relaciones de amistad.

```
1 | SELECT U.nombre
2 |
3 | FROM Usuarios U
4 |
5 | JOIN Relaciones R ON U.id_usuario = R.id_usuario1
6 |
7 | GROUP BY U.nombre
8 |
9 | HAVING COUNT(R.id_usuario2) = 3;
```

22. Nombre y edad de los usuarios.

(Suponiendo una columna de fecha de nacimiento fecha_nacimiento en Usuarios):



```
1 | SELECT nombre, EXTRACT(YEAR FROM SYSDATE) - EXTRACT(YEAR FROM fecha_nacimiento) AS edad
2 |
3 | FROM Usuarios;
```

23. Usuarios y la fecha de su primer comentario.

```
1 | SELECT U.nombre, MIN(C.fecha_comentario) AS primer_comentario
2 |
3 | FROM Usuarios U
4 |
5 | JOIN Comentarios C ON U.id_usuario = C.id_usuario
6 |
7 | GROUP BY U.nombre;
```

24. Número de mensajes enviados por cada usuario.

```
1 | SELECT U.nombre, COUNT(M.id_mensaje) AS total_mensajes
2 |
3 | FROM Usuarios U
4 |
5 | LEFT JOIN Mensajes M ON U.id_usuario = M.id_usuario_emisor
6 |
7 | GROUP BY U.nombre;
```

25. Usuarios que han trabajado (comentado) en más de 5 publicaciones.

```
1 | SELECT U.nombre
2 |
3 | FROM Usuarios U
4 |
5 | JOIN Comentarios C ON U.id_usuario = C.id_usuario
6 |
7 | GROUP BY U.nombre
```




```
8  
9 | HAVING COUNT(C.id_comentario) > 5;
```

26. Usuarios y cantidad de comentarios recibidos en sus publicaciones.

```
1 | SELECT U.nombre, COUNT(C.id_comentario) AS total_comentarios  
2 |  
3 | FROM Usuarios U  
4 |  
5 | JOIN Publicaciones P ON U.id_usuario = P.id_usuario  
6 |  
7 | LEFT JOIN Comentarios C ON P.id_publicacion = C.id_publicacion  
8 |  
9 | GROUP BY U.nombre;
```

27. Usuarios que han recibido mensajes de más de 3 usuarios diferentes.

```
1 | SELECT U.nombre  
2 |  
3 | FROM Usuarios U  
4 |  
5 | JOIN Mensajes M ON U.id_usuario = M.id_usuario_receptor  
6 |  
7 | GROUP BY U.nombre  
8 |  
9 | HAVING COUNT(DISTINCT M.id_usuario_emisor) > 3;
```

28. Publicaciones con más de 5 comentarios y sus autores.

```
1 | SELECT U.nombre, P.contenido, COUNT(C.id_comentario) AS total_comentarios  
2 |  
3 | FROM Usuarios U  
4 |  
5 | JOIN Publicaciones P ON U.id_usuario = P.id_usuario  
6 |
```



```
6 |  
7 | JOIN Comentarios C ON P.id_publicacion = C.id_publicacion  
8 |  
9 | GROUP BY U.nombre, P.contenido  
10 |  
11 | HAVING COUNT(C.id_comentario) > 5;
```

29. Usuario con más relaciones de amistad.

```
1 | SELECT U.nombre, COUNT(R.id_usuario2) AS total_relaciones  
2 |  
3 | FROM Usuarios U  
4 |  
5 | JOIN Relaciones R ON U.id_usuario = R.id_usuario1  
6 |  
7 | GROUP BY U.nombre  
8 |  
9 | ORDER BY COUNT(R.id_usuario2) DESC  
10 |  
11 | FETCH FIRST 1 ROW ONLY;
```

30. Usuario con más comentarios en total.

```
1 | SELECT U.nombre, COUNT(C.id_comentario) AS total_comentarios  
2 |  
3 | FROM Usuarios U  
4 |  
5 | JOIN Comentarios C ON U.id_usuario = C.id_usuario  
6 |  
7 | GROUP BY U.nombre  
8 |  
9 | ORDER BY COUNT(C.id_comentario) DESC  
10 |  
11 | FETCH FIRST 1 ROW ONLY;
```



Obra publicada con Licencia Creative Commons Reconocimiento Compartir igual 4.0
(<http://creativecommons.org/licenses/by-sa/4.0/>).