

### CHECK III

DH Tube - 07/02/2022

Integrantes: Sergio Guerrero, Victor Valencia, Esteban Jiménez, Indira Réquiz

-- 1 Generar un reporte que indique la cantidad de usuarios por país, mostrando el nombre del país y su cantidad.

```
SELECT pais.nombre, count(usuario.idUsuario) AS cantidad_usuario FROM usuario
JOIN pais ON usuario.pais_idpais = pais.idPais
GROUP BY usuario.pais_idpais;
```

```
1      -- 1
2 •    SELECT pais.nombre, count(usuario.idUsuario) AS cantidad_usuario FROM usuario
3      JOIN pais ON usuario.pais_idpais = pais.idPais
4      GROUP BY usuario.pais_idpais;
5
```

Result Grid		Filter Rows:	Export:	Wrap Cell Content:
nombre	cantidad_usuario			
Alemania	1			
Andorra	1			
Antigua y Barbuda	1			
Argelia	1			
Argentina	8			
Brasil	8			

-- 2 Mostrar el Top 5 de avatars utilizados por los usuarios, listando el nombre del avatar y la cantidad de veces utilizado.

```
SELECT avatar.nombre, COUNT(usuario.avatar_idavatar) AS cantidad
FROM avatar
INNER JOIN usuario ON avatar.idavatar = usuario.avatar_idavatar
GROUP BY usuario.avatar_idavatar
ORDER BY cantidad DESC
LIMIT 5;
```

```
>      -- 2
6 •    SELECT avatar.nombre, COUNT(usuario.avatar_idavatar) AS cantidad
7      FROM avatar
8      INNER JOIN usuario ON avatar.idavatar = usuario.avatar_idavatar
9      GROUP BY usuario.avatar_idavatar
10     ORDER BY cantidad DESC
11     LIMIT 5;
12
13
```

Result Grid		Filter Rows:	Export:	Wrap Cell Content:
nombre	cantidad			
San Hill	2			
Dexter Jettster	2			
Ki-Adi-Mundi	2			
Clegg Lars	2			
Chewbacca	1			

-- 3 Emitir un listado de todas las playlists públicas, informando el nombre del usuario que la creó, el título de cada video que posee y el nombre y año de creación de la playlist

```
SELECT playlist.idPlaylist AS ID, playlist.privado, usuario.nombre AS UsuarioCreador, video.titulo AS Videos, playlist.nombre AS Playlist, playlist.fechaCreacion AS "Fecha de creacion"
FROM usuario
INNER JOIN playlist ON usuario.idUsuario = Usuario_idUsuario
INNER JOIN playlist_video ON idPlaylist = Playlist_idPlaylist
INNER JOIN video ON Video_idVideo = idVideo
WHERE playlist.privado = 0;
```

```
14 -- 3
15 • SELECT playlist.idPlaylist AS ID, playlist.privado, usuario.nombre AS UsuarioCreador, video.titulo AS Videos,
16     playlist.nombre AS Playlist, playlist.fechaCreacion AS "Fecha de creacion"
17 FROM usuario
18 INNER JOIN playlist ON usuario.idUsuario = Usuario_idUsuario
19 INNER JOIN playlist_video ON idPlaylist = Playlist_idPlaylist
20 INNER JOIN video ON Video_idVideo = idVideo
21 WHERE playlist.privado = 0;
22
```

Result Grid						
Filter Rows: <input type="text"/> Export:  Wrap Cell Content: <input type="checkbox"/>						
	ID	privado	UsuarioCreador	Videos	Playlist	Fecha de creacion
▶	2	0	MARIA PAULA GONZALEZ	Python for beginners	Para Estudiar	2021-05-31 00:00:00
	2	0	MARIA PAULA GONZALEZ	Sorry	Para Estudiar	2021-05-31 00:00:00
	4	0	PATRICIA SOTO	Python for beginners	Las mejores videos	2021-02-14 00:00:00
	4	0	PATRICIA SOTO	Programacion back end	Las mejores videos	2021-02-14 00:00:00
	4	0	PATRICIA SOTO	Sugar	Las mejores videos	2021-02-14 00:00:00
	5	0	MARIA CARMEN SOSA	Gangnam Style	Mis videos favoritos	2021-05-03 00:00:00
	5	0	MARIA CARMEN SOSA	Despacito	Mis videos favoritos	2021-05-03 00:00:00
	5	0	MARIA CARMEN SOSA	Sorry	Mis videos favoritos	2021-05-03 00:00:00
	5	0	MARIA CARMEN SOSA	Tutorial de SQL 1	Mis videos favoritos	2021-05-03 00:00:00
	5	0	MARIA CARMEN SOSA	Eliminaciones Sudamericanas - Colombia vs Arge...	Mis videos favoritos	2021-05-03 00:00:00
	5	0	MARIA CARMEN SOSA	Thinking Out Loud	Mis videos favoritos	2021-05-03 00:00:00

-- 4 Listar las 10 etiquetas menos usadas.

```
SELECT etiqueta.nombre, COUNT(video_etiqueta.Etiqueta_idEtiqueta) AS cantidad_usadas
FROM video_etiqueta
JOIN etiqueta ON etiqueta.idetiqueta = video_etiqueta.etiqueta_idetiqueta
GROUP BY etiqueta.nombre
ORDER BY cantidad_usadas
LIMIT 10;
```

```
22 -- 4
23 • SELECT etiqueta.nombre, COUNT(video_etiqueta.Etiqueta_idEtiqueta) AS cantidad_usadas
24 FROM video_etiqueta
25 JOIN etiqueta ON etiqueta.idetiqueta = video_etiqueta.etiqueta_idetiqueta
26 GROUP BY etiqueta.nombre
27 ORDER BY cantidad_usadas
28 LIMIT 10;
```

Result Grid		
Filter Rows: <input type="text"/> Export:  Wrap Cell Content: <input type="checkbox"/>		
	nombre	cantidad_usadas
▶	Shape Of You Lyrics	1
	Ed Sheerin	1
	getting started with python	1
	python crash course	1
	Shape Of You	1
	Ed Sheeran Official Video	1
	Mark Ronson essentials	1
	featuring Bruno Mars	1
	python course	1
	Ed Sheeran Shape Of You	1

-- 5 Generar un reporte de las últimas 10 reacciones, listando la fecha, el nombre de la reacción, el id y nombre de usuario, y el título del video.

```
SELECT idReaccion, fecha, tiporeaccion.nombre, usuario.idUsuario, usuario.nombre, titulo
FROM reaccion
INNER JOIN tiporeaccion ON reaccion.TipoReaccion_idTipoReaccion = tiporeaccion.idTipoReaccion
INNER JOIN usuario ON usuario.idUsuario = reaccion.Usuario_idUsuario
INNER JOIN video ON reaccion.Video_idVideo = video.idVideo
ORDER BY fecha DESC
LIMIT 10;
```

```
33 -- 5 Generar un reporte de las últimas 10 reacciones, listando la fecha, el nombre de la reacción, el id y nom
34 • SELECT idReaccion, fecha, tiporeaccion.nombre, usuario.idUsuario, usuario.nombre, titulo
35 FROM reaccion
36 INNER JOIN tiporeaccion ON reaccion.TipoReaccion_idTipoReaccion = tiporeaccion.idTipoReaccion
37 INNER JOIN usuario ON usuario.idUsuario = reaccion.Usuario_idUsuario
38 INNER JOIN video ON reaccion.Video_idVideo = video.idVideo
39 ORDER BY fecha DESC
40 LIMIT 10;
```

idReaccion	fecha	nombre	idUsuario	nombre	titulo
32	2021-05-29 00:00:00	DisLike	1	MARIA ORTIZ	Eliminaciones Sudamericanas - Colombia vs Arge...
15	2021-05-29 00:00:00	Like	13	JUAN SANCHEZ	Dark Horse
12	2021-05-25 00:00:00	DisLike	16	JUAN SEBASTIAN LOPEZ	Uptown Funk
58	2021-05-24 00:00:00	DisLike	19	JHON WATSON	Sorry
63	2021-05-24 00:00:00	Like	1	MARIA ORTIZ	Tutorial de SQL 2
8	2021-05-23 00:00:00	Like	10	ANTONIO GARCIA	Counting Stars
75	2021-05-22 00:00:00	DisLike	10	ANTONIO GARCIA	Programacion front end
47	2021-05-22 00:00:00	DisLike	17	SANTIAGO MARTINEZ	Dark Horse
4	2021-05-18 00:00:00	DisLike	8	MARY SMITH	Shake It Off

– 6. Mostrar por usuario la cantidad de playlists que posee, pero solo para aquellos que tengan más de 1 playlist creada en el año 2021.

```
SELECT * FROM playlist
WHERE YEAR(fechaCreacion) >= 2021
ORDER BY fechaCreacion ASC;
```

```
1 -- 6. Mostrar por usuario la cantidad de playlists que posee, pero solo para aquellos que tengan más de 1 playlist creada en el año 21
2 • SELECT * FROM playlist
3 WHERE YEAR(fechaCreacion) >= 2021
4 ORDER BY fechaCreacion ASC;
5
```

idPlaylist	nombre	fechaCreacion	privado	Usuario_idUsuario
7	Mi top 10	2021-01-21 00:00:00	1	15
1	Para correr	2021-01-29 00:00:00	1	15
4	Las mejores videos	2021-02-14 00:00:00	0	9
9	Musica Latina	2021-03-26 00:00:00	0	10
3	Para Gym	2021-04-04 00:00:00	1	6
8	Lo mejor del pop	2021-04-30 00:00:00	0	14
6	Top 20	2021-05-02 00:00:00	0	5
5	Mis videos favoritos	2021-05-03 00:00:00	0	7
10	Rock	2021-05-21 00:00:00	1	11
2	Para Estudiar	2021-05-31 00:00:00	0	4
	TOTAL	TOTAL	TOTAL	TOTAL

– 7. Generar un reporte de las reacciones generadas en el 2021, con el siguiente formato :

```
SELECT tiporeaccion.nombre AS 'Nombre de Reaccion', usuario.nombre AS 'Nombre de Usuario',
video.titulo AS 'Titulo de Video', reaccion.fecha AS 'Fecha'
FROM video
INNER JOIN usuario ON video.Usuario_idUsuario = usuario.idUsuario
INNER JOIN reaccion ON usuario.idUsuario = reaccion.Usuario_idUsuario
INNER JOIN tiporeaccion ON reaccion.TipoReaccion_idTipoReaccion = tiporeaccion.idTipoReaccion
WHERE YEAR(fecha) >= 2021
ORDER BY fecha ASC;
```

```
6 /* 7. Generar un reporte de las reacciones generadas en el 2021, con el siguiente formato :*/
7 • SELECT tiporeaccion.nombre AS 'Nombre de Reaccion', usuario.nombre AS 'Nombre de Usuario', video.titulo AS 'Titulo de Video', reaccion.fecha AS 'Fecha'
8 FROM video
9 INNER JOIN usuario ON video.Usuario_idUsuario = usuario.idUsuario
10 INNER JOIN reaccion ON usuario.idUsuario = reaccion.Usuario_idUsuario
11 INNER JOIN tiporeaccion ON reaccion.TipoReaccion_idTipoReaccion = tiporeaccion.idTipoReaccion
12 WHERE YEAR(fecha) >= 2021
13 ORDER BY fecha ASC;
```

Nombre de Reaccion	Nombre de Usuario	Titulo de Video	Fecha
DisLike	LUISA GOMEZ	Tutorial de SQL 3	2021-01-08 00:00:00
DisLike	JOSE MARTINEZ	Programacion back end	2021-01-09 00:00:00
DisLike	JOSE MARTINEZ	See You Again	2021-01-09 00:00:00
DisLike	MARIA CARMEN SOSA	Eliminaciones Sudamericanas - Colombia vs Arge...	2021-01-13 00:00:00
DisLike	JOSE MARTINEZ	Programacion back end	2021-01-14 00:00:00
DisLike	JOSE MARTINEZ	See You Again	2021-01-14 00:00:00
DisLike	JUAN SANCHEZ	Thinking Out Loud	2021-01-16 00:00:00
Like	JUAN ESTEBAN DIAZ	Python for beginners	2021-01-18 00:00:00
Like	JUAN ESTEBAN DIAZ	Sorry	2021-01-18 00:00:00
Like	JUAN ESTEBAN DIAZ	Despacito	2021-01-18 00:00:00
Like	CARMEN RAMIREZ	Shake It Off	2021-01-19 00:00:00
Like	CARMEN RAMIREZ	Shake It Off	2021-01-19 00:00:00

8. Listar la cantidad de videos según sean públicos o privados.

```
SELECT COUNT(*) AS 'Cantidad de Videos',
CASE
WHEN privado = 1 THEN 'Privado'
WHEN privado = 0 THEN 'Publico'
END AS Visibilidad
FROM video
GROUP BY privado;
```

```
14
15 -- 8. Listar la cantidad de videos según sean públicos o privados.
16 • SELECT count(*) AS 'Cantidad de Videos',
17 CASE
18 WHEN privado = 1 THEN 'Privado'
19 WHEN privado = 0 THEN 'Publico'
20 END AS Visibilidad
21 FROM video
22 GROUP BY privado;
```

Cantidad de Videos	Visibilidad
12	Privado
8	Publico

## 9. Generar un reporte con los nombres de los usuario que no poseen ninguna Playlist.

```

SELECT usuario.nombre, playlist.idplaylist
FROM usuario
LEFT JOIN playlist ON usuario.idUsuario = playlist.Usuario_idUsuario
WHERE playlist.Usuario_idUsuario IS NULL;
--
27 -- 9. Generar un reporte con los nombres de los usuario que no poseen ninguna Playlist.
28 • SELECT usuario.nombre, playlist.idplaylist
29 FROM usuario
30 LEFT JOIN playlist ON usuario.idUsuario = playlist.Usuario_idUsuario
31 WHERE playlist.Usuario_idUsuario IS NULL;
32

```

nombre	idplaylist
MARIA ORTIZ	NULL
ISABEL BALLESTEROS	NULL
CARMEN RAMIREZ	NULL
MARY SMITH	NULL
FRANCISCO LOPEZ	NULL
JUAN SANCHEZ	NULL
JUAN SEBASTIAN LOPEZ	NULL
SANTIAGO MARTINEZ	NULL
DAVID RUBIO	NULL
JHON WATSON	NULL
Juan Jose Batzal	NULL

## 10. Listar todos los videos hayan o no recibido reacciones en el último mes; indicar cuántas reacciones tuvieron y si han sido reproducidos o no. El listado debe estar ordenado alfabéticamente por título del video.

```

SELECT v.*,
COUNT(r.idReaccion) AS Cantidad_Reacciones,
CASE
WHEN LENGTH(v.cantidadReproducciones) > 0 THEN 'Si'
ELSE 'No'
END AS Reproducido
FROM video v
LEFT JOIN reaccion r ON r.video_idvideo = v.idvideo
WHERE MONTH(r.fecha) = 01
GROUP BY v.idVideo
ORDER BY v.titulo;

```

```

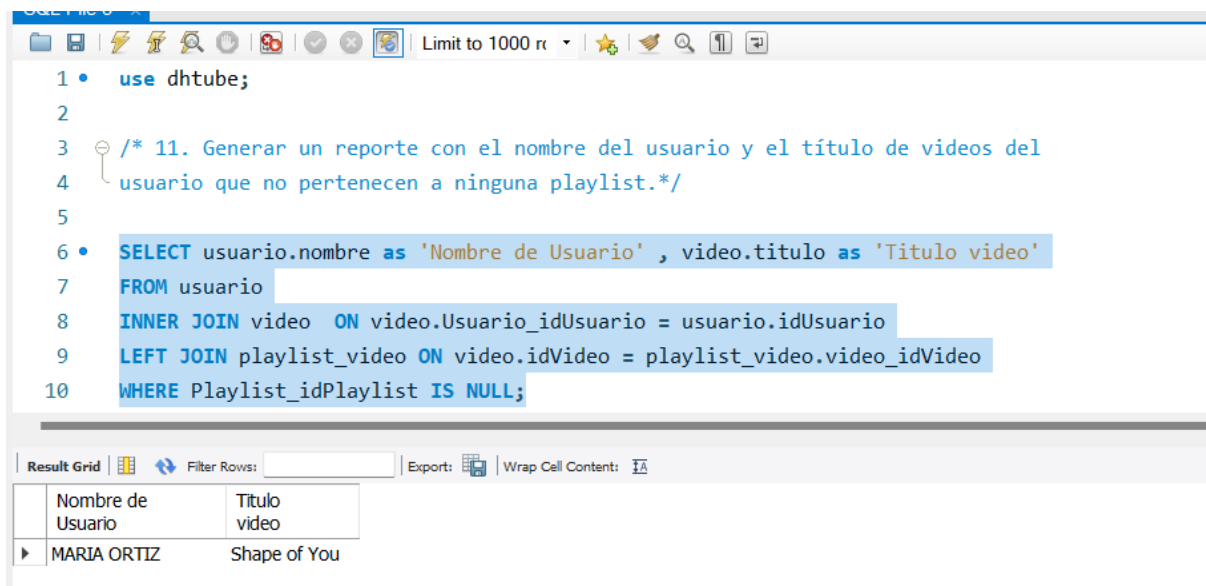
77 -- 10. Listar todos los videos hayan o no recibido reacciones en el último mes; indicar cuántas reacciones tuvieron y si han sido reproducidos o no. El listado debe
78 • SELECT v.*,
79 COUNT(r.idReaccion) AS Cantidad_Reacciones,
80 CASE
81 WHEN LENGTH(v.cantidadReproducciones) > 0 THEN 'Si'
82 ELSE 'No'
83 END AS Reproducido
84 FROM video v
85 LEFT JOIN reaccion r ON r.video_idvideo = v.idvideo
86 WHERE MONTH(r.fecha) = 01
87 GROUP BY v.idVideo
88 ORDER BY v.titulo;

```

idVideo	Usuario_idUsuario	titulo	descripcion	tamario	nombreArchivo	duracion	imagen
5	9	Counting Stars	Counting Stars by OneRepublic	614438	countingstars.mp4	839763	https://www.google.com,
20	7	Eliminaciones Sudamericanas - Colombia vs Arge...	Eliminaciones Sudamericanas - Colombia vs Arge...	544447	eliminaciones sudamericanas colombia vs argentin...	574099	https://www.google.com,
2	4	Faded	Faded by Alan Walker	589666	faded.mp4	696245	https://www.google.com,
19	11	Programacion back end	Programacion back end	518272	programacionbackend.mp4	609856	https://www.google.com,
18	2	Programacion front end	Programacion front end	815173	programacionfrontend.mp4	846190	https://www.google.com,

11. Generar un reporte con el nombre del usuario y el título de videos del usuario que no pertenecen a ninguna playlist.

```
SELECT usuario.nombre AS 'Nombre de Usuario' , video.titulo AS 'Titulo video'
FROM usuario
INNER JOIN video ON video.Usuario_idUsuario = usuario.idUsuario
LEFT JOIN playlist_video ON video.idVideo = playlist_video.video_idVideo
WHERE Playlist_idPlaylist IS NULL;
```



The screenshot shows a SQL IDE window with a query editor and a result grid. The query editor contains the following SQL code:

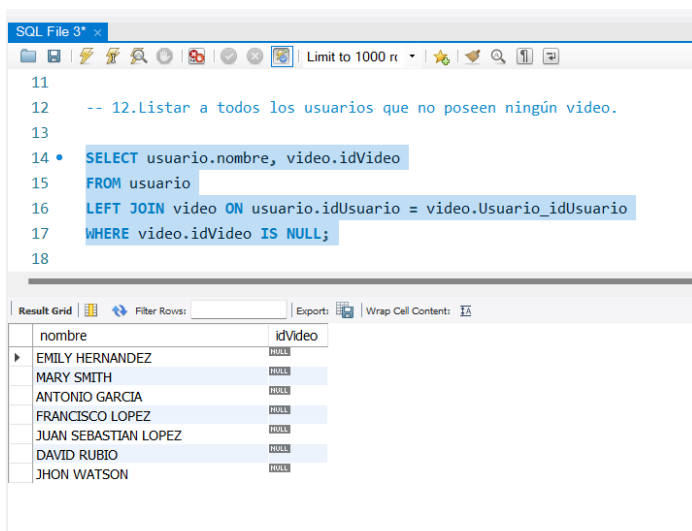
```
1 • use dhtube;
2
3 /* 11. Generar un reporte con el nombre del usuario y el título de videos del
4 usuario que no pertenecen a ninguna playlist.*/
5
6 • SELECT usuario.nombre as 'Nombre de Usuario' , video.titulo as 'Titulo video'
7 FROM usuario
8 INNER JOIN video ON video.Usuario_idUsuario = usuario.idUsuario
9 LEFT JOIN playlist_video ON video.idVideo = playlist_video.video_idVideo
10 WHERE Playlist_idPlaylist IS NULL;
```

The result grid shows the following data:

Nombre de Usuario	Titulo video
MARIA ORTIZ	Shape of You

12. Listar a todos los usuarios que no poseen ningún video.

```
SELECT usuario.nombre, video.idVideo
FROM usuario
LEFT JOIN video ON usuario.idUsuario = video.Usuario_idUsuario
WHERE video.idVideo IS NULL;
```



The screenshot shows a SQL IDE window with a query editor and a result grid. The query editor contains the following SQL code:

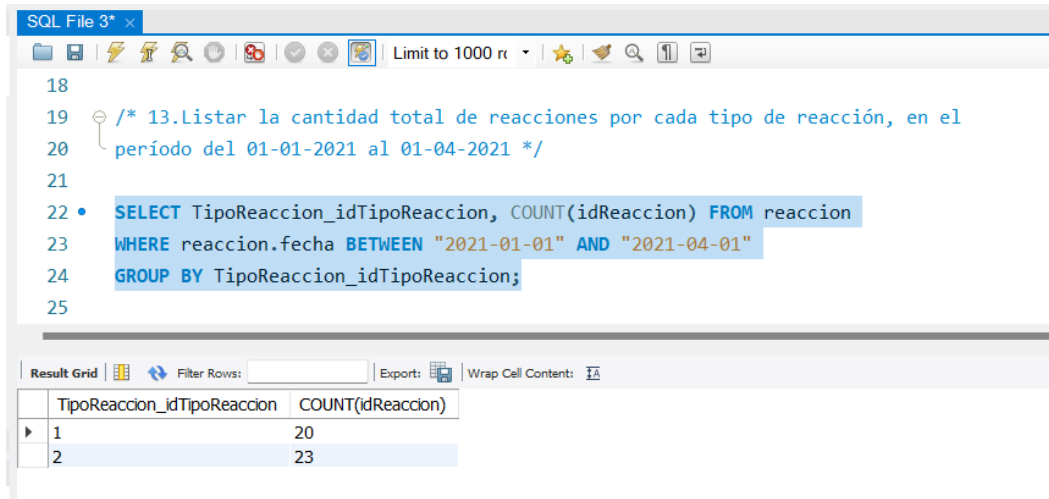
```
11
12 -- 12. Listar a todos los usuarios que no poseen ningún video.
13
14 • SELECT usuario.nombre, video.idVideo
15 FROM usuario
16 LEFT JOIN video ON usuario.idUsuario = video.Usuario_idUsuario
17 WHERE video.idVideo IS NULL;
18
```

The result grid shows the following data:

nombre	idVideo
EMILY HERNANDEZ	NULL
MARY SMITH	NULL
ANTONIO GARCIA	NULL
FRANCISCO LOPEZ	NULL
JUAN SEBASTIAN LOPEZ	NULL
DAVID RUBIO	NULL
JHON WATSON	NULL

13. Listar la cantidad total de reacciones por cada tipo de reacción, en el período del 01-01-2021 al 01-04-2021

```
SELECT TipoReaccion_idTipoReaccion, COUNT(idReaccion) FROM reaccion
WHERE reaccion.fecha BETWEEN "2021-01-01" AND "2021-04-01"
GROUP BY TipoReaccion_idTipoReaccion;
```



The screenshot shows a SQL editor window titled "SQL File 3\*" with a toolbar and a "Limit to 1000 rows" dropdown. The SQL query is as follows:

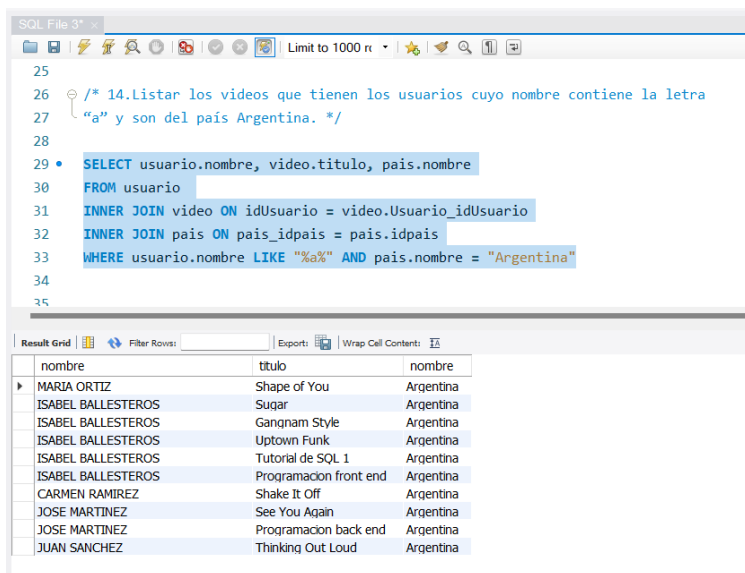
```
18
19 /* 13. Listar la cantidad total de reacciones por cada tipo de reacción, en el
20 período del 01-01-2021 al 01-04-2021 */
21
22 • SELECT TipoReaccion_idTipoReaccion, COUNT(idReaccion) FROM reaccion
23 WHERE reaccion.fecha BETWEEN "2021-01-01" AND "2021-04-01"
24 GROUP BY TipoReaccion_idTipoReaccion;
25
```

Below the query, the "Result Grid" is displayed with the following data:

TipoReaccion_idTipoReaccion	COUNT(idReaccion)
1	20
2	23

14. Listar los videos que tienen los usuarios cuyo nombre contiene la letra "a" y son del país Argentina.

```
SELECT usuario.nombre, video.titulo, pais.nombre
FROM usuario
INNER JOIN video ON idUsuario = video.Usuario_idUsuario
INNER JOIN pais ON pais_idpais = pais.idpais
WHERE usuario.nombre LIKE "%a%" AND pais.nombre = "Argentina"
```



The screenshot shows a SQL editor window titled "SQL File 3\*" with a toolbar and a "Limit to 1000 rows" dropdown. The SQL query is as follows:

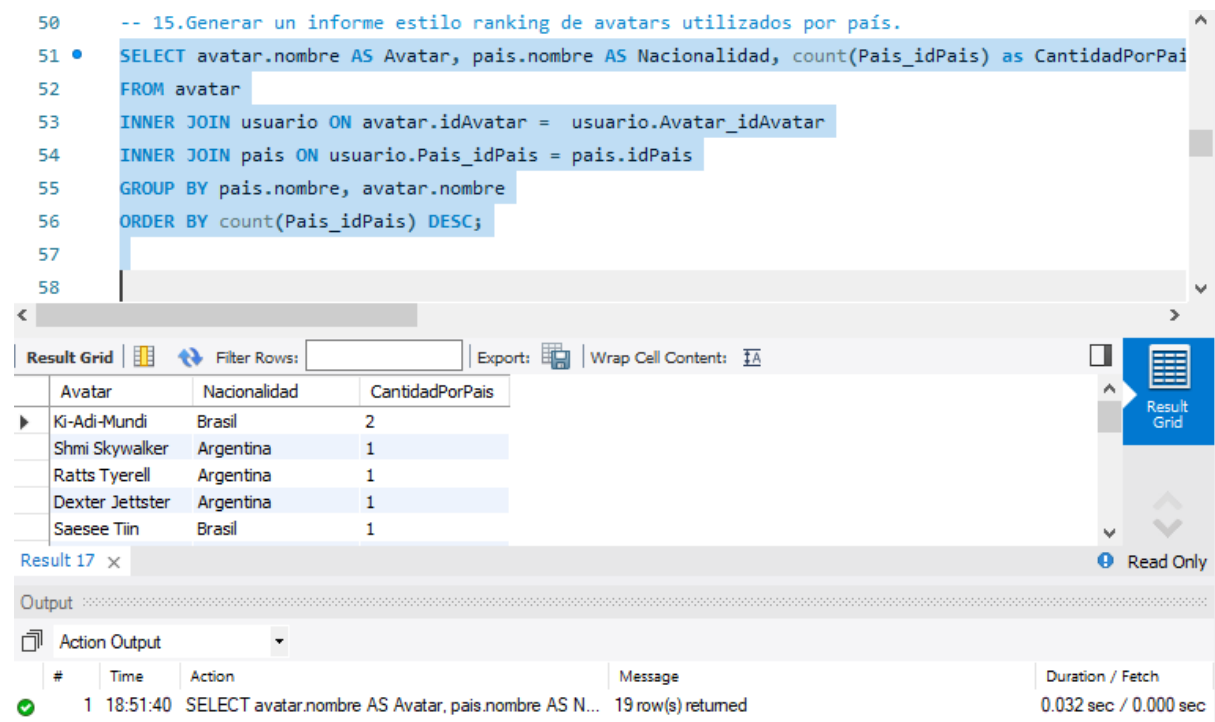
```
25
26 /* 14. Listar los videos que tienen los usuarios cuyo nombre contiene la letra
27 "a" y son del país Argentina. */
28
29 • SELECT usuario.nombre, video.titulo, pais.nombre
30 FROM usuario
31 INNER JOIN video ON idUsuario = video.Usuario_idUsuario
32 INNER JOIN pais ON pais_idpais = pais.idpais
33 WHERE usuario.nombre LIKE "%a%" AND pais.nombre = "Argentina"
34
35
```

Below the query, the "Result Grid" is displayed with the following data:

nombre	titulo	nombre
MARIA ORTIZ	Shape of You	Argentina
ISABEL BALLESTEROS	Sugar	Argentina
ISABEL BALLESTEROS	Gangnam Style	Argentina
ISABEL BALLESTEROS	Uptown Funk	Argentina
ISABEL BALLESTEROS	Tutorial de SQL 1	Argentina
ISABEL BALLESTEROS	Programacion front end	Argentina
CARMEN RAMIREZ	Shake It Off	Argentina
JOSE MARTINEZ	See You Again	Argentina
JOSE MARTINEZ	Programacion back end	Argentina
JUAN SANCHEZ	Thinking Out Loud	Argentina

### 15. Generar un informe estilo ranking de avatars utilizados por país.

```
SELECT avatar.nombre AS Avatar, pais.nombre AS Nacionalidad, COUNT(Pais_idPais) AS
CantidadPorPais
FROM avatar
INNER JOIN usuario ON avatar.idAvatar = usuario.Avatar_idAvatar
INNER JOIN pais ON usuario.Pais_idPais = pais.idPais
GROUP BY pais.nombre, avatar.nombre
ORDER BY count(Pais_idPais) DESC;
```



```
50 -- 15. Generar un informe estilo ranking de avatars utilizados por país.
51 • SELECT avatar.nombre AS Avatar, pais.nombre AS Nacionalidad, count(Pais_idPais) as CantidadPorPais
52 FROM avatar
53 INNER JOIN usuario ON avatar.idAvatar = usuario.Avatar_idAvatar
54 INNER JOIN pais ON usuario.Pais_idPais = pais.idPais
55 GROUP BY pais.nombre, avatar.nombre
56 ORDER BY count(Pais_idPais) DESC;
57
58
```

Avatar	Nacionalidad	CantidadPorPais
Ki-Adi-Mundi	Brasil	2
Shmi Skywalker	Argentina	1
Ratts Tyerell	Argentina	1
Dexter Jettster	Argentina	1
Saesee Tiin	Brasil	1

Result 17 x

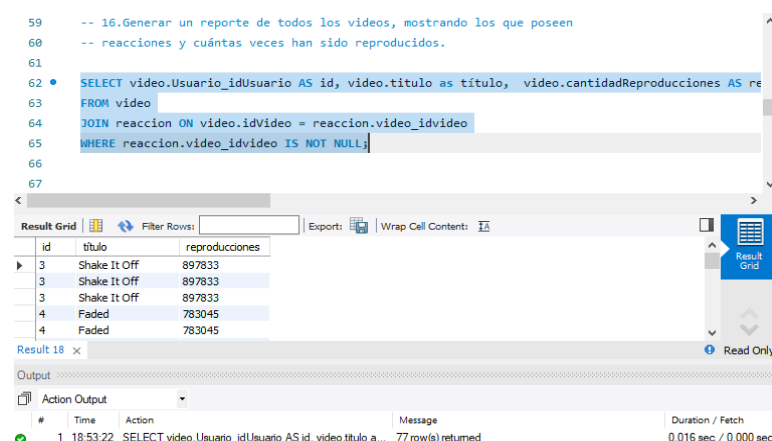
Output

Action Output

#	Time	Action	Message	Duration / Fetch
1	18:51:40	SELECT avatar.nombre AS Avatar, pais.nombre AS N...	19 row(s) returned	0.032 sec / 0.000 sec

### 16. Generar un reporte de todos los videos, mostrando los que poseen reacciones y cuántas veces han sido reproducidos.

```
SELECT video.Usuario_idUsuario AS id, video.titulo as título, video.cantidadReproducciones AS re
reproducciones
FROM video
JOIN reaccion ON video.idVideo = reaccion.video_idvideo
WHERE reaccion.video_idvideo IS NOT NULL;
```



```
59 -- 16. Generar un reporte de todos los videos, mostrando los que poseen
60 -- reacciones y cuántas veces han sido reproducidos.
61
62 • SELECT video.Usuario_idUsuario AS id, video.titulo as título, video.cantidadReproducciones AS re
63 FROM video
64 JOIN reaccion ON video.idVideo = reaccion.video_idvideo
65 WHERE reaccion.video_idvideo IS NOT NULL;
66
67
```

id	título	reproducciones
3	Shake It Off	897833
3	Shake It Off	897833
3	Shake It Off	897833
4	Faded	783045
4	Faded	783045

Result 18 x

Output

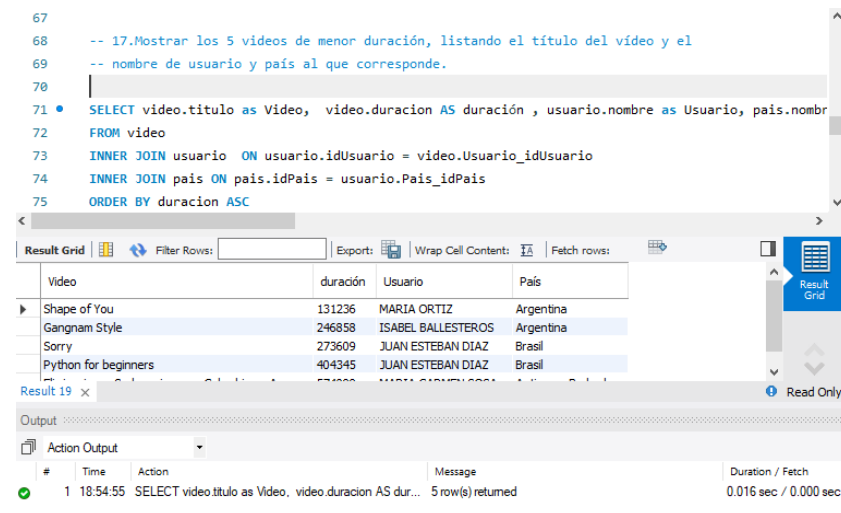
Action Output

#	Time	Action	Message	Duration / Fetch
1	18:53:22	SELECT video.Usuario_idUsuario AS id, video.titulo a...	77 row(s) returned	0.016 sec / 0.000 sec



**17. Mostrar los 5 videos de menor duración, listando el título del vídeo y el nombre de usuario y país al que corresponde.**

```
SELECT video.titulo AS Video, video.duracion AS duración , usuario.nombre AS Usuario,
pais.nombre AS País
FROM video
INNER JOIN usuario ON usuario.idUsuario = video.Usuario_idUsuario
INNER JOIN pais ON pais.idPais = usuario.Pais_idPais
ORDER BY duracion ASC;
```



```
67
68 -- 17. Mostrar los 5 videos de menor duración, listando el título del vídeo y el
69 -- nombre de usuario y país al que corresponde.
70
71 • SELECT video.titulo as Video, video.duracion AS duración , usuario.nombre as Usuario, pais.nombre
72 FROM video
73 INNER JOIN usuario ON usuario.idUsuario = video.Usuario_idUsuario
74 INNER JOIN pais ON pais.idPais = usuario.Pais_idPais
75 ORDER BY duracion ASC
```

Video	duración	Usuario	País
Shape of You	131236	MARIA ORTIZ	Argentina
Gangnam Style	246858	ISABEL BALLESTEROS	Argentina
Sorry	273609	JUAN ESTEBAN DIAZ	Brasil
Python for beginners	404345	JUAN ESTEBAN DIAZ	Brasil

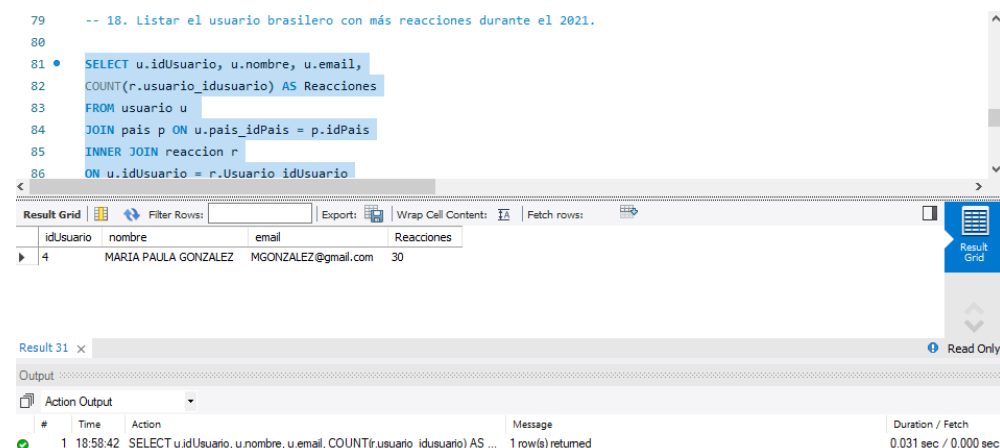
Result 19 x Read Only

Output

#	Time	Action	Message	Duration / Fetch
1	18:54:55	SELECT video.titulo as Video, video.duracion AS dur...	5 row(s) returned	0.016 sec / 0.000 sec

**18. Listar el usuario brasilero con más reacciones durante el 2021.**

```
SELECT u.idUsuario, u.nombre, u.email,
COUNT(r.usuario_idusuario) AS Reacciones
FROM usuario u
JOIN pais p ON u.pais_idPais = p.idPais
INNER JOIN reaccion r
ON u.idUsuario = r.Usuario_idUsuario
WHERE p.nombre = "Brasil"
ORDER BY reacciones
LIMIT 1;
```



```
79 -- 18. Listar el usuario brasilero con más reacciones durante el 2021.
80
81 • SELECT u.idUsuario, u.nombre, u.email,
82 COUNT(r.usuario_idusuario) AS Reacciones
83 FROM usuario u
84 JOIN pais p ON u.pais_idPais = p.idPais
85 INNER JOIN reaccion r
86 ON u.idUsuario = r.Usuario_idUsuario
```

idUsuario	nombre	email	Reacciones
4	MARIA PAULA GONZALEZ	MGONZALEZ@gmail.com	30

Result 31 x Read Only

Output

#	Time	Action	Message	Duration / Fetch
1	18:58:42	SELECT u.idUsuario, u.nombre, u.email, COUNT(r.usuario_idusuario) AS ...	1 row(s) returned	0.031 sec / 0.000 sec

## 19. Generar un reporte listando los usuarios, sus canales, playlists y los videos que integran esas playlists.

```
SELECT usuario.nombre AS Usuario , canal.nombre AS canal, playlist.nombre AS playlist ,
video.titulo as video
FROM usuario
INNER JOIN canal ON usuario.idUsuario = canal.Usuario_idUsuario
INNER JOIN playlist ON playlist.Usuario_idUsuario = usuario.idUsuario
INNER JOIN playlist_video ON playlist_video.Playlist_idPlaylist = playlist.idPlaylist
INNER JOIN video ON playlist_video.Video_idVideo = video.idVideo;
```

The screenshot shows a SQL IDE with a query editor and a results pane. The query is as follows:

```
-- 19. Generar un reporte listando los usuarios, sus canales, playlists y los
-- videos que integran esas playlists.
SELECT usuario.nombre as Usuario , canal.nombre as canal, playlist.nombre as playlist , video.titulo as video
FROM usuario
INNER JOIN canal ON usuario.idUsuario = canal.Usuario_idUsuario
INNER JOIN playlist ON playlist.Usuario_idUsuario = usuario.idUsuario
INNER JOIN playlist_video ON playlist_video.Playlist_idPlaylist = playlist.idPlaylist
INNER JOIN video ON playlist_video.Video_idVideo = video.idVideo;
```

The results pane displays a table with 4 columns: Usuario, canal, playlist, and video. The data is as follows:

Usuario	canal	playlist	video
EMILY HERNANDEZ	Lo Mejor de la musica	Top 20	Sugar
EMILY HERNANDEZ	Lo Mejor de la musica	Top 20	Sugar
EMILY HERNANDEZ	Lo Mejor de la musica	Top 20	Tutorial de SQL 1
EMILY HERNANDEZ	Lo Mejor de la musica	Top 20	Dark Horse
EMILY HERNANDEZ	Lo Mejor de la musica	Top 20	See You Again

The results pane also shows a message: "SELECT usuario.nombre as Usuario , canal.nombre as canal, playlist.nombre as playlist, video.titulo as video" and a duration of 0.000 sec / 0.000 sec.

## 20. Listar todas las playlists que posean menos de 3 videos, mostrando el nombre de usuario y el avatar que posee.

```
SELECT COUNT(playlist_video.Video_idVideo) AS Cantidad_videos, playlist.nombre, avatar.nombre
FROM playlist
INNER JOIN usuario ON usuario.idUsuario = playlist.Usuario_idUsuario
INNER JOIN avatar ON avatar.idAvatar = usuario.Avatar_idAvatar
INNER JOIN playlist_video ON playlist.idPlaylist = playlist_video.Playlist_idPlaylist
INNER JOIN video ON playlist_video.Video_idVideo = video.idVideo
GROUP BY playlist.idPlaylist
HAVING COUNT(playlist_video.Video_idVideo) < 3;
```

The screenshot shows a SQL IDE with a query editor and a results pane. The query is as follows:

```
SELECT count(playlist_video.Video_idVideo) AS Cantidad_videos, playlist.nombre, avatar.nombre FROM playlist
JOIN usuario ON usuario.idUsuario = playlist.Usuario_idUsuario
JOIN avatar ON avatar.idAvatar = usuario.Avatar_idAvatar
JOIN playlist_video ON playlist.idPlaylist = playlist_video.Playlist_idPlaylist
JOIN video ON playlist_video.Video_idVideo = video.idVideo
GROUP BY playlist.idPlaylist
HAVING count(playlist_video.Video_idVideo) < 3;
```

The results pane displays a table with 3 columns: Cantidad\_videos, nombre, and nombre. The data is as follows:

Cantidad_videos	nombre	nombre
-----------------	--------	--------