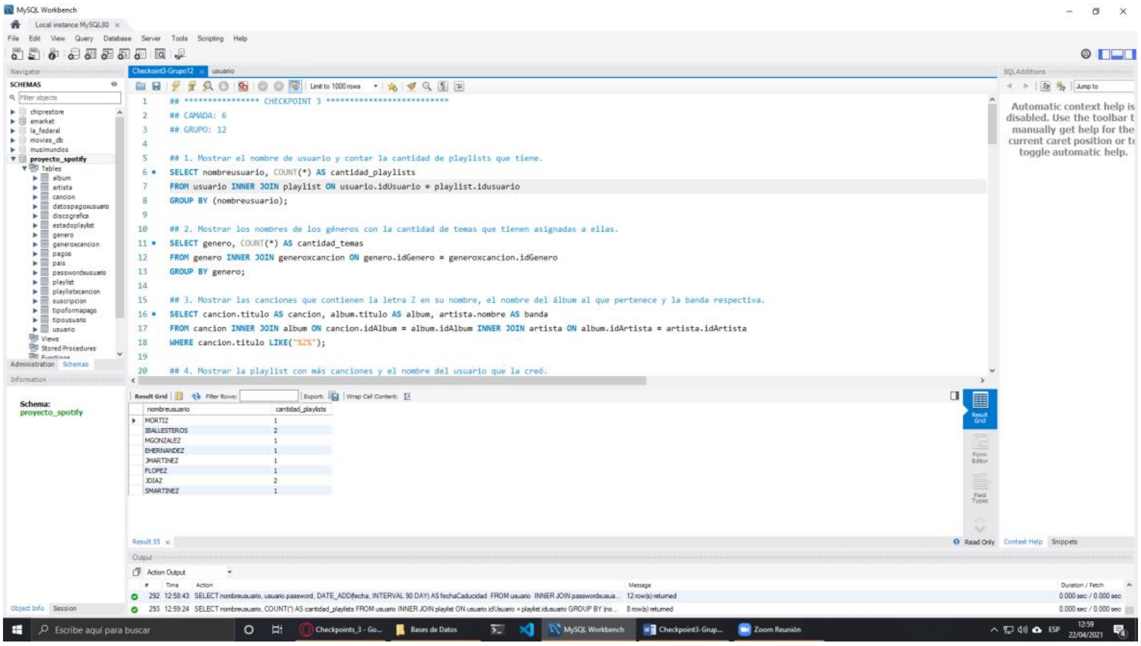
1)

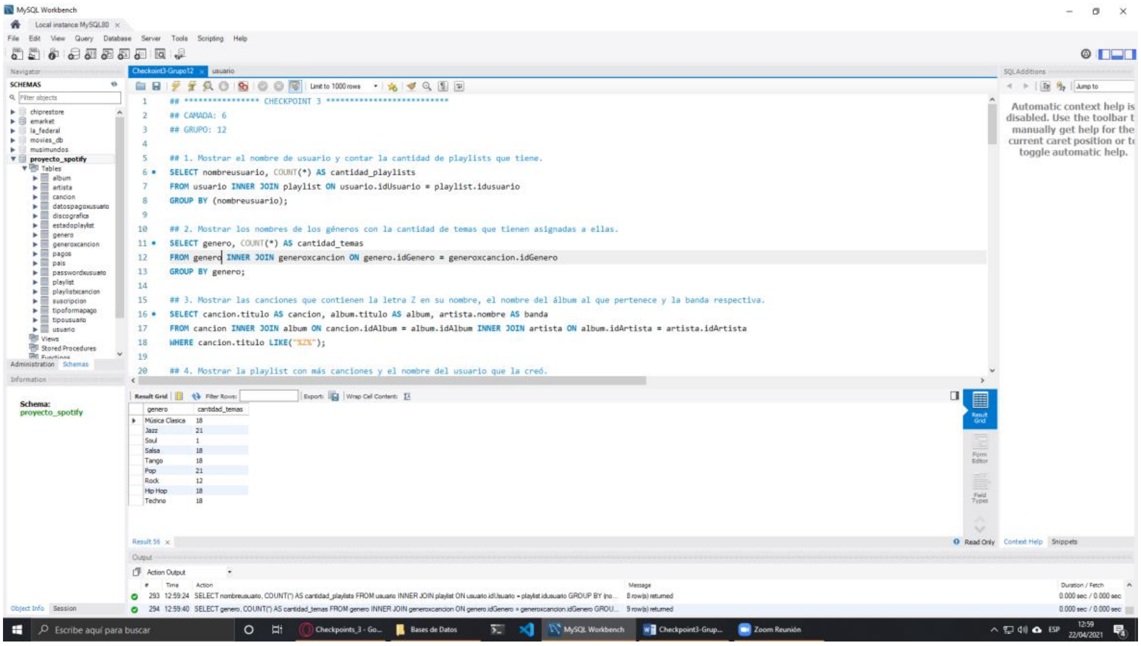


```sql
## *************** CHECKPOINT 3 ************************
## CAMADA: 6
## GRUPO: 12

## 1. Mostrar el nombre de usuario y contar la cantidad de playlists que tiene.
SELECT nombreusuario, COUNT(*) AS cantidad_playlists
FROM usuario INNER JOIN playlist ON usuario.idUsuario = playlist.idusuario
GROUP BY (nombreusuario);

## 2. Mostrar los nombres de los géneros con la cantidad de temas que tienen asignadas a ellas.
SELECT genero, COUNT(*) AS cantidad_temas
FROM genero INNER JOIN generoxcancion ON genero.idGenero = generoxcancion.idGenero
GROUP BY genero;

## 3. Mostrar las canciones que contienen la letra Z en su nombre, el nombre del álbum al que pertenece y la banda respectiva.
SELECT cancion.titulo AS cancion, album.titulo AS album, artista.nombre AS banda
FROM cancion INNER JOIN album ON cancion.idAlbum = album.idAlbum INNER JOIN artista ON album.idArtista = artista.idArtista
WHERE cancion.titulo LIKE("%Z%");

## 4. Mostrar la playlist con más canciones y el nombre del usuario que la creó.
```

nombreusuario	cantidad_playlists
MORTIZ	1
IBALLESTEROS	2
MGONZALEZ	1
EHERNANDEZ	1
JMARTINEZ	1
FLOPEZ	1
JDIAZ	2
SMARTINEZ	1

```
Output
Action Output
#    Time      Action                                                                                                             Message              Duration / Fetch
292  12:58:43  SELECT nombreusuario, usuario.password, DATE_ADD(fecha, INTERVAL 90 DAY) AS fechaCaducidad  FROM usuario  INNER JOIN passwordxusua...  12 row(s) returned  0.000 sec / 0.000 sec
293  12:59:24  SELECT nombreusuario, COUNT(*) AS cantidad_playlists FROM usuario INNER JOIN playlist ON usuario.idUsuario = playlist.idusuario GROUP BY (no...  8 row(s) returned  0.000 sec / 0.000 sec
```

2)

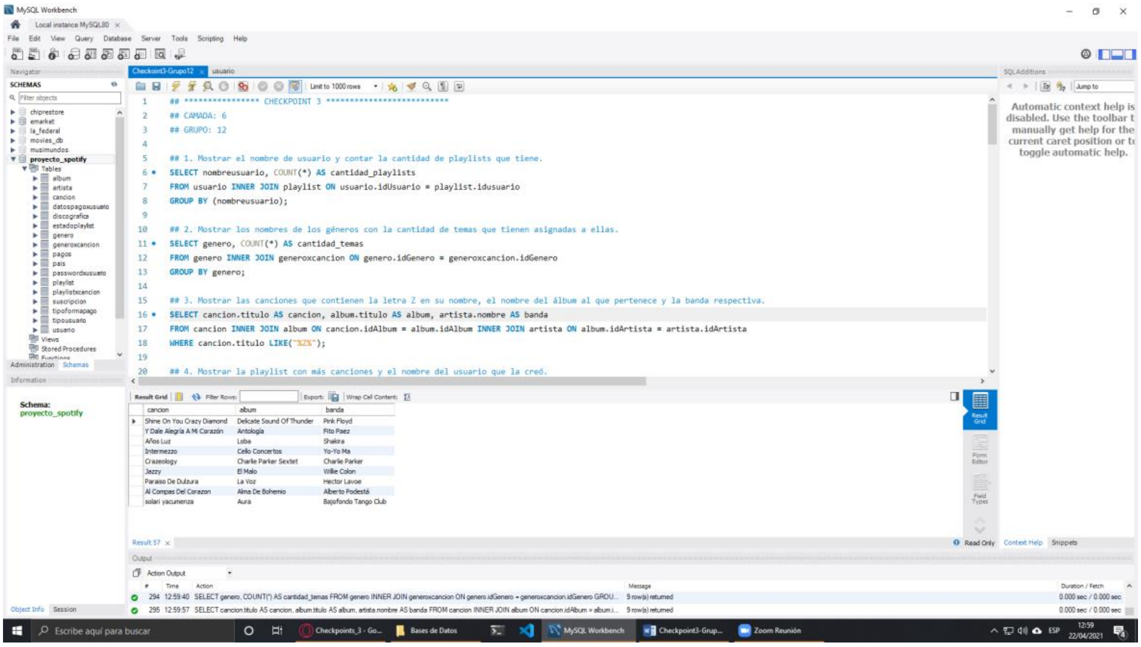


```sql
## *************** CHECKPOINT 3 ************************
## CAMADA: 6
## GRUPO: 12

## 1. Mostrar el nombre de usuario y contar la cantidad de playlists que tiene.
SELECT nombreusuario, COUNT(*) AS cantidad_playlists
FROM usuario INNER JOIN playlist ON usuario.idUsuario = playlist.idusuario
GROUP BY (nombreusuario);

## 2. Mostrar los nombres de los géneros con la cantidad de temas que tienen asignadas a ellas.
SELECT genero, COUNT(*) AS cantidad_temas
FROM genero INNER JOIN generoxcancion ON genero.idGenero = generoxcancion.idGenero
GROUP BY genero;

## 3. Mostrar las canciones que contienen la letra Z en su nombre, el nombre del álbum al que pertenece y la banda respectiva.
SELECT cancion.titulo AS cancion, album.titulo AS album, artista.nombre AS banda
FROM cancion INNER JOIN album ON cancion.idAlbum = album.idAlbum INNER JOIN artista ON album.idArtista = artista.idArtista
WHERE cancion.titulo LIKE("%Z%");

## 4. Mostrar la playlist con más canciones y el nombre del usuario que la creó.
```

genero	cantidad_temas
Música Clasica	18
Jazz	21
Soul	1
Salsa	18
Tango	18
Pop	21
Rock	12
Hip Hop	18
Techno	18

```
Output
Action Output
#    Time      Action                                                                                                             Message              Duration / Fetch
293  12:59:24  SELECT nombreusuario, COUNT(*) AS cantidad_playlists FROM usuario INNER JOIN playlist ON usuario.idUsuario = playlist.idusuario GROUP BY (no...  8 row(s) returned  0.000 sec / 0.000 sec
294  12:59:40  SELECT genero, COUNT(*) AS cantidad_temas FROM genero INNER JOIN generoxcancion ON genero.idGenero = generoxcancion.idGenero GROU...  9 row(s) returned  0.000 sec / 0.000 sec
```

3)



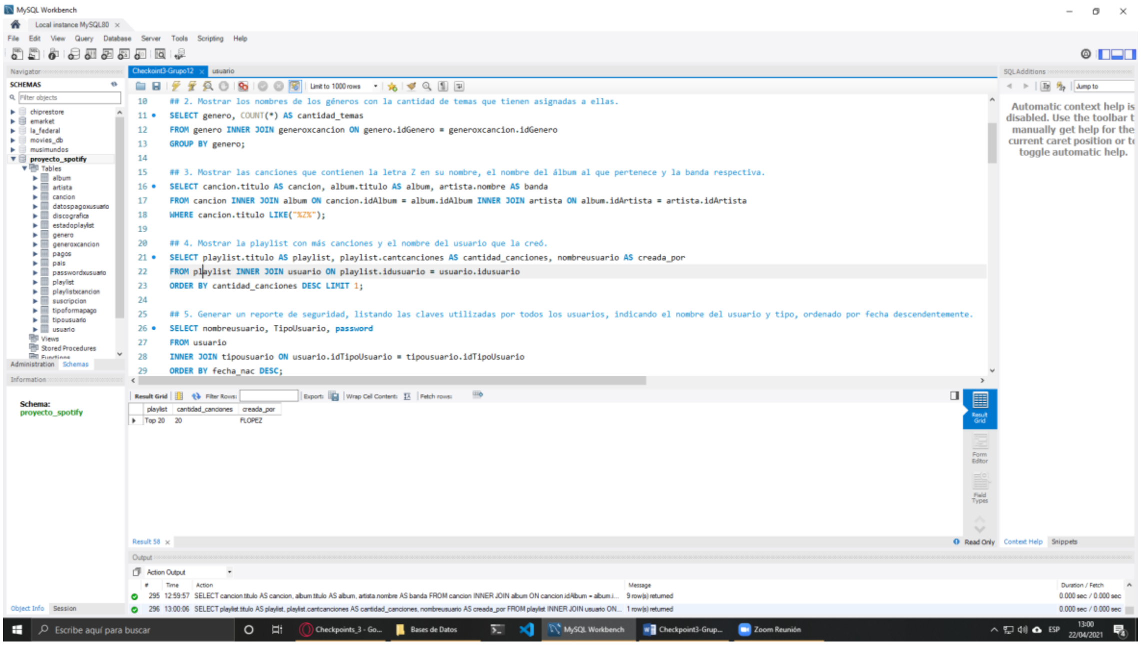
```sql
## **************** CHECKPOINT 3 *************************
## CAMADA: 6
## GRUPO: 12

## 1. Mostrar el nombre de usuario y contar la cantidad de playlists que tiene.
SELECT nombreusuario, COUNT(*) AS cantidad_playlists
FROM usuario INNER JOIN playlist ON usuario.idUsuario = playlist.idusuario
GROUP BY (nombreusuario);

## 2. Mostrar los nombres de los géneros con la cantidad de temas que tienen asignadas a ellas.
SELECT genero, COUNT(*) AS cantidad_temas
FROM genero INNER JOIN generoxcancion ON genero.idGenero = generoxcancion.idGenero
GROUP BY genero;

## 3. Mostrar las canciones que contienen la letra Z en su nombre, el nombre del álbum al que pertenece y la banda respectiva.
SELECT cancion.titulo AS cancion, album.titulo AS album, artista.nombre AS banda
FROM cancion INNER JOIN album ON cancion.idAlbum = album.idAlbum INNER JOIN artista ON album.idArtista = artista.idArtista
WHERE cancion.titulo LIKE("%Z%");

## 4. Mostrar la playlist con más canciones y el nombre del usuario que la creó.
```

cancion	album	banda
Shine On You Crazy Diamond	Delicate Sound Of Thunder	Pink Floyd
Y Dale Alegría A Mi Corazón	Antología	Fito Paez
Años Luz	Loba	Shakira
Intermezzo	Cello Concertos	Yo-Yo Ma
Crazeology	Charlie Parker Sextet	Charlie Parker
Jazzy	El Malo	Willie Colon
Paraiso De Dulzura	La Voz	Hector Lavoe
Al Compas Del Corazon	Alma De Bohemio	Alberto Podestá
solari yacumenza	Aura	Bajofondo Tango Club

Result 57

```
#    Time      Action                                                                                                   Message              Duration / Fetch
294  12:59:40  SELECT genero, COUNT(*) AS cantidad_temas FROM genero INNER JOIN generoxcancion ON genero.idGenero = generoxcancion.idGenero GROU...  9 row(s) returned  0.000 sec / 0.000 sec
295  12:59:57  SELECT cancion.titulo AS cancion, album.titulo AS album, artista.nombre AS banda FROM cancion INNER JOIN album ON cancion.idAlbum = album.i...  9 row(s) returned  0.000 sec / 0.000 sec
```

4)



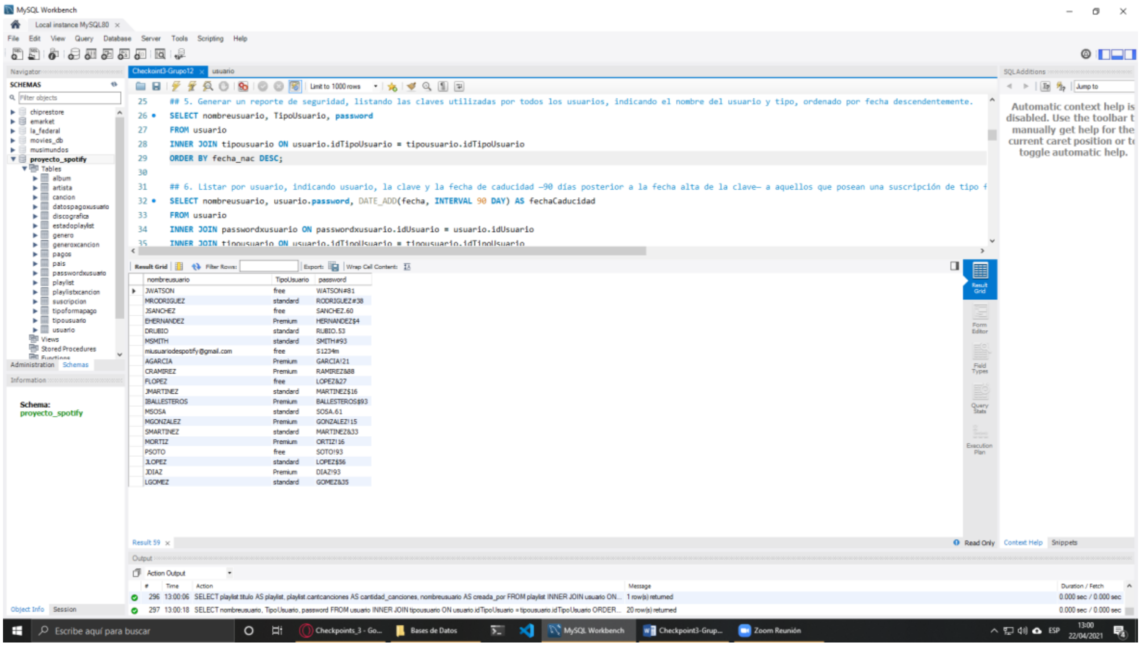
```sql
## 2. Mostrar los nombres de los géneros con la cantidad de temas que tienen asignadas a ellas.
SELECT genero, COUNT(*) AS cantidad_temas
FROM genero INNER JOIN generoxcancion ON genero.idGenero = generoxcancion.idGenero
GROUP BY genero;

## 3. Mostrar las canciones que contienen la letra Z en su nombre, el nombre del álbum al que pertenece y la banda respectiva.
SELECT cancion.titulo AS cancion, album.titulo AS album, artista.nombre AS banda
FROM cancion INNER JOIN album ON cancion.idAlbum = album.idAlbum INNER JOIN artista ON album.idArtista = artista.idArtista
WHERE cancion.titulo LIKE("%Z%");

## 4. Mostrar la playlist con más canciones y el nombre del usuario que la creó.
SELECT playlist.titulo AS playlist, playlist.cantcanciones AS cantidad_canciones, nombreusuario AS creada_por
FROM playlist INNER JOIN usuario ON playlist.idusuario = usuario.idusuario
ORDER BY cantidad_canciones DESC LIMIT 1;

## 5. Generar un reporte de seguridad, listando las claves utilizadas por todos los usuarios, indicando el nombre del usuario y tipo, ordenado por fecha descendentemente.
SELECT nombreusuario, TipoUsuario, password
FROM usuario
INNER JOIN tipousuario ON usuario.idTipoUsuario = tipousuario.idTipoUsuario
ORDER BY fecha_nac DESC;
```

playlist	cantidad_canciones	creada_por
Top 20	20	FLOPEZ

Result 58

```
#    Time      Action                                                                                                   Message              Duration / Fetch
295  12:59:57  SELECT cancion.titulo AS cancion, album.titulo AS album, artista.nombre AS banda FROM cancion INNER JOIN album ON cancion.idAlbum = album.i...  9 row(s) returned  0.000 sec / 0.000 sec
296  13:00:06  SELECT playlist.titulo AS playlist, playlist.cantcanciones AS cantidad_canciones, nombreusuario AS creada_por FROM playlist INNER JOIN usuario ON...  1 row(s) returned  0.000 sec / 0.000 sec
```

5)

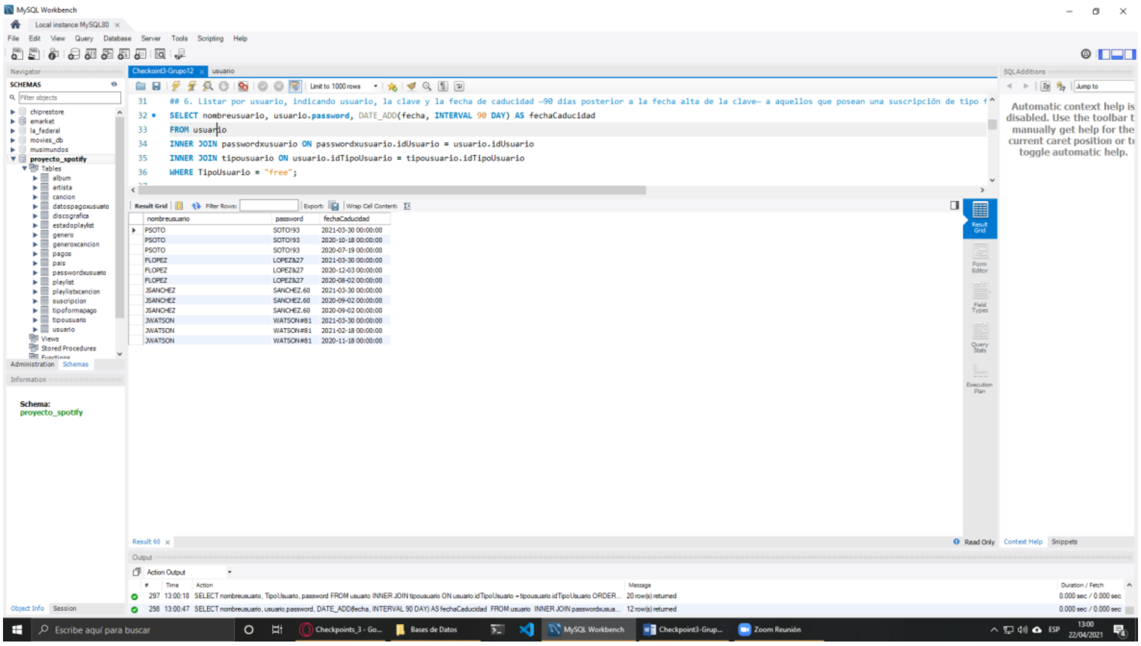


```sql
## 5. Generar un reporte de seguridad, listando las claves utilizadas por todos los usuarios, indicando el nombre del usuario y tipo, ordenado por fecha descendentemente.
SELECT nombreusuario, TipoUsuario, password
FROM usuario
INNER JOIN tipousuario ON usuario.idTipoUsuario = tipousuario.idTipoUsuario
ORDER BY fecha_nac DESC;

## 6. Listar por usuario, indicando usuario, la clave y la fecha de caducidad -90 días posterior a la fecha alta de la clave- a aquellos que posean una suscripción de tipo f
SELECT nombreusuario, usuario.password, DATE_ADD(fecha, INTERVAL 90 DAY) AS fechaCaducidad
FROM usuario
INNER JOIN passwordxusuario ON passwordxusuario.idUsuario = usuario.idUsuario
INNER JOIN tipousuario ON usuario.idTipoUsuario = tipousuario.idTipoUsuario
```

nombreusuario	TipoUsuario	password
JWATSON	free	WATSON#81
MRODRIGUEZ	standard	RODRIGUEZ#38
JSANCHEZ	free	SANCHEZ.60
EHERNANDEZ	Premium	HERNANDEZ$4
DRUBIO	standard	RUBIO.53
MSMITH	standard	SMITH#93
miusuariodespotify@gmail.com	free	S1234m
AGARCIA	Premium	GARCIA!21
CRAMIREZ	Premium	RAMIREZ&88
FLOPEZ	free	LOPEZ&27
JMARTINEZ	standard	MARTINEZ$16
IBALLESTEROS	Premium	BALLESTEROS$93
MSOSA	standard	SOSA.61
MGONZALEZ	Premium	GONZALEZ!15
SMARTINEZ	standard	MARTINEZ&33
MORTIZ	Premium	ORTIZ!16
PSOTO	free	SOTO!93
JLOPEZ	standard	LOPEZ$56
JDIAZ	Premium	DIAZ!93
LGOMEZ	standard	GOMEZ&35

```
#    Time      Action                                                                                                              Message            Duration / Fetch
296  13:00:06  SELECT playlist.titulo AS playlist, playlist.cantcanciones AS cantidad_canciones, nombreusuario AS creada_por FROM playlist INNER JOIN usuario ON...  1 row(s) returned  0.000 sec / 0.000 sec
297  13:00:18  SELECT nombreusuario, TipoUsuario, password FROM usuario INNER JOIN tipousuario ON usuario.idTipoUsuario = tipousuario.idTipoUsuario ORDER...  20 row(s) returned  0.000 sec / 0.000 sec
```

6)

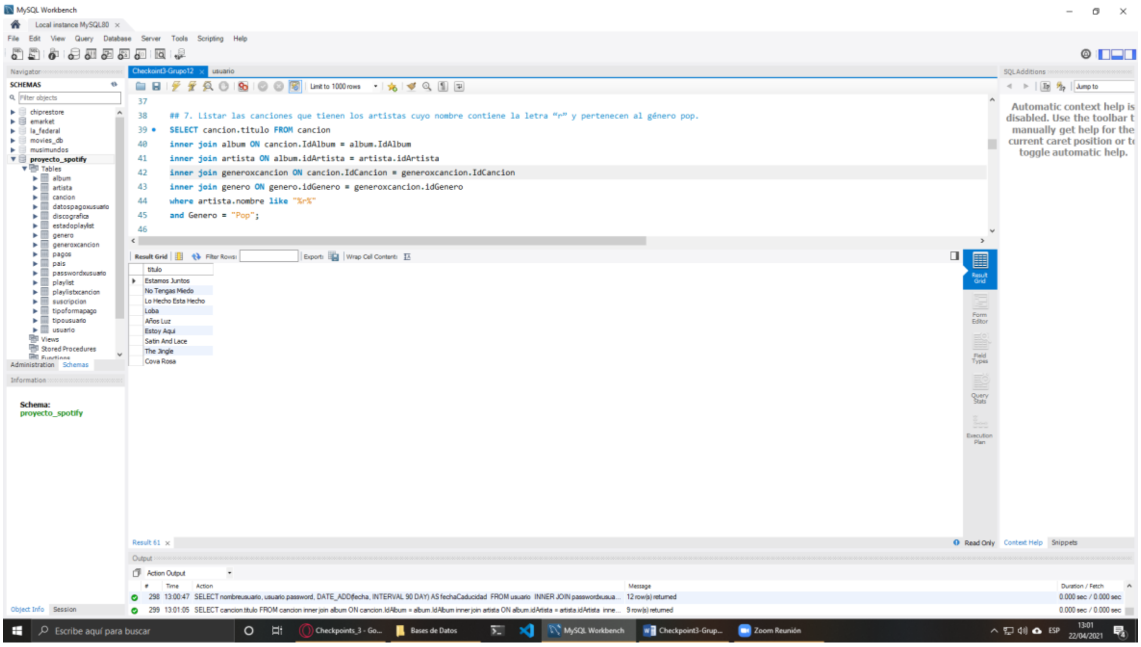


```sql
## 6. Listar por usuario, indicando usuario, la clave y la fecha de caducidad -90 días posterior a la fecha alta de la clave- a aquellos que posean una suscripción de tipo f
SELECT nombreusuario, usuario.password, DATE_ADD(fecha, INTERVAL 90 DAY) AS fechaCaducidad
FROM usuario
INNER JOIN passwordxusuario ON passwordxusuario.idUsuario = usuario.idUsuario
INNER JOIN tipousuario ON usuario.idTipoUsuario = tipousuario.idTipoUsuario
WHERE TipoUsuario = "free";
```

nombreusuario	password	fechaCaducidad
PSOTO	SOTO!93	2021-03-30 00:00:00
PSOTO	SOTO!93	2020-10-18 00:00:00
PSOTO	SOTO!93	2020-07-19 00:00:00
FLOPEZ	LOPEZ&27	2021-03-30 00:00:00
FLOPEZ	LOPEZ&27	2020-12-03 00:00:00
FLOPEZ	LOPEZ&27	2020-08-02 00:00:00
JSANCHEZ	SANCHEZ.60	2021-03-30 00:00:00
JSANCHEZ	SANCHEZ.60	2020-09-02 00:00:00
JSANCHEZ	SANCHEZ.60	2020-09-02 00:00:00
JWATSON	WATSON#81	2021-03-30 00:00:00
JWATSON	WATSON#81	2021-02-18 00:00:00
JWATSON	WATSON#81	2020-11-18 00:00:00

```
#    Time      Action                                                                                                              Message            Duration / Fetch
297  13:00:18  SELECT nombreusuario, TipoUsuario, password FROM usuario INNER JOIN tipousuario ON usuario.idTipoUsuario = tipousuario.idTipoUsuario ORDER...  20 row(s) returned  0.000 sec / 0.000 sec
298  13:00:47  SELECT nombreusuario, usuario.password, DATE_ADD(fecha, INTERVAL 90 DAY) AS fechaCaducidad  FROM usuario  INNER JOIN passwordxusua...  12 row(s) returned  0.000 sec / 0.000 sec
```

7)

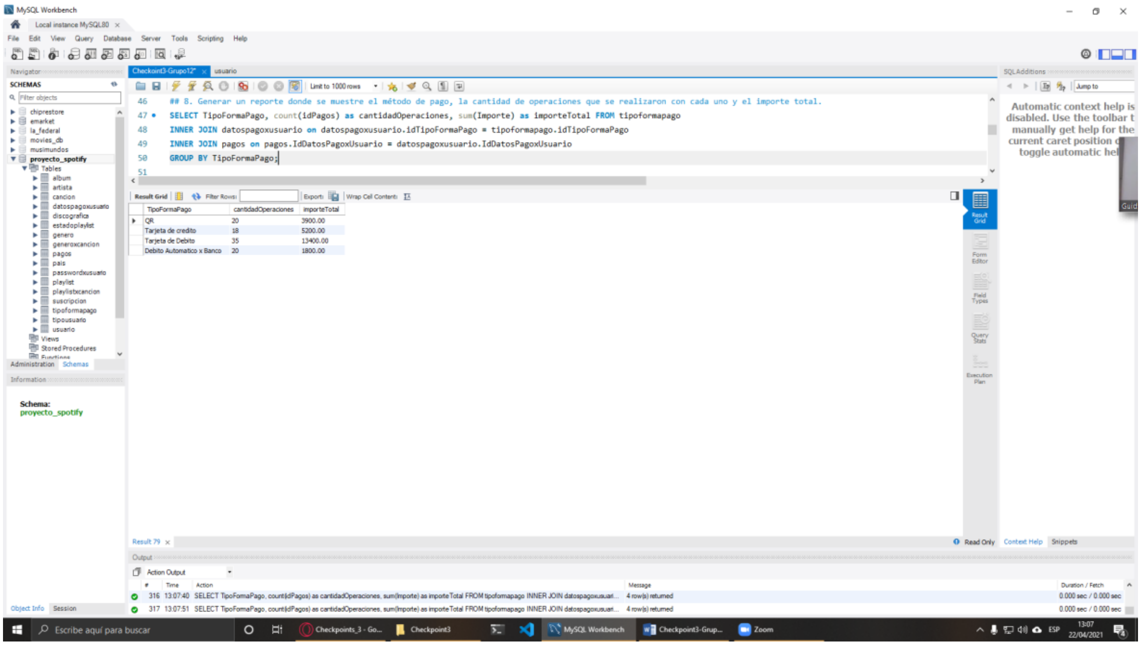


```sql
## 7. Listar las canciones que tienen los artistas cuyo nombre contiene la letra “r” y pertenecen al género pop.
SELECT cancion.titulo FROM cancion
inner join album ON cancion.IdAlbum = album.IdAlbum
inner join artista ON album.idArtista = artista.idArtista
inner join generoxcancion ON cancion.IdCancion = generoxcancion.IdCancion
inner join genero ON genero.idGenero = generoxcancion.idGenero
where artista.nombre like "%r%"
and Genero = "Pop";
```

titulo
Estamos Juntos
No Tengas Miedo
Lo Hecho Esta Hecho
Loba
Años Luz
Estoy Aqui
Satin And Lace
The Jingle
Cova Rosa

```
#    Time      Action                                                                                                         Message               Duration / Fetch
298  13:00:47  SELECT nombreusuario, usuario.password, DATE_ADD(fecha, INTERVAL 90 DAY) AS fechaCaducidad  FROM usuario  INNER JOIN passwordxusua...  12 row(s) returned  0.000 sec / 0.000 sec
299  13:01:05  SELECT cancion.titulo FROM cancion inner join album ON cancion.IdAlbum = album.IdAlbum inner join artista ON album.idArtista = artista.idArtista  inne...  9 row(s) returned  0.000 sec / 0.000 sec
```

8)

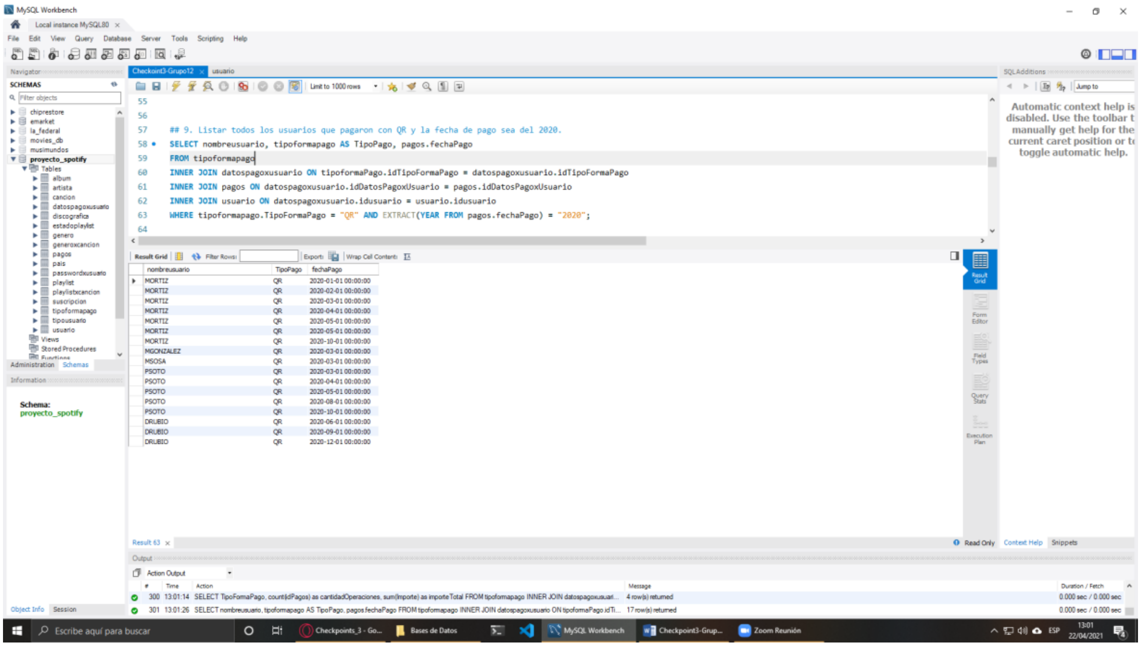


```sql
## 8. Generar un reporte donde se muestre el método de pago, la cantidad de operaciones que se realizaron con cada uno y el importe total.
SELECT TipoFormaPago, count(idPagos) as cantidadOperaciones, sum(Importe) as importeTotal FROM tipoformapago
INNER JOIN datospagoxusuario on datospagoxusuario.idTipoFormaPago = tipoformapago.idTipoFormaPago
INNER JOIN pagos on pagos.IdDatosPagoxUsuario = datospagoxusuario.IdDatosPagoxUsuario
GROUP BY TipoFormaPago;
```

TipoFormaPago	cantidadOperaciones	importeTotal
QR	20	3900.00
Tarjeta de credito	18	5200.00
Tarjeta de Debito	35	13400.00
Debito Automatico x Banco	20	1800.00

```
#    Time      Action                                                                                                         Message              Duration / Fetch
316  13:07:40  SELECT TipoFormaPago, count(idPagos) as cantidadOperaciones, sum(Importe) as importeTotal FROM tipoformapago INNER JOIN datospagoxusuari...  4 row(s) returned  0.000 sec / 0.000 sec
317  13:07:51  SELECT TipoFormaPago, count(idPagos) as cantidadOperaciones, sum(Importe) as importeTotal FROM tipoformapago INNER JOIN datospagoxusuari...  4 row(s) returned  0.000 sec / 0.000 sec
```

9)

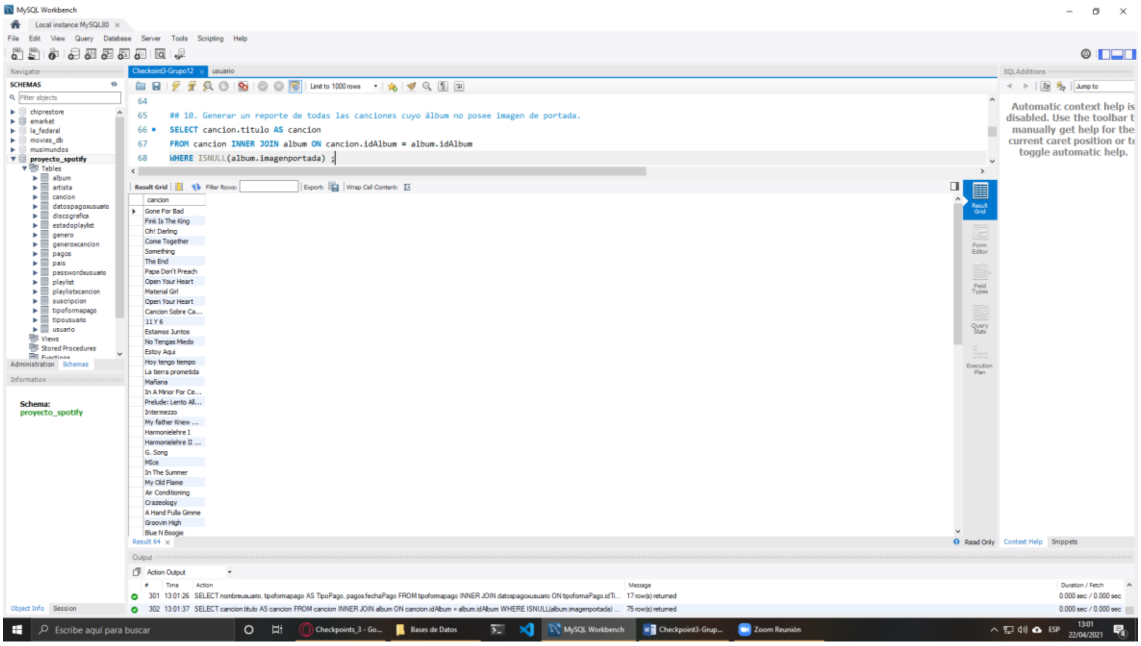


```sql
## 9. Listar todos los usuarios que pagaron con QR y la fecha de pago sea del 2020.
SELECT nombreusuario, tipoformapago AS TipoPago, pagos.fechaPago
FROM tipoformapago
INNER JOIN datospagoxusuario ON tipoformaPago.idTipoFormaPago = datospagoxusuario.idTipoFormaPago
INNER JOIN pagos ON datospagoxusuario.idDatosPagoxUsuario = pagos.idDatosPagoxUsuario
INNER JOIN usuario ON datospagoxusuario.idusuario = usuario.idusuario
WHERE tipoformapago.TipoFormaPago = "QR" AND EXTRACT(YEAR FROM pagos.fechaPago) = "2020";
```

nombreusuario	TipoPago	fechaPago
MORTIZ	QR	2020-01-01 00:00:00
MORTIZ	QR	2020-02-01 00:00:00
MORTIZ	QR	2020-03-01 00:00:00
MORTIZ	QR	2020-04-01 00:00:00
MORTIZ	QR	2020-05-01 00:00:00
MORTIZ	QR	2020-05-01 00:00:00
MORTIZ	QR	2020-10-01 00:00:00
MGONZALEZ	QR	2020-03-01 00:00:00
MSOSA	QR	2020-03-01 00:00:00
PSOTO	QR	2020-03-01 00:00:00
PSOTO	QR	2020-04-01 00:00:00
PSOTO	QR	2020-05-01 00:00:00
PSOTO	QR	2020-08-01 00:00:00
PSOTO	QR	2020-10-01 00:00:00
DRUBIO	QR	2020-06-01 00:00:00
DRUBIO	QR	2020-09-01 00:00:00
DRUBIO	QR	2020-12-01 00:00:00

```
#    Time      Action                                                                                                                        Message             Duration / Fetch
300  13:01:14  SELECT TipoFormaPago, count(idPagos) as cantidadOperaciones, sum(Importe) as importeTotal FROM tipoformapago INNER JOIN datospagoxusuari...  4 row(s) returned   0.000 sec / 0.000 sec
301  13:01:26  SELECT nombreusuario, tipoformapago AS TipoPago, pagos.fechaPago FROM tipoformapago INNER JOIN datospagoxusuario ON tipoformaPago.idTi...  17 row(s) returned  0.000 sec / 0.000 sec
```

10)

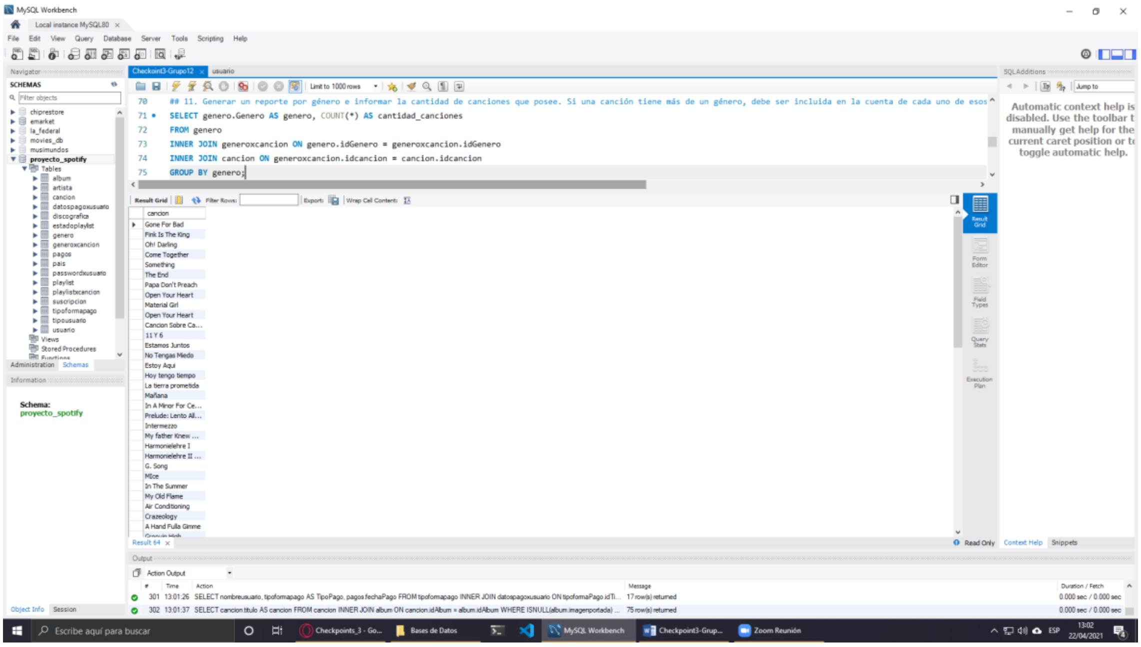


```sql
## 10. Generar un reporte de todas las canciones cuyo álbum no posee imagen de portada.
SELECT cancion.titulo AS cancion
FROM cancion INNER JOIN album ON cancion.idAlbum = album.idAlbum
WHERE ISNULL(album.imagenportada) ;
```

cancion
Gone For Bad
Fink Is The King
Oh! Darling
Come Together
Something
The End
Papa Don't Preach
Open Your Heart
Material Girl
Open Your Heart
Cancion Sobre Ca...
11 Y 6
Estamos Juntos
No Tengas Miedo
Estoy Aqui
Hoy tengo tiempo
La tierra prometida
Mañana
In A Minor For Ce...
Prelude: Lento All...
Intermezzo
My father Knew ...
Harmonielehre I
Harmonielehre II ...
G. Song
MIce
In The Summer
My Old Flame
Air Conditioning
Crazeology
A Hand Fulla Gimme
Groovin High
Blue N Boogie

```
#    Time      Action                                                                                                                        Message             Duration / Fetch
301  13:01:26  SELECT nombreusuario, tipoformapago AS TipoPago, pagos.fechaPago FROM tipoformapago INNER JOIN datospagoxusuario ON tipoformaPago.idTi...  17 row(s) returned  0.000 sec / 0.000 sec
302  13:01:37  SELECT cancion.titulo AS cancion FROM cancion INNER JOIN album ON cancion.idAlbum = album.idAlbum WHERE ISNULL(album.imagenportada) ...  75 row(s) returned  0.000 sec / 0.000 sec
```

11)

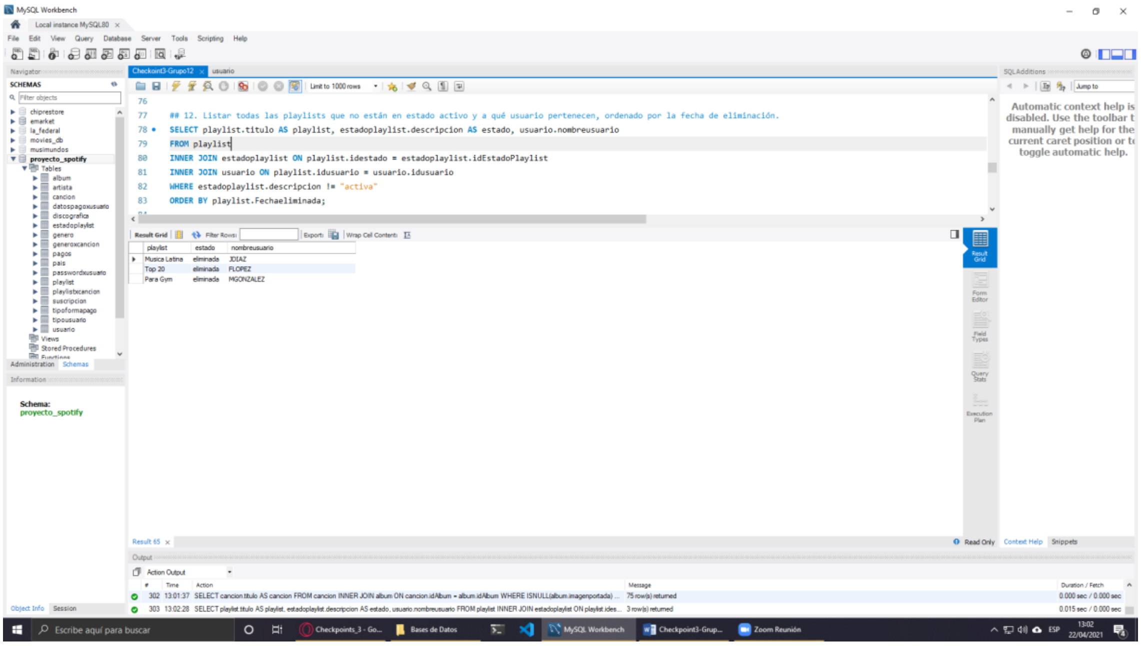


```sql
## 11. Generar un reporte por género e informar la cantidad de canciones que posee. Si una canción tiene más de un género, debe ser incluida en la cuenta de cada uno de esos
SELECT genero.Genero AS genero, COUNT(*) AS cantidad_canciones
FROM genero
INNER JOIN generoxcancion ON genero.idGenero = generoxcancion.idGenero
INNER JOIN cancion ON generoxcancion.idcancion = cancion.idcancion
GROUP BY genero;
```

cancion
Gone For Bad
Fink Is The King
Oh! Darling
Come Together
Something
The End
Papa Don't Preach
Open Your Heart
Material Girl
Open Your Heart
Cancion Sobre Ca...
11 Y 6
Estamos Juntos
No Tengas Miedo
Estoy Aqui
Hoy tengo tiempo
La tierra prometida
Mañana
In A Minor For Ce...
Prelude: Lento All...
Intermezzo
My father Knew ...
Harmonielehre I
Harmonielehre II ...
G. Song
MIce
In The Summer
My Old Flame
Air Conditioning
Crazeology
A Hand Fulla Gimme
Groovin High

```
#    Time      Action                                                                                                              Message              Duration / Fetch
301  13:01:26  SELECT nombreusuario, tipoformapago AS TipoPago, pagos.fechaPago FROM tipoformapago INNER JOIN datospagoxusuario ON tipoformaPago.idTi...  17 row(s) returned  0.000 sec / 0.000 sec
302  13:01:37  SELECT cancion.titulo AS cancion FROM cancion INNER JOIN album ON cancion.idAlbum = album.idAlbum WHERE ISNULL(album.imagenportada) ...  75 row(s) returned  0.000 sec / 0.000 sec
```

12)

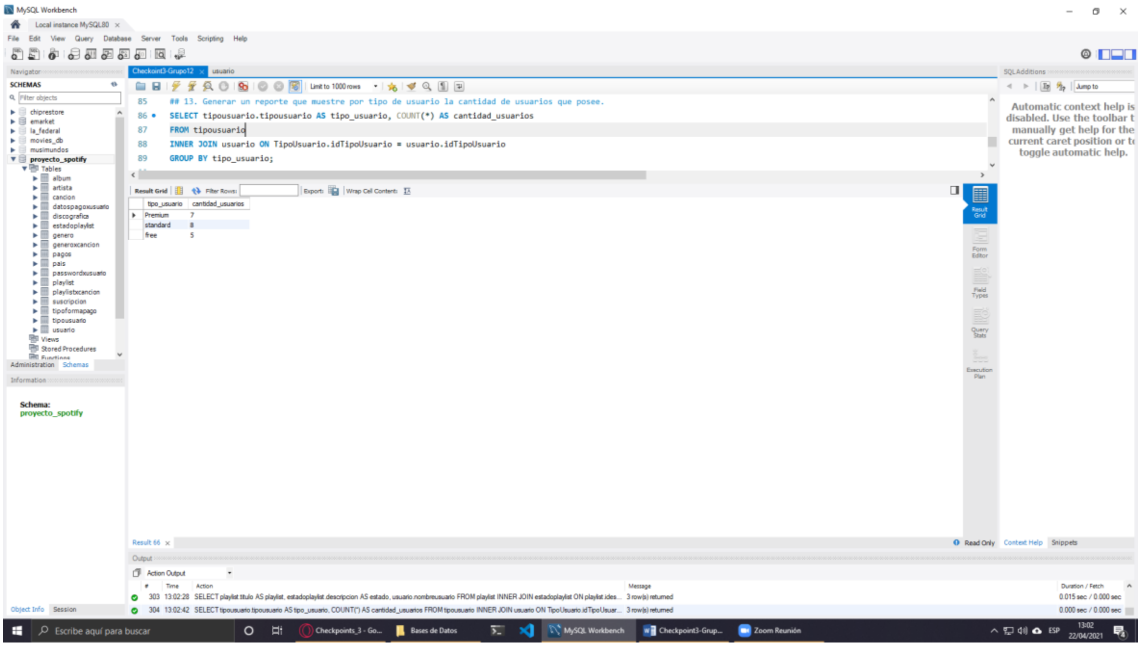


```sql
## 12. Listar todas las playlists que no están en estado activo y a qué usuario pertenecen, ordenado por la fecha de eliminación.
SELECT playlist.titulo AS playlist, estadoplaylist.descripcion AS estado, usuario.nombreusuario
FROM playlist
INNER JOIN estadoplaylist ON playlist.idestado = estadoplaylist.idEstadoPlaylist
INNER JOIN usuario ON playlist.idusuario = usuario.idusuario
WHERE estadoplaylist.descripcion != "activa"
ORDER BY playlist.Fechaeliminada;
```

playlist	estado	nombreusuario
Musica Latina	eliminada	JDIAZ
Top 20	eliminada	FLOPEZ
Para Gym	eliminada	MGONZALEZ

```
#    Time      Action                                                                                                              Message              Duration / Fetch
302  13:01:37  SELECT cancion.titulo AS cancion FROM cancion INNER JOIN album ON cancion.idAlbum = album.idAlbum WHERE ISNULL(album.imagenportada) ...  75 row(s) returned  0.000 sec / 0.000 sec
303  13:02:28  SELECT playlist.titulo AS playlist, estadoplaylist.descripcion AS estado, usuario.nombreusuario FROM playlist INNER JOIN estadoplaylist ON playlist.ides...  3 row(s) returned  0.015 sec / 0.000 sec
```

13)

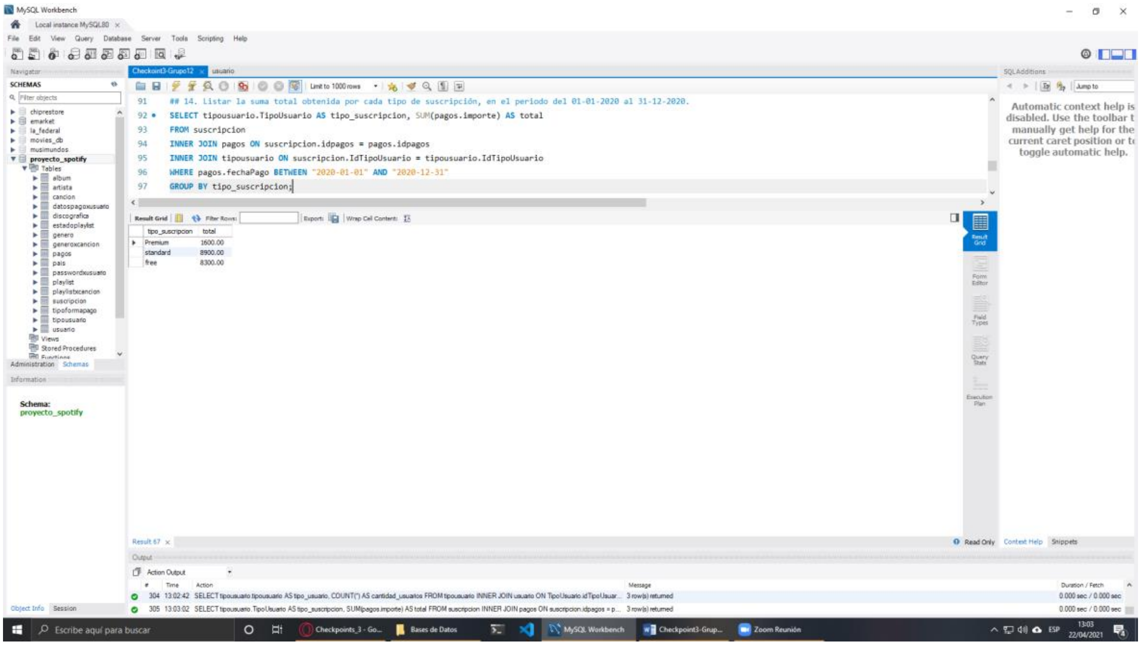


```sql
## 13. Generar un reporte que muestre por tipo de usuario la cantidad de usuarios que posee.
SELECT tipousuario.tipousuario AS tipo_usuario, COUNT(*) AS cantidad_usuarios
FROM tipousuario
INNER JOIN usuario ON TipoUsuario.idTipoUsuario = usuario.idTipoUsuario
GROUP BY tipo_usuario;
```

tipo_usuario	cantidad_usuarios
Premium	7
standard	8
free	5

```
#    Time      Action                                                                                                                          Message            Duration / Fetch
303  13:02:28  SELECT playlist.titulo AS playlist, estadoplaylist.descripcion AS estado, usuario.nombreusuario FROM playlist INNER JOIN estadoplaylist ON playlist.ides...   3 row(s) returned   0.015 sec / 0.000 sec
304  13:02:42  SELECT tipousuario.tipousuario AS tipo_usuario, COUNT(*) AS cantidad_usuarios FROM tipousuario INNER JOIN usuario ON TipoUsuario.idTipoUsuar...   3 row(s) returned   0.000 sec / 0.000 sec
```

14)

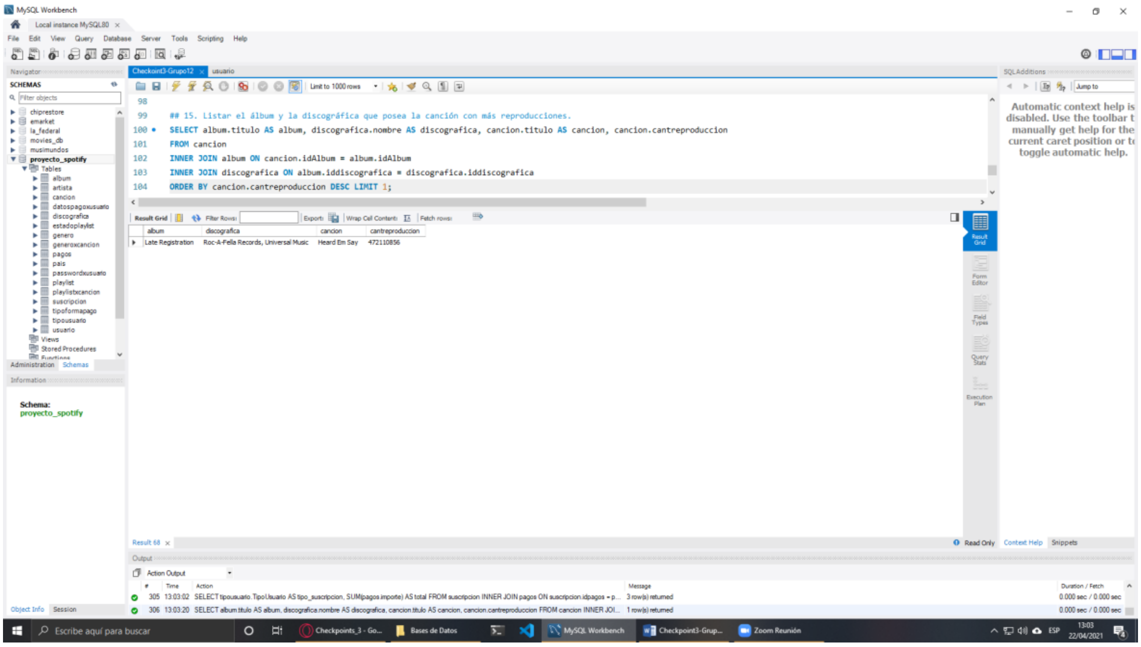


```sql
## 14. Listar la suma total obtenida por cada tipo de suscripción, en el periodo del 01-01-2020 al 31-12-2020.
SELECT tipousuario.TipoUsuario AS tipo_suscripcion, SUM(pagos.importe) AS total
FROM suscripcion
INNER JOIN pagos ON suscripcion.idpagos = pagos.idpagos
INNER JOIN tipousuario ON suscripcion.IdTipoUsuario = tipousuario.IdTipoUsuario
WHERE pagos.fechaPago BETWEEN "2020-01-01" AND "2020-12-31"
GROUP BY tipo_suscripcion;
```

tipo_suscripcion	total
Premium	1600.00
standard	8900.00
free	8300.00

```
#    Time      Action                                                                                                                          Message            Duration / Fetch
304  13:02:42  SELECT tipousuario.tipousuario AS tipo_usuario, COUNT(*) AS cantidad_usuarios FROM tipousuario INNER JOIN usuario ON TipoUsuario.idTipoUsuar...   3 row(s) returned   0.000 sec / 0.000 sec
305  13:03:02  SELECT tipousuario.TipoUsuario AS tipo_suscripcion, SUM(pagos.importe) AS total FROM suscripcion INNER JOIN pagos ON suscripcion.idpagos = p...   3 row(s) returned   0.000 sec / 0.000 sec
```

15)

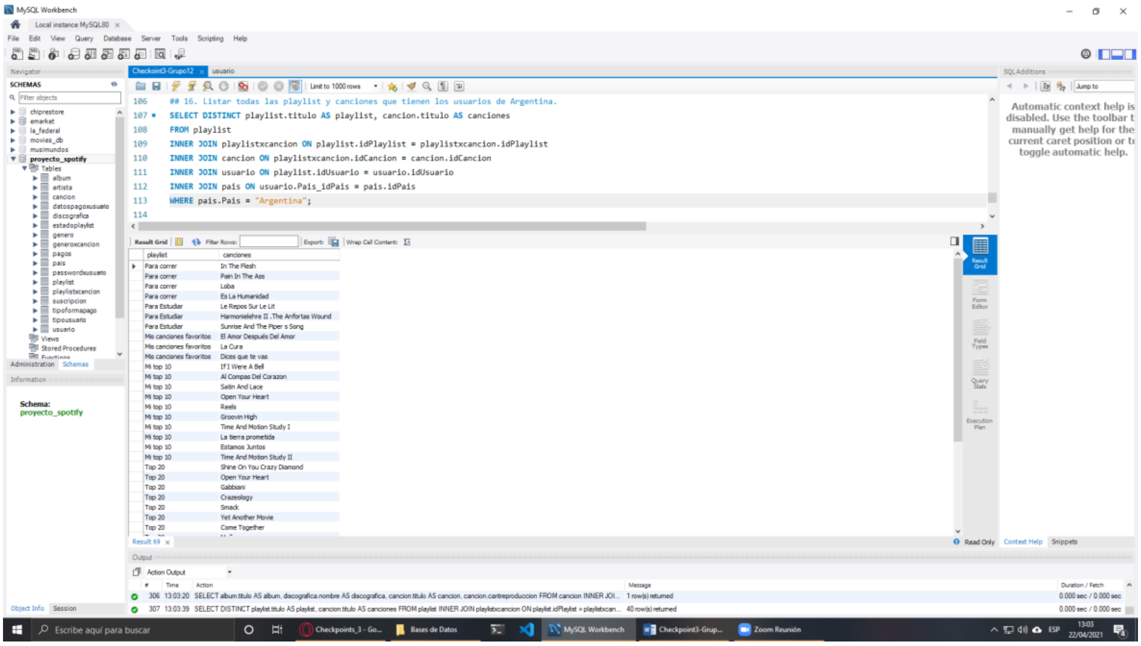


```sql
## 15. Listar el álbum y la discográfica que posea la canción con más reproducciones.
SELECT album.titulo AS album, discografica.nombre AS discografica, cancion.titulo AS cancion, cancion.cantreproduccion
FROM cancion
INNER JOIN album ON cancion.idAlbum = album.idAlbum
INNER JOIN discografica ON album.iddiscografica = discografica.iddiscografica
ORDER BY cancion.cantreproduccion DESC LIMIT 1;
```

album	discografica	cancion	cantreproduccion
Late Registration	Roc-A-Fella Records, Universal Music	Heard Em Say	472110856

```
#    Time      Action                                                                                                            Message              Duration / Fetch
305  13:03:02  SELECT tipousuario.TipoUsuario AS tipo_suscripcion, SUM(pagos.importe) AS total FROM suscripcion INNER JOIN pagos ON suscripcion.idpagos = p...  3 row(s) returned  0.000 sec / 0.000 sec
306  13:03:20  SELECT album.titulo AS album, discografica.nombre AS discografica, cancion.titulo AS cancion, cancion.cantreproduccion FROM cancion INNER JOI...  1 row(s) returned  0.000 sec / 0.000 sec
```

16)

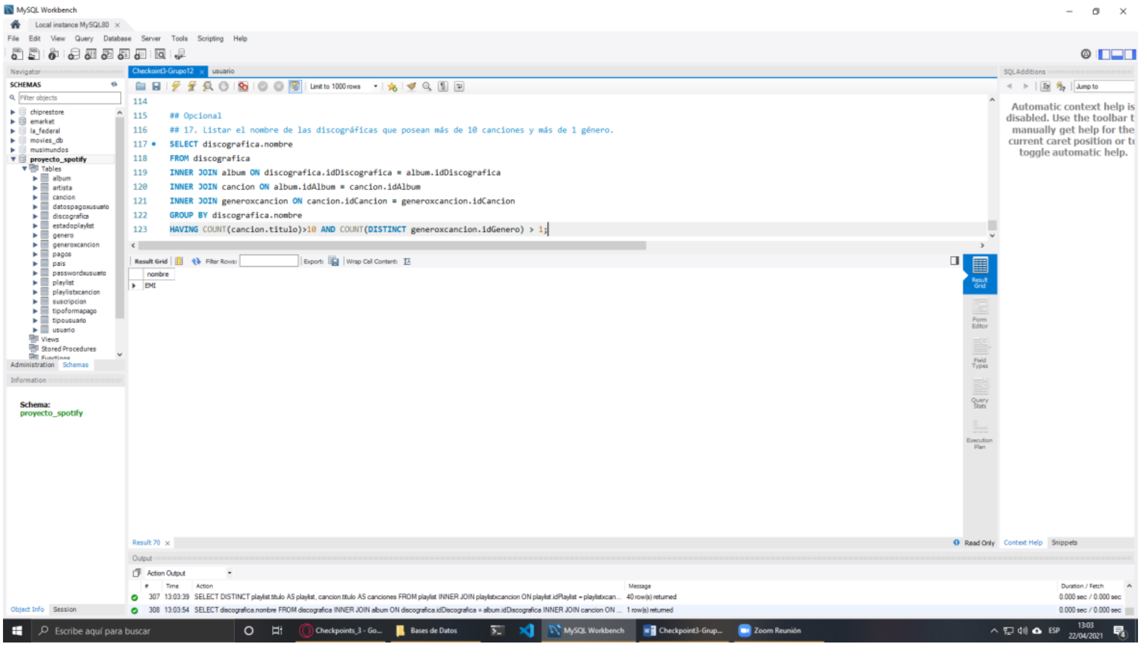


```sql
## 16. Listar todas las playlist y canciones que tienen los usuarios de Argentina.
SELECT DISTINCT playlist.titulo AS playlist, cancion.titulo AS canciones
FROM playlist
INNER JOIN playlistxcancion ON playlist.idPlaylist = playlistxcancion.idPlaylist
INNER JOIN cancion ON playlistxcancion.idCancion = cancion.idCancion
INNER JOIN usuario ON playlist.idUsuario = usuario.idUsuario
INNER JOIN pais ON usuario.Pais_idPais = pais.idPais
WHERE pais.Pais = "Argentina";
```

playlist	canciones
Para correr	In The Flesh
Para correr	Pain In The Ass
Para correr	Loba
Para correr	Es La Humanidad
Para Estudiar	Le Repos Sur Le Lit
Para Estudiar	Harmonielehre II .The Anfortas Wound
Para Estudiar	Sunrise And The Piper s Song
Mis canciones favoritos	El Amor Después Del Amor
Mis canciones favoritos	La Cura
Mis canciones favoritos	Dices que te vas
Mi top 10	If I Were A Bell
Mi top 10	Al Compas Del Corazon
Mi top 10	Satin And Lace
Mi top 10	Open Your Heart
Mi top 10	Reels
Mi top 10	Groovin High
Mi top 10	Time And Motion Study I
Mi top 10	La tierra prometida
Mi top 10	Estamos Juntos
Mi top 10	Time And Motion Study II
Top 20	Shine On You Crazy Diamond
Top 20	Open Your Heart
Top 20	Gabbiani
Top 20	Crazeology
Top 20	Smack
Top 20	Yet Another Movie
Top 20	Come Together

```
#    Time      Action                                                                                                            Message               Duration / Fetch
306  13:03:20  SELECT album.titulo AS album, discografica.nombre AS discografica, cancion.titulo AS cancion, cancion.cantreproduccion FROM cancion INNER JOI...  1 row(s) returned   0.000 sec / 0.000 sec
307  13:03:39  SELECT DISTINCT playlist.titulo AS playlist, cancion.titulo AS canciones FROM playlist INNER JOIN playlistxcancion ON playlist.idPlaylist = playlistxcan...  40 row(s) returned  0.000 sec / 0.000 sec
```

17)



```sql
## Opcional
## 17. Listar el nombre de las discográficas que posean más de 10 canciones y más de 1 género.
SELECT discografica.nombre
FROM discografica
INNER JOIN album ON discografica.idDiscografica = album.idDiscografica
INNER JOIN cancion ON album.idAlbum = cancion.idAlbum
INNER JOIN generoxcancion ON cancion.idCancion = generoxcancion.idCancion
GROUP BY discografica.nombre
HAVING COUNT(cancion.titulo)>10 AND COUNT(DISTINCT generoxcancion.idGenero) > 1;
```

nombre
EMI

Schema:
proyecto_spotify

```
#    Time      Action                                                                                                                      Message             Duration / Fetch
307  13:03:39  SELECT DISTINCT playlist.titulo AS playlist, cancion.titulo AS canciones FROM playlist INNER JOIN playlistxcancion ON playlist.idPlaylist = playlistxcan...   40 row(s) returned   0.000 sec / 0.000 sec
308  13:03:54  SELECT discografica.nombre FROM discografica INNER JOIN album ON discografica.idDiscografica = album.idDiscografica INNER JOIN cancion ON ...   1 row(s) returned   0.000 sec / 0.000 sec
```