

Carrera: Certified Tech Developer

Materia: Bases de Datos I

Checkpoint III

Grupo: 3.

Integrantes:

Florencia Cadelli	florcadelli@gmail.com
Maria Florencia De Mollein	florenciademollein@gmail.com
Sofia Viola	sofiaviola524@gmail.com
Agustín Nicolás Bravo Pérez	anbravoperez@gmail.com

1. Mostrar el nombre de usuario y contar la cantidad de playlists que tiene.

```
SELECT usuario.nombreusuario, COUNT(playlist.idPlaylist) AS "Cantidad Playlist" from playlist  
inner join usuario on playlist.idusuario= usuario.idUsuario  
group by nombreusuario;
```

```
1  /* 1. Mostrar el nombre de usuario y contar la cantidad de  
2  playlists que tiene*/  
3  • SELECT usuario.nombreusuario, COUNT(playlist.idPlaylist) AS "Cantidad Playlist" from playlist  
4  inner join usuario on playlist.idusuario= usuario.idUsuario  
5  group by nombreusuario;  
6  
7  
8  
9  
10  
11
```

Result Grid		Filter Rows:	Export:	Wrap Cell Content:
nombreusuario	Cantidad Playlist			
MORTIZ	1			
IBALLESTEROS	2			
MGONZALEZ	1			
EHERNANDEZ	1			

2. Mostrar los nombres de los géneros con la cantidad de temas que tienen asignadas a ellas.

```
12 /*2 Mostrar los nombres de los géneros con la cantidad de temas  
13 que tienen asignadas a ellas.*/  
14 • SELECT genero.genero, COUNT(IdCancion) AS "CantidadDeTemas" from generoxcancion  
15 inner join genero on generoxcancion.idgenero=genero.idgenero  
16 group by genero.genero;
```

Result Grid		Filter Rows:	Export:	Wrap Cell Content:
genero	CantidadDeTemas			
Rock	12			
Soul	1			
Pop	21			
Música Clasica	18			
Jazz	21			

3. Mostrar las canciones que contienen la letra Z en su nombre, el nombre del álbum al que pertenece y la banda respectiva.

```

24  /*3 Mostrar las canciones que contienen la letra Z en su nombre,
25  el nombre del álbum al que pertenece y la banda respectiva.*/
26  • SELECT cancion.titulo AS Canciones, album.titulo AS Album, artista.Nombre AS Artista FROM cancion
27  INNER JOIN album on cancion.IdAlbum=album.idAlbum
28  INNER join artista on album.idArtista=artista.IdArtista
29  WHERE cancion.titulo LIKE "%Z%";
30
31
32
33
34

```

Result Grid		
Filter Rows:	Export:	Wrap Cell Content:
Canciones	Album	Artista
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

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

```

select playlist.titulo, playlist.cantcanciones, usuario.nombreusuario from
playlist
inner join usuario on playlist.idusuario = usuario.idUsuario
order by cantcanciones desc
limit 1

```

```

13
14 • select playlist.titulo, playlist.cantcanciones, usuario.nombreusuario from playlist
15 inner join usuario on playlist.idusuario = usuario.idUsuario
16 order by cantcanciones desc
17 limit 1

```

Result Grid			
Filter Rows:	Export:	Wrap Cell Content:	Fetch rows:
titulo	cantcanciones	nombreusuario	
Top 20	20	FLOPEZ	

5. Generar un reporte de seguridad, listando las claves utilizadas por todos los usuarios, indicando el nombre del usuario y tipo, ordenado por fecha descendientemente.

```

select usuario.nombreusuario,
tipousuario.TipoUsuario,passwordxusuario.Password,
passwordxusuario.Fecha from usuario

```

```

inner join passwordxusuario on passwordxusuario.idUsuario =
usuario.idUsuario
inner join tipousuario on tipousuario.idTipoUsuario = usuario.IdTipoUsuario
order by passwordxusuario.Fecha

```

48

```

49 • select usuario.nombreusuario, tipousuario.TipoUsuario, passwordxusuario.Password, passwordxusuario.Fecha from usu
50 inner join passwordxusuario on passwordxusuario.idUsuario = usuario.idUsuario
51 inner join tipousuario on tipousuario.idTipoUsuario = usuario.IdTipoUsuario
52 order by passwordxusuario.Fecha desc

```

Result Grid

	nombreusuario	TipoUsuario	Password	Fecha
▶	MORTIZ	Premium	123456789	2021-01-01 00:00:00
	AGARCIA	Premium	GARCIA!2134	2020-12-30 00:00:00
	SMARTINEZ	standard	MARTINEZ&3332	2020-12-30 00:00:00
	EHERNANDEZ	Premium	HERNANDEZ\$491	2020-12-30 00:00:00
	JLOPEZ	standard	LOPEZ\$5696	2020-12-30 00:00:00
	MGONZALEZ	Premium	GONZALEZ11545	2020-12-30 00:00:00
	MRODRIGUEZ	standard	123456789	2020-12-30 00:00:00
	DAMIDEF7	Premium	DAMIDEF7&8864	2020-12-30 00:00:00

Result 69 ×

Read Only

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 free.

```

SELECT u.idUsuario, pxu.Password, MAX(pxu.Fecha) AS FechaDeAlta,
DATE_ADD(MAX(pxu.Fecha), INTERVAL 90 DAY) AS FechaDeCaducacion
FROM usuario u
INNER JOIN tipousuario tu ON u.idTipoUsuario = tu.idTipoUsuario
INNER JOIN passwordxusuario pxu ON pxu.idUsuario = u.idUsuario
WHERE tu.idTipoUsuario = 3
GROUP BY u.idUsuario;

```

```

1  -- 6
2  • SELECT u.idUsuario, pxu.Password, MAX(pxu.Fecha) AS FechaDeAlta,
3  DATE_ADD(MAX(pxu.Fecha), INTERVAL 90 DAY) AS FechaDeCaducacion
4  FROM usuario u
5  INNER JOIN tipousuario tu ON u.idTipoUsuario = tu.idTipoUsuario
6  INNER JOIN passwordxusuario pxu ON pxu.idUsuario = u.idUsuario
7  WHERE tu.idTipoUsuario = 3
8  GROUP BY u.idUsuario;
9
10

```

Result Grid | Filter Rows: | Export: | Wrap Cell Content: |

	idUsuario	Password	FechaDeAlta	FechaDeCaducacion
▶	9	SOTO!9372	2020-12-30 00:00:00	2021-03-30 00:00:00
	12	LOPEZ&2768	2020-12-30 00:00:00	2021-03-30 00:00:00
	13	SANCHEZ.6058	2020-12-30 00:00:00	2021-03-30 00:00:00
	19	WATSON#8199	2020-12-30 00:00:00	2021-03-30 00:00:00

Result 73 x

7. Listar las canciones que tienen los artistas cuyo nombre contiene la letra “r” y pertenecen al género pop.

```

select cancion.*, artista.Nombre, genero.Genero from artista
inner join album on artista.idArtista = album.idArtista
inner join cancion on album.idAlbum = cancion.IdAlbum
inner join generoxcancion on cancion.idCancion = generoxcancion.IdCancion
inner join genero on genero.idGenero = generoxcancion.IdGenero
where genero.Genero = "Pop" and artista.Nombre like "%r%"

```

```

38 • select cancion.*, artista.Nombre, genero.Genero from artista
39 inner join album on artista.idArtista = album.idArtista
40 inner join cancion on album.idAlbum = cancion.IdAlbum
41 inner join generoxcancion on cancion.idCancion = generoxcancion.IdCancion
42 inner join genero on genero.idGenero = generoxcancion.IdGenero
43 where genero.Genero = "Pop" and artista.Nombre like "%r%"
44

```

idCancion	titulo	duracion	cantreproduccion	cantlikes	IdAlbum	Nombre	Genero
19	Estamos Juntos	4.39	389555	12488	12	Diego Torres	Pop
20	No Tengas Miedo	4.25	258456	5247	12	Diego Torres	Pop
21	Lo Hecho Esta Hecho	3.13	986444	657112	13	Shakira	Pop
22	Loba	3.09	3150441	1244523	13	Shakira	Pop
23	Años Luz	3.41	1335054	485777	13	Shakira	Pop
24	Estoy Aqui	3.52	845300	247712	14	Shakira	Pop
85	Satin And Lace	4.25	233200	95	35	Tito Rodriguez	Pop
130	The Jingle	4.38	440523	5352	50	Marco Carola	Pop
133	Cova Rosa	5.31	543440	642	51	Oscar Mulero	Pop

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.TipoFormaPago, count(*), sum(pagos.Importe) from pagos
inner join datospagoxusuario on pagos.IdDatosPagoxUsuario =
datospagoxusuario.idDatosPagoxUsuario
inner join tipoformapago on datospagoxusuario.idTipoFormaPago =
tipoformapago.idTipoFormaPago
group by tipoformapago.TipoFormaPago

```

```

24 • select tipoformapago.TipoFormaPago, count(*), sum(pagos.Importe) from pagos
25 inner join datospagoxusuario on pagos.IdDatosPagoxUsuario = datospagoxusuario.idDatosPagoxUsuario
26 inner join tipoformapago on datospagoxusuario.idTipoFormaPago = tipoformapago.idTipoFormaPago
27 group by tipoformapago.TipoFormaPago
28
29

```

TipoFormaPago	count(*)	sum(pagos.Importe)
Efectivo	20	3900
Tarjeta de credito	18	5200
Tarjeta de Debito	35	13400
Debito Automatico x Banco	20	1800

9. Listar todos los usuarios que pagaron con efectivo y la fecha de pago sea del 2020.

```
SELECT DISTINCT u.*
FROM usuario u
INNER JOIN datospagoxusuario dpu ON u.idUsuario = dpu.idusuario
INNER JOIN tipoformapago tfp ON tfp.idTipoFormaPago =
dpu.idTipoFormaPago
INNER JOIN pagos p ON p.IdDatosPagoxUsuario =
dpu.idDatosPagoxUsuario
WHERE tfp.idTipoFormaPago = 1 AND (p.fechaPago BETWEEN
"2020-01-01" AND "2020-12-31");
```

```
1  -- 9
2  • SELECT DISTINCT u.*
3  FROM usuario u
4  INNER JOIN datospagoxusuario dpu ON u.idUsuario = dpu.idusuario
5  INNER JOIN tipoformapago tfp ON tfp.idTipoFormaPago = dpu.idTipoFormaPago
6  INNER JOIN pagos p ON p.IdDatosPagoxUsuario = dpu.idDatosPagoxUsuario
7  WHERE tfp.idTipoFormaPago = 1 AND (p.fechaPago BETWEEN "2020-01-01" AND "2020-12-31");
8
```

Result Grid								
Filter Rows: <input type="text"/> Export: <input type="button" value=""/> Wrap Cell Content: <input type="button" value="A"/>								
idUsuario	nombreusuario	nyap	fecha_nac	sexo	CP	password	Pais_idPais	I
1	MORTIZ	MARIA ORTIZ	1975-09-27	F	1001	ORTIZ!16	1	1
4	MGONZALEZ	MARIA PAULA GONZALEZ	1981-03-27	F	118942	GONZALEZ!15	2	1
7	MSOSA	MARIA CARMEN SOSA	1981-07-16	F	3	SOSA.61	6	2
9	PSOTO	PATRICIA SOTO	1974-07-12	F	832 0000	SOTO!93	8	3
18	DRUBIO	DAVID RUBIO	2001-01-17	M	60000	RUBIO.53	3	2

10. Generar un reporte de todas las canciones cuyo álbum no posee imagen de portada.

```
SELECT c.*, a.imagenportada
FROM cancion c
INNER JOIN album a ON a.idAlbum = c.idAlbum
WHERE a.imagenportada IS NULL;
```

```

9      -- 10
10 •   SELECT c.*, a.imagenportada
11      FROM cancion c
12      INNER JOIN album a ON a.idAlbum = c.idAlbum
13      WHERE a.imagenportada IS NULL;

```

Result Grid Filter Rows: Export: Wrap Cell Content:							
	idCancion	titulo	duracion	cantreproduccion	cantlikes	IdAlbum	imagenportada
▶	3	Gone For Bad	3.58	1200400	6500	2	NULL
	4	Fink Is The King	3.15	218500	8600	2	NULL
	7	Oh! Darling	3.26	1598634	256986	4	NULL
	8	Come Together	4.2	3568946	103569	4	NULL
	9	Something	3.03	628634	5698	4	NULL
	10	The End	2.05	68946	3569	4	NULL
	11	Papa Don't Preach	4.29	NULL	NULL	8	NULL
	12	Open Your Heart	4.13	2500245	1785444	8	NULL

Result 26 x

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 géneros.

```

SELECT g.idGenero, g.Genero, COUNT(*)
FROM cancion c
INNER JOIN generoxcancion gxc ON gxc.idCancion = c.idCancion
INNER JOIN genero g ON g.idGenero = gxc.idGenero
GROUP BY g.idGenero;

```

```

15      -- 11
16 •   SELECT g.idGenero, g.Genero, COUNT(*)
17      FROM cancion c
18      INNER JOIN generoxcancion gxc ON gxc.idCancion = c.idCancion
19      INNER JOIN genero g ON g.idGenero = gxc.idGenero
20      GROUP BY g.idGenero;

```

Result Grid Filter Rows: Export: Wrap Cell Content:			
	idGenero	Genero	COUNT(*)
▶	1	Música Clasica	18
	2	Jazz	21
	3	Soul	1
	6	Salsa	18
	8	Tango	18
	9	Pop	21
	11	Rock	12
	15	Hip Hop	18

Result 35 x

12. Listar todas las playlists que no están en estado activo y a qué usuario pertenecen, ordenado por la fecha de eliminación.


```

SELECT p.*, u.nombreusuario
FROM playlist p
INNER JOIN usuario u ON p.idusuario = u.idUsuario
INNER JOIN estadoplaylist ep ON ep.idEstadoPlaylist = p.idestado
WHERE ep.idEstadoPlaylist != 1
ORDER BY p.Fechaeliminada;

```

```

22      -- 12
23      SELECT p.*, u.nombreusuario
24      FROM playlist p
25      INNER JOIN usuario u ON p.idusuario = u.idUsuario
26      INNER JOIN estadoplaylist ep ON ep.idEstadoPlaylist = p.idestado
27      WHERE ep.idEstadoPlaylist != 1
28      ORDER BY p.Fechaeliminada;
29

```

	idPlaylist	idusuario	titulo	cantcanciones	idestado	Fechacreacion	Fechaeliminada	nombreu
▶	9	15	Musica Latina	5	2	2016-12-11 00:00:00	2016-02-19 00:00:00	JDIAZ
	6	12	Top 20	20	2	2016-06-06 00:00:00	2016-04-12 00:00:00	FLOPEZ
	3	4	Para Gym	15	2	2020-03-07 00:00:00	2020-04-10 00:00:00	MGONZAI

13. Generar un reporte que muestre por tipo de usuario la cantidad de usuarios que posee.*

```

select tipousuario.tipousuario, count(usuario.idusuario) from tipousuario
inner join usuario on tipousuario.idTipoUsuario = usuario.IdTipoUsuario
group by tipousuario.tipousuario

```

```

4 • select tipousuario.tipousuario, count(usuario.idusuario) from tipousuario
5 inner join usuario on tipousuario.idTipoUsuario = usuario.IdTipoUsuario
6 group by tipousuario.tipousuario
7
8 /*
9 14. Listar la suma total obtenida por cada tipo de suscripción, en el

```

Result Grid

	tipousuario	count(usuario.idusuario)
▶	free	4
	Premium	7
	standard	8

14. Listar la suma total obtenida por cada tipo de suscripción, en el periodo del 01-01-2020 al 31-12-2020.

Esto lo interpretamos de dos formas: Agrupar según el tipo de suscripción:

```

SELECT s.idTipoUsuario, SUM(p.Importe) AS SumaTotal
FROM suscripcion s
INNER JOIN pagos p ON p.idPagos = s.idPagos
WHERE p.fechaPago BETWEEN "2020-01-01" AND "2020-12-31"
GROUP BY s.idTipoUsuario;

```

```

48 -- 14
49 • SELECT s.idTipoUsuario, SUM(p.Importe) AS SumaTotal
50 FROM suscripcion s
51 INNER JOIN pagos p ON p.idPagos = s.idPagos
52 WHERE p.fechaPago BETWEEN "2020-01-01" AND "2020-12-31"
53 GROUP BY s.idTipoUsuario;
54

```

Result Grid

	idTipoUsuario	SumaTotal
▶	1	1600
	2	8900
	3	8300

Result 86 x

O Agrupar según idSuscripcion:

```
select suscripcion.idSuscripcion, sum(pagos.importe), pagos.fechaPago from
pagos
inner join suscripcion on suscripcion.idpagos = pagos.idPagos
where pagos.fechaPago between '2020-01-01 00:00:00'and '2020-12-31 00:00:00'
group by suscripcion.idSuscripcion
```

11

12 • `select suscripcion.idSuscripcion, sum(pagos.importe), pagos.fechaPago from pagos`

13 `inner join suscripcion on suscripcion.idpagos = pagos.idPagos`

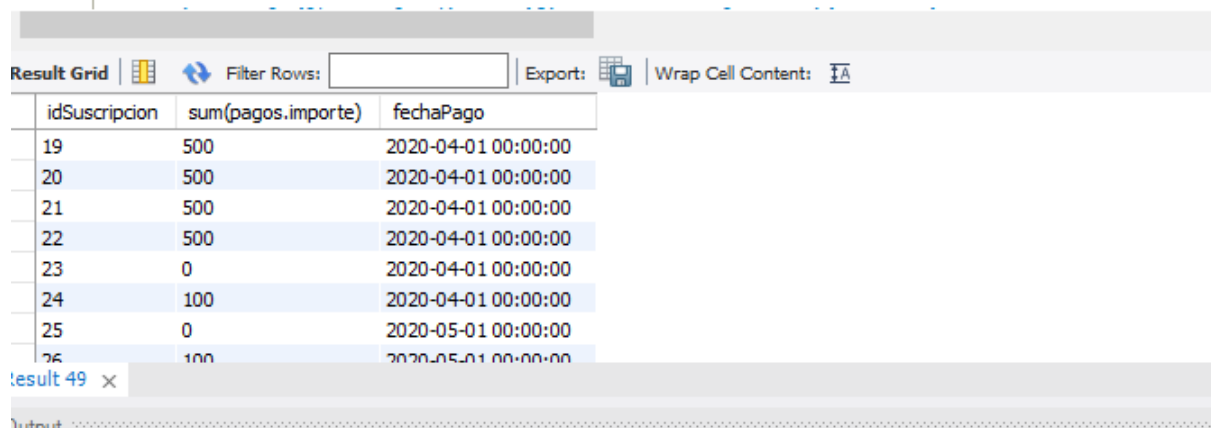
14 `where pagos.fechaPago between '2020-01-01 00:00:00'and '2020-12-31 00:00:00'`

15 `group by suscripcion.idSuscripcion`

16

17

18 `/*`



idSuscripcion	sum(pagos.importe)	fechaPago
19	500	2020-04-01 00:00:00
20	500	2020-04-01 00:00:00
21	500	2020-04-01 00:00:00
22	500	2020-04-01 00:00:00
23	0	2020-04-01 00:00:00
24	100	2020-04-01 00:00:00
25	0	2020-05-01 00:00:00
26	100	2020-05-01 00:00:00

15. Listar el álbum y la discográfica que posea la canción con más reproducciones.

```
select album.titulo as album , discografica.nombre as discografica ,
sum(cancion.cantreproduccion) as "cantidad de reproducciones" from discografica
inner join album on album.iddiscografica = discografica.idDiscografica
inner join cancion on album.idAlbum = cancion.IdAlbum
group by album.titulo
order by sum(cancion.cantreproduccion) desc
limit 1
```

```

25
26 • select album.titulo as album , discografica.nombre as discografica , sum(cancion.cantreproduccion) as "
27 inner join album on album.iddiscografica = discografica.idDiscografica
28 inner join cancion on album.idAlbum = cancion.IdAlbum
29 group by album.titulo
30 order by sum(cancion.cantreproduccion) desc
31 limit 1
32
33

```

album	discografica	cantidad de reproducciones
Late Registration	Roc-A-Fella Records, Universal Music	472976693

16. Listar todas las playlist y canciones que tienen los usuarios de Argentina.

```

select playlist.titulo, cancion.titulo, usuario.Pais_idPais, pais.Pais from usuario
inner join playlist on playlist.idusuario = usuario.idUsuario
inner join playlistxcancion on playlistxcancion.IdPlaylist = playlist.idPlaylist
inner join cancion on playlistxcancion.Idcancion = cancion.idCancion
inner join pais on pais.idPais = usuario.Pais_idPais
where pais.Pais = 'argentina';

```

```

37 16. Listar todas las playlist y canciones que tienen los usuarios de
38 Argentina.*/
39
40 • select playlist.titulo, cancion.titulo, usuario.Pais_idPais, pais.Pais from usuario
41 inner join playlist on playlist.idusuario = usuario.idUsuario
42 inner join playlistxcancion on playlistxcancion.IdPlaylist = playlist.idPlaylist
43 inner join cancion on playlistxcancion.Idcancion = cancion.idCancion
44 inner join pais on pais.idPais = usuario.Pais_idPais
45 where pais.Pais = 'argentina';

```

titulo	titulo	Pais_idPais	Pais
Para Estudiar	Sunrise And The Piper s Song	1	Argentina
Mis canciones favoritos	El Amor Después Del Amor	1	Argentina
Mis canciones favoritos	La Cura	1	Argentina
Mis canciones favoritos	Dices que te vas	1	Argentina
Top 20	Shine On You Crazy Diamond	1	Argentina
Top 20	Open Your Heart	1	Argentina
Top 20	Gabbiani	1	Argentina
Top 20	Crazytown	1	Argentina

17. Listar el nombre de las discográficas que posean más de 10 canciones y más de 1 género.

```

SELECT d.nombre, COUNT(c.idCancion), COUNT(DISTINCT gxc.idGenero)
FROM discografica d

```

```



INNER JOIN album a ON a.iddiscografica = d.iddiscografica
INNER JOIN cancion c ON c.idAlbum = a.idAlbum
INNER JOIN generoxcancion gxc ON c.idCancion = gxc.idCancion
GROUP BY d.idDiscografica
HAVING COUNT(c.idCancion) > 10 AND COUNT(DISTINCT gxc.idGenero) >
1;

```

```

30  -- 17
31  • SELECT d.nombre, COUNT(c.idCancion), COUNT(DISTINCT gxc.idGenero)
32  FROM discografica d
33  INNER JOIN album a ON a.iddiscografica = d.iddiscografica
34  INNER JOIN cancion c ON c.idAlbum = a.idAlbum
35  INNER JOIN generoxcancion gxc ON c.idCancion = gxc.idCancion
36  GROUP BY d.idDiscografica
37  HAVING COUNT(c.idCancion) > 10 AND COUNT(DISTINCT gxc.idGenero) > 1;
38
39

```

Result Grid			
Filter Rows: <input type="text"/>			
Export: 			
Wrap Cell Content: 			
	nombre	COUNT(c.idCancion)	COUNT(DISTINCT gxc.idGenero)
▶	EMI	12	3

Result 56 × 