

PROYECTO BASE DE DATOS

LIGA FANTASY



FANTASY



Alfonso López García

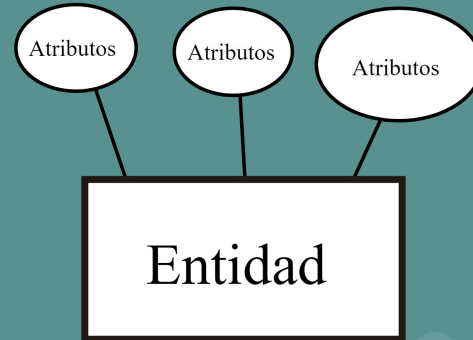
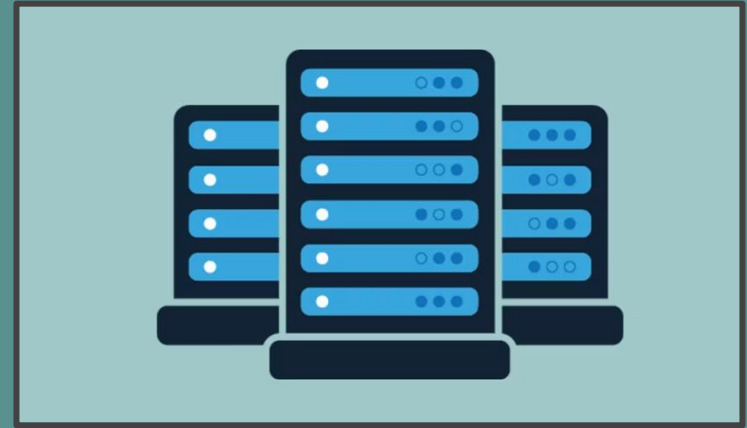
1ºDAW

IES ALIXAR

29/03/2025

ÍNDICE

1. **Introducción**
2. **Modelo Entidad/Relación**
3. **Modelo Relacional**
4. **Carga Masiva de Datos**
5. **Consultas**
6. **Vistas y triggers**
7. **Procedimientos y funciones**
8. **Conclusión**



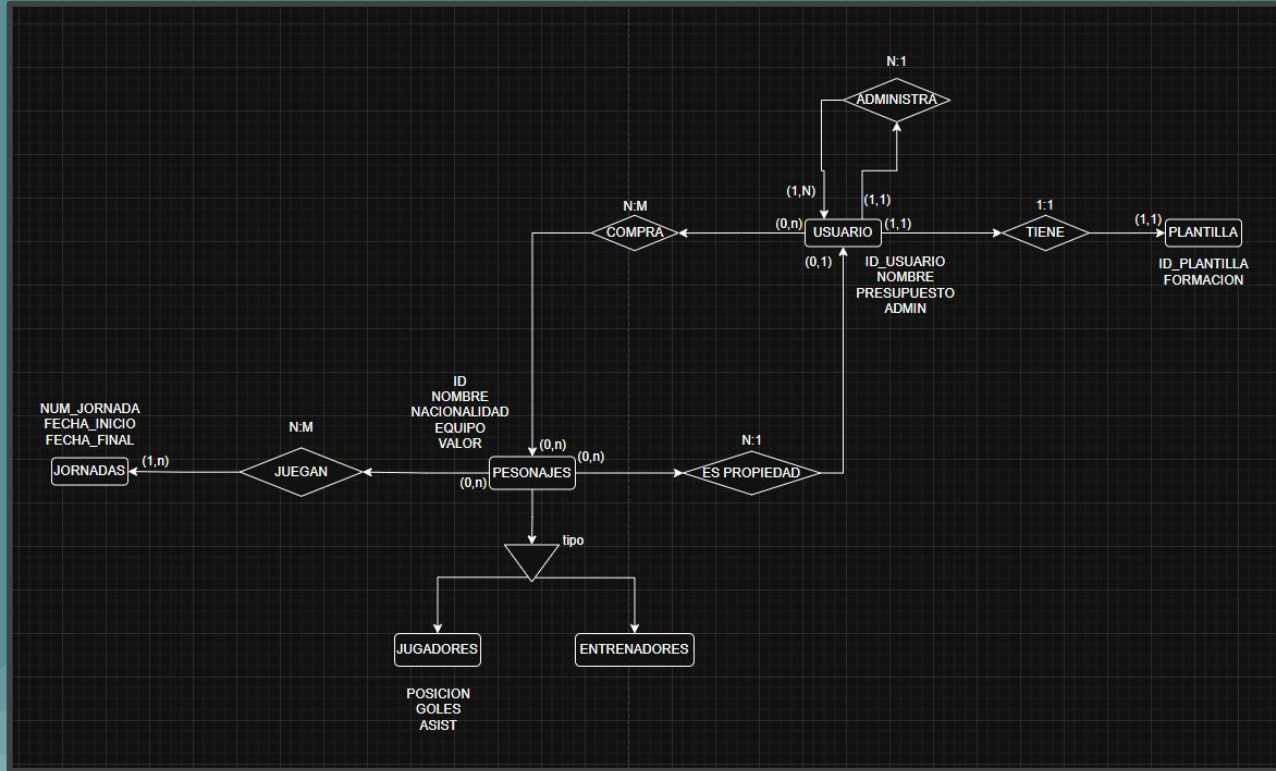
1. INTRODUCCIÓN

En este proyecto hemos creado una base de datos sobre una **Liga Fantasy** de la Liga Española de Fútbol. En ella habrá registros como los **nombres de los usuarios** o los **jugadores registrados** en la competición con sus respectivos **dueños**, su **valor de mercado** y sus demás datos. También se guardarán los **traspasos** que se hagan entre los usuarios y los **puntos** que hagan los jugadores cada jornada.



2. MODELO ENTIDAD/RELACIÓN

En el Modelo Entidad/Relación tenemos varias entidades con sus respectivas claves primarias y foráneas. Entre todas estas encontramos **entidades débiles**, una **herencia**, una **relación recíproca** y varias **relaciones N:M**.

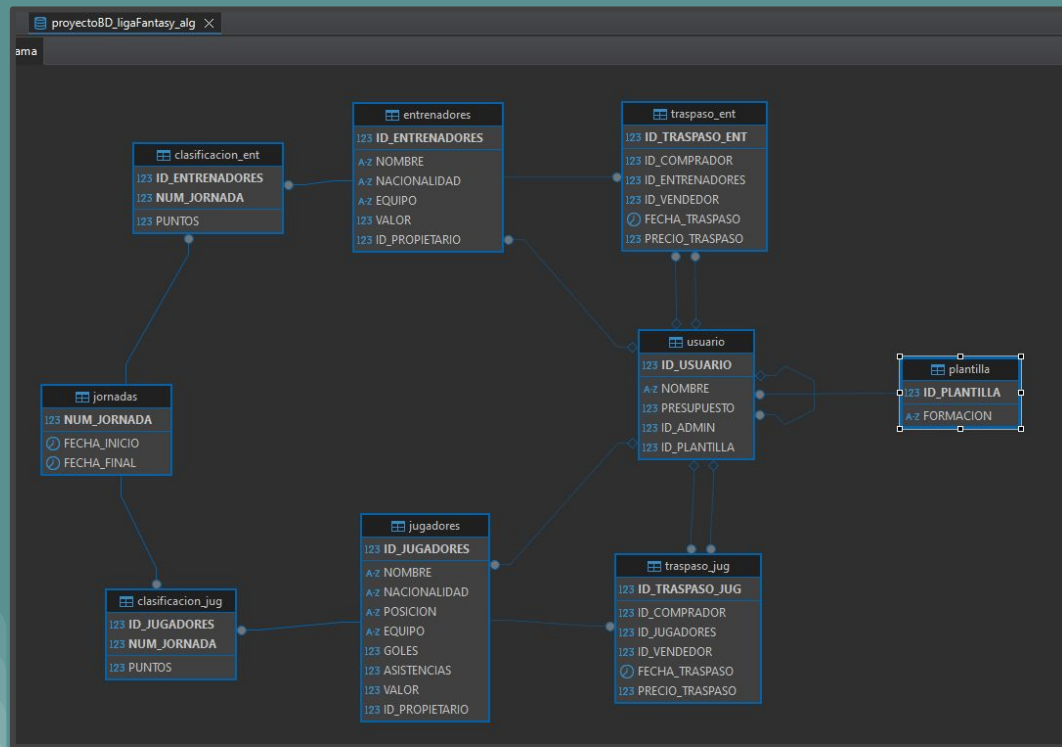


3. MODELO RELACIONAL

En el modelo relacional ya podremos apreciar mejor las tablas que se crean con las **relaciones entre entidades**, las **claves primarias**, las **claves foráneas** y las **entidades débiles**.

Cuando tengamos el modelo relacional hecho, podremos pasarlo a la herramienta de Bases de Datos **DBeaver** gracias a un Script que podemos generar desde el **Workbench de MySQL**.

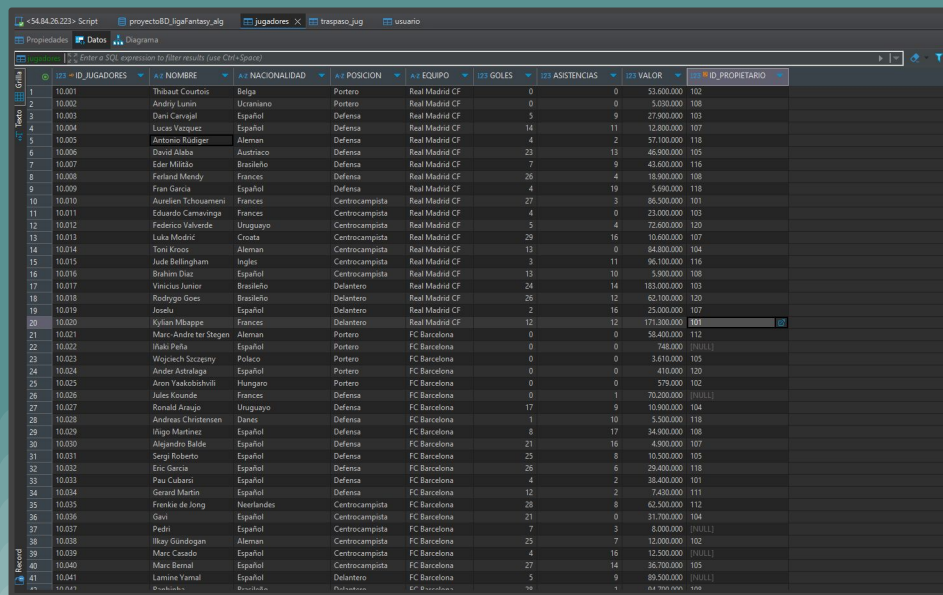
Cuando tengamos ese Script, seremos capaces de iniciar nuestra Base de Datos **vacía** en cualquier editor.



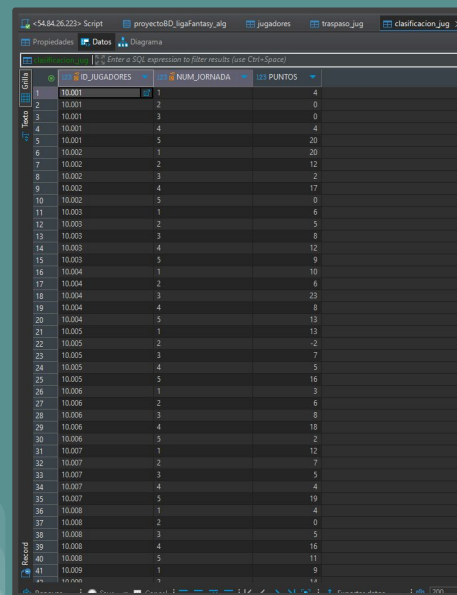
4. CARGA MASIVA DE DATOS

Para la carga masiva de datos hemos creado un [excel](#) donde hemos recopilado toda la información necesaria para nuestro proyecto. Para su inserción en la Base de Datos, lo que hemos hecho ha sido pasar las tablas del excel por un [convertidor online](#) para generar un **Script** con los datos de nuestras tablas.

Las dos tablas que más datos se les han introducido han sido las de **jugadores** y **clasificación de jugadores**.



ID	NOMBRE	NACIONALIDAD	POSICION	EQUIPO	GOLES	ASISTENCIAS	VALOR	ID_PROPIETARIO
1	Thibaut Courtois	Belga	Portero	Real Madrid CF	0	0	53.600.000	102
2	Andriy Lunin	Ucraniano	Portero	Real Madrid CF	0	0	5.000.000	108
3	Dani Carvajal	Español	Defensa	Real Madrid CF	5	9	27.900.000	103
4	Lucas Vázquez	Español	Defensa	Real Madrid CF	14	11	12.800.000	107
5	Antonio Rüdiger	Alemán	Defensa	Real Madrid CF	4	2	57.100.000	118
6	David Alaba	Austriaco	Defensa	Real Madrid CF	23	13	48.900.000	105
7	Éder Militão	Brasilíelo	Defensa	Real Madrid CF	7	9	42.600.000	116
8	Ferland Mendy	Frances	Defensa	Real Madrid CF	26	4	18.900.000	108
9	Fran García	Español	Defensa	Real Madrid CF	4	19	5.690.000	118
10	Aurélien Tchouaméni	Frances	Centrocampista	Real Madrid CF	27	3	86.500.000	101
11	Edouardo Camavinga	Frances	Centrocampista	Real Madrid CF	4	0	23.000.000	103
12	Federico Valverde	Uruguayo	Centrocampista	Real Madrid CF	5	4	72.600.000	120
13	Luka Modrić	Croata	Centrocampista	Real Madrid CF	29	16	10.600.000	107
14	Toni Kroos	Alemán	Centrocampista	Real Madrid CF	13	0	84.800.000	104
15	Jude Bellingham	Inglés	Centrocampista	Real Madrid CF	3	11	96.100.000	116
16	Brahim Díaz	Español	Centrocampista	Real Madrid CF	13	10	5.900.000	108
17	Vincius Junior	Brasilíelo	Delantero	Real Madrid CF	24	14	183.000.000	103
18	Rodrygo Goes	Brasilíelo	Delantero	Real Madrid CF	26	12	62.100.000	120
19	Joselu	Español	Delantero	Real Madrid CF	2	16	25.000.000	107
20	Kilian Kilipaz	Frances	Delantero	Real Madrid CF	10	12	171.300.000	100
21	Marc-André ter Stegen	Alemán	Portero	FC Barcelona	0	0	58.400.000	112
22	Waki Peña	Español	Portero	FC Barcelona	0	0	748.000	[NULL]
23	Wojciech Szczęsny	Polaco	Portero	FC Barcelona	0	0	3.610.000	105
24	Andrés Iniesta	Español	Portero	FC Barcelona	0	0	410.000	120
25	Áron Yakobshvíl	Húngaro	Portero	FC Barcelona	0	0	579.000	102
26	Jules Koundé	Frances	Defensa	FC Barcelona	0	1	70.200.000	[NULL]
27	Ronald Araújo	Uruguayo	Defensa	FC Barcelona	17	9	10.900.000	104
28	Andreas Christensen	Danés	Defensa	FC Barcelona	11	10	5.500.000	118
29	Hugo Martínez	Español	Defensa	FC Barcelona	8	17	34.900.000	108
30	Allegandro Balde	Español	Defensa	FC Barcelona	21	16	4.900.000	107
31	Sergi Roberto	Español	Defensa	FC Barcelona	25	8	10.500.000	105
32	Eric García	Español	Defensa	FC Barcelona	26	6	29.400.000	118
33	Pau Cubarsi	Español	Defensa	FC Barcelona	4	2	38.400.000	101
34	Gerard Martín	Español	Defensa	FC Barcelona	12	2	7.430.000	111
35	Frenkie de Jong	Neerlandés	Centrocampista	FC Barcelona	28	8	62.500.000	112
36	Gavi	Español	Centrocampista	FC Barcelona	21	0	31.700.000	104
37	Pedri	Español	Centrocampista	FC Barcelona	7	3	8.900.000	[NULL]
38	Ilkay Gündogan	Alemán	Centrocampista	FC Barcelona	25	7	12.000.000	102
39	Marc Casado	Español	Centrocampista	FC Barcelona	4	16	12.500.000	[NULL]
40	Marc Bernal	Español	Centrocampista	FC Barcelona	27	14	36.700.000	105
41	Lamine Yamal	Español	Delantero	FC Barcelona	5	9	80.500.000	[NULL]
42	Robert Lewandowski	Polaco	Delantero	FC Barcelona	14	1	64.700.000	106



ID	NUM_JORNADA	PUNTOS
1	10.001	4
2	10.001	2
3	10.001	3
4	10.001	4
5	10.001	5
6	10.002	1
7	10.002	2
8	10.002	3
9	10.002	4
10	10.002	5
11	10.003	1
12	10.003	2
13	10.003	3
14	10.003	4
15	10.003	5
16	10.004	1
17	10.004	2
18	10.004	3
19	10.004	4
20	10.004	5
21	10.005	1
22	10.005	2
23	10.005	3
24	10.005	4
25	10.005	5
26	10.006	1
27	10.006	2
28	10.006	3
29	10.006	4
30	10.006	5
31	10.007	1
32	10.007	2
33	10.007	3
34	10.007	4
35	10.007	5
36	10.008	1
37	10.008	2
38	10.008	3
39	10.008	4
40	10.008	5
41	10.009	1
42	10.009	2

5. CONSULTAS

CONSULTA 1

LOS JUGADORES QUE HAN HECHO MÁS PUNTOS EN CADA JORNADA

```
select cj.num_jornada, j.nombre as jugador, cj.puntos
from clasificacion_jug cj inner join jugadores j
    on cj.id_jugadores = j.id_jugadores
where cj.puntos = (select max(cj2.puntos)
    from clasificacion_jug cj2
    where cj2.num_jornada = cj.num_jornada )
order by cj.num_jornada;
```

The screenshot shows a database IDE with a script editor and a results window. The script editor contains the following SQL code:

```
use proyectoBD_ligaFantasy_alg;

--
-- Consulta 1
-- Los jugadores que hayan hecho mas puntos en cada jornada

select cj.num_jornada, j.nombre as jugador, cj.puntos
from clasificacion_jug cj inner join jugadores j
    on cj.id_jugadores = j.id_jugadores
where cj.puntos = (select max(cj2.puntos)
    from clasificacion_jug cj2
    where cj2.num_jornada = cj.num_jornada )
order by cj.num_jornada;
```

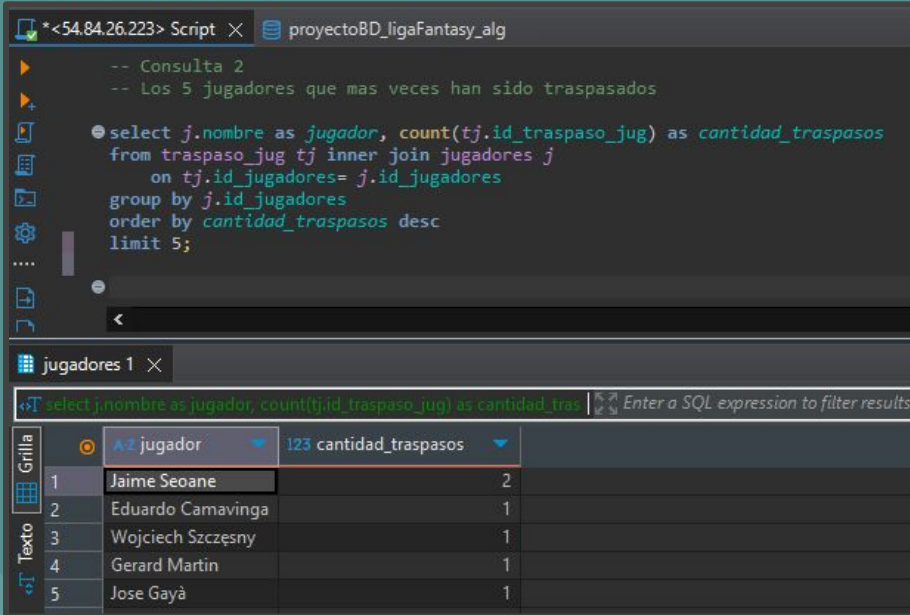
The results window shows the output of the query, displaying a table with 3 columns: num_jornada, jugador, and puntos. The table contains 24 rows of data, showing the top player for each jornada.

num_jornada	jugador	puntos
1	Ruben Gomez	25
2	Antonio Sivera	25
3	Yuri Berchiche	25
4	Vedat Muriqi	25
5	Carlos Dominguez	25
6	Fran Garcia	20
7	Gerard Martin	20
8	Yeray alvarez	20
9	Kang-In Lee	20
10	Abdon Prats	20
11	Dani Parejo	20
12	Egoitz Arana	20
13	Jon Karrikaburu	20
14	Ricardo Rodriguez	20
15	Giovani Lo Celso	20
16	Isco	20
17	Jesus Navas	20
18	Sander Berge	20
19	Radamel Falcao	20
20	Munir El Haddadi	20
21	Luis Milla	20
22	Aleksandar Sedlar	20
23	Sergi Gomez	20
24	Alvaro Tejero	20

CONSULTA 2

LOS 5 JUGADORES QUE MÁS VECES HAN SIDO TRASPASADOS

```
select j.nombre as jugador, count(tj.id_traspaso_jug) as
cantidad_traspasos
from traspaso_jug tj inner join jugadores j
    on tj.id_jugadores= j.id_jugadores
group by j.id_jugadores
order by cantidad_traspasos desc
limit 5;
```



The screenshot shows a database IDE with a script editor and a results viewer. The script editor contains the following SQL query:

```
-- Consulta 2
-- Los 5 jugadores que mas veces han sido traspasados

select j.nombre as jugador, count(tj.id_traspaso_jug) as cantidad_traspasos
from traspaso_jug tj inner join jugadores j
    on tj.id_jugadores= j.id_jugadores
group by j.id_jugadores
order by cantidad_traspasos desc
limit 5;
```

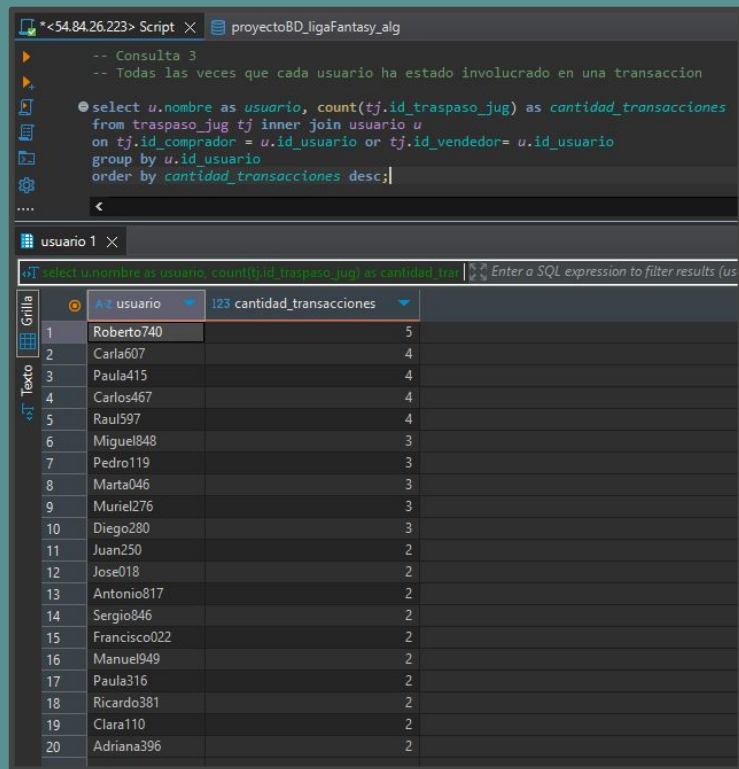
The results viewer shows the following table:

	A2 jugador	123 cantidad_traspasos
1	Jaime Seoane	2
2	Eduardo Camavinga	1
3	Wojciech Szczesny	1
4	Gerard Martin	1
5	Jose Gayà	1

CONSULTA 3

TODAS LAS VECES QUE CADA USUARIO HA ESTADO INVOLUCRADO EN UNA TRANSACCIÓN

```
select u.nombre as usuario, count(tj.id_traspaso_jug) as
cantidad_transacciones
from traspaso_jug tj inner join usuario u
on tj.id_comprador = u.id_usuario or tj.id_vendedor= u.id_usuario
group by u.id_usuario
order by cantidad_transacciones desc;
```



The screenshot shows a database IDE with a script editor and a results grid. The script editor contains the following SQL query:

```
-- Consulta 3
-- Todas las veces que cada usuario ha estado involucrado en una transaccion

select u.nombre as usuario, count(tj.id_traspaso_jug) as cantidad_transacciones
from traspaso_jug tj inner join usuario u
on tj.id_comprador = u.id_usuario or tj.id_vendedor= u.id_usuario
group by u.id_usuario
order by cantidad_transacciones desc;
```

The results grid shows the following data:

	usuario	cantidad_transacciones
1	Roberto740	5
2	Carla607	4
3	Paula415	4
4	Carlos467	4
5	Raul597	4
6	Miguel848	3
7	Pedro119	3
8	Marta046	3
9	Muriel276	3
10	Diego280	3
11	Juan250	2
12	Jose018	2
13	Antonio817	2
14	Sergio846	2
15	Francisco022	2
16	Manuel949	2
17	Paula316	2
18	Ricardo381	2
19	Clara110	2
20	Adriana396	2

CONSULTA 4

TODOS LOS JUGADORES ESPAÑOLES QUE HAYAN ESTADO INVOLUCRADOS EN ALGUNA TRANSACCIÓN Y QUE JUEGUEN EN UNA PLANTILLA CON UNA FORMACIÓN 4-3-3

```
select j.id_jugadores as jugador, j.nacionalidad, j.equipo,
       tj.fecha_traspaso, tj.precio_traspaso, u.nombre as usuario
from traspaso_jug tj inner join jugadores j
     on tj.id_jugadores= j.id_jugadores
inner join usuario u
     on j.id_propietario= u.id_usuario
inner join plantilla p
     on u.id_plantilla= p.id_plantilla
where j.nacionalidad= 'Español' and p.formacion = '4-3-3';
```

<54.84.26.223> Script X proyectoBD_ligaFantasy_alg

```
-- Consulta 4
-- Todos los jugadores españoles que hayan estado involucrados en alguna transaccion y que jueguen en una plantilla con una formacion 4-3-3

select j.id_jugadores as jugador, j.nacionalidad, j.equipo, tj.fecha_traspaso, tj.precio_traspaso, u.nombre as usuario
from traspaso_jug tj inner join jugadores j
     on tj.id_jugadores= j.id_jugadores
inner join usuario u
     on j.id_propietario= u.id_usuario
inner join plantilla p
     on u.id_plantilla= p.id_plantilla
where j.nacionalidad= 'Español' and p.formacion = '4-3-3';
```

jugadores(+) 1 X

select j.id_jugadores as jugador, j.nacionalidad, j.equipo, tj.fecha_tras

	123 jugador	A-Z nacionalidad	A-Z equipo	fecha_traspaso	123 precio_traspaso	A-Z usuario
1	10.245	Español	Sevilla FC	2024-12-12 15:45:30	56.000.000	Pedro119
2	10.302	Español	Rayo Vallecano	2024-11-02 14:58:34	88.000.000	Muriel276
3	10.356	Español	Getafe CF	2024-12-08 21:00:45	15.000.000	Muriel276
4	10.356	Español	Getafe CF	2024-12-03 13:25:15	33.000.000	Muriel276
5	10.176	Español	CA Osasuna	2024-12-07 20:20:10	12.000.000	Francisco022
6	10.178	Español	CA Osasuna	2024-11-18 10:15:00	32.000.000	Adriana396

CONSULTA 5

TODOS LOS JUGADORES QUE HAN SIDO COMPRADOS POR USUARIOS QUE HAN PAGADO UNA CANTIDAD DE TRASPASO MAYOR AL PROMEDIO DE PRESUPUESTO DE TODOS LOS USUARIOS

```
select j.nombre as jugador, u.nombre as comprador, tj.precio_traspaso
from traspaso_jug tj inner join jugadores j
    on tj.id_jugadores= j.id_jugadores
inner join usuario u
    on tj.id_comprador= u.id_usuario
where tj.precio_traspaso > (
    select avg(presupuesto)
    from usuario);
```

The screenshot shows a SQL IDE window titled 'proyectoBD_ligaFantasy_alg'. The script editor contains the following SQL query:

```
--
-- Consulta 5
-- Todos los jugadores que han sido comprados por usuarios que han pagado
-- una cantidad mayor al promedio de presupuesto de todos los usuarios

select j.nombre as jugador, u.nombre as comprador, tj.precio_traspaso
from traspaso_jug tj inner join jugadores j
    on tj.id_jugadores= j.id_jugadores
inner join usuario u
    on tj.id_comprador= u.id_usuario
where tj.precio_traspaso > (
    select avg(presupuesto)
    from usuario);
```

Below the script editor, the results are displayed in a table titled 'jugadores(+1) X'. The table has 5 columns: a selection icon, 'A-Z jugador', 'A-Z comprador', '123 precio_traspaso', and an empty column. The results are as follows:

	A-Z jugador	A-Z comprador	123 precio_traspaso	
1	Raul Albiol	Diego280	89.500.000	
2	Salvi Sanchez	Muriel276	88.000.000	
3	Marc Vidal	Diego280	73.000.000	
4	Martin Zubimendi	Manuel949	72.000.000	

6. VISTAS Y TRIGGERS

Triggers para actualizar el **ID_PROPIETARIO** de los **jugadores** y **entrenadores** cuando se produzca un traspaso.

```
-- TRIGGER PARA ACTUALIZAR EL ID_PROPIETARIO DE JUGADOR

delimiter &&

❶ create trigger actualizar_propietario_jugador
after insert on traspaso_jug
for each row
begin
    update jugadores
    set id_propietario = new.id_comprador
    where id_jugadores = new.id_jugadores;
end &&

delimiter ;

-- TRIGGER PARA ACTUALIZAR EL ID_PROPIETARIO DE ENTRENADOR

delimiter &&

❷ create trigger actualizar_propietario_entrenador
after insert on traspaso_ent
for each row
begin
    update entrenadores
    set id_propietario = new.id_comprador
    where id_entrenadores = new.id_entrenadores;
end &&

delimiter ;
```

Triggers para actualizar el **PRESUPUESTO** de los **usuarios** cuando se involucren en un traspaso y verificar si tienen esa cantidad de dinero (Si algún comprador no tuviera suficiente saltaría un error).

```
delimiter &&

create trigger actualizar_presupuesto_traspaso_jug
after insert on traspaso_jug
for each row
begin
    declare nuevo_presupuesto int;

    -- calculamos el nuevo presupuesto del comprador antes de actualizar
    select presupuesto - new.precio_traspaso into nuevo_presupuesto
    from usuario
    where id_usuario = new.id_comprador;

    -- verificamos si el presupuesto quedaría en negativo
    if nuevo_presupuesto < 0 then
        signal sqlstate '45000'
        set message_text = 'error: el comprador no tiene suficiente presupuesto para este traspaso.';
    end if;

    -- actualizamos los presupuestos si la validación se pasó
    update usuario
    set presupuesto = presupuesto - new.precio_traspaso
    where id_usuario = new.id_comprador;

    update usuario
    set presupuesto = presupuesto + new.precio_traspaso
    where id_usuario = new.id_vendedor;

end &&

delimiter ;
```

```
delimiter &&

❶ create trigger actualizar_presupuesto_traspaso_ent
after insert on traspaso_ent
for each row
begin
    declare nuevo_presupuesto int;

    ❷ -- calculamos el nuevo presupuesto del comprador antes de actualizar
    select presupuesto - new.precio_traspaso into nuevo_presupuesto
    from usuario
    where id_usuario = new.id_comprador;

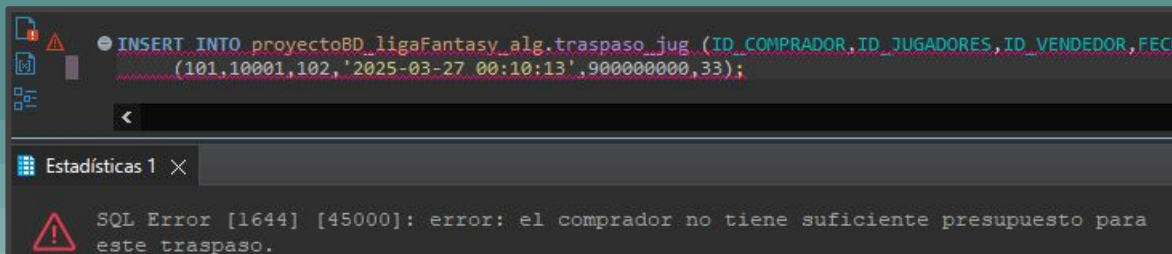
    ❸ -- verificamos si el presupuesto quedaría en negativo
    if nuevo_presupuesto < 0 then
        signal sqlstate '45000'
        set message_text = 'error: el comprador no tiene suficiente presupuesto para este traspaso.';
    end if;

    ❹ -- actualizamos los presupuestos si la validación se pasó
    update usuario
    set presupuesto = presupuesto - new.precio_traspaso
    where id_usuario = new.id_comprador;

    ❺ update usuario
    set presupuesto = presupuesto + new.precio_traspaso
    where id_usuario = new.id_vendedor;

end &&

delimiter ;
```



DATOS ANTES Y DESPUÉS DE LA EJECUCIÓN DE LOS TRIGGERS

●

INSERT INTO proyectoBD_ligaFantasy_alg.traspaso_jug (ID_COMPRAADOR,ID_JUGADORES,ID_VENDEADOR,FECHA_TRASPASO,PRECIO_TRASPASO,ID_TRASPASO_JUG) VALUES (102,10020,101,'2025-01-27 00:10:13',100000000,32);

Estadísticas 1

Name	Value
Updated Rows	1
Execute time	0.199s
Start time	Sat Mar 29 18:36:49 CET 2025
Finish time	Sat Mar 29 18:36:49 CET 2025
Query	INSERT INTO proyectoBD_ligaFantasy_alg.traspaso_jug (ID_COMPRAADOR,ID_JUGADORES,ID_VENDEADOR,FECHA_TRASPASO,PRECIO_TRASPASO,ID_TRASPASO_JUG) VALUES (102,10020,101,'2025-01-27 00:10:13',100000000,32)

19	10.020	Jugador	Español	Delantero	Real Madrid CF	12	12	171.300.000	101
20	10.020	Kylian Mbappe	Frances	Delantero	Real Madrid CF	12	12	171.300.000	102
21	10.021	M. A. L. S. C.	Al	Delantero	Real Madrid CF	12	12	171.300.000	103

19	10.020	Jugador	Español	Delantero	Real Madrid CF	12	12	171.300.000	101
20	10.020	Kylian Mbappe	Frances	Delantero	Real Madrid CF	12	12	171.300.000	102
21	10.021	M. A. L. S. C.	Al	Delantero	Real Madrid CF	12	12	171.300.000	103

Grilla	<div><div></div><div>123 ID_USUARIO</div><div></div></div>	<div><div></div><div>A-Z NOMBRE</div><div></div></div>	<div><div></div><div>123 PRESUPUESTO</div><div></div></div>	<div><div></div><div>123 ID_ADMIN</div><div></div></div>	<div><div></div><div>123 ID_PLANTILLA</div><div></div></div>
1	101	Diego280	123.230.154	[NULL]	1.001
2	102	Carla607	121.940.104	101	1.002

Grilla	<div><div></div><div>123 ID_USUARIO</div><div></div></div>	<div><div></div><div>A-Z NOMBRE</div><div></div></div>	<div><div></div><div>123 PRESUPUESTO</div><div></div></div>	<div><div></div><div>123 ID_ADMIN</div><div></div></div>	<div><div></div><div>123 ID_PLANTILLA</div><div></div></div>
1	101	Diego280	223.230.154	[NULL]	1.001
2	102	Carla607	21.940.104	101	1.002

Vistas para ver los puntos totales de cada jugador y para ver la cantidad gastada por usuario.

```
-- VER PUNTOS TOTALES DE LOS JUGADORES
```

```
● create view vista_clasificacion_jugadores as
select j.id_jugadores, j.nombre as jugador, j.equipo, sum(c.puntos) as puntos_totales
from jugadores j join clasificacion_jug c
on j.id_jugadores = c.id_jugadores
group by j.id_jugadores, j.nombre, j.equipo
order by puntos_totales desc;
```

```
-- VISTA PARA VER CANTIDAD GASTADA POR USUARIO
```

```
● create view vista_resumen_traspasos_usuario as
select u.id_usuario, u.nombre as usuario, sum(t.precio_traspaso) as total_gastado
from traspaso_jug t inner join usuario u
on t.id_comprador = u.id_usuario
group by u.id_usuario, u.nombre
order by total_gastado desc;
```

	123 id_jugadores	A-Z jugador	A-Z equipo	123 puntos_totales
1	10.416	Darwin Machis	Real Valladolid	78
2	10.081	Iñigo Lekue	Athletic Club	77
3	10.164	Jon Karrikaburu	Real Sociedad	76
4	10.142	alex Remiro	Real Sociedad	74
5	10.154	Martin Zubimendi	Real Sociedad	74
6	10.184	Iker Muñoz	CA Osasuna	74
7	10.408	Mario Martín	Real Valladolid	74
8	10.065	Antoine Griezmann	Atletico de Madrid	73
9	10.220	Vladyslav Krapavtsov	Girona FC	73

	123 id_usuario	A-Z usuario	123 total_gastado
1	101	Diego280	272.000.000
2	102	Carla607	135.800.000
3	112	Muriel276	121.000.000
4	106	Miguel848	96.000.000
5	114	Francisco022	82.000.000
6	115	Manuel949	72.000.000
7	111	Paula415	69.000.000
8	107	Pedro119	68.500.000
9	119	Clara110	66.000.000
10	113	Sergio846	65.000.000
11	120	Adriana396	52.000.000
12	108	Juan250	51.000.000
13	103	Antonio817	46.000.000
14	117	Roberto740	35.000.000
15	109	Marta046	24.000.000
16	105	Carlos467	13.900.000
17	116	Paula316	12.000.000
18	110	Jose018	8.000.000
19	118	Ricardo381	7.500.000

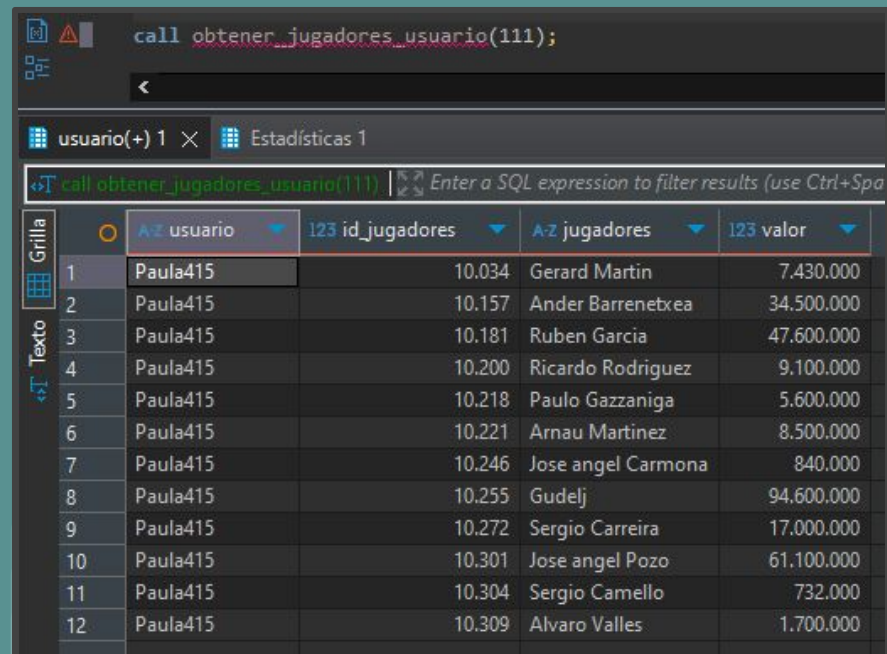
7. PROCEDIMIENTOS Y FUNCIONES

Procedimiento para ver todos los jugadores de un usuario.

```
-- VER TODOS LOS JUGADORES DE UN USUARIO
delimiter &&

create procedure obtener_jugadores_usuario(in id_usuario int)
begin
    select u.nombre as usuario, j.id_jugadores, j.nombre as jugadores, j.valor
    from jugadores j inner join usuario u
        on u.id_usuario = j.id_propietario
    where j.id_propietario = id_usuario and id_usuario= u.id_usuario;
end &&

delimiter ;
```



The screenshot shows a database client interface. At the top, a SQL command is entered: `call obtener_jugadores_usuario(111);`. Below the command bar, the results are displayed in a table. The table has four columns: 'usuario', 'id_jugadores', 'jugadores', and 'valor'. The results show 12 rows of data for user 111, all with the username 'Paula415'. The players listed are Gerard Martin, Ander Barrenetxea, Ruben Garcia, Ricardo Rodriguez, Paulo Gazzaniga, Arnau Martinez, Jose angel Carmona, Gudelj, Sergio Carreira, Jose angel Pozo, Sergio Camello, and Alvaro Valles, with their respective values.

	A-Z usuario	123 id_jugadores	A-Z jugadores	123 valor
1	Paula415	10.034	Gerard Martin	7.430.000
2	Paula415	10.157	Ander Barrenetxea	34.500.000
3	Paula415	10.181	Ruben Garcia	47.600.000
4	Paula415	10.200	Ricardo Rodriguez	9.100.000
5	Paula415	10.218	Paulo Gazzaniga	5.600.000
6	Paula415	10.221	Arnau Martinez	8.500.000
7	Paula415	10.246	Jose angel Carmona	840.000
8	Paula415	10.255	Gudelj	94.600.000
9	Paula415	10.272	Sergio Carreira	17.000.000
10	Paula415	10.301	Jose angel Pozo	61.100.000
11	Paula415	10.304	Sergio Camello	732.000
12	Paula415	10.309	Alvaro Valles	1.700.000

Procedimiento para ver el valor de los equipos de los usuarios.

call obtener_precioEquipos_usuario();

usuario 1 × Estadísticas 1

call obtener_precioEquipos_usuario() | Enter a SQL expression to filter results (use Ctrl+Space)

	id_usuario	A-Z usuario	precio_total_equipo	numero_jugadores
1	117	Roberto740	845.010.000	16
2	120	Adriana396	783.262.000	24
3	102	Carla607	705.255.000	20
4	101	Diego280	672.656.000	19
5	112	Muriel276	607.000.000	16
6	114	Francisco022	606.996.000	19
7	118	Ricardo381	583.261.000	18
8	107	Pedro119	537.146.000	23
9	106	Miguel848	533.974.000	15
10	109	Marta046	477.760.000	14
11	103	Antonio817	475.985.000	16
12	104	Raul597	418.681.000	20
13	119	Clara110	372.960.000	18
14	116	Paula316	350.252.000	10
15	110	Jose018	316.694.000	15
16	108	Juan250	307.570.000	13
17	111	Paula415	288.702.000	12
18	105	Carlos467	285.433.000	12
19	113	Sergio846	245.632.000	12
20	115	Manuel949	132.500.400	14

```
-- VER LOS EQUIPOS MAS CAROS
```

```
delimiter &&
```

```
create procedure obtener_precioEquipos_usuario()
```

```
begin
```

```
    select u.id_usuario, u.nombre as usuario, sum(j.valor) as precio_total_equipo, count(j.id_jugadores) as numero_jugadores
    from usuario u left join jugadores j
        on u.id_usuario = j.id_propietario
    group by u.id_usuario, u.nombre
    order by precio_total_equipo desc;
```

```
end &&
```

```
delimiter ;
```

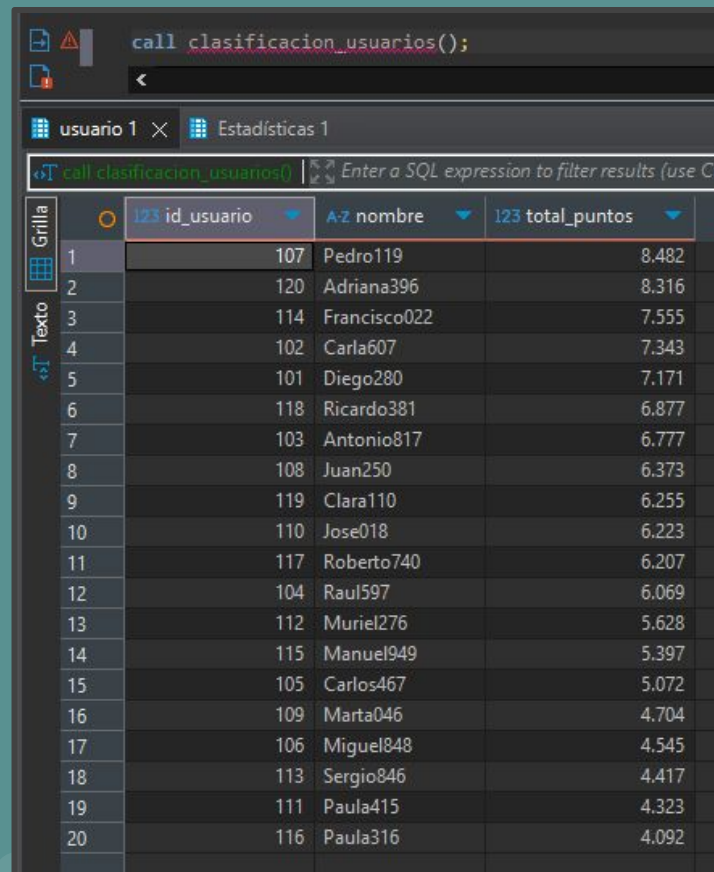
Procedimiento para ver la clasificación de los usuarios.
(Procedimiento con una función implementada).

```
-- VER CLASIFICACION USUARIOS

delimiter &&

create procedure clasificacion_usuarios()
begin
    select u.id_usuario, u.nombre, obtener_puntos_usuario(u.id_usuario) as total_puntos
    from usuario u
    order by total_puntos desc;
end &&

delimiter ;
```



call clasificacion_usuarios();

usuario 1 x Estadísticas 1

call clasificacion_usuarios() | Enter a SQL expression to filter results (use C

	id_usuario	nombre	total_puntos
1	107	Pedro119	8.482
2	120	Adriana396	8.316
3	114	Francisco022	7.555
4	102	Carla607	7.343
5	101	Diego280	7.171
6	118	Ricardo381	6.877
7	103	Antonio817	6.777
8	108	Juan250	6.373
9	119	Clara110	6.255
10	110	Jose018	6.223
11	117	Roberto740	6.207
12	104	Raul597	6.069
13	112	Muriel276	5.628
14	115	Manuel949	5.397
15	105	Carlos467	5.072
16	109	Marta046	4.704
17	106	Miguel848	4.545
18	113	Sergio846	4.417
19	111	Paula415	4.323
20	116	Paula316	4.092

Función para ver los puntos de un usuario.

```
-- FUNCION VER PUNTOS DEL USUARIO

delimiter &&

create function obtener_puntos_usuario(id_usuario_param int)
returns int
deterministic
begin
    declare total_puntos int default 0;

    select ifnull(sum(cj.puntos), 0) + ifnull(sum(ce.puntos), 0) into total_puntos
    from usuario u left join jugadores j
        on u.id_usuario = j.id_propietario
    left join clasificacion_jug cj on j.id_jugadores = cj.id_jugadores
    left join entrenadores e on u.id_usuario = e.id_propietario
    left join clasificacion_ent ce on e.id_entrenadores = ce.id_entrenadores
    where u.id_usuario = id_usuario_param;

    return total_puntos;
end &&

delimiter ;

select obtener_puntos_usuario(110);
```

```
select obtener_puntos_usuario(110);|
```



tados 1



obtener_puntos_usuario(110) | Enter a SQL expres

123 obtener_puntos_usuario(110)

6.223

Función para ver la cantidad de jugadores de un usuario.

```
-- VER CANTIDAD DE JUGADORES DE UN USUARIO

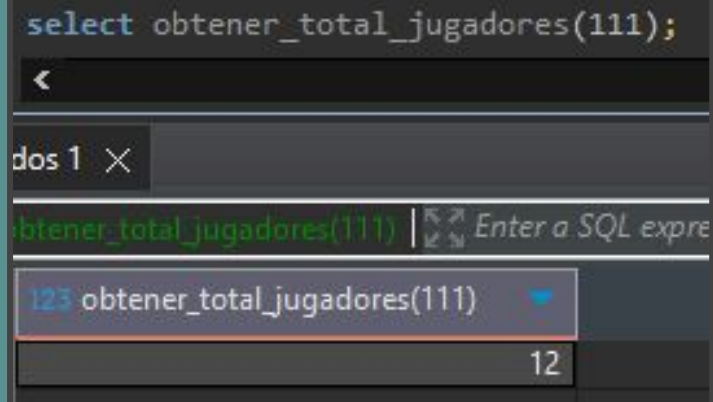
delimiter &&

create function obtener_total_jugadores(id_usuario int)
returns int
deterministic
begin
    declare total_jugadores int default 0;

    select count(*)
    into total_jugadores
    from jugadores j
    where j.id_propietario = id_usuario;

    return total_jugadores;
end &&

delimiter ;
```



The screenshot shows a SQL client window with a query editor and a results pane. The query editor contains the SQL statement `select obtener_total_jugadores(111);`. Below the editor, there is a tab labeled "dos 1" and a search bar. The results pane displays the output of the function call, which is the number 12.

```
select obtener_total_jugadores(111);
```

obtener_total_jugadores(111)
12

6. CONCLUSIÓN

En este proyecto se ha puesto en práctica todos los conocimientos que hemos adquirido durante el curso en la asignatura de Base de Datos, reforzando lo aprendido.

Desde mi parecer, este ejercicio me ha sido muy útil para establecer conexiones entre la materia que hemos aprendido. Además de usar los conocimientos aprendidos, también me ha ayudado para conseguir soltura a la hora de trabajar con las herramientas que nos han aportado y con el entorno de trabajo.

El trabajo es mejorable en ciertos aspectos, con la experiencia necesaria y el conocimiento se podrían implementar y mejorar características como los triggers, algunas tablas o incluso incrementar la carga de datos. Aún así, pienso que el proyecto ha quedado bien y me gustaría seguir trabajando en él.

Enlace de GitHub:

