

Liga Española de Fútbol

Base de Datos

Proyecto de Base de Datos

Álvaro Labrador

IES Alixar

16/06/2024

Índice

1. DESCRIPCIÓN	3
2. MODELO ENTIDAD RELACIÓN	3
3. MODELO RELACIONAL	4
4. DISEÑO RELACIONAL	5
5. CARGA MASIVA DE DATOS	6
6. CONSULTAS	8
7. VISTAS	11
8. FUNCIONES	11
9. PROCEDIMIENTOS	14
10. TRIGGERS	17
11. ENLACE GITHUB	19
12. CONCLUSIÓN	19

1. DESCRIPCIÓN

Nuestro proyecto se centra en el diseño e implementación de una base de datos relacional que aborda la gestión de datos de una liga de fútbol. Este trabajo busca ofrecer una herramienta eficaz para almacenar, organizar y acceder a la información para el análisis estadístico de la Liga Española de Fútbol.

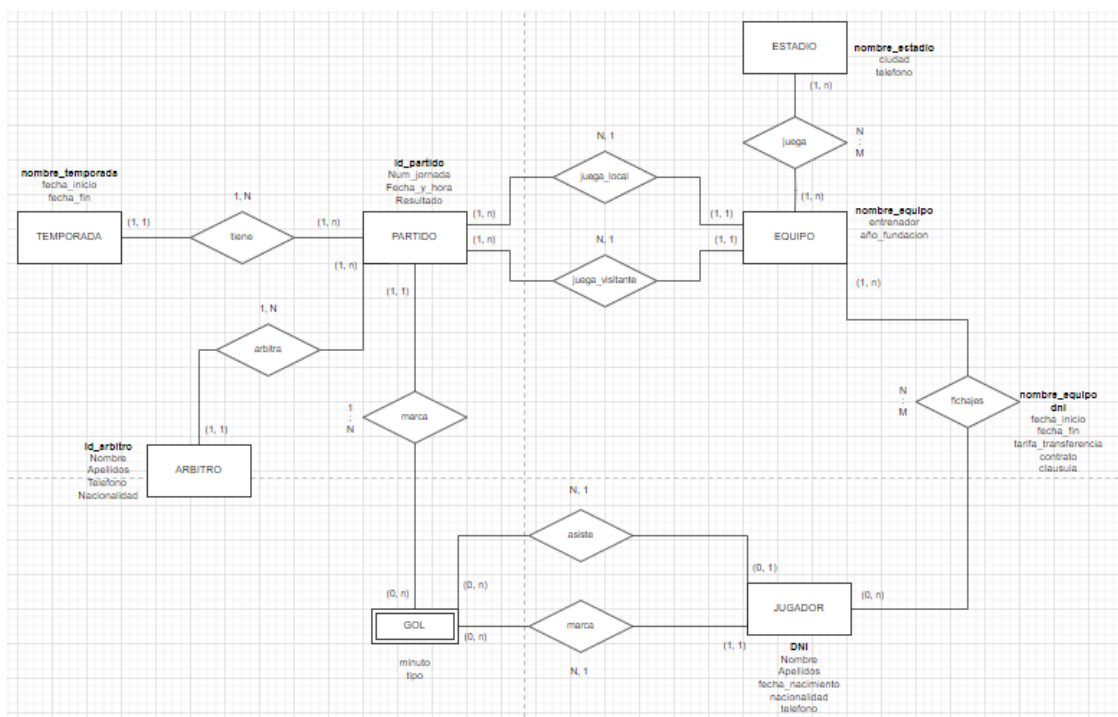
El proyecto se desarrolla paso a paso desde el diseño de la base de datos, su implementación en el SGBD MySql Server, carga masiva de datos y ejecución de consultas, funciones, procedimientos y triggers. Finalizamos el proyecto con unas conclusiones y valoración personal.

La información que queremos almacenar está relacionada con las temporadas, equipos, jugadores, partidos, árbitros y estadios dentro del ámbito de la Liga Española de Fútbol.

2. MODELO ENTIDAD RELACIÓN

Para realizar nuestro proyecto, debemos empezar con el modelo entidad relación.

Sus entidades más importantes son **partido**, **equipo** y **jugador**. En una temporada se juegan diferentes partidos, en estos partidos juegan un equipo de local y otro de visitante. Los equipos juegan en un **estadio** y fichan a jugadores, que pueden marcar o asistir un **gol** en un determinado partido que son arbitrados por un **árbitro**. El modelo sería el siguiente:



3. MODELO RELACIONAL

Tras realizar el modelo entidad relacional, realizaremos el modelo relacional, el cual está compuesto por 9 tablas. Con este modelo se consigue proporcionar una manera más clara de representar datos y sus correspondientes Primary Keys(PK) y Foreign Keys(FK):

TEMPORADA (nombre_temporada, fecha_inicio, fecha_fin)

PK: nombre_temporada

PARTIDO(id_partido, num_jornada, fecha_y_hora, resultado, nombre_temporada, id_arbitro, equipo_local, equipo_visitante)

PK: id_partido

FK: nombre_temporada REFERENCIA a TEMPORADA

FK: id_arbitro REFERENCIA a ÁRBITRO

FK: equipo_local REFERENCIA a EQUIPO

FK: equipo_visitante REFERENCIA a EQUIPO

EQUIPO(nombre_equipo, entrenador, año_fundación)

PK: nombre_equipo

JUEGA(nombre_equipo, nombre_estadio)

PK: nombre_equipo, nombre_estadio

FK: nombre_equipo REFERENCIA a EQUIPO

FK: nombre_estadio REFERENCIA a ESTADIO

ESTADIO(nombre_estadio, ciudad, teléfono)

PK: nombre_estadio

JUGADOR(dni, nombre, apellidos, fecha_nacimiento, nacionalidad, teléfono)

PK: dni

FICHAJES(nombre_equipo, dni, fecha_inicio, fecha_fin, tarifa_transferencia, contrato, clausula)

PK: dni, nombre_equipo

FK: dni REFERENCIA a JUGADOR

FK: nombre_equipo REFERENCIA a EQUIPO

ÁRBITRO(id_arbitro, nombre, apellidos, teléfono, nacionalidad)

PK: id_arbitro

GOL(minuto, tipo, **jugador_marca**, **jugador_asiste**, **id_partido**)

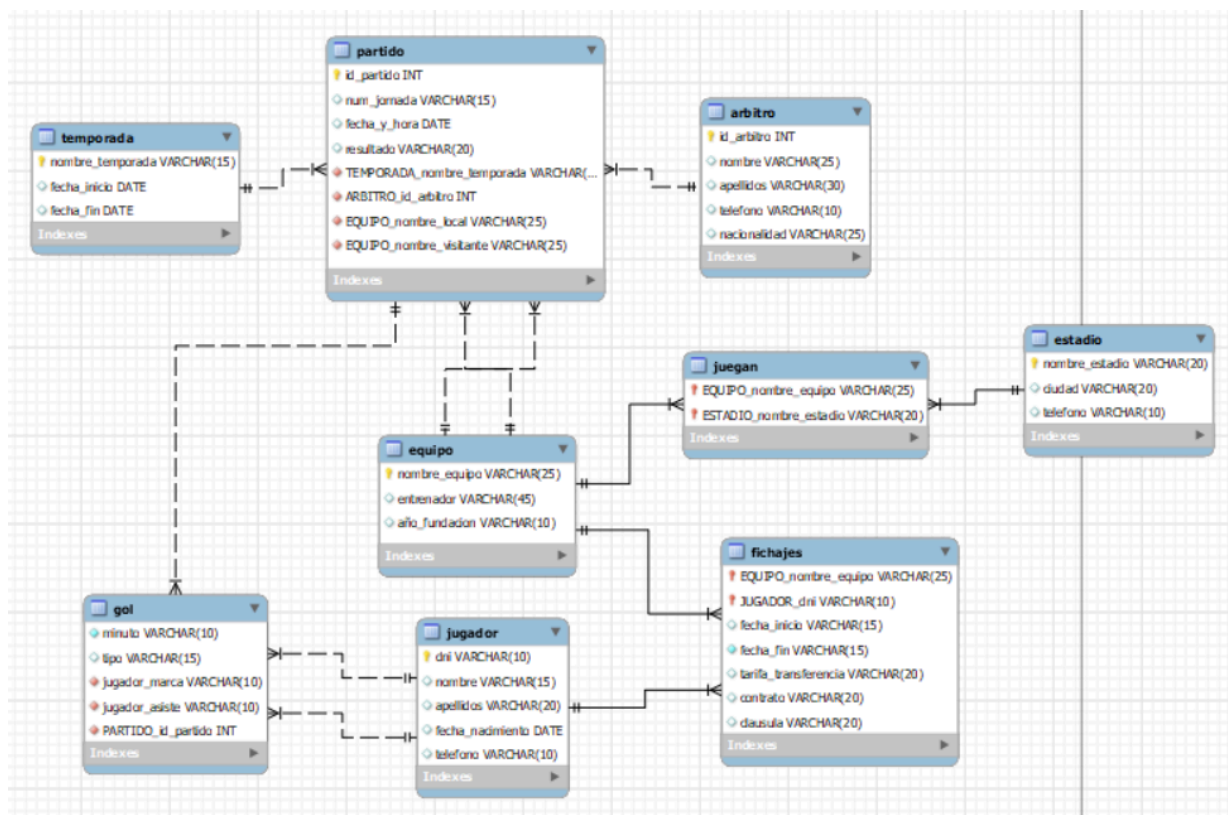
FK: jugador_marca REFERENCIA a JUGADOR

FK: jugador_asiste REFERENCIA a JUGADOR

FK: id_partido REFERENCIA a PARTIDO

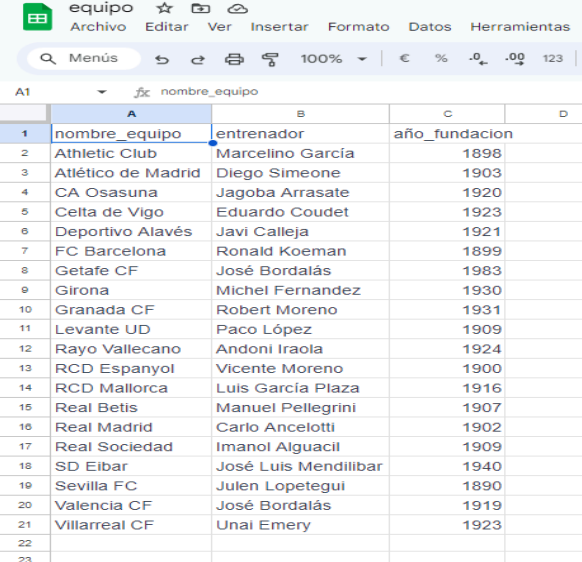
4. DISEÑO RELACIONAL

Una vez hecho el modelo relacional lo pasamos a MySQL Workbench. El diseño sería el siguiente:



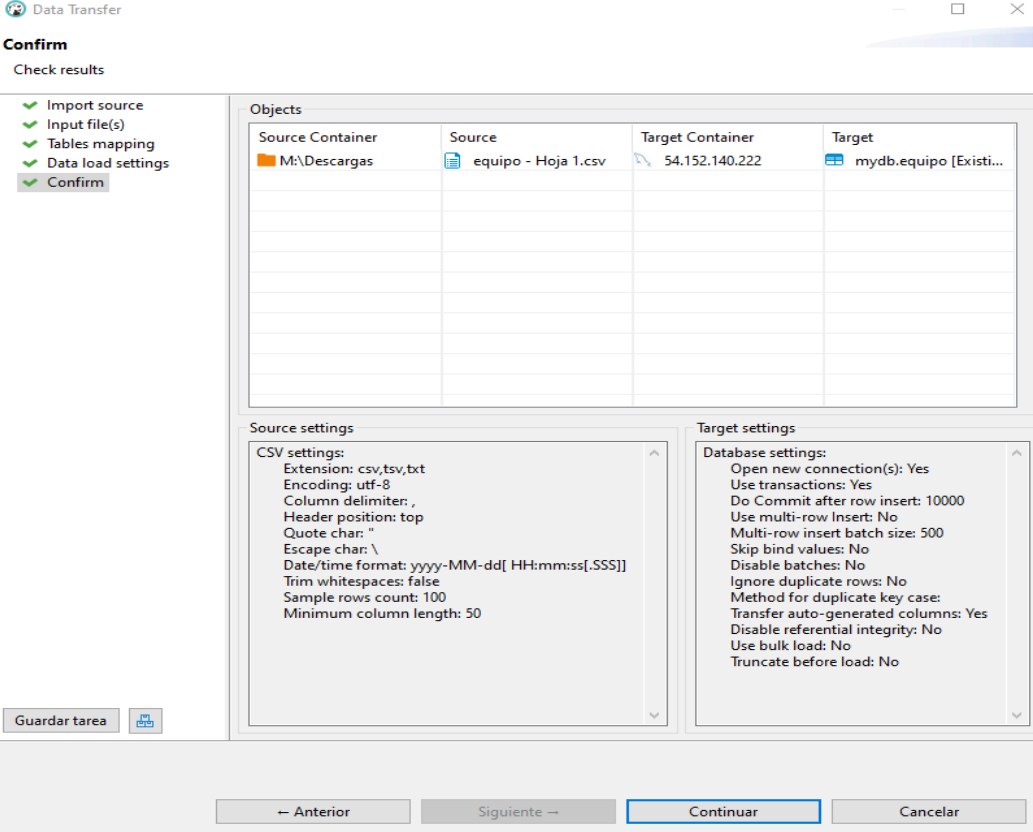
5. CARGA MASIVA DE DATOS

Para realizar consultas en nuestra base de datos es necesario tener datos en cada una de las tablas. Para ello, hemos obtenido los datos gracias a la herramienta chatgpt. Luego, estos datos los hemos pasado a una hoja de cálculo y hemos descargado su csv para posteriormente importarlo en cada una de las tablas.



The screenshot shows a spreadsheet application with a menu bar (Archivo, Editar, Ver, Insertar, Formato, Datos, Herramientas) and a toolbar. The active sheet is named 'nombre_equipo'. The table contains 21 rows of data, with columns for team name, coach, and founding year.

	A	B	C	D
1	nombre_equipo	entrenador	año_fundacion	
2	Athletic Club	Marcelino García	1898	
3	Atlético de Madrid	Diego Simeone	1903	
4	CA Osasuna	Jagoba Arrasate	1920	
5	Celta de Vigo	Eduardo Coudet	1923	
6	Deportivo Alavés	Javi Calleja	1921	
7	FC Barcelona	Ronald Koeman	1899	
8	Getafe CF	José Bordalás	1983	
9	Girona	Michel Fernandez	1930	
10	Granada CF	Robert Moreno	1931	
11	Levante UD	Paco López	1909	
12	Rayo Vallecano	Andoni Iraola	1924	
13	RCD Espanyol	Vicente Moreno	1900	
14	RCD Mallorca	Luis García Plaza	1916	
15	Real Betis	Manuel Pellegrini	1907	
16	Real Madrid	Carlo Ancelotti	1902	
17	Real Sociedad	Imanol Alguacil	1909	
18	SD Eibar	José Luis Mendilibar	1940	
19	Sevilla FC	Julen Lopetegui	1890	
20	Valencia CF	José Bordalás	1919	
21	Villarreal CF	Unai Emery	1923	
22				
23				



The screenshot shows the 'Data Transfer' tool interface. The 'Confirm' step is active, showing a list of objects to be transferred. The source is 'M:\Descargas' and the target is 'mydb.equipo [Existi...]'.

Confirm
Check results

- ✓ Import source
- ✓ Input file(s)
- ✓ Tables mapping
- ✓ Data load settings
- ✓ Confirm

Source Container	Source	Target Container	Target
M:\Descargas	equipo - Hoja 1.csv	54.152.140.222	mydb.equipo [Existi...]

Source settings

CSV settings:

- Extension: csv,tsv,txt
- Encoding: utf-8
- Column delimiter: ,
- Header position: top
- Quote char: "
- Escape char: \
- Date/time format: yyyy-MM-dd[HH:mm:ss[.SSS]]
- Trim whitespaces: false
- Sample rows count: 100
- Minimum column length: 50

Target settings

Database settings:

- Open new connection(s): Yes
- Use transactions: Yes
- Do Commit after row insert: 10000
- Use multi-row Insert: No
- Multi-row insert batch size: 500
- Skip bind values: No
- Disable batches: No
- Ignore duplicate rows: No
- Method for duplicate key case: Transfer auto-generated columns: Yes
- Disable referential integrity: No
- Use bulk load: No
- Truncate before load: No

Guardar tarea

Anterior Siguiente Continuar Cancelar

6. CONSULTAS

-- 1. Mostrar los nombres de los equipos junto con la suma de los minutos en los que sus jugadores han anotado goles y lo ordena en orden descendiente según la cantidad total de minutos de goles anotados.

```
select e.nombre_equipo, SUM(g.minuto) as suma_minutos
from equipo e inner join fichajes f
on e.nombre_equipo = f.EQUIPO_nombre_equipo inner join jugador j
on f.JUGADOR_dni = j.dni left join gol g
on j.dni = g.jugador_marca
group by e.nombre_equipo
order by suma_minutos desc;
```

	nombre_equipo	suma_minutos
1	Villarreal CF	132
2	Granada CF	112
3	Sevilla FC	100
4	RCD Mallorca	90
5	Atlético de Madrid	87
6	Celta de Vigo	77
7	Girona	67
8	Real Betis	54
9	Getafe CF	43
10	RCD Espanyol	39
11	Real Sociedad	36
12	FC Barcelona	[NULL]
13	Deportivo Alavés	[NULL]

```
-- 2. Obtener el nombre del estadio y la cantidad total de partidos jugados en ese
estadio donde su numero de telefono se encuentra entre 100000000 y 200000000
select nombre_estadio, telefono, count(*) as total_partidos
from estadio e inner join juegan j
on e.nombre_estadio = j.ESTADIO_nombre_estadio inner join partido p
on j.EQUIPO_nombre_equipo = p.EQUIPO_nombre_local or
j.EQUIPO_nombre_equipo = p.EQUIPO_nombre_visitante
group by nombre_estadio
having e.telefono BETWEEN 100000000 and 200000000;
```

	nombre_estadio	telefono	total_partidos
1	Ciutat de València	109876543	114
2	Pizjuán	109876543	114
3	San Mamés	123456789	114

```
-- 3. Obtener los jugadores que su nombre comienza por la letra 'J' y cuya tarifa es
mayor que el promedio de la tarifa de todos los jugadores
```

```
select j.nombre, j.apellidos, f.EQUIPO_nombre_equipo
from jugador j inner join fichajes f
on j.dni = f.JUGADOR_dni
where f.tarifa_transferencia > (
    SELECT avg(f2.tarifa_transferencia)
    from fichajes f2
) and j.nombre like 'J%';
```

	nombre	apellidos	EQUIPO_nombre
1	Javier	Avila	Real Madrid
2	Jaime	Ulloa	Real Madrid
3	Juan	Rodríguez	Athletic Club
4	Jesús	Delgado	Atlético de Madrid
5	Javier	Blanco	Atlético de Madrid
6	Juan Pablo	Escobar	Sevilla FC
7	Juan Carlos	Santander	Real Sociedad
8	Jaime	Lira	Getafe CF

-- 4. Obtener la cantidad de partidos jugados por cada equipo en una temporada específica y el resultado de cada partido

```
select nombre_equipo, t.nombre_temporada, count(p.id_partido) as
Partidos_jugados, group_concat(p.resultado) as Resultados
from equipo e inner join partido p
on e.nombre_equipo = p.EQUIPO_nombre_local or e.nombre_equipo =
p.EQUIPO_nombre_visitante inner join temporada t
on p.TEMPORADA_nombre_temporada = t.nombre_temporada
group by nombre_equipo , nombre_temporada;
```

	nombre_equipo	nombre_temporada	Partidos_jugados	Resultados
1	Athletic Club	Temporada 1	38	0-2,2-1,1-1,2-0,1-0,0-1,2-1,0
2	Athletic Club	Temporada 2	38	1-2,1-0,1-1,2-0,1-0,0-1,2-1,0
3	Athletic Club	Temporada 3	38	1-2,1-0,2-1,2-0,1-0,0-1,2-1,0
4	Atlético de Madrid	Temporada 1	38	1-0,1-0,2-0,1-2,0-1,1-1,2-0,0
5	Atlético de Madrid	Temporada 2	38	0-2,1-0,2-0,1-2,0-1,1-1,2-0,0
6	Atlético de Madrid	Temporada 3	38	0-2,0-2,0-0,1-2,0-1,1-1,2-0,0
7	CA Osasuna	Temporada 1	38	2-0,2-1,0-1,2-1,2-0,0-2,2-0,0
8	CA Osasuna	Temporada 2	38	0-0,1-0,0-1,2-1,2-0,0-2,2-0,0

-- 5. Obtener al jugador que ha marcado 2 goles en la Temporada 1

```
select nombre, nombre_temporada, count(g.jugador_marca) as
Goles_marcados
from jugador j inner join gol g
on j.dni = g.jugador_marca inner join partido p
on g.PARTIDO_id_partido = p.id_partido inner join temporada t
on p.TEMPORADA_nombre_temporada = t.nombre_temporada
group by j.nombre, t.nombre_temporada
having Goles_marcados='2' and nombre_temporada='Temporada 1';
```

	nombre	nombre_temporada	Goles_marcados
1	Marcos	Temporada 1	2
2	Humberto	Temporada 1	2

7. VISTAS

-- Primera vista

```
create view primera as
    select e.nombre_equipo, SUM(g.minuto) as suma_minutos
    from equipo e inner join fichajes f
    on e.nombre_equipo = f.EQUIPO_nombre_equipo inner join jugador
j
    on f.JUGADOR_dni = j.dni inner join gol g
    on j.dni = g.jugador_marca
    group by e.nombre_equipo
    order by suma_minutos desc;
```

-- Segunda vista

```
create view segunda as
    select nombre_estadio, telefono, count(*) as total_partidos
    from estadio e inner join juegan j
    on e.nombre_estadio = j.ESTADIO_nombre_estadio inner join
partido p
    on j.EQUIPO_nombre_equipo = p.EQUIPO_nombre_local or
    j.EQUIPO_nombre_equipo = p.EQUIPO_nombre_visitante
    group by nombre_estadio
    having e.telefono BETWEEN 100000000 and 200000000;
```

8. FUNCIONES

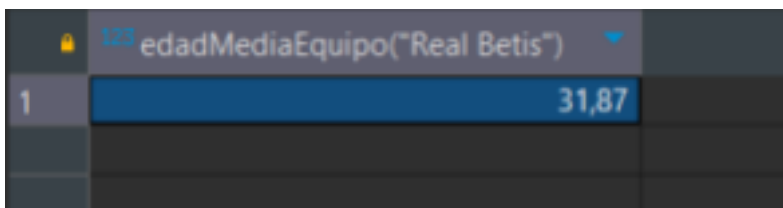
-- 1. Esta función toma el nombre del equipo como parámetro y devuelve la edad media de los jugadores del equipo

```
delimiter $$
create function edadMediaEquipo(nombre_equipo varchar(25))
returns decimal(10, 2)
deterministic
begin
    declare edad_total decimal(10, 2);
    declare cantidad_jugadores int;

    select sum(timestampdiff(year, fecha_nacimiento, curdate())) into
edad_total
    from jugador
    where dni in (select jugador_dni from fichajes where
equipo_nombre_equipo = nombre_equipo);

    select count(*) into cantidad_jugadores
    from fichajes
    where equipo_nombre_equipo = nombre_equipo;

    return edad_total / cantidad_jugadores;
end $$
delimiter ;
select edadMediaEquipo("Real Betis");
```



The screenshot shows a database query result in a dark-themed interface. The query is 'edadMediaEquipo("Real Betis")'. The result is a single row with the value '31,87'.

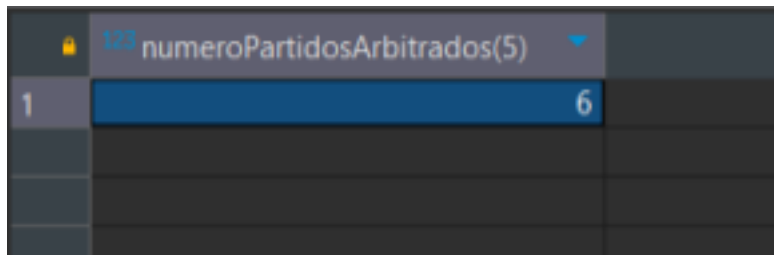
	edadMediaEquipo("Real Betis")
1	31,87

-- 2. Esta función toma el id del árbitro y devuelve el número de partidos que ha arbitrado ese árbitro

```
delimiter $$
create function numeroPartidosArbitrados(arbitro_id int)
returns int
deterministic
begin
    declare num_partidos int;

    select count(*) into num_partidos
    from arbitro
    where id_arbitro = arbitro_id;

    if num_partidos > 0 then
        select count(*)
        into num_partidos
        from partido
        where arbitro_id_arbitro = arbitro_id;
        return num_partidos;
    else
        return null;
    end if;
end $$
delimiter ;
select numeroPartidosArbitrados(5);
```



	123 numeroPartidosArbitrados(5)	
1	6	

9. PROCEDIMIENTOS

-- 1. Procedimiento para seleccionar el nombre de los equipos, el nombre del entrenador y la cantidad de jugadores que tiene cada equipo

```
delimiter $$
create procedure consultarEquipos()
begin
    select
        equipo.nombre_equipo,
        equipo.entrenador,
        count(jugador.dni) as cantidad_jugadores
    from
        equipo
    inner join
        fichajes on equipo.nombre_equipo =
fichajes.equipo_nombre_equipo
    inner join
        jugador on fichajes.jugador_dni = jugador.dni
    group by equipo.nombre_equipo, equipo.entrenador
    order by equipo.nombre_equipo;
end $$
delimiter ;
call consultarequipos();
```

	nombre_equipo	entrenador	cantidad_jugadores	
1	Athletic Club	Marcelino García	15	
2	Atlético de Madrid	Diego Simeone	15	
3	CA Osasuna	Jagoba Arrasate	15	
4	Celta de Vigo	Eduardo Coudet	15	
5	Deportivo Alavés	Javi Calleja	15	
6	FC Barcelona	Ronald Koeman	15	
7	Getafe CF	José Bordalás	15	
8	Girona	Michel Fernandez	15	

```

-- 2. Procedimiento que utiliza un cursor para hacer un informe de los estadios
delimiter $$
create procedure resumenEstadios()
begin
    declare done int default false;
    declare estadio_nombre varchar(20);
    declare estadio_ciudad varchar(20);
    declare estadio_telefono varchar(20);
    declare resumen varchar(5000) default '';
    declare contador int default 1;
    -- Cursor para recorrer los estadios
    declare cursor_estadios cursor for
        select nombre_estadio, ciudad, telefono from estadio;

    -- Manejador para el fin del cursor
    declare continue handler for not found set done = true;
    -- Encabezado
    set resumen = concat(resumen, '===== RESUMEN DE
ESTADIOS =====\n');
    open cursor_estadios;

    leer_estadios: loop
        fetch cursor_estadios into estadio_nombre, estadio_ciudad,
estadio_telefono;
        if done then
            leave leer_estadios;
        end if;

        -- Informacion de cada estadio
        set resumen = concat(resumen, '\n', contador, '. nombre: ',
estadio_nombre, ', ciudad: ', estadio_ciudad, ', teléfono: ',
estadio_telefono);
        set contador = contador + 1;
    end loop leer_estadios;
    -- Mostramos el informe
    select resumen;

    close cursor_estadios;
end $$
delimiter ;
call resumenEstadios();

```

```

===== RESUMEN DE ESTADIOS =====
1. Nombre: Benito Villamarín, Ciudad: Sevilla, Teléfono: 876543210
2. Nombre: Camp Nou, Ciudad: Barcelona, Teléfono: 234567890
3. Nombre: Cerámica, Ciudad: Villarreal, Teléfono: 567890123
4. Nombre: Ciutat de València, Ciudad: Valencia, Teléfono: 109876543
5. Nombre: Coliseum, Ciudad: Getafe, Teléfono: 567890123
6. Nombre: El Sadar, Ciudad: Pamplona, Teléfono: 765432109
7. Nombre: Estadio de Balaídos, Ciudad: Vigo, Teléfono: 345678901
8. Nombre: Estadio de Ipurua, Ciudad: Eibar, Teléfono: 456789012
9. Nombre: Estadio de Vallecas, Ciudad: Madrid, Teléfono: 210987654
10. Nombre: La Rosaleda, Ciudad: Málaga, Teléfono: 123456789
11. Nombre: Los Cármenes, Ciudad: Granada, Teléfono: 654321098
12. Nombre: Mendizorrosa, Ciudad: Vitoria, Teléfono: 789012345
13. Nombre: Mestalla, Ciudad: Valencia, Teléfono: 987654321
14. Nombre: Montilivi, Ciudad: Girona, Teléfono: 901234567
15. Nombre: Pizjuán, Ciudad: Sevilla, Teléfono: 109876543
16. Nombre: Príncipe de Asturias, Ciudad: Sanlúcar la Mayor, Teléfono: 123456789
17. Nombre: Príncipe de Asturias, Ciudad: Sanlúcar la Mayor, Teléfono: 123456789
18. Nombre: RCDE Stadium, Ciudad: Barcelona, Teléfono: 890123456
19. Nombre: Reale Arena, Ciudad: San Sebastián, Teléfono: 876543210
20. Nombre: San Mamés, Ciudad: Bilbao, Teléfono: 123456789
21. Nombre: Santiago Bernabéu, Ciudad: Madrid, Teléfono: 543210987
22. Nombre: Son Moix, Ciudad: Palma de Mallorca, Teléfono: 321098765
23. Nombre: Wanda Metropolitano, Ciudad: Madrid, Teléfono: 987654321

```

-- 3. Procedimiento en el que se utiliza la función 'edadMediaEquipo' en el que se calcula la edad media del equipo que juega en el estadio que se recibe como parámetro

delimiter \$\$

create procedure calcularEdadMediaEquipoEnEstadio(**in** nombre_estadio **varchar**(20))

begin

declare nombre_equipo_en_estadio **varchar**(25);

declare edad_media **decimal**(10, 2);

select equipo_nombre_equipo **into** nombre_equipo_en_estadio

from juegan

where estadio_nombre_estadio = nombre_estadio;

set edad_media = edadMediaEquipo(nombre_equipo_en_estadio);

select concat('La edad media de los jugadores del equipo que juega en el estadio ', nombre_estadio, ' es: ', edad_media, ' años') **as** resultado;

end \$\$

delimiter ;

call calcularEdadMediaEquipoEnEstadio('Benito Villamarín');

resultado	
1	La edad media de los jugadores del equipo que juega en el estadio Benito Villamarín es: 31.87 años

10. TRIGGERS

-- 1. Creamos un trigger en el que al insertar un jugador, si su nombre es nulo mandará un mensaje de error personalizado.

```
delimiter $$
create trigger before_insert_jugador
before insert on jugador
for each row
begin

    if new.nombre is null then
        signal sqlstate '45000' set message_text = 'Es
obligatorio introducir el nombre del jugador';
    end if;

end $$
delimiter ;

insert into jugador (dni, nombre, apellidos, fecha_nacimiento,
telefono)
values ('43218401F', null, 'Sanchez', '2003-04-27', '123456789');
```



SQL Error [1644] [45000]: Es obligatorio introducir el nombre del jugador



Detalles ↓



En el caso de no tener el trigger en nuestra base de datos, mandaría el siguiente error ya que en las propiedades de la tabla de jugador, específicamente en el nombre tenemos activado que no sea null:



SQL Error [1048] [23000]: Column 'nombre' cannot be null



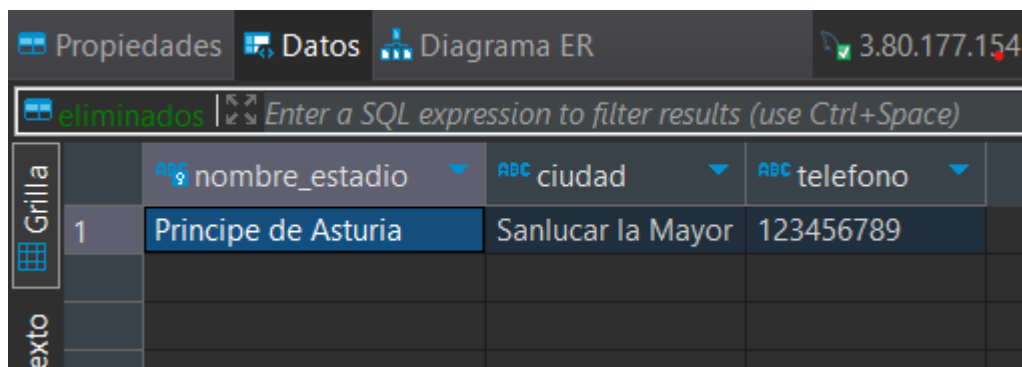
Detalles ↓



Column Name	#	Data Type	Not Null	Auto Increment	Key
ABC dni	1	varchar(10)	[v]	[]	PRI
ABC nombre	2	varchar(25)	[v]	[]	
ABC apellidos	3	varchar(20)	[]	[]	
🕒 fecha_naci...	4	date	[]	[]	
ABC telefono	5	varchar(10)	[]	[]	

-- 2. Creamos un trigger en el que antes de eliminar un estadio, esa eliminación se guarda en una nueva tabla (eliminados)

```
create table eliminados (  
    nombre_estadio varchar(20) primary key,  
    ciudad varchar(20),  
    telefono varchar(10)  
);  
  
delimiter $$  
create trigger before_estadio_delete  
before delete on estadio  
for each row  
begin  
    insert into eliminados (nombre_estadio, ciudad, telefono)  
    values(old.nombre_estadio, old.ciudad, old.telefono);  
end $$  
delimiter ;  
delete from estadio where nombre_estadio = 'Principe de Asturia';
```



The screenshot shows a database management interface with tabs for 'Propiedades', 'Datos', and 'Diagrama ER'. The 'Datos' tab is active, displaying a table named 'eliminados'. The table has three columns: 'nombre_estadio', 'ciudad', and 'telefono'. A single row is visible with the values 'Principe de Asturia', 'Sanlucar la Mayor', and '123456789'. The interface also shows a search bar and a sidebar with icons for 'Grilla' and 'exto'.

	nombre_estadio	ciudad	telefono
1	Principe de Asturia	Sanlucar la Mayor	123456789

11. ENLACE GITHUB

Para obtener el esquema, los datos y las consultas, vistas, funciones, procedimientos y triggers que he realizado, os dejo un aquí un enlace a mi GitHub en el que podéis encontrar los diferentes ficheros:

<https://github.com/Alvaroo5/BBDDRec.git>

12. CONCLUSIÓN

Para concluir, gracias a este proyecto he aprendido muchas cosas en el ámbito de Base de Datos ya que al principio de curso cuando nos comunicaron que tendríamos que realizar un proyecto de una base de datos, al principio me sorprendí ya que no tenía ningún tipo de conocimiento y pensaba cómo podría lograr realizarlo. Aunque con el paso del tiempo y yendo poco a poco fui adquiriendo nuevos conocimientos en clase e iba dándole forma en cada uno de los trimestres del curso hasta en este último que logramos terminarlo.

La verdad es que me siento orgulloso porque con un gran esfuerzo y aprendiendo nuevos conceptos aunque me hayan resultado con mayor o menor dificultad he conseguido sacarlo adelante.

Con más tiempo y más tranquilidad podríamos mejorar algunos aspectos como sacando mayor provecho a algunas consultas o con una mejor documentación, comentando algunos puntos, mejoraríamos su explicación.