

Proyecto **KINGS LEAGUE**

Una puerta hacia la gestión de bases de datos.

Proyecto de Certificación en SQL basado en la Kings League. Una inmersión total en el mundo de las Bases de Datos y el Deporte.

Presentado por:
Felipe Chavez



ÍNDICE

Contenido del documento

<u>Introducción</u>	3
<u>Objetivos</u>	3
<u>Situación problemática</u>	4
<u>Modelo de negocio</u>	4
<u>Diagrama entidad relación</u>	5
<u>D.E.R Ingeniería Inversa</u>	6
<u>Descripción de las tablas</u>	7
<u>Vistas implementadas</u>	13
<u>Funciones Implementadas</u>	16
<u>Stored Procedures Implementados</u>	18
<u>Triggers Implementados</u>	19
<u>Scripts</u>	20
<u>Informes</u>	21
<u>Herramientas utilizadas</u>	22

INTRODUCCIÓN

Para la concepción de este Proyecto decidí apoyarme en mi apasionado gusto por el hermoso deporte del Fútbol, específicamente decidí tomar inspiración de un formato o competición destacada en los últimos meses como lo es la Kings League. Una liga privada de fútbol 7 fundada por el exfutbolista Gerard Piqué en 2022, en esta competición participan personalidades de internet y streamers, así como algunos exjugadores profesionales.

La liga tiene como objetivo ofrecer un espectáculo alternativo al fútbol tradicional, con reglas especiales y un formato innovador. Como cualquier otra liga de futbol, maneja grandes cantidades de información, por nombrar algunos ejemplos relacionados con el futbol tradicional se pueden encontrar aspectos como: Jugadores, Patrocinadores, Equipos, Presidentes, etc. Con tan solo estos ejemplos es posible concebir algunas posibles entidades para dar forma al proyecto, siendo una de las principales ventajas por la cual escogí esta temática.

Es por ello por lo que mi proyecto se basa específicamente en la información generada y administrada dentro de la Kings League, intentando asemejarse lo más posible a una base de datos útil y funcional que podría ser utilizada dentro de la competición, cubriendo todos los aspectos claves, además de las necesidades de la liga.

OBJETIVOS

Partiendo del conocimiento previo acerca de la Kings League y tras haber identificado las necesidades de la liga durante estos meses de competición, se definieron los siguientes objetivos a cumplir con el desarrollo e implementación del proyecto:

- i. Unificar toda la información referente a la competición en una única base de datos centralizada.
- ii. Almacenar y administrar la información fundamental requerida para el desarrollo de la competición.
- iii. Diseñar y configurar la base de datos de manera óptima para garantizar un rendimiento eficiente.
- iv. Construir una base de datos escalable que pueda adaptarse al crecimiento futuro del proyecto deportivo.
- v. Establecer políticas de respaldo periódico y procedimientos de recuperación ante situaciones de pérdida o corrupción de datos.

SITUACIÓN PROBLEMÁTICA

Considerando que la Kings League constituye una competición innovadora, con apenas unos meses desde su establecimiento, es evidente que su formato, reglas y estructura difieren significativamente de las ligas de fútbol convencionales. Resulta inviable el empleo de una base de datos basada en el fútbol tradicional para almacenar su información, pues a pesar de que puedan existir algunas similitudes, es altamente probable que los modelos de bases de datos ya implementados no abarquen todas las necesidades particulares de la liga, lo que desencadenaría en pérdidas de información e inconsistencias en los datos que lograran ser almacenados.

Partiendo de la necesidad de concebir una base de datos completamente nueva, que tenga en consideración las entidades y relaciones únicas de la Kings League, pues la base de datos debe ser adaptada en su totalidad al formato y estructura de la competición.

Con la implementación de este proyecto se busca asegurar el almacenamiento de toda la información necesaria y pertinente, adicionalmente uno de los aspectos críticos a tener en cuenta es la escalabilidad de la base de datos, garantizando la capacidad de expandirse y adaptarse sin dificultades ante el crecimiento futuro de la liga.

MODELO DE NEGOCIO

Considerando que la Kings League constituye una competición innovadora, con apenas unos meses desde su establecimiento, es evidente que su formato, reglas y estructura difieren significativamente de las ligas de fútbol convencionales. Resulta inviable el empleo de una base de datos basada en el fútbol tradicional para almacenar su información, pues a pesar de que puedan existir algunas similitudes, es altamente probable que los modelos de bases de datos ya implementados no abarquen todas las necesidades particulares de la liga, lo que desencadenaría en pérdidas de información e inconsistencias en los datos que lograran ser almacenados.

Partiendo de la necesidad de concebir una base de datos completamente nueva, que tenga en consideración las entidades y relaciones únicas de la Kings League, pues la base de datos debe ser adaptada en su totalidad al formato y estructura de la competición.

Con la implementación de este proyecto se busca asegurar el almacenamiento de toda la información necesaria y pertinente, adicionalmente uno de los aspectos críticos a tener en cuenta es la escalabilidad de la base de datos, garantizando la capacidad de expandirse y adaptarse sin dificultades ante el crecimiento futuro de la liga.

DIAGRAMA ENTIDAD RELACIÓN

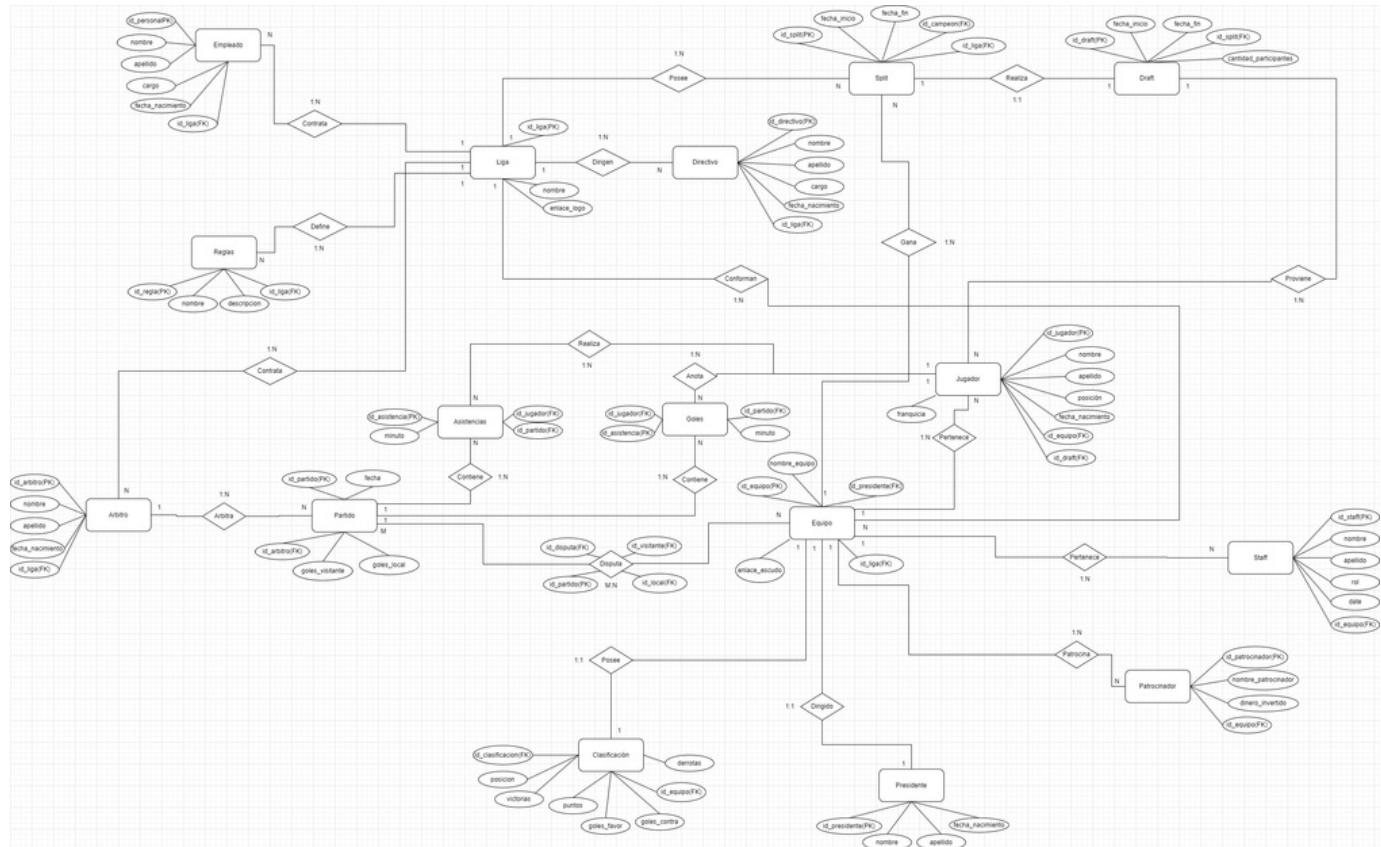
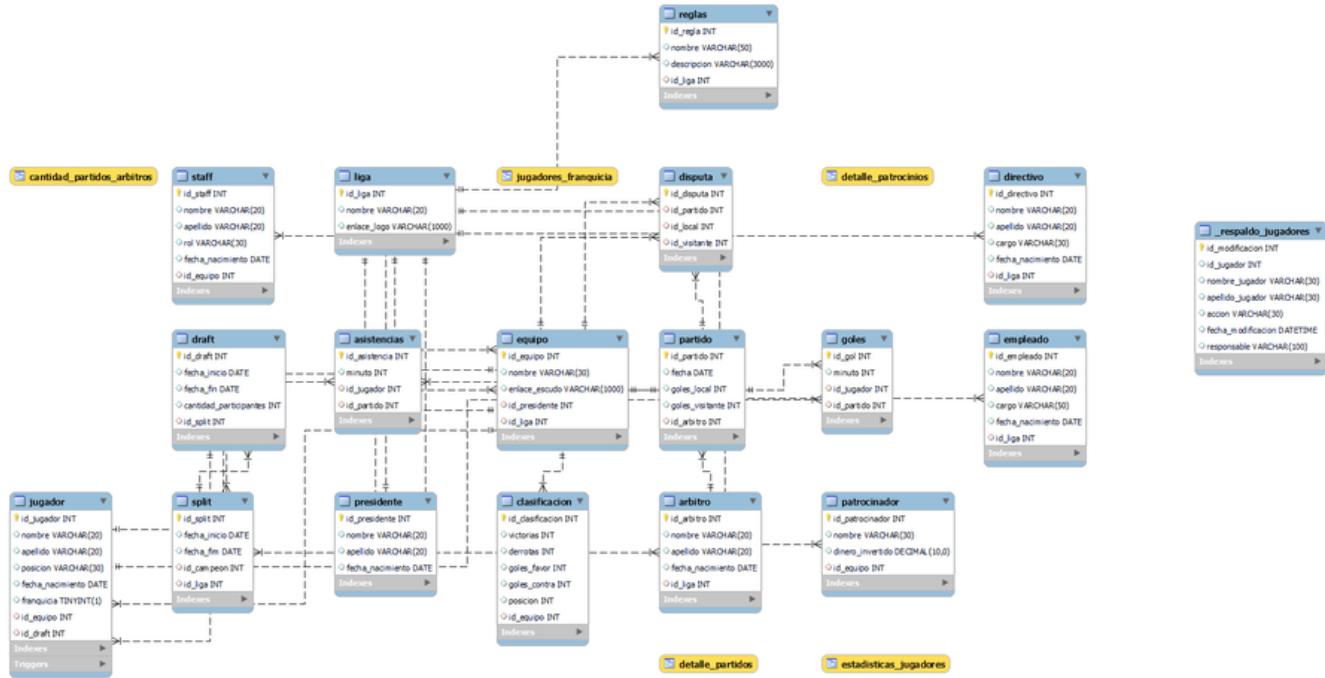


DIAGRAMA ENTIDAD RELACIÓN

GENERADO CON INGENIERÍA INVERSA



DESCRIPCIÓN DE LAS TABLAS IMPLEMENTADAS

Liga				
<i>En esta tabla se encuentran los datos básicos de la liga.</i>				
Nombre del campo	Abreviatura	Tipo de datos	Lenght	Tipos de claves
ID de la liga	id_liga	int		PK
Nombre de la liga	nombre	varchar	20	
Enlace al logo	enlace_logo	varchar	1000	

Split				
<i>En esta tabla se encuentran los datos correspondientes a los dos splits realizados anualmente.</i>				
Nombre del campo	Abreviatura	Tipo de datos	Lenght	Tipos de claves
ID del split	id_split	int		PK
Fecha de inicio	fecha_inicio	date		
Fecha de finalización	fecha_fin	date		
ID del campeón	id_campeon	int		FK
ID de la liga	id_liga	int		FK

Directivo				
<i>En esta tabla se encuentra toda la información referente a los directivos de la organización.</i>				
Nombre del campo	Abreviatura	Tipo de datos	Lenght	Tipos de claves
ID del directivo	id_directivo	int		PK
Nombre del directivo	nombre	varchar	20	
Apellido del directivo	apellido	varchar	20	
Cargo del directivo	cargo	varchar	30	
Fecha de Nacimiento	fecha_nacimiento	date		
ID de la liga	id_liga	int		FK

DESCRIPCIÓN DE LAS TABLAS IMPLEMENTADAS

Empleado				
<i>En esta tabla se encuentra toda la información referente al personal que trabaja para la organización.</i>				
Nombre del campo	Abreviatura	Tipo de datos	Lenght	Tipos de claves
ID del empleado	id_empleado	int		PK
Nombre del empleado	nombre	varchar	20	
Apellido del empleado	apellido	varchar	20	
Cargo del empleado	cargo	varchar	30	
Fecha de Nacimiento	fecha_nacimiento	date		
ID de la liga	id_liga	int		FK

Reglas				
<i>En esta tabla se encuentra toda la información referente a las reglas de la competición.</i>				
Nombre del campo	Abreviatura	Tipo de datos	Lenght	Tipos de claves
ID de la regla	id_regla	int		PK
Nombre de la regla	nombre	varchar	50	
Descripción de la regla	descripcion	varchar	3000	
ID de la liga	id_liga	int		FK

Goles				
<i>En esta tabla se encuentran los goles realizados en todos los partidos disputados durante el split.</i>				
Nombre del campo	Abreviatura	Tipo de datos	Lenght	Tipos de claves
ID del gol	id_gol	int		PK
Minuto en el que se anotó el gol	minuto	time		
ID del jugador	id_jugador	int		FK
ID del partido	id_partido	int		FK

DESCRIPCIÓN DE LAS TABLAS IMPLEMENTADAS

Arbitro				
<i>En esta tabla se encuentra toda la información referente a los árbitros que trabajan para la organización.</i>				
Nombre del campo	Abreviatura	Tipo de datos	Lenght	Tipos de claves
ID del árbitro	id_arbitro	int		
Nombre del árbitro	nombre	varchar	20	
Apellido del árbitro	apellido	varchar	20	
Cargo del árbitro	cargo	varchar	30	
Fecha de Nacimiento	fecha_nacimiento	date		
ID de la liga	id_liga	int		FK

Partido				
<i>En esta tabla se encuentran la información de cada uno de los partidos disputados durante el split.</i>				
Nombre del campo	Abreviatura	Tipo de datos	Lenght	Tipos de claves
ID del partido	id_resultado	int		PK
Fecha del partido	fecha	date		
Goles del equipo local	goles_local	int		
Goles del equipo visitante	goles_visitante	int		
ID del arbitro	goles_visitante	int		FK

Asistencias				
<i>En esta tabla se encuentran las asistencias realizadas en todos los partidos disputados durante el split.</i>				
Nombre del campo	Abreviatura	Tipo de datos	Lenght	Tipos de claves
ID de la asistencia	id_asistencia	int		PK
Minuto en el que se realizó la asistencia	minuto	time		
ID del jugador	id_jugador	int		FK
ID del partido	id_partido	int		FK

DESCRIPCIÓN DE LAS TABLAS IMPLEMENTADAS

Jugador				
<i>En esta tabla se encuentra toda la información referente a los jugadores que participan en la competición.</i>				
Nombre del campo	Abreviatura	Tipo de datos	Lenght	Tipos de claves
ID del jugador	id_jugador	int		PK
Nombre del jugador	nombre	varchar	20	
Apellido del jugador	apellido	varchar	20	
Posición del jugador	posición	varchar	30	
Fecha de Nacimiento	fecha_nacimiento	date		
Franquicia	jugador_franquicia	boolean		
ID del equipo	id_equipo	int		FK
ID del draft	id_draft	int		FK

Equipo				
<i>En esta tabla se encuentra toda la información referente a los equipos que conforman la competición.</i>				
Nombre del campo	Abreviatura	Tipo de datos	Lenght	Tipos de claves
ID del equipo	id_equipo	int		PK
Nombre del equipo	nombre	varchar	30	
Enlace al escudo	enlace_escudo	varchar	1000	
ID del presidente	id_presidente	int		FK
ID de la liga	id_liga	int		FK

DESCRIPCIÓN DE LAS TABLAS IMPLEMENTADAS

Presidente				
<i>En esta tabla se encuentra toda la información referente a los presidentes o propietarios de los equipos.</i>				
Nombre del campo	Abreviatura	Tipo de datos	Lenght	Tipos de claves
ID del presidente	id_presidente	int		PK
Nombre del presidente	nombre	varchar	20	
Apellido del presidente	apellido	varchar	20	
Fecha de Nacimiento	fecha_nacimiento	date		

Clasificación				
<i>En esta tabla se encuentra toda la información referente a la clasificación de los equipos.</i>				
Nombre del campo	Abreviatura	Tipo de datos	Lenght	Tipos de claves
ID de la clasificación	id_clasificación	int		PK
Cantidad de victorias	victorias	int		
Cantidad de derrotas	derrotas	int		
Goles a favor	goles_favor	int		
Goles en contra	goles_contra	int		
Puntos	puntos	int		
Posición	posicion	int		
ID del equipo	id_equipo	int		FK

Patrocinador				
<i>En esta tabla se encuentra toda la información referente a los patrocinadores de cada uno de los equipos.</i>				
Nombre del campo	Abreviatura	Tipo de datos	Lenght	Tipos de claves
ID del patrocinador	id_patrocinador	int		PK
Nombre del patrocinador	nombre	varchar	30	
Dinero invertido	dinero_invertido	decimal		
ID del equipo	id_equipo	int		FK

DESCRIPCIÓN DE LAS TABLAS IMPLEMENTADAS

Staff				
<i>En esta tabla se encuentra toda la información referente al staff que trabaja para cada uno de los equipos.</i>				
Nombre del campo	Abreviatura	Tipo de datos	Lenght	Tipos de claves
ID del staff	id_staff	int		PK
Nombre del staff	nombre	varchar	20	
Apellido del staff	apellido	varchar	20	
Rol dentro del staff	rol	varchar	30	
Fecha de Nacimiento	fecha_nacimiento	date		
ID del equipo	id_equipo	int		FK

Draft				
<i>En esta tabla se encuentran los datos correspondientes al draft realizado anualmente.</i>				
Nombre del campo	Abreviatura	Tipo de datos	Lenght	Tipos de claves
ID del draft	id_draft	int		PK
Fecha de inicio	fecha_inicio	date		
Fecha de finalización	fecha_fin	date		
Cantidad de participantes	cantidad_participantes	int		
ID del split	id_split	int		FK

Disputa				
<i>En esta tabla se encuentran los datos de la relación N:M generada entre las entidades partido y equipo.</i>				
Nombre del campo	Abreviatura	Tipo de datos	Lenght	Tipos de claves
ID de la disputa	id_disputa	int		PK
ID del partido	id_partido	int		FK
ID del equipo local	id_local	int		FK
ID del equipo visitante	id_visitante	int		FK

VIEWS IMPLEMENTADAS

Vista 1: Estadísticas Jugadores

Descripción: En esta vista pueden ser consultados los goles y asistencias de cada jugador, se encuentra ordenada por la cantidad de goles.

Objetivo: El objetivo de esta vista es poder consultar de forma sencilla y rápida las estadísticas de cada uno de los jugadores en lo que a goles y asistencias hace referencia, adicionalmente se excluyen a los jugadores que no hayan marcado goles o asistencias con el fin de filtrar la información.

Tablas/Datos: Jugador, Goles, Asistencias y Equipo

```
CREATE OR REPLACE VIEW estadisticas_jugadores AS
  (SELECT
    CONCAT(j.nombre, ' ', j.apellido) Nombre,
    IFNULL(g.cantidad_goles, 0) Cantidad_Goles,
    IFNULL(a.cantidad_asistencias, 0) Cantidad_Asistencias,
    e.nombre AS Equipo
  FROM
    jugador j
    LEFT JOIN
    (SELECT
      id_jugador, COUNT(*) cantidad_goles
    FROM
      goles
    GROUP BY id_jugador) g ON j.id_jugador = g.id_jugador
    LEFT JOIN
    (SELECT
      id_jugador, COUNT(*) cantidad_asistencias
    FROM
      asistencias
    GROUP BY id_jugador) a ON j.id_jugador = a.id_jugador
    JOIN
      equipo e ON e.id_equipo = j.id_equipo
  WHERE
    g.cantidad_goles > 0
    OR a.cantidad_asistencias > 0
  ORDER BY g.cantidad_goles DESC);
```

Vista 2: Jugadores Franquicia

```
CREATE OR REPLACE VIEW jugadores_franquicia AS
  (SELECT
    CONCAT(j.nombre, ' ', j.apellido)
  Nombre_Jugador_Franquicia,
    e.nombre Equipo
  FROM
    jugador j
    JOIN
      equipo e ON e.id_equipo = j.id_equipo
  WHERE
    franquicia = 1);
```

Descripción: En esta vista se pueden conocer los nombres y apellidos de todos los jugadores franquicia y el equipo al que pertenecen.

Objetivo: El objetivo de esta vista es permitir de forma rápida y sencilla conocer los jugadores franquicia que participan en la liga, adicionalmente conocer a qué equipo pertenecen sin necesidad de realizar consultas u operaciones adicionales.

Tablas/Datos: Equipo y Jugador

VIEWS IMPLEMENTADAS

Vista 3: Detalle Partidos

Descripción: En esta vista pueden ser consultados el resultado de los partidos, los equipos enfrentados y adicionalmente son ordenados por fecha.

Objetivo: El objetivo de esta vista es poder conocer el resultado final de cada uno de los partidos disputados en el transcurso de la liga, el nombre de los equipos enfrentados y la fecha en la que se disputó el partido, todo con el fin de consultar esta información de forma rápida, breve y resumida. Esta vista puede ser útil al momento de ejecutar resúmenes de las jornadas o highlights.

Tablas/Datos: Partido, Disputa y Equipo

```
CREATE OR REPLACE VIEW detalle_partidos AS
  (SELECT
      CONCAT(l.nombre, ' - ', v.nombre) Equipos,
      CONCAT(p.goles_local, ' - ', p.goles_visitante)
    Goles_Partido,
      fecha Fecha
   FROM
      partido p
      JOIN
      disputa d ON p.id_partido = d.id_partido
      JOIN
      equipo l ON d.id_local = l.id_equipo
      JOIN
      equipo v ON d.id_visitante = v.id_equipo
   ORDER BY fecha DESC);
franquicia = 1);
```

Vista 4: Detalle Patrocinios

Descripción: En esta vista se pueden consultar los detalles de los patrocinadores de cada uno de los equipos incluyendo: nombre, presupuesto y equipo patrocinado. Adicionalmente, la información es ordenada de acuerdo al presupuesto invertido por cada uno.

Objetivo: El objetivo de esta vista es poder visualizar de forma breve y resumida la información de cada uno de los patrocinadores junto con el equipo patrocinado y el presupuesto invertido, esta información puede ser consultada por equipos de ética y competencia para garantizar que los patrocinadores cumplan con los rangos presupuestales adicionalmente esta información puede ser utilizada por el equipo contable o de impuestos.

```
CREATE OR REPLACE VIEW detalle_patrocinios AS
  (SELECT
      e.nombre Equipo,
      p.nombre Patrocinador,
      CONCAT('€ ', p.dinero_invertido) Dinero_Invertido
    FROM
      equipo e
      JOIN
      patrocinador p ON e.id_equipo = p.id_equipo
  ORDER BY p.dinero_invertido DESC);
franquicia = 1);
```

Tablas/Datos: Equipo y Patrocinador

VIEWS IMPLEMENTADAS

Vista 5: Cantidad Partidos Arbitrados

Descripción: En esta tabla es posible consultar la cantidad de partidos arbitrados por cada uno de los integrantes del cuerpo arbitral.

Objetivo: El objetivo de esta tabla es llevar un conteo y control de los partidos arbitrados por cada integrante del cuerpo arbitral a lo largo de la temporada.

Tablas/Datos: Arbitro y Partido

```
CREATE OR REPLACE VIEW cantidad_partidos_arbitros AS
  (SELECT
    CONCAT(nombre, ' ', apellido) Nombre_Arbitro,
    COUNT(p.id_arbitro) Cantidad_Partidos
  FROM
    arbitro a
    JOIN
    partido p ON a.id_arbitro = p.id_arbitro
  GROUP BY a.id_arbitro
  ORDER BY Cantidad_Partidos DESC);
```

FUNCIONES IMPLEMENTADAS

Función 1: Mayor Goleador Asistidor

Descripción: Esta función permite ingresar opciones o parámetros (goleador y asistidor) y retorna el nombre del jugador que más goles o asistencias ha realizado hasta la fecha, de acuerdo con la opción seleccionada. Adicionalmente, tiene una gestión en caso de errores con el parámetro ingresado.

Objetivo: El objetivo de esta función es obtener los nombre(s) del máximo goleador(es), para que cualquier persona pueda tener acceso a la información de los jugadores más destacados y con mejor rendimiento de la liga.

Tablas/Datos: Vista estadisticas_jugadores.

```
- □ ×
DROP FUNCTION IF EXISTS mayor_goleador_asistidor;
DELIMITER //
CREATE FUNCTION mayor_goleador_asistidor(opcion VARCHAR(20))
RETURNS VARCHAR(100)
DETERMINISTIC
BEGIN
    DECLARE resultado VARCHAR(100);
    CASE
        WHEN opcion LIKE 'goleador' THEN
            SELECT GROUP_CONCAT(nombre SEPARATOR ', ')
            INTO resultado
            FROM estadisticas_jugadores
            WHERE cantidad_goles = (SELECT MAX(cantidad_goles)
            FROM estadisticas_jugadores);
        WHEN opcion LIKE 'asistidor' THEN
            SELECT GROUP_CONCAT(nombre SEPARATOR ', ')
            INTO resultado
            FROM estadisticas_jugadores
            WHERE cantidad_asistencias = (SELECT
                MAX(cantidad_asistencias) FROM estadisticas_jugadores);
        ELSE
            SET resultado = 'Por favor ingresa un valor válido,
            las opciones son: Goleador - Asistidor. Gracias!';
    END CASE;
    RETURN resultado;
END //
DELIMITER ;
```

Función 2: Buscador ID Jugador

```
- □ ×
DROP FUNCTION IF EXISTS buscador_id_jugador;
DELIMITER $$
CREATE FUNCTION buscador_id_jugador(opcion INT) RETURNS
VARCHAR(100)
DETERMINISTIC
BEGIN
    DECLARE resultado_equipo VARCHAR(30);
    DECLARE nombre_jugador VARCHAR(100);

    SELECT
        e.nombre
    INTO resultado_equipo
    FROM
        jugador j
        JOIN
        equipo e ON j.id_equipo = e.id_equipo
    WHERE
        j.id_jugador = opcion;
    SELECT
        CONCAT(j.nombre, ' ', j.apellido)
    INTO nombre_jugador
    FROM
        jugador j
        JOIN
        equipo e ON j.id_equipo = e.id_equipo
    WHERE
        j.id_jugador = opcion;
    IF resultado_equipo IS NOT NULL THEN
        RETURN CONCAT('El jugador ', nombre_jugador, ' pertenece a:
        ', resultado_equipo);
    ELSE
        RETURN CONCAT('No se encontró un jugador dentro de la Kings
        League con el ID ingresado: ', opcion);
    END IF;
END $$
DELIMITER ;
```

Descripción: Esta función permite conocer el nombre y el equipo al que pertenece un jugador con tan solo ingresar su ID único. Adicionalmente, tiene una gestión en caso de errores con el parámetro ingresado.

Objetivo: El objetivo de esta función es ser utilizada como un buscador únicamente ingresando el ID del jugador, permitiendo que cualquier fanático pueda conocer la información principal de cada uno de los jugadores, adicionalmente también podría ser utilizada por empleados dentro de la Kings League.

Tablas/Datos: Jugador

FUNCIONES IMPLEMENTADAS

Función 3: Buscador Nombre Jugador

Descripción: Esta función permite conocer el nombre y el equipo al que pertenece un jugador con tan solo ingresar su nombre y apellido. Adicionalmente, tiene una gestión en caso de errores con el parámetro ingresado.

Objetivo: El objetivo de esta función es ser utilizada como un buscador únicamente ingresando el nombre y apellido del jugador, permitiendo que cualquier fanático pueda conocer la información principal de cada uno de los jugadores, adicionalmente también podría ser utilizada por empleados dentro de la Kings League.

Tablas/Datos: Jugador

```

DROP FUNCTION IF EXISTS buscador_nombre_jugador;
DELIMITER $$ 
CREATE FUNCTION buscador_nombre_jugador(nombre varchar(50))
RETURNS VARCHAR(100)
DETERMINISTIC
BEGIN
DECLARE resultado_equipo VARCHAR(30);
DECLARE nombre_jugador VARCHAR(100);

SELECT
    e.nombre
INTO resultado_equipo FROM
    jugador j
    JOIN
        equipo e ON j.id_equipo = e.id_equipo
WHERE
    CONCAT(j.nombre, ' ', j.apellido) LIKE nombre;
SELECT
    CONCAT(j.nombre, ' ', j.apellido)
INTO nombre_jugador FROM
    jugador j
    JOIN
        equipo e ON j.id_equipo = e.id_equipo
WHERE
    CONCAT(j.nombre, ' ', j.apellido) LIKE nombre;
IF resultado_equipo IS NOT NULL THEN
    RETURN CONCAT('El jugador ',nombre_jugador,' pertenece a:',
',resultado_equipo');
ELSE
    RETURN CONCAT('No se encontró un jugador dentro de la Kings
League con el nombre ingresado: ', nombre);
END IF;
END $$ 
DELIMITER ;

```

STORED PROCEDURES IMPLEMENTADOS

Stored Procedure 1: Orden Jugadores

Descripción: Este Stored Procedure permite ordenar el listado de todos los jugadores de la liga de acuerdo al campo de ordenamiento y el tipo de ordenamiento ingresado.

Objetivo: Tiene como objetivo obtener de manera rápida y sencilla la lista ordenada de jugadores ascendente o descendente de acuerdo al campo requerido en cada caso de uso.

Tablas/Datos: Jugador

```
- □ ×
DROP PROCEDURE IF EXISTS orden_jugadores;
DELIMITER //
CREATE PROCEDURE orden_jugadores(
    IN campo VARCHAR(20),
    IN orden VARCHAR(20)
)
BEGIN
    IF orden = 'ascendente' THEN
        SET orden = 'ASC';
    ELSEIF orden = 'descendente' THEN
        SET orden = 'DESC';
    ELSE
        SET orden = 'DESC';
    END IF;
    ##Aqui se aplica el uso de cláusulas para efectuar el SP
    SET @jugadores = CONCAT('SELECT * FROM jugador ORDER BY ', campo, ' ', orden);
    PREPARE stmt FROM @jugadores;
    EXECUTE stmt;
    DEALLOCATE PREPARE stmt;
END //
```

Stored Procedure 2: Insert Patrocinador

```
- □ ×
DROP PROCEDURE IF EXISTS sp_insert_patrocinador;
DELIMITER $$
CREATE PROCEDURE sp_insert_patrocinador(IN nombre VARCHAR(50),
    IN dinero INT, IN equipo INT, OUT output VARCHAR(100))
BEGIN
    IF (nombre <> '') AND (dinero AND equipo != 0) THEN
        INSERT INTO PATROCINADOR
        (id_patrocinador, nombre, dinero_invertido, id_equipo) VALUES
        (null, UCASE(nombre), dinero, equipo);
        SET output = 'Inserción del patrocinador exitosa.';
    ELSE
        SET output = 'ERROR: no se pudo registrar al
        patrocinador del equipo.';
    END IF;
END
$$
```

Descripción: Este Stored Procedure facilita la inserción de registros de patrocinadores, adicionalmente genera respuestas de acuerdo al resultado de la inserción.

Objetivo: Tiene como objetivo facilitar la inserción de registros a la tabla Patrocinador, únicamente requiere la información y por defecto genera la sintaxis e ingresa los registros.

Tablas/Datos: Patrocinador

TRIGGERS IMPLEMENTADOS

Trigger 1: Control de Logs

Descripción: Este trigger permite registrar, gestionar y actualizar logs con respecto a la modificación de una tabla.

Objetivo: Tiene como objetivo gestionar y actualizar una tabla de respaldo con respecto a la inserción, modificación o eliminación de jugadores, dada la cantidad de registros es importante garantizar un respaldo óptimo de los mismos.

```
-- □ ×

##Creacion del trigger para la insercion de jugadores
CREATE
    TRIGGER triger_insert_jugadores
    AFTER INSERT ON jugador FOR EACH ROW
        INSERT INTO _respaldo_jugadores (id_jugador , nombre_jugador
        , apellido_jugador , accion , fecha_modificacion , responsable)
        VALUES (NEW.id_jugador , NEW.nombre , NEW.apellido , 'INSERT' ,
        CURRENT_TIMESTAMP() , USER()); 

##Creacion del trigger para la actualización de jugadores
CREATE
    TRIGGER triger_update_jugadores
    AFTER UPDATE ON jugador FOR EACH ROW
        INSERT INTO _respaldo_jugadores (id_jugador , nombre_jugador
        , apellido_jugador , accion , fecha_modificacion , responsable)
        VALUES (NEW.id_jugador , NEW.nombre , NEW.apellido , 'UPDATE' ,
        CURRENT_TIMESTAMP() , USER()); 

##Creacion del trigger para la eliminació de jugadores
CREATE
    TRIGGER triger_delete_jugadores
    AFTER DELETE ON jugador FOR EACH ROW
        INSERT INTO _respaldo_jugadores (id_jugador , nombre_jugador
        , apellido_jugador , accion , fecha_modificacion , responsable)
        VALUES (OLD.id_jugador , OLD.nombre , OLD.apellido , 'DELETE' ,
        CURRENT_TIMESTAMP() , USER());
```

Trigger 2: Control edad jugadores.

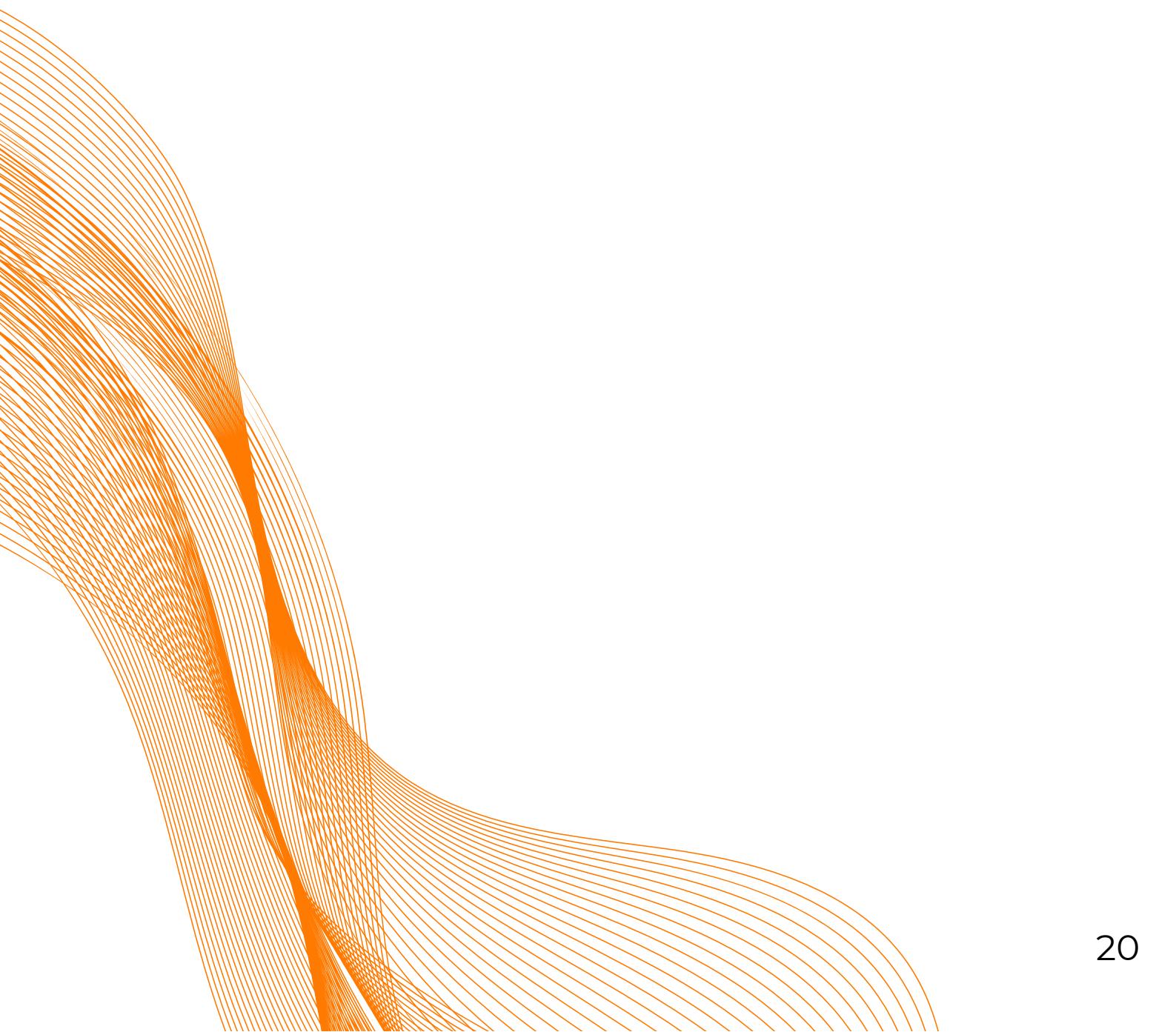
```
-- □ ×

DELIMITER //
CREATE TRIGGER control_edad_jugador
BEFORE INSERT ON jugador
FOR EACH ROW
BEGIN
    ##Declaracion de variable que almacenara la edad
    DECLARE edad INT;
    ##Seteamos a la variable el resultado de la funcion
    TIMESTAMPDIFF que habiamos utilizado previamente para calcular
    diferencias entre fechas
    SET edad = TIMESTAMPDIFF(YEAR, NEW.fecha_nacimiento,
    CURDATE());
    ##Usamos un condicional para realizar la validación
    IF edad < 18 THEN
        ##Utilice una señal personalizada para que se ilustre un
        error personalizado en el output de MYSQL
        SIGNAL SQLSTATE '45000'
        ##Modifico el mensaje para que sea acorde con lo que
        necesito
        SET MESSAGE_TEXT = 'No se puede registrar jugadores
        menores de 18 años.';
    END IF;
END;
//
```

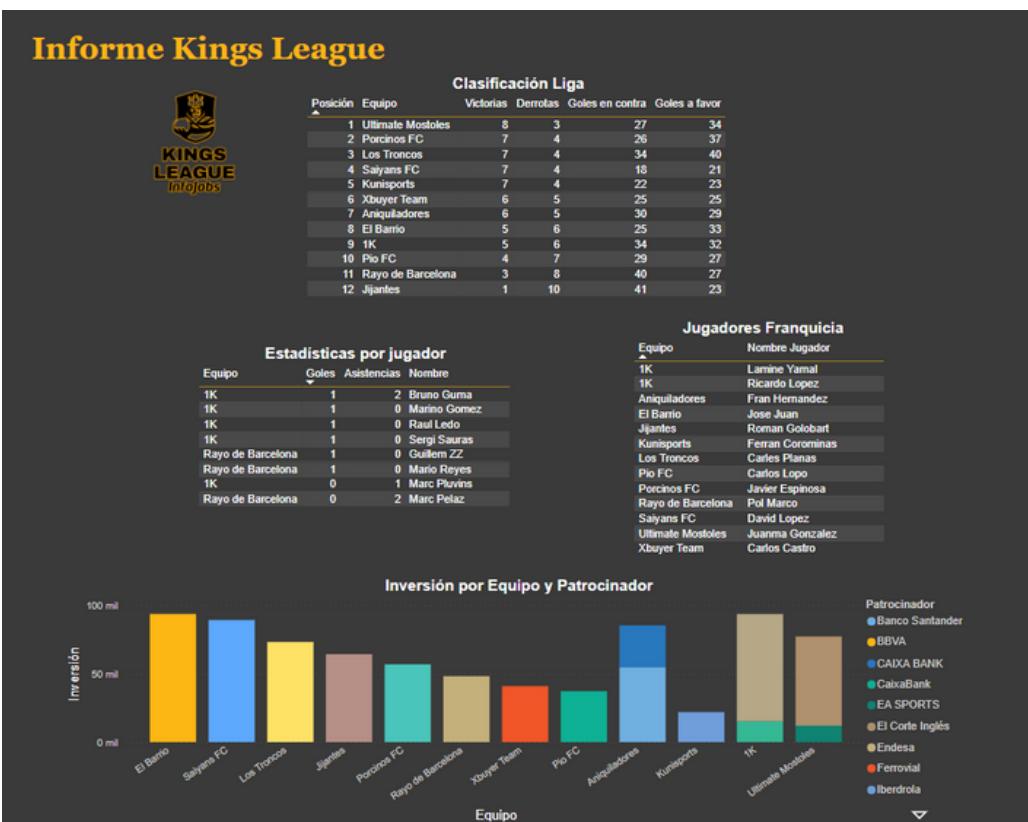
Descripción: Este trigger otorga un control sobre la edad de los jugadores a ingresar en la base de datos.

Objetivo: Tiene como objetivo evitar que se ingresen jugadores menores de edad en la base de datos.

SCRIPTS

- Script Creación Base de Datos (Todos los objetos).
 - Script Inserción de datos.
 - Script Test, incluye testeos sobre los objetos creados previamente.
- 

INFORME



La siguiente captura corresponde a un Dashboard realizado en Power BI, con el objetivo de realizar un análisis y monitoreo de las estadísticas de la liga en tiempo real.

En el Dashboard se puede encontrar información como:

1. Clasificación general de la liga.
2. Estadísticas de goles y asistencias por jugador.
3. Jugadores franquicia por equipo.
4. Inversión de los patrocinadores a cada uno de los equipos.

Esta herramienta facilita el análisis y comprensión de los datos, adicionalmente se encuentra sincronizada en tiempo real con los datos. Obviamente, existe una oportunidad de mejora con respecto a la utilización de objetos visuales, sin embargo, es un buen primer acercamiento con respecto al análisis de la información alojada hasta el momento.

HERRAMIENTAS UTILIZADAS

MySQL

MySQL es un sistema de gestión de bases de datos ampliamente utilizado en aplicaciones web y empresariales. Ofrece una alta velocidad y escalabilidad, junto con una gran comunidad de usuarios y soporte. MySQL es de código abierto, lo que lo hace accesible y asequible, y se utiliza para almacenar y recuperar datos de manera eficiente en una amplia gama de aplicaciones.



Power BI

Power BI es una herramienta de inteligencia empresarial de Microsoft que permite visualizar y analizar datos de manera efectiva. Permite conectar diversas fuentes de datos, transformarlos y crear visualizaciones interactivas y tableros de control. Con su interfaz amigable, incluso los no técnicos pueden explorar datos y tomar decisiones informadas.



Draw.io

Draw.io es una aplicación en línea de diagramación y diseño gráfico. Con una interfaz intuitiva y una amplia biblioteca de formas, permite crear diagramas, organigramas y más de manera profesional. Facilita la colaboración en tiempo real y la exportación en múltiples formatos, siendo ideal para comunicación visual y planificación de proyectos. Como herramienta basada en la nube, es accesible desde cualquier lugar.



Canva

Canva es una plataforma de diseño gráfico en línea que simplifica la creación de contenido visual de alta calidad. Ofrece una amplia gama de plantillas personalizables para diseño gráfico, redes sociales, presentaciones y más. Su interfaz intuitiva y herramientas de edición accesibles hacen que la creación de diseños sea fácil y efectiva.



Git Hub

GitHub es una plataforma líder en gestión de código fuente y colaboración para desarrolladores. Permite alojar, revisar y colaborar en proyectos de software de forma eficiente. Con características como control de versiones, seguimiento de problemas y colaboración en tiempo real, GitHub es esencial para equipos de desarrollo.



¡GRACIAS! |

Felipe Chavez 

