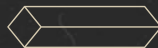




Entrega Final Curso de SQL

Premier League DB

Alexis Santa Ana
Comisión 53170



Índice

01

Introducción

02

Objetivos

03

Diagrama Entidad-Relación

04

Listado de Tablas

05

Vistas

06

Funciones

07

Stored Procedures

08

Triggers

09

DCL

10

Backup

11

Herramientas utilizadas





Introducción

Esta base de datos llamada "Premier_League_DB" guarda información detallada sobre los equipos, jugadores, partidos y entrenadores de la liga.

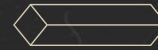
También incluye vistas que simplifican la obtención de datos específicos, como la cantidad de jugadores por equipo y la nacionalidad de los jugadores.



Además, hay funciones para buscar jugadores por nombre y obtener detalles sobre estadios y ciudades.

Es como tener un backstage de la liga más emocionante del fútbol inglés, ¡todo organizado y accesible en un solo lugar!



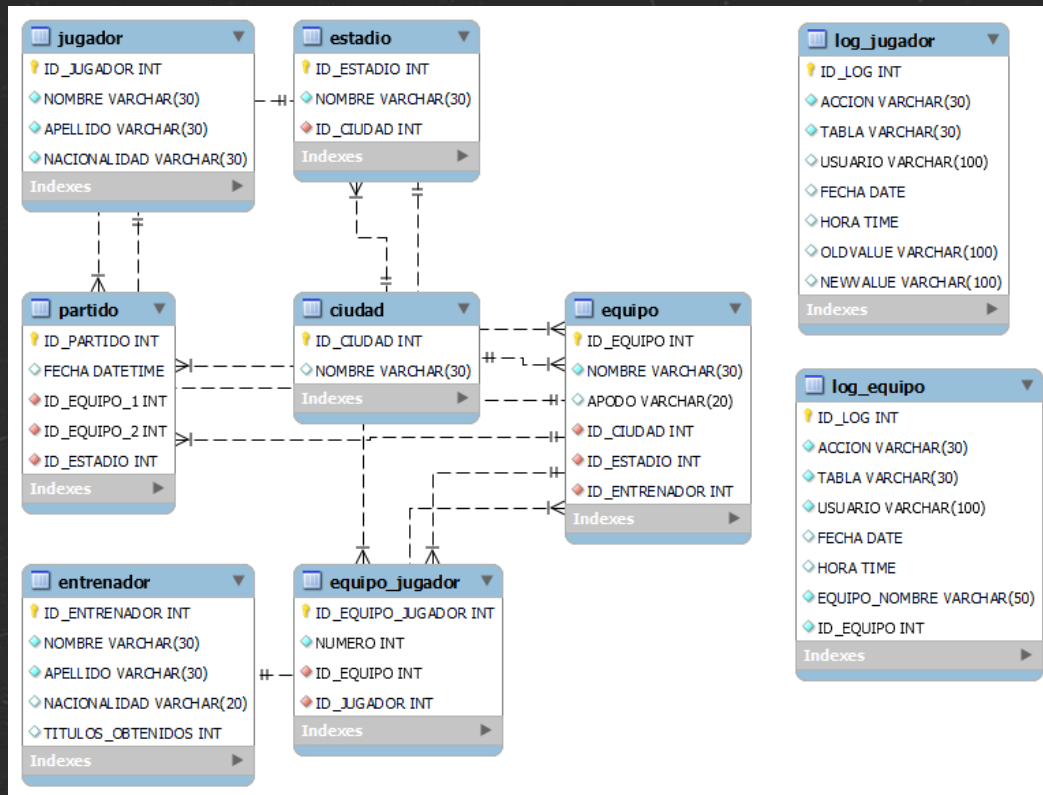


Objetivos

1. Seguir la información detallada de equipos, jugadores, partidos y entrenadores de la Premier League.
2. Facilitar el acceso a datos específicos, como la cantidad de jugadores por equipo y la nacionalidad de los jugadores, a través de vistas.
3. Proporcionar funciones para buscar jugadores por nombre y obtener detalles sobre estadios y ciudades.
4. Establecer relaciones entre diferentes entidades, como equipos, ciudades y estadios, para un seguimiento completo de la liga.
5. Permitir la gestión de usuarios con diferentes niveles de acceso y permisos.
6. Mantener la integridad de los datos mediante el uso de claves foráneas y restricciones.
7. Ofrecer procedimientos almacenados para realizar tareas específicas, como ordenar equipos por diferentes criterios y agregar nuevas ciudades.
8. Gestionar transacciones de manera segura para mantener la consistencia de los datos.

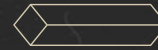


Diagrama Entidad-Relación





Listado de Tablas



EQUIPO: Contiene detalles y atributos de los equipos de la Premier League.

equipo
ID_EQUIPO INT
NOMBRE VARCHAR(30)
APODO VARCHAR(20)
ID_CIUADAD INT
ID_ESTADIO INT
ID_ENTRENADOR INT
Indexes

ID_EQUIPO: Identificador único del equipo (Integer).

NOMBRE: Nombre del equipo (VARCHAR).

APODO: Apodo del equipo, si lo tiene (VARCHAR).

ID_CIUADAD: Identificador de la ciudad donde se encuentra el equipo (Integer).

ID_ESTADIO: Identificador del estadio donde juega el equipo (Integer).

ID_ENTRENADOR: Identificador del entrenador del equipo (Integer).



CIUDAD: Almacena registros relacionados con las ciudades asociadas a los equipos.

ciudad
ID_CIUADAD INT
NOMBRE VARCHAR(30)
Indexes

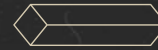
ID_CIUADAD: Identificador único de la ciudad (Integer).

NOMBRE: Nombre de la ciudad (VARCHAR).

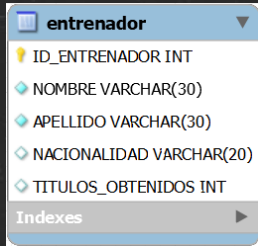




Listado de Tablas



ENTRENADOR: Guarda información sobre los entrenadores de los equipos de fútbol.



entrenador	
ID_ENTRENADOR	INT
NOMBRE	VARCHAR(30)
APELLIDO	VARCHAR(30)
NACIONALIDAD	VARCHAR(20)
TITULOS_OBTENIDOS	INT
Indexes	

ID_ENTRENADOR: Identificador único del entrenador (Integer).

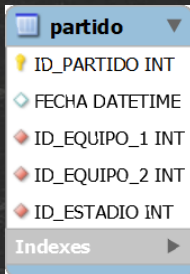
NOMBRE: Nombre del entrenador (VARCHAR).

APELLIDO: Apellido del entrenador (VARCHAR).

NACIONALIDAD: Nacionalidad del entrenador (VARCHAR).

TITULOS_OBTENIDOS: Número de títulos obtenidos por el entrenador (Integer).

PARTIDO: Registra datos sobre los encuentros disputados en la liga.



partido	
ID_PARTIDO	INT
FECHA	DATETIME
ID_EQUIPO_1	INT
ID_EQUIPO_2	INT
ID_ESTADIO	INT
Indexes	

ID_PARTIDO: Identificador único del partido (Integer).

FECHA: Fecha y hora del partido (DATETIME).

ID_EQUIPO_1: Identificador del primer equipo (Integer).

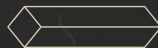
ID_EQUIPO_2: Identificador del segundo equipo (Integer).

ID_ESTADIO: Identificador del estadio donde se juega el partido (Integer).





Listado de Tablas



ESTADIO: Ofrece detalles sobre los estadios donde se realizan los partidos.

estadio	
ID_ESTADIO INT	
NOMBRE VARCHAR(30)	
ID_CIUADAD INT	
Indexes	

ID_ESTADIO: Identificador único del estadio (Integer).

NOMBRE: Nombre del estadio (VARCHAR).

ID_CIUADAD: Identificador de la ciudad donde se encuentra el estadio (Integer).

EQUIPO_JUGADOR: Establece la relación entre equipos y sus respectivos jugadores.

equipo_jugador	
ID_EQUIPO_JUGADOR INT	
NUMERO INT	
ID_EQUIPO INT	
ID_JUGADOR INT	
Indexes	

ID_EQUIPO_JUGADOR: Identificador único de la relación equipo-jugador (Integer).

NUMERO: Número de camiseta del jugador en el equipo (Integer).

ID_EQUIPO: Identificador del equipo (Integer).

ID_JUGADOR: Identificador del jugador (Integer).

JUGADOR: Contiene perfiles detallados de los futbolistas participantes en la liga.

jugador	
ID_JUGADOR INT	
NOMBRE VARCHAR(30)	
APELLIDO VARCHAR(30)	
NACIONALIDAD VARCHAR(30)	
Indexes	

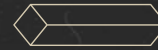
ID_JUGADOR: Identificador único del jugador (Integer).

NOMBRE: Nombre del jugador (VARCHAR).

APELLIDO: Apellido del jugador (VARCHAR).

NACIONALIDAD: Nacionalidad del jugador (VARCHAR).





Vistas

1. VW_EQUIPO_CIUDAD:

- Muestra el nombre del equipo junto con la ciudad correspondiente.

2. VW_EQUIPO_CANTIDAD_JUGADORES:

- Indica la cantidad de jugadores por cada equipo.

3. VW_JUGADORES_NACIONALIDAD:

- Muestra la cantidad de jugadores por nacionalidad, ordenados de forma descendente.

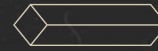
4. VW_DETALLES_PARTIDOS:

- Proporciona detalles de los partidos, incluyendo la fecha, los equipos enfrentados y el estadio.

5. VW_ENTRENADORES_EQUIPOS:

- Presenta el nombre completo de los entrenadores junto con el equipo que dirigen actualmente, ordenados por la cantidad de trofeos obtenidos.





Funciones

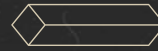
1. FN_JUGADOR_RETORNAR:

- Retorna el nombre completo de un jugador buscando por caracteres ingresados como parámetro.

2. FN_ESTADIO_CIUADAD:

- Retorna el nombre del estadio y la ciudad a la que pertenece, dados el ID del estadio como parámetro.





Stored Procedures

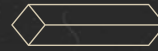
1. SP_LISTA_NOMBRES:

- Ingresa por parámetro el nombre de una tabla disponible para que devuelva una lista con todos los nombres de los registros existentes.

2. SP_INSERTAR_CIUADAD:

- Inserta una nueva ciudad en la tabla de ciudades, utilizando el nombre de la ciudad como parámetro de entrada.





Triggers

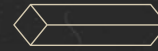
1. TRG_LOG_EQUIPO:

- Desencadenado después de la inserción en la tabla de equipos. Registra la acción de inserción en un registro de log.

2. TRG_LOG_JUGADOR:

- Desencadenado antes de una actualización en la tabla de jugadores. Registra la acción de actualización en un registro de log, capturando el valor antiguo y nuevo del nombre completo del jugador.





DCL (Data Control Lenguaje)

1. Creación de Usuarios:

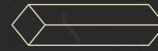
- Se crean dos usuarios: READONLY y HRDEPARTMENT, con contraseñas asociadas.

2. Asignación de Permisos:

- El usuario READONLY tiene permisos de solo lectura en todas las bases de datos.
- El usuario HRDEPARTMENT tiene permisos de selección, inserción y actualización en todas las bases de datos.

Estas acciones de Control de Lenguaje de Datos (DCL) aseguran que los usuarios tengan los permisos adecuados para interactuar con la base de datos según sus roles y responsabilidades.



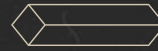


Backup

Se ha realizado un backup completo de manera de tener respaldo de la base de datos completa.

Se ha guardado en el archivo `backup_premier_league_db.sql`





Herramientas Utilizadas

MySQL Workbench

ChatGPT 3.5

Powerpoint

GitHub

