

Bases de datos

Ian Andres Abadia Alba
Ruben Alejandro Diaz Cardenas
Rosa Isabel Rosero Cartagena
Ana Laura Valencia Martinez
Carlos Andres Viveros

November 2024

1. Creacion de tablas

Tabla Posiciones

Código de la Tabla

```
-- Active: 1733013713611@@127.0.0.1@3306@futbol

CREATE TABLE posiciones (
    id_posicion CHAR(2) NOT NULL,
    posicion VARCHAR(50) NOT NULL,
    CONSTRAINT pk_id_posicion PRIMARY KEY (id_posicion
)
);
```

CREATE TABLE posiciones: Esta parte de la consulta indica que se está creando una nueva tabla llamada **posiciones**.

Columnas de la Tabla:

id_posicion CHAR(2) NOT NULL:

id_posicion es una columna de tipo **CHAR** con una longitud fija de 2 caracteres. La restricción **NOT NULL** significa que esta columna no puede contener valores nulos, es decir, debe tener un valor asignado en cada fila de la tabla.

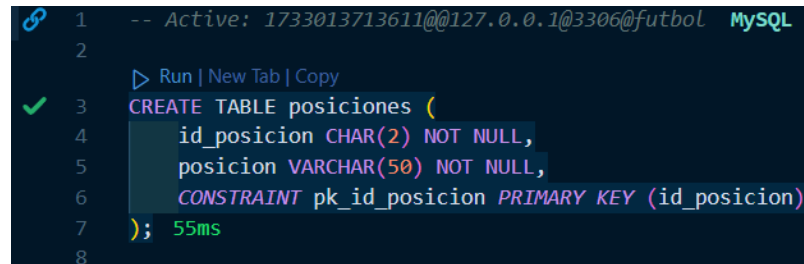
posicion VARCHAR(50) NOT NULL:

posicion es una columna de tipo **VARCHAR**, que permite almacenar cadenas de texto de longitud variable, con un máximo de 50 caracteres. Al igual que **id_posicion**, esta columna también tiene la restricción **NOT NULL**.

Restricción de Clave Primaria:

CONSTRAINT pk_id_posicion PRIMARY KEY (id_posicion):

Esta línea define una restricción de clave primaria llamada **pk_id_posicion** sobre la columna **id_posicion**. La clave primaria asegura que cada valor en esta columna sea único y no nulo, lo que permite identificar de manera única cada fila de la tabla.



```
1  -- Active: 1733013713611@@127.0.0.1@3306@futbol  MySQL
2
3  Run | New Tab | Copy
4  CREATE TABLE posiciones (
5      id_posicion CHAR(2) NOT NULL,
6      posicion VARCHAR(50) NOT NULL,
7      CONSTRAINT pk_id_posicion PRIMARY KEY (id_posicion)
8  ); 55ms
```

Resumen

En resumen, la consulta crea una tabla **posiciones** que almacena información sobre diferentes posiciones, donde cada posición tiene un identificador único de 2 caracteres y un nombre de posición de hasta 50 caracteres. La clave primaria garantiza la unicidad de los identificadores de posición, lo que es fundamental para mantener la integridad de los datos en la base de datos.

Tabla Estadios

Código de la Tabla

```
-- Active: 1733013713611@@127.0.0.1@3306@futbol

CREATE TABLE estadios (
    id_estadio CHAR(2) NOT NULL,
    aforo INT NOT NULL,
    ano_de_fundacion DATE NOT NULL,
    ciudad VARCHAR(150) NOT NULL,
    CONSTRAINT pk_id_estadio PRIMARY KEY (id_estadio)
);
```

CREATE TABLE estadios: Esta parte de la consulta indica que se está creando una nueva tabla llamada **estadios**.

Columnas de la Tabla:

id_estadio CHAR(2) NOT NULL:

id_estadio es una columna de tipo **CHAR** con una longitud fija de 2 caracteres. La restricción **NOT NULL** significa que esta columna no puede contener valores nulos, es decir, debe tener un valor asignado en cada fila de la tabla.

aforo INT NOT NULL:

aforo es una columna de tipo **INT**, que se utiliza para almacenar números enteros. Esta columna representa la capacidad máxima del estadio. También tiene la restricción **NOT NULL**, lo que significa que debe contener un valor en cada fila.

ano_de_fundacion DATE NOT NULL:

ano_de_fundacion es una columna de tipo **DATE**, que almacena fechas. Esta columna indica el año en que se fundó el estadio. Al igual que las anteriores, esta columna también tiene la restricción **NOT NULL**.

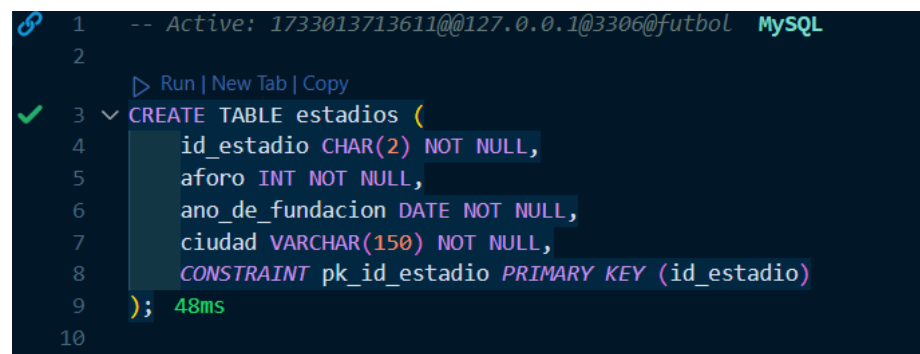
ciudad VARCHAR(150) NOT NULL:

ciudad es una columna de tipo **VARCHAR**, que permite almacenar cadenas de texto de longitud variable, con un máximo de 150 caracteres. Esta columna almacena el nombre de la ciudad donde se encuentra el estadio. Esta columna también tiene la restricción **NOT NULL**.

Restricción de Clave Primaria:

CONSTRAINT pk_id_estadio PRIMARY KEY (id_estadio):

Esta línea define una restricción de clave primaria llamada **pk_id_estadio** sobre la columna **id_estadio**. La clave primaria asegura que cada valor en esta columna sea único y no nulo, lo que permite identificar de manera única cada fila de la tabla.



```
1  -- Active: 1733013713611@@127.0.0.1@3306@futbol  MySQL
2
3  Run | New Tab | Copy
4  ✓ CREATE TABLE estadios (
5      id_estadio CHAR(2) NOT NULL,
6      aforo INT NOT NULL,
7      ano_de_fundacion DATE NOT NULL,
8      ciudad VARCHAR(150) NOT NULL,
9      CONSTRAINT pk_id_estadio PRIMARY KEY (id_estadio)
10 ); 48ms
```

Resumen

En resumen, la consulta crea una tabla **estadios** que almacena información sobre diferentes estadios, donde cada estadio tiene un identificador único de 2 caracteres, un aforo (capacidad) que es un número entero, un año de fundación

que es una fecha, y una ciudad que es una cadena de texto de hasta 150 caracteres. La clave primaria garantiza la unicidad de los identificadores de estadio, lo que es fundamental para mantener la integridad de los datos en la base de datos.

Tabla Equipos

Código de la Tabla

```
-- Active: 1733013713611@@127.0.0.1@3306@futbol

CREATE TABLE equipos (
    id_equipo INT AUTO_INCREMENT,
    nombre_del_equipo VARCHAR(255) NOT NULL,
    id_estadio CHAR(2) NOT NULL,
    CONSTRAINT pk_id_equipo PRIMARY KEY (id_equipo),
    CONSTRAINT fk_id_estadio FOREIGN KEY (id_estadio)
        REFERENCES estadios (id_estadio)
);
```

CREATE TABLE equipos: Esta parte de la consulta indica que se está creando una nueva tabla llamada **equipos**.

Columnas de la Tabla:

id_equipo INT AUTO_INCREMENT:

id_equipo es una columna de tipo **INT** que se utiliza para almacenar números enteros. La opción **AUTO_INCREMENT** significa que este campo se incrementará automáticamente con cada nueva fila insertada, lo que permite generar un identificador único para cada equipo sin necesidad de especificarlo manualmente.

nombre_del_equipo VARCHAR(255) NOT NULL:

nombre_del_equipo es una columna de tipo **VARCHAR**, que permite almacenar cadenas de texto de longitud variable, con un máximo de 255 caracteres. Esta columna almacena el nombre del equipo. La restricción **NOT NULL** significa que esta columna no puede contener valores nulos, es decir, debe tener un valor asignado en cada fila de la tabla.

id_estadio CHAR(2) NOT NULL:

id_estadio es una columna de tipo **CHAR** con una longitud fija de 2 caracteres. Esta columna se utiliza para relacionar el equipo con un estadio específico. También tiene la restricción **NOT NULL**, lo que significa que debe contener un valor en cada fila.

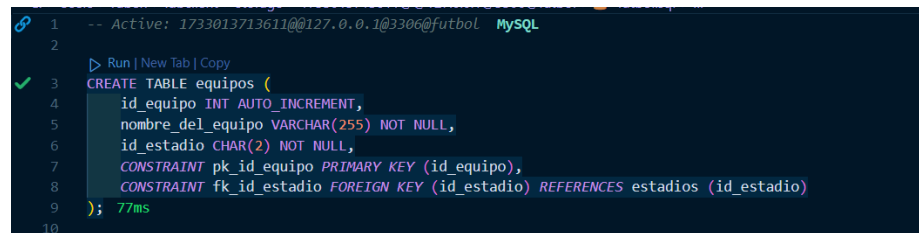
Restricciones de Clave Primaria y Foránea:

CONSTRAINT pk_id_equipo PRIMARY KEY (id_equipo):

Esta línea define una restricción de clave primaria llamada `pk_id_equipo` sobre la columna `id_equipo`. La clave primaria asegura que cada valor en esta columna sea único y no nulo, lo que permite identificar de manera única cada fila de la tabla.

CONSTRAINT fk_id_estadio FOREIGN KEY (id_estadio) REFERENCES estadios (id_estadio):

Esta línea define una restricción de clave foránea llamada `fk_id_estadio` sobre la columna `id_estadio`. Esto establece una relación entre la tabla `equipos` y la tabla `estadios`, asegurando que cualquier valor en `id_estadio` en la tabla `equipos` debe existir en la columna `id_estadio` de la tabla `estadios`. Esto ayuda a mantener la integridad referencial entre las dos tablas.



```
1  -- Active: 1733013713611@@127.0.0.1@3306@futbol  MySQL
2
3  Run | New Tab | Copy
4  CREATE TABLE equipos (
5      id_equipo INT AUTO_INCREMENT,
6      nombre_del_equipo VARCHAR(255) NOT NULL,
7      id_estadio CHAR(2) NOT NULL,
8      CONSTRAINT pk_id_equipo PRIMARY KEY (id_equipo),
9      CONSTRAINT fk_id_estadio FOREIGN KEY (id_estadio) REFERENCES estadios (id_estadio)
10 ); 77ms
```

Resumen

En resumen, la consulta crea una tabla `equipos` que almacena información sobre diferentes equipos, donde cada equipo tiene un identificador único que se incrementa automáticamente, un nombre que es una cadena de texto de hasta 255 caracteres, y un identificador de estadio que se relaciona con la tabla `estadios`. Las restricciones de clave primaria y foránea garantizan la unicidad de los identificadores de equipo y la integridad de las relaciones entre las tablas, respectivamente.

Tabla Jugadores

Código de la Tabla

```
-- Active: 1733013713611@@127.0.0.1@3306@futbol

CREATE TABLE jugadores (
    id_jugador INT AUTO_INCREMENT,
    tipo_de_documento CHAR(3) NOT NULL,
    numero_de_documento INT NOT NULL,
    nombre VARCHAR(150) NOT NULL,
    fecha_de_nacimiento DATE NOT NULL CHECK (
        fecha_de_nacimiento < '2024-1-1'),
    id_posicion CHAR(2) NOT NULL,
    id_equipo INT NOT NULL,
    CONSTRAINT pk_id_jugador PRIMARY KEY (id_jugador),
    CONSTRAINT fk_id_posicion FOREIGN KEY (id_posicion
    ) REFERENCES posiciones (id_posicion),
    CONSTRAINT fk_id_equipo_jugador FOREIGN KEY (
    id_equipo) REFERENCES equipos (id_equipo)
);
```

CREATE TABLE jugadores: Esta parte de la consulta indica que se está creando una nueva tabla llamada `jugadores`.

Columnas de la Tabla:

id_jugador INT AUTO_INCREMENT:

`id_jugador` es una columna de tipo `INT` que se utiliza para almacenar números enteros. La opción `AUTO_INCREMENT` significa que este campo se incrementará automáticamente con cada nueva fila insertada, lo que permite generar un identificador único para cada jugador sin necesidad de especificarlo manualmente.

tipo_de_documento CHAR(3) NOT NULL:

`tipo_de_documento` es una columna de tipo `CHAR` con una longitud fija de 3 caracteres. Esta columna se utiliza para almacenar el tipo de documento del jugador (por ejemplo, DNI, PAS, etc.). La restricción `NOT NULL` significa que esta columna no puede contener valores nulos, es decir, debe tener un valor asignado en cada fila de la tabla.

numero_de_documento INT NOT NULL:

`numero_de_documento` es una columna de tipo `INT` que almacena el número del documento del jugador. Tiene la restricción `NOT NULL`, lo que significa que debe contener un valor en cada fila.

nombre VARCHAR(150) NOT NULL:

`nombre` es una columna de tipo `VARCHAR`, que permite almacenar cadenas de texto de longitud variable, con un máximo de 150 caracteres. Esta columna

almacena el nombre del jugador. La restricción NOT NULL significa que esta columna no puede contener valores nulos, es decir, debe tener un valor asignado en cada fila de la tabla.

fecha_de_nacimiento DATE NOT NULL CHECK (fecha_de_nacimiento < '2024-01-01');

fecha_de_nacimiento es una columna de tipo DATE, que almacena fechas. Esta columna indica la fecha de nacimiento del jugador. Tiene la restricción NOT NULL, lo que significa que debe contener un valor en cada fila. Además, el constraint CHECK asegura que la fecha de nacimiento sea anterior al 1 de enero de 2024, lo que significa que solo se pueden registrar jugadores nacidos antes de esa fecha.

id_posicion CHAR(2) NOT NULL:

id_posicion es una columna de tipo CHAR con una longitud fija de 2 caracteres. Esta columna se utiliza para relacionar al jugador con una posición específica en el campo. Tiene la restricción NOT NULL, lo que significa que debe contener un valor en cada fila.

id_equipo INT NOT NULL:

id_equipo es una columna de tipo INT que se utiliza para relacionar al jugador con un equipo específico. También tiene la restricción NOT NULL, lo que significa que debe contener un valor en cada fila.

Restricciones de Clave Primaria y Foráneas:

CONSTRAINT pk_id_jugador PRIMARY KEY (id_jugador):

Esta línea define una restricción de clave primaria llamada **pk_id_jugador** sobre la columna **id_jugador**. La clave primaria asegura que cada valor en esta columna sea único y no nulo, lo que permite identificar de manera única cada fila de la tabla.

CONSTRAINT fk_id_posicion FOREIGN KEY (id_posicion) REFERENCES posiciones (id_posicion):

Esta línea define una restricción de clave foránea llamada **fk_id_posicion** sobre la columna **id_posicion**. Esto establece una relación entre la tabla **jugadores** y la tabla **posiciones**, asegurando que cualquier valor en **id_posicion** en la tabla **jugadores** debe existir en la columna **id_posicion** de la tabla **posiciones**. Esto ayuda a mantener la integridad referencial entre las dos tablas.

CONSTRAINT fk_id_equipo_jugador FOREIGN KEY (id_equipo) REFERENCES equipos (id_equipo):

Esta línea define una restricción de clave foránea llamada **fk_id_equipo_jugador** sobre la columna **id_equipo**. Esto establece una relación entre la tabla **jugadores** y la tabla **equipos**, asegurando que cualquier valor en **id_equipo** en la tabla **jugadores** debe existir en la columna **id_equipo** de la tabla **equipos**.

```

1  -- Active: 1733013713611@@127.0.0.1@3306@futbol MySQL
2
3  Run | New Tab | Copy
4  CREATE TABLE jugadores (
5      id_jugador INT AUTO_INCREMENT,
6      tipo_de_documento CHAR(3) NOT NULL,
7      numero_de_documento INT NOT NULL,
8      nombre VARCHAR(150) NOT NULL,
9      fecha_de_nacimiento DATE NOT NULL CHECK (fecha_de_nacimiento < '2024-1-1'),
10     id_posicion CHAR(2) NOT NULL,
11     id_equipo INT NOT NULL,
12     CONSTRAINT pk_id_jugador PRIMARY KEY (id_jugador),
13     CONSTRAINT fk_id_posicion FOREIGN KEY (id_posicion) REFERENCES posiciones (id_posicion),
14     CONSTRAINT fk_id_equipo_jugador FOREIGN KEY (id_equipo) REFERENCES equipos (id_equipo)
15 ); 99ms

```

Resumen

En resumen, la consulta crea una tabla `jugadores` que almacena información sobre los jugadores de fútbol, donde cada jugador tiene un identificador único que se incrementa automáticamente, un tipo y número de documento, su nombre, fecha de nacimiento, y se relaciona con una posición y un equipo específico. Las restricciones de clave primaria y foránea garantizan la unicidad de los identificadores de jugador y la integridad de las relaciones entre las tablas, respectivamente.

Tabla Presidentes

Código de la Tabla

```

-- Active: 1733013713611@@127.0.0.1@3306@futbol

CREATE TABLE presidentes (
    id_presidente INT AUTO_INCREMENT,
    id_equipo INT NOT NULL,
    tipo_de_documento CHAR(3) NOT NULL,
    numero_de_documento INT NOT NULL,
    nombre VARCHAR(150) NOT NULL,
    apellido VARCHAR(150) NOT NULL,
    fecha_de_nacimiento DATE NOT NULL,
    fecha_de_eleccion DATE NOT NULL,
    CONSTRAINT pk_id_presidente PRIMARY KEY (
        id_presidente),
    CONSTRAINT fk_id_equipo_presidente FOREIGN KEY (
        id_equipo) REFERENCES equipos (id_equipo)
);

```


CREATE TABLE presidentes: Esta parte de la consulta indica que se está creando una nueva tabla llamada **presidentes**.

Columnas de la Tabla:

id_presidente INT AUTO_INCREMENT:

id_presidente es una columna de tipo INT que se utiliza para almacenar números enteros. La opción **AUTO_INCREMENT** significa que este campo se incrementará automáticamente con cada nueva fila insertada, lo que permite generar un identificador único para cada presidente sin necesidad de especificarlo manualmente.

id_equipo INT NOT NULL:

id_equipo es una columna de tipo INT que se utiliza para relacionar al presidente con un equipo específico. Tiene la restricción **NOT NULL**, lo que significa que debe contener un valor en cada fila.

tipo_de_documento CHAR(3) NOT NULL:

tipo_de_documento es una columna de tipo CHAR con una longitud fija de 3 caracteres. Esta columna se utiliza para almacenar el tipo de documento del presidente (por ejemplo, DNI, PAS, etc.). También tiene la restricción **NOT NULL**, lo que significa que debe contener un valor en cada fila.

numero_de_documento INT NOT NULL:

numero_de_documento es una columna de tipo INT que almacena el número del documento del presidente. Tiene la restricción **NOT NULL**, lo que significa que debe contener un valor en cada fila.

nombre VARCHAR(150) NOT NULL:

nombre es una columna de tipo VARCHAR, que permite almacenar cadenas de texto de longitud variable, con un máximo de 150 caracteres. Esta columna almacena el nombre del presidente. La restricción **NOT NULL** significa que esta columna no puede contener valores nulos, es decir, debe tener un valor asignado en cada fila de la tabla.

apellido VARCHAR(150) NOT NULL:

apellido es una columna de tipo VARCHAR, que permite almacenar cadenas de texto de longitud variable, con un máximo de 150 caracteres. Esta columna almacena el apellido del presidente. También tiene la restricción **NOT NULL**, lo que significa que debe contener un valor en cada fila.

fecha_de_nacimiento DATE NOT NULL:

fecha_de_nacimiento es una columna de tipo DATE, que almacena fechas. Esta columna indica la fecha de nacimiento del presidente. Tiene la restricción **NOT NULL**, lo que significa que debe contener un valor en cada fila.

fecha_de_eleccion DATE NOT NULL:

fecha_de_eleccion es una columna de tipo DATE, que almacena fechas. Esta columna indica la fecha en que el presidente fue elegido. También tiene la restricción **NOT NULL**, lo que significa que debe contener un valor en cada fila.

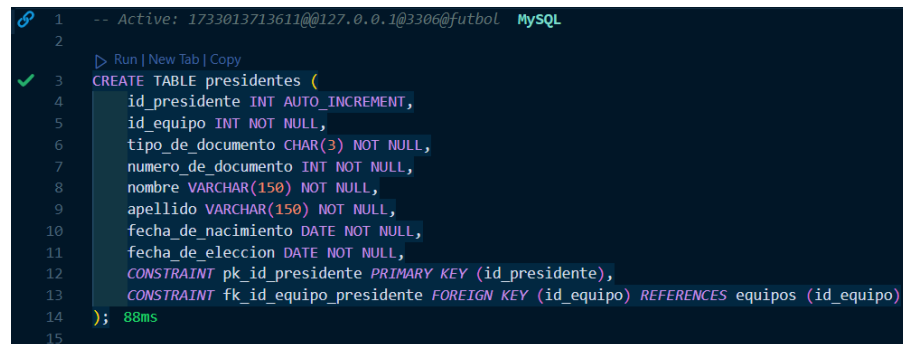
Restricciones de Clave Primaria y Foráneas:

CONSTRAINT pk_id_presidente PRIMARY KEY (id_presidente):

Esta línea define una restricción de clave primaria llamada `pk_id_presidente` sobre la columna `id_presidente`. La clave primaria asegura que cada valor en esta columna sea único y no nulo, lo que permite identificar de manera única cada fila de la tabla.

CONSTRAINT fk_id_equipo_presidente FOREIGN KEY (id_equipo) REFERENCES equipos (id_equipo):

Esta línea define una restricción de clave foránea llamada `fk_id_equipo_presidente` sobre la columna `id_equipo`. Esto establece una relación entre la tabla `presidentes` y la tabla `equipos`, asegurando que cualquier valor en `id_equipo` en la tabla `presidentes` debe existir en la columna `id_equipo` de la tabla `equipos`. Esto ayuda a mantener la integridad referencial entre las dos tablas.



```
1  -- Active: 1733013713611@@127.0.0.1@3306@futbol MySQL
2
3  > Run | New Tab | Copy
4  CREATE TABLE presidentes (
5      id_presidente INT AUTO_INCREMENT,
6      id_equipo INT NOT NULL,
7      tipo_de_documento CHAR(3) NOT NULL,
8      numero_de_documento INT NOT NULL,
9      nombre VARCHAR(150) NOT NULL,
10     apellido VARCHAR(150) NOT NULL,
11     fecha_de_nacimiento DATE NOT NULL,
12     fecha_de_eleccion DATE NOT NULL,
13     CONSTRAINT pk_id_presidente PRIMARY KEY (id_presidente),
14     CONSTRAINT fk_id_equipo_presidente FOREIGN KEY (id_equipo) REFERENCES equipos (id_equipo)
15 ); 88ms
```

Resumen

En resumen, la consulta crea una tabla `presidentes` que almacena información sobre los presidentes de los equipos, donde cada presidente tiene un identificador único que se incrementa automáticamente, un identificador de equipo que se relaciona con la tabla `equipos`, un tipo y número de documento, así como su nombre, apellido, fecha de nacimiento y fecha de elección. Las restricciones de clave primaria y foránea garantizan la unicidad de los identificadores de presidente y la integridad de las relaciones entre las tablas, respectivamente.

Tabla Goles

Código de la Tabla

```
-- Active: 1733013713611@@127.0.0.1@3306@futbol

CREATE TABLE goles (
  id_goles CHAR(2) NOT NULL,
  id_jugador INT NOT NULL,
  id_equipo INT NOT NULL,
  id_partido INT NOT NULL,
  minuto_del_gol INT NOT NULL,
  descripcion VARCHAR(255) NOT NULL,
  CONSTRAINT pk_id_goles PRIMARY KEY (id_goles),
  CONSTRAINT fk_id_jugador FOREIGN KEY (id_jugador)
    REFERENCES jugadores (id_jugador),
  CONSTRAINT fk_id_equipo FOREIGN KEY (id_equipo)
    REFERENCES equipos (id_equipo),
  CONSTRAINT fk_id_partido FOREIGN KEY (id_partido)
    REFERENCES partidos (id_partido)
);
```

CREATE TABLE goles: Esta parte de la consulta indica que se está creando una nueva tabla llamada goles.

Columnas de la Tabla:

id_goles CHAR(2) NOT NULL:

id_goles es una columna de tipo CHAR con una longitud fija de 2 caracteres. Esta columna se utiliza para almacenar un identificador único para cada gol. La restricción NOT NULL significa que esta columna no puede contener valores nulos, es decir, debe tener un valor asignado en cada fila de la tabla.

id_jugador INT NOT NULL:

id_jugador es una columna de tipo INT que se utiliza para relacionar el gol con un jugador específico. Tiene la restricción NOT NULL, lo que significa que debe contener un valor en cada fila.

id_equipo INT NOT NULL:

id_equipo es una columna de tipo INT que se utiliza para relacionar el gol con un equipo específico. También tiene la restricción NOT NULL, lo que significa que debe contener un valor en cada fila.

id_partido INT NOT NULL:

id_partido es una columna de tipo INT que se utiliza para relacionar el gol con un partido específico. Tiene la restricción NOT NULL, lo que significa que debe contener un valor en cada fila.

minuto_del_gol INT NOT NULL:

`minuto_del_gol` es una columna de tipo `INT` que almacena el minuto en el que se anotó el gol durante el partido. También tiene la restricción `NOT NULL`, lo que significa que debe contener un valor en cada fila.

descripcion VARCHAR(255) NOT NULL:

`descripcion` es una columna de tipo `VARCHAR`, que permite almacenar cadenas de texto de longitud variable, con un máximo de 255 caracteres. Esta columna puede almacenar información adicional sobre el gol, como el tipo de gol o una breve descripción. La restricción `NOT NULL` significa que esta columna no puede contener valores nulos, es decir, debe tener un valor asignado en cada fila de la tabla.

Restricciones de Clave Primaria y Foráneas:

CONSTRAINT pk_id_goles PRIMARY KEY (id_goles):

Esta línea define una restricción de clave primaria llamada `pk_id_goles` sobre la columna `id_goles`. La clave primaria asegura que cada valor en esta columna sea único y no nulo, lo que permite identificar de manera única cada fila de la tabla.

CONSTRAINT fk_id_jugador FOREIGN KEY (id_jugador) REFERENCES jugadores (id_jugador):

Esta línea define una restricción de clave foránea llamada `fk_id_jugador` sobre la columna `id_jugador`. Esto establece una relación entre la tabla `goles` y la tabla `jugadores`, asegurando que cualquier valor en `id_jugador` en la tabla `goles` debe existir en la columna `id_jugador` de la tabla `jugadores`. Esto ayuda a mantener la integridad referencial entre las dos tablas.

CONSTRAINT fk_id_equipo FOREIGN KEY (id_equipo) REFERENCES equipos (id_equipo):

Esta línea define una restricción de clave foránea llamada `fk_id_equipo` sobre la columna `id_equipo`. Esto establece una relación entre la tabla `goles` y la tabla `equipos`, asegurando que cualquier valor en `id_equipo` en la tabla `goles` debe existir en la columna `id_equipo` de la tabla `equipos`.

CONSTRAINT fk_id_partido FOREIGN KEY (id_partido) REFERENCES partidos (id_partido):

Esta línea define una restricción de clave foránea llamada `fk_id_partido` sobre la columna `id_partido`. Esto establece una relación entre la tabla `goles` y la tabla `partidos`, asegurando que cualquier valor en `id_partido` en la tabla `goles` debe existir en la columna `id_partido` de la tabla `partidos`.

```

1  -- Active: 1733013713611@@127.0.0.1@3306@futbol  MySQL
2
3  > Run | New Tab | Copy
4  CREATE TABLE goles (
5      id_goles CHAR(2) NOT NULL,
6      id_jugador INT NOT NULL,
7      id_equipo INT NOT NULL,
8      id_partido INT NOT NULL,
9      minuto_del_gol INT NOT NULL,
10     descripcion VARCHAR(255) NOT NULL,
11     CONSTRAINT pk_id_goles PRIMARY KEY (id_goles),
12     CONSTRAINT fk_id_jugador FOREIGN KEY (id_jugador) REFERENCES jugadores (id_jugador),
13     CONSTRAINT fk_id_equipo FOREIGN KEY (id_equipo) REFERENCES equipos (id_equipo),
14     CONSTRAINT fk_id_partido FOREIGN KEY (id_partido) REFERENCES partidos (id_partido)
15 ); 126ms

```

Resumen

En resumen, la consulta crea una tabla **goles** que almacena información sobre los goles anotados en los partidos, donde cada gol tiene un identificador único, se relaciona con un jugador, un equipo y un partido específico, y contiene información sobre el minuto en que se anotó el gol y una descripción adicional. Las restricciones de clave primaria y foránea garantizan la unicidad de los identificadores de goles y la integridad de las relaciones entre las tablas, respectivamente.

Tabla Partidos

Código de la Tabla

```

-- Active: 1733013713611@@127.0.0.1@3306@futbol

CREATE TABLE partidos (
    id_partido INT AUTO_INCREMENT,
    fecha_del_partido DATE NOT NULL,
    goles_equipo_local INT NOT NULL,
    goles_equipo_visitante INT NOT NULL,
    CONSTRAINT pk_id_partido PRIMARY KEY (id_partido)
);

```

CREATE TABLE partidos: Esta parte de la consulta indica que se está creando una nueva tabla llamada **partidos**.

Columnas de la Tabla:

id_partido INT AUTO_INCREMENT:
id_partido es una columna de tipo INT que se utiliza para almacenar números

enteros. La opción `AUTO_INCREMENT` significa que este campo se incrementará automáticamente con cada nueva fila insertada, lo que permite generar un identificador único para cada partido sin necesidad de especificarlo manualmente.

`id_equipo_local INT NOT NULL:`

`id_equipo_local` es una columna de tipo `INT` que almacena el identificador del equipo local que participa en el partido. La restricción `NOT NULL` indica que este campo es obligatorio y no puede estar vacío.

`id_equipo_visitante INT NOT NULL:`

`id_equipo_visitante` es una columna de tipo `INT` que almacena el identificador del equipo visitante que participa en el partido. También tiene la restricción `NOT NULL`, lo que significa que debe contener un valor en cada fila.

`fecha_del_partido DATE NOT NULL:`

`fecha_del_partido` es una columna de tipo `DATE`, que almacena fechas. Esta columna indica la fecha en la que se jugó el partido. Tiene la restricción `NOT NULL`, lo que significa que debe contener un valor en cada fila.

`goles_equipo_local INT NOT NULL:`

`goles_equipo_local` es una columna de tipo `INT` que almacena el número de goles anotados por el equipo local en el partido. También tiene la restricción `NOT NULL`, lo que significa que debe contener un valor en cada fila.

`goles_equipo_visitante INT NOT NULL:`

`goles_equipo_visitante` es una columna de tipo `INT` que almacena el número de goles anotados por el equipo visitante en el partido. Tiene la restricción `NOT NULL`, lo que significa que debe contener un valor en cada fila.

Restricciones de Clave Primaria:

`CONSTRAINT pk_id_partido PRIMARY KEY (id_partido):`

Esta línea define una restricción de clave primaria llamada `pk_id_partido` sobre la columna `id_partido`. La clave primaria asegura que cada valor en esta columna sea único y no nulo, lo que permite identificar de manera única cada fila de la tabla.

```
1  -- Active: 1733013713611@@127.0.0.1@3306@futbol MySQL
2
3  Run | New Tab | Copy
4  CREATE TABLE partidos (
5      id_partido INT AUTO_INCREMENT,
6      id_equipo_local INT NOT NULL,
7      id_equipo_visitante INT NOT NULL,
8      fecha_del_partido DATE NOT NULL,
9      goles_equipo_local INT NOT NULL,
10     goles_equipo_visitante INT NOT NULL,
11     CONSTRAINT pk_id_partido PRIMARY KEY (id_partido)
12 ); 56ms
```

Resumen

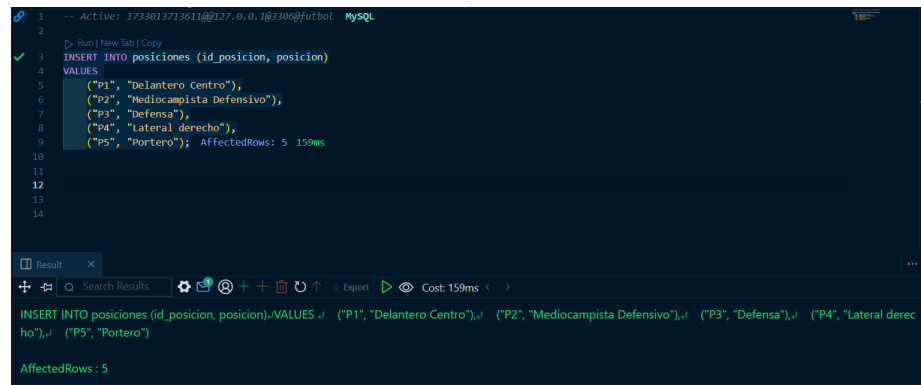
En resumen, la consulta crea una tabla **partidos** que almacena información sobre los partidos de fútbol, donde cada partido tiene un identificador único que se incrementa automáticamente, los identificadores de los equipos local y visitante, la fecha en la que se jugó el partido, y el número de goles anotados por cada equipo. La restricción de clave primaria garantiza la unicidad de los identificadores de partido.

2. Insercion de Datos

2.1. Insertar Posiciones

```
-- Active: 1733013713611@@127.0.0.1@3306@futbol

INSERT INTO posiciones (id_posicion, posicion)
VALUES
    ("P1", "Delantero_Centro"),
    ("P2", "Mediocampista_Defensivo"),
    ("P3", "Defensa"),
    ("P4", "Lateral_derecho"),
    ("P5", "Portero");
```



The screenshot shows a MySQL command window with the following content:

```
1 -- Active: 1733013713611@@127.0.0.1@3306@futbol
2
3 INSERT INTO posiciones (id_posicion, posicion)
4 VALUES
5     ("P1", "Delantero_Centro"),
6     ("P2", "Mediocampista_Defensivo"),
7     ("P3", "Defensa"),
8     ("P4", "Lateral_derecho"),
9     ("P5", "Portero"); AffectedRows: 5 159ms
```

Below the command window, the 'Result' tab is open, showing the output of the INSERT statement:

```
INSERT INTO posiciones (id_posicion, posicion) VALUES ("P1", "Delantero_Centro"), ("P2", "Mediocampista_Defensivo"), ("P3", "Defensa"), ("P4", "Lateral derec
ho"), ("P5", "Portero")
AffectedRows: 5
```

2.2. Insertar Estadios

```
-- Active: 1733013713611@@127.0.0.1@3306@futbol

INSERT INTO estadios (id_estadio, aforo,
    ano_de_fundacion, ciudad)
VALUES
    ("E1", 52000, '2010-02-21', "Palmira"),
    ("E2", 37899, '1937-06-20', "Santiago_de_Cali"),
    ("E3", 46000, '1953-03-18', "Medellin"),
    ("E4", 46000, '1953-03-18', "Medellin"),
    ("E5", 39000, '1938-08-10', "Bogota_D.C.");
```



```

1  -- Active: 1733013713611@@127.0.0.1@3306@futbol  MySQL
2
3  > Run | New Tab | Copy
4  INSERT INTO estadios (id_estadio, aforo, ano_de_fundacion, ciudad)
5  VALUES
6  ("E1", 52000, '2010-02-21', "Palmira"),
7  ("E2", 37899, '1937-06-20', "Santiago de Cali"),
8  ("E3", 46000, '1953-03-18', "Medellin"),
9  ("E4", 46000, '1953-03-18', "Medellin"),
10 ("E5", 39000, '1938-08-10', "Bogota D.C."); AffectedRows: 5 30ms

```

2.3. Insertar Equipos

```

-- Active: 1733013713611@@127.0.0.1@3306@futbol

INSERT INTO equipos (nombre_del_equipo, id_estadio)
VALUES
  ("Deportivo_Cali", "E1"),
  ("America_de_Cali", "E2"),
  ("Nacional", "E3"),
  ("Independiente_Medellin", "E4"),
  ("Millonarios", "E5");

```

```

1  -- Active: 1733013713611@@127.0.0.1@3306@futbol  MySQL
2
3  > Run | New Tab | Copy
4  INSERT INTO equipos (nombre_del_equipo, id_estadio)
5  VALUES
6  ("Deportivo Cali", "E1"),
7  ("America de Cali", "E2"),
8  ("Nacional", "E3"),
9  ("Independiente Medellin", "E4"),
10 ("Millonarios", "E5"); 26ms AffectedRows: 5

```

2.4. Insertar Jugadores

```
-- Active: 1733013713611@@127.0.0.1@3306@futbol

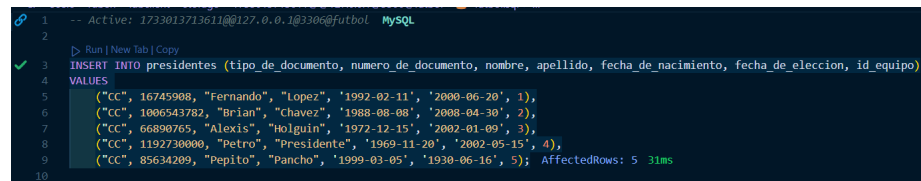
INSERT INTO jugadores (tipo_de_documento,
    numero_de_documento, nombre,
    fecha_de_nacimiento, id_posicion, id_equipo)
VALUES
    ("CC", 12345678, " J u a n Perez ", '1995-03-12', "P1", 1),
    ("CC", 87654321, "Marí a G ómez", '1992-07-25', "P2", 2),
    ("CC", 98765432, " C a r l o s Rodr íguez", '1988-11-05', "P3", 3),
    ("CC", 65432198, " A n a Torres ", '1990-01-15', "P4", 4),
    ("CC", 76543210, " A n a Ramirez ", '2000-12-31', "P5", 5);
```

```
1 -- Active: 1733013713611@@127.0.0.1@3306@futbol MySQL
2
3 > Run | New Tab | Copy
4 INSERT INTO jugadores (tipo_de_documento, numero_de_documento, nombre, fecha_de_nacimiento, id_posicion, id_equipo)
5 VALUES
6     ("CC", 12345678, "Juan Perez", '1995-03-12', "P1", 1),
7     ("CC", 87654321, "Maria Gómez", '1992-07-25', "P2", 2),
8     ("CC", 98765432, "Carlos Rodríguez", '1988-11-05', "P3", 3),
9     ("CC", 65432198, "Ana Torres", '1990-01-15', "P4", 4),
10    ("CC", 76543210, "Ana Ramirez", '2000-12-31', "P5", 5);
```

2.5. Insertar Presidentes

```
-- Active: 1733013713611@@127.0.0.1@3306@futbol

INSERT INTO presidentes (tipo_documento,
    numero_documento, nombre, apellido,
    fecha_de_nacimiento, fecha_de_eleccion, id_equipo)
VALUES
    ("CC", 16745908, "Fernando", "Lopez", '1992-02-11',
        '2000-06-20', 1),
    ("CC", 1006543782, "Brian", "Chavez", '1988-08-08',
        '2008-04-30', 2),
    ("CC", 66890765, "Alexis", "Holguin", '1972-12-15',
        '2002-01-09', 3),
    ("CC", 1192730000, "Petro", "Presidente", '
        1969-11-20', '2002-05-15', 4),
    ("CC", 85634209, "Pepito", "Pancho", '1999-03-05',
        '1930-06-16', 5);
```



```
1 -- Active: 1733013713611@@127.0.0.1@3306@futbol MySQL
2
3 Run | New Tab | Copy
4 INSERT INTO presidentes (tipo_documento, numero_documento, nombre, apellido, fecha_de_nacimiento, fecha_de_eleccion, id_equipo)
5 VALUES
6     ("CC", 16745908, "Fernando", "Lopez", '1992-02-11', '2000-06-20', 1),
7     ("CC", 1006543782, "Brian", "Chavez", '1988-08-08', '2008-04-30', 2),
8     ("CC", 66890765, "Alexis", "Holguin", '1972-12-15', '2002-01-09', 3),
9     ("CC", 1192730000, "Petro", "Presidente", '1969-11-20', '2002-05-15', 4),
10    ("CC", 85634209, "Pepito", "Pancho", '1999-03-05', '1930-06-16', 5); AffectedRows: 5 31ms
```

2.6. Insertar Partidos

```
-- Active: 1733013713611@@127.0.0.1@3306@futbol

INSERT INTO partidos (id_equipo_local,
    id_equipo_visitante, fecha_del_partido,
    goles_equipo_local, goles_equipo_visitante)
VALUES
    (1, 5, '2020-12-06', 2, 2),
    (2, 4, '2000-02-10', 3, 3),
    (3, 1, '2017-02-11', 1, 2),
    (2, 3, '2011-02-12', 2, 2),
    (5, 2, '2024-02-13', 4, 4);
```

```
2
3  Run | New Tab | Copy
4  INSERT INTO partidos (id_equipo_local, id_equipo_visitante, fecha_del_partido,
5  goles_equipo_local, goles_equipo_visitante)
6  VALUES
7  (1, 5, '2020-12-06', 2, 2),
8  (2, 4, '2000-02-10', 3, 3),
9  (3, 1, '2017-02-11', 1, 2),
10 (2, 3, '2011-02-12', 2, 2),
11 (5, 2, '2024-02-13', 4, 4); AffectedRows: 5 14ms
```

2.7. Insertar Goles

```
-- Active: 1733013713611@@127.0.0.1@3306@futbol

INSERT INTO goles (id_goles, id_jugador, id_equipo,
id_partido, minuto_del_gol, descripcion)
VALUES
("G1", 1, 1, 1, 5, "Remate cruzado desde la
derecha"),
("G2", 3, 3, 2, 20, "Gol olimpico"),
("G3", 5, 5, 3, 40, "Penal"),
("G4", 2, 2, 4, 47, "Media volea"),
("G5", 4, 4, 5, 74, "Cabeza");
```

```
1  -- Active: 1733013713611@@127.0.0.1@3306@futbol MySQL
2
3  Run | New Tab | Copy
4  INSERT INTO goles (id_goles, id_jugador, id_equipo, id_partido, minuto_del_gol, descripcion)
5  VALUES
6  ("G1", 1, 1, 1, 5, "Remate cruzado desde la derecha"),
7  ("G2", 3, 3, 2, 20, "Gol olimpico"),
8  ("G3", 5, 5, 3, 40, "Penal"),
9  ("G4", 2, 2, 4, 47, "Media volea"),
10 ("G5", 4, 4, 5, 74, "Cabeza"); AffectedRows: 5 30ms
```

3. Procedimiento almacenado

3.1. Consultar Juan

```
-- Active: 1733013713611@@127.0.0.1@3306@futbol

CREATE PROCEDURE ConsultarGolesJugador(IN
    numero_documento INT)
BEGIN
    SELECT
        j.numero_de_documento AS Documento_Jugador,
        j.nombre AS Nombre_Jugador,
        e.nombre_del_equipo AS Equipo_Perteneciente,
        CONCAT(p.nombre, '□', p.apellido) AS
            Director_Tecnico,
        COUNT(g.id_goles) AS Total_Goles_Realizados
    FROM
        jugadores j
        INNER JOIN equipos e ON j.id_equipo = e.
            id_equipo
        INNER JOIN presidentes p ON e.id_equipo = p.
            id_equipo
        LEFT JOIN goles g ON j.id_jugador = g.
            id_jugador
    WHERE
        j.id_jugador = numero_documento
    GROUP BY
        j.id_jugador,
        j.nombre,
        e.nombre_del_equipo,
        p.nombre,
        p.apellido;
END
```

```

1  -- Active: 1733013713611@@127.0.0.1@3306@futbol  MySQL
2
3  > Run | Copy
4  ✓ CREATE PROCEDURE ConsultarGolesJugador(IN numero_documento INT)
5  BEGIN
6      SELECT
7          j.numero_de_documento AS Documento_Jugador,
8          j.nombre AS Nombre_Jugador,
9          e.nombre_del_equipo AS Equipo_Perteneciente,
10         CONCAT(p.nombre, ' ', p.apellido) AS Director_Tecnico,
11         COUNT(g.id_goles) AS Total_Goles_Realizados
12     FROM
13         jugadores j
14         INNER JOIN equipos e ON j.id_equipo = e.id_equipo
15         INNER JOIN presidentes p ON e.id_equipo = p.id_equipo
16         LEFT JOIN goles g ON j.id_jugador = g.id_jugador
17     WHERE
18         j.id_jugador = numero_documento
19     GROUP BY
20         j.id_jugador,
21         j.nombre,
22         e.nombre_del_equipo,
23         p.nombre,
24         p.apellido;
25 END 28ms

```

4. Ejecución de Procedimiento Almacenado (CALL)

4.1. CALL 1

```
USE FUTBOL;  
  
CALL obtener_info(1);
```

```
1 • USE futbol;  
2  
3 • CALL ConsultarGolesJugador(1);  
4  
5
```

Documento_Jugador	Nombre_Jugador	Equipo_Perteneciente	Director_Tecnico	Total_Goles_Realizados
12345678	Juan Perez	Deportivo Cali	Fernando Lopez	1

#	Time	Action	Message
1	19:09:48	USE futbol	0 row(s) affected
2	19:09:48	CALL ConsultarGolesJugador(1)	1 row(s) returned

4.2. CALL 2

```
USE FUTBOL;  
  
CALL obtener_info(2);
```

```

1 • USE futbol;
2
3 • CALL ConsultarGolesJugador(2);
4
5 |

```

Documento_Jugador	Nombre_Jugador	Equipo_Perteneciente	Director_Tecnico	Total_Goles_Realizados
87654321	Maria Gómez	América de Cali	Brian Chavez	1

#	Time	Action	Message
1	19:10:24	USE futbol	0 row(s) affected
2	19:10:24	CALL ConsultarGolesJugador(2)	1 row(s) returned

4.3. CALL 3

```

USE FUTBOL;

CALL obtener_info(3);

```

```

1 • USE futbol;
2
3 • CALL ConsultarGolesJugador(3);
4
5 |

```

Documento_Jugador	Nombre_Jugador	Equipo_Perteneciente	Director_Tecnico	Total_Goles_Realizados
98765432	Carlos Rodríguez	Nacional	Alexis Holguin	1

#	Time	Action	Message
1	19:10:45	USE futbol	0 row(s) affected
2	19:10:45	CALL ConsultarGolesJugador(3)	1 row(s) returned

4.4. CALL 4

```
USE FUTBOL;  
  
CALL obtener_info(4);
```

```
1 • USE futbol;  
2  
3 • CALL ConsultarGolesJugador(4);  
4  
5
```


Documento_Jugador	Nombre_Jugador	Equipo_Perteneciente	Director_Tecnico	Total_Goles_Realizados
65432198	Ana Torres	Independiente Medellin	Petro Presidente	1

#	Time	Action	Message
1	19:11:10	USE futbol	0 row(s) affected
2	19:11:10	CALL ConsultarGolesJugador(4)	1 row(s) returned

4.5. CALL 5

```
USE FUTBOL;  
  
CALL obtener_info(5);
```

```
1 • USE futbol;
2
3 • CALL ConsultarGolesJugador(5);
4
5
```

Result Grid				
Filter Rows: <input type="text"/>				
Export:  Wrap Cell Content: <input type="checkbox"/>				
Documento_Jugador	Nombre_Jugador	Equipo_Perteneiente	Director_Tecnico	Total_Goles_Realizados
76543210	Ana Ramirez	Milenarios	Pepito Pancho	1

Result 15		
Output		
Action Output		
#	Time	Message
1	19:11:36	USE futbol 0 row(s) affected
2	19:11:36	CALL ConsultarGolesJugador(5) 1 row(s) returned

5. Creacion de Nuevo Usuario

Nuevo Usuario

Código de creación

```
select * from mysql.user; -- muestra los usuarios
                             existentes
CREATE USER 'gestor'@'localhost' IDENTIFIED BY 'pio123
'; -- crea un nuevo usuario llamado 'gestor'
GRANT SELECT, INSERT, UPDATE, DELETE ON futbol.* TO '
gestor'@'localhost'; -- le da permisos especificos
a la base de datos futbol
FLUSH PRIVILEGES; -- ejecuta los cambios sin necesidad
de reiniciar el servidor
```

Gestión de Usuarios y Permisos en MySQL

```
SELECT * FROM mysql.user; -- muestra los usuarios
                             existentes
```

1. SELECT * FROM mysql.user: - Este comando selecciona y muestra todos los registros de la tabla 'user' en la base de datos 'mysql', que contiene información sobre todos los usuarios existentes en el sistema MySQL. Esto permite al administrador ver qué usuarios están configurados y sus detalles.

```
CREATE USER 'gestor'@'localhost' IDENTIFIED BY 'pio123
'; -- crea un nuevo usuario llamado 'gestor'
```

2. CREATE USER 'gestor'@'localhost' IDENTIFIED BY 'pio123': - Este comando crea un nuevo usuario llamado 'gestor' que puede conectarse desde 'localhost' (la máquina local). - 'IDENTIFIED BY 'pio123'' establece la contraseña del usuario como 'pio123'. Es importante elegir contraseñas seguras para proteger el acceso a la base de datos.

```
GRANT SELECT, INSERT, UPDATE, DELETE ON futbol.* TO 'gestor'@'localhost'; -- le da permisos específicos a la base de datos futbol
```

3. GRANT SELECT, INSERT, UPDATE, DELETE ON futbol.* TO 'gestor'@'localhost': - Este comando otorga permisos específicos al usuario 'gestor' para realizar operaciones de 'SELECT', 'INSERT', 'UPDATE' y 'DELETE' en todas las tablas de la base de datos 'futbol'. - 'futbol.*' indica que los permisos se aplican a todas las tablas dentro de la base de datos 'futbol'. Esto permite al usuario gestionar datos en esa base de datos sin tener acceso a otras bases de datos, lo que es una buena práctica de seguridad.

```
FLUSH PRIVILEGES; -- ejecuta los cambios sin necesidad de reiniciar el servidor
```

4. FLUSH PRIVILEGES: - Este comando se utiliza para recargar los privilegios en MySQL. Después de crear un nuevo usuario o modificar permisos, es necesario ejecutar este comando para que los cambios surtan efecto sin necesidad de reiniciar el servidor MySQL. - Esto asegura que el sistema reconozca inmediatamente los nuevos permisos asignados al usuario.

```

1  > Run | ⚡ Reset | JSON
2  select * from mysql.user; -- muestra los usuarios existentes
3  > Run | ⚡ Reset
4  CREATE USER 'gestor'@'localhost' IDENTIFIED BY 'pio123'; -- crea un nuevo usuario llamado 'gestor'
5  > Run | ⚡ Reset
6  GRANT SELECT, INSERT, UPDATE, DELETE ON futbol.* TO 'gestor'@'localhost'; -- le da permisos específicos a la base de datos futbol
7  > Run | ⚡ Reset
8  FLUSH PRIVILEGES; 26ms -- ejecuta los cambios sin necesidad de reiniciar el servidor
9

```

Result X

CREATE USER 'gestor'@'localhost' IDENTIFIED BY 'pio123'

AffectedRows: 0

Cost: 114ms

Resumen

En resumen, estos comandos permiten al administrador de la base de datos:

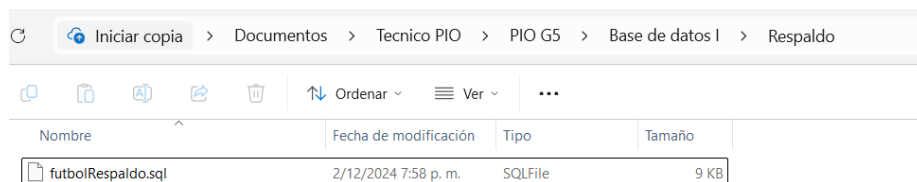
- Ver la lista de usuarios existentes.
- Crear un nuevo usuario con una contraseña específica.

- Asignar permisos específicos a ese usuario para interactuar con una base de datos particular (**futbol**).
- Aplicar los cambios de permisos sin reiniciar el servidor.

6. Respaldo y Restauración

6.1. Creación del respaldo

```
C:\Users\ruben>mysqldump -u root -p futbol >"C:\Users\ruben\Documents\Tecnico PIO\PIO G5\Base de datos I\Respaldo\futbol
Respaldo.sql"
Enter password: *****
C:\Users\ruben>
```



Nombre	Fecha de modificación	Tipo	Tamaño
futbolRespaldo.sql	2/12/2024 7:58 p. m.	SQLFile	9 KB

6.2. Restauración del respaldo

```
mysql> source C:\Users\ruben\Documents\Tecnico PIO\PIO G5\Base de datos I\Respaldo\futbolrespaldo.sql
Query OK, 0 rows affected (0.00 sec)

Query OK, 0 rows affected (0.00 sec)

Query OK, 0 rows affected (0.00 sec)

Query OK, 0 rows affected (0.00 sec)

Query OK, 0 rows affected (0.00 sec)

Query OK, 0 rows affected (0.00 sec)

Query OK, 0 rows affected (0.00 sec)

Query OK, 0 rows affected (0.00 sec)

Query OK, 0 rows affected (0.00 sec)

Query OK, 0 rows affected (0.00 sec)

Query OK, 0 rows affected (0.03 sec)

Query OK, 0 rows affected (0.00 sec)

Query OK, 0 rows affected (0.00 sec)

Query OK, 0 rows affected (0.14 sec)

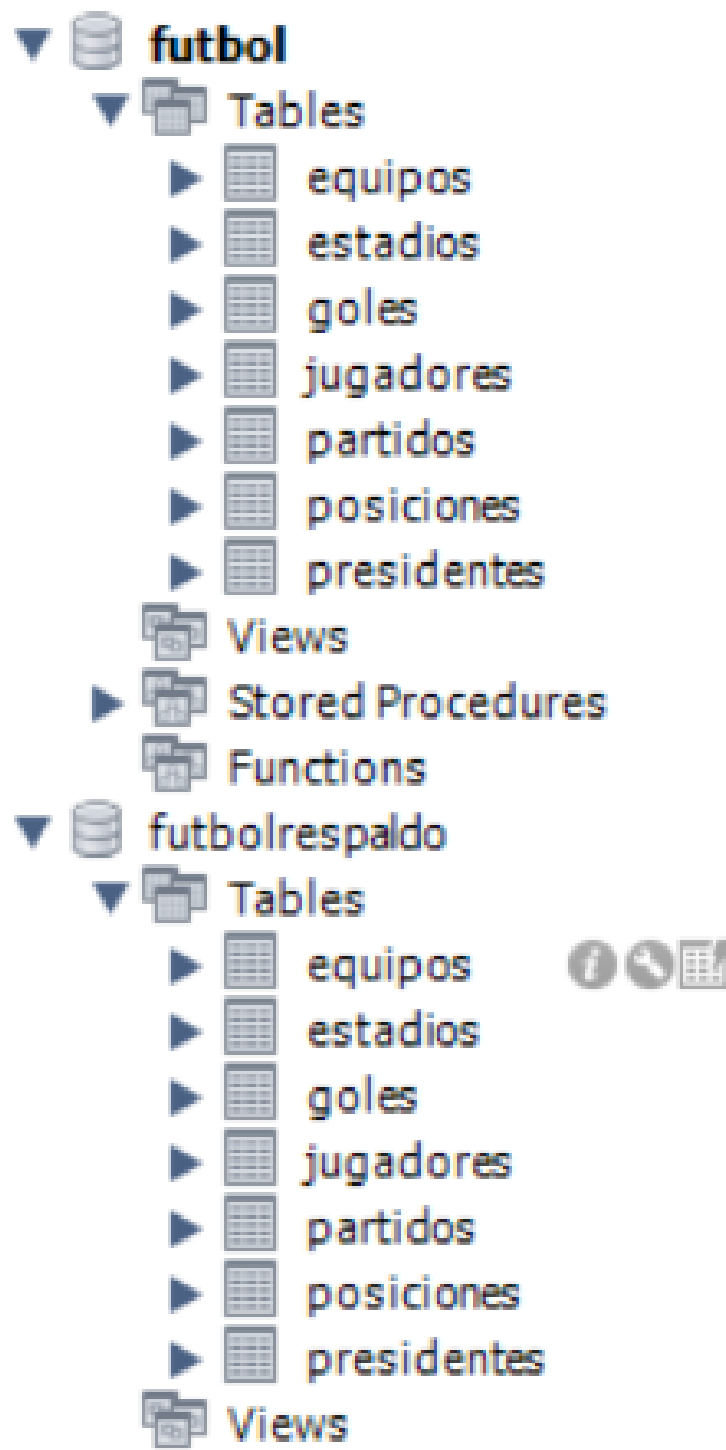
Query OK, 0 rows affected (0.00 sec)

Query OK, 0 rows affected (0.01 sec)

Query OK, 0 rows affected (0.01 sec)

Query OK, 5 rows affected (0.01 sec)
Records: 5 Duplicates: 0 Warnings: 0

Query OK, 0 rows affected (0.01 sec)
```



7. Modelo Lógico de la Base de datos

