BBDD F1



***url Github***

***https://github.com/Abraham-HUB-777/Tarefa-Final-BD-1-DAM***

**Nome Alumno/a: Abraham Fernández Bande**

***Abraham Fernández Bande***

**Curso: *1º DAM* Materia:** ***Bases de Datos – Proyecto Final 24/25***

Contido

[1. Introducción 2](#_Toc198814045)

[2. Descripción del Problema / Requisitos 2](#_Toc198814046)

[3. Modelo Conceptual 5](#_Toc198814047)

[4. Modelo Relacional 6](#_Toc198814048)

[5. Proceso de Normalización 0](#_Toc198814049)

[6. Script de Creación de la Base de Datos 1](#_Toc198814050)

[7. Carga de Datos Inicial 8](#_Toc198814051)

[8. Funciones y Procedimientos Almacenados 28](#_Toc198814052)

[9. Triggers 34](#_Toc198814053)

[10. Consultas SQL 36](#_Toc198814054)

[11. Casos de Prueba y Simulación 41](#_Toc198814055)

[12. Resultados y Verificación 43](#_Toc198814056)

[13. Capturas de Pantalla (opcional) 44](#_Toc198814057)

[14. Conclusiones y Mejoras Futuras 51](#_Toc198814058)

[15. Enlace al Repositorio en GitHub 52](#_Toc198814059)

# Introducción

 El cliente necesita ofrecer a los aficionados y analistas datos precisos sobre resultados, tiempos de vuelta, posiciones en la parrilla, estrategias de carrera y estadísticas completas de pilotos y equipos. Esta información no solo enriquece la experiencia del espectador, sino que también permite un análisis más profundo del rendimiento en la pista.

# Descripción del Problema / Requisitos

En el mundo de los deportes, la gestión y análisis de datos es esencial para los equipos, medios y analistas.

La cantidad de información generada en la F1 para los analistas es extensa y compleja.

Este proyecto busca resolver esa necesidad mediante el diseño y construcción de una base de datos relacional robusta y bien estructurada, que permita almacenar y consultar de manera eficiente toda la información relevante relacionada con:

Pilotos y sus características individuales (nacionalidad, peso, altura, historial, etc.).

Equipos y su desempeño a lo largo de distintas temporadas.

Temporadas, incluyendo ganadores, rendimiento por año y cambios en las alineaciones.

Coches y sus especificaciones técnicas por temporada.

Gran premios, circuitos, condiciones climáticas y resultados por carrera.

Clasificaciones de pilotos y equipos, tanto por carrera como por temporada.

Datos de clasificación (Qualy) y estrategias de carrera como uso de neumáticos, paradas en pits y penalizaciones.

**Requisitos Funcionales:**

Almacenamiento de datos históricos de temporadas, pilotos, equipos, circuitos y carreras.

Consulta de estadísticas por piloto, equipo o temporada.

Visualización de resultados detallados de grandes premios: posiciones finales, mejores vueltas, penalizaciones y condiciones de carrera.

Seguimiento de alineaciones de pilotos y coches por temporada y equipo.

Registro de estrategias de carrera, incluyendo paradas, neumáticos y penalizaciones.

Gestión de clasificaciones actualizadas por temporada para pilotos y equipos.

**Requisitos No Funcionales:**

Estructura relacional normalizada que evite redundancia y mantenga integridad referencial.

Escalabilidad para incorporar futuras temporadas y nuevas métricas.

Mantenibilidad y claridad en la estructura de datos para facilitar futuras extensiones (por ejemplo, prácticas libres, sprint races, etc.).

Compatibilidad con herramientas de análisis y visualización externa.

**Para crear esta BD necesitaremos tablas para almacenar**

**- Piloto**(id, nombre, abreviatura, nacionalidad, altura, peso, fecha\_nacimiento, numero\_licencia, foto\_url);

**- Equipo**(id, nombre, pais, color, sede\_principal, director\_equipo, año\_fundacion, logo\_url);

**- Temporada**(id, año, idPilotoGANADOR, idEquipoGANADOR);

**- CocheTemporada**(id, nombre, motor, idEquipo, idTemporada, especificaciones\_tecnicas, peso\_coche, potencia);

**- PilotoTemporadaEquipo**(id, idTemporada, idPiloto, idEquipo, idCocheTemporada, numeroCoche);

**- Circuito**(id, nombre, localizacion, pais, kilometros, numero\_vueltas, record\_vuelta, tipo\_circuito, longitud\_pit\_lane);

**- GranPremio**(id, nombreGranPremio, idTemporada, idCircuito, fecha, clima, hora\_inicio, vueltas\_completadas, estado);

**- ClasificacionPiloto**(id, idTemporada, idPiloto, posicion, puntos);

**- ClasificacionEquipo**(id, idTemporada, idEquipo, posicion, puntos);

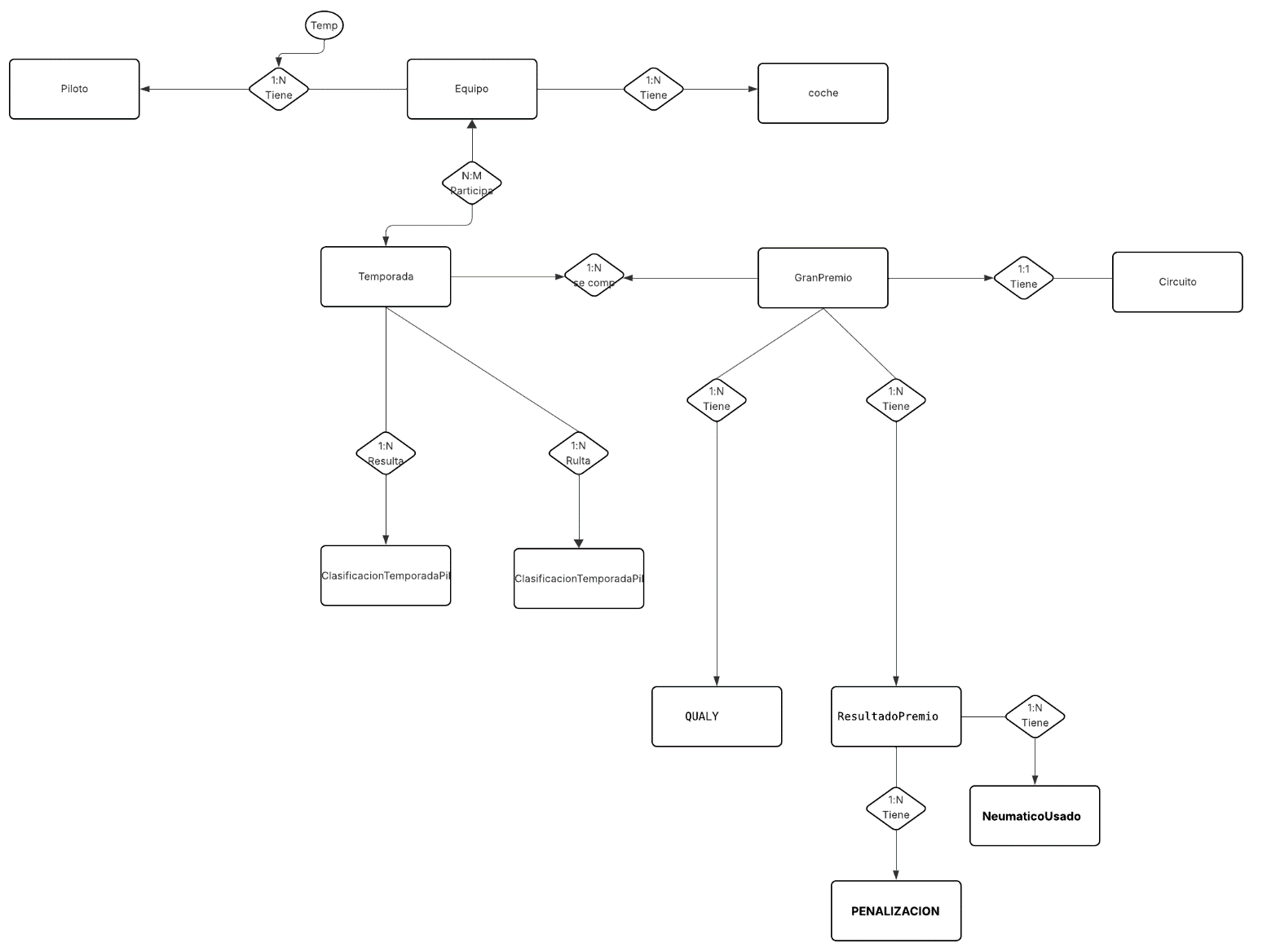
**- Qualy**(id, idGranPremio, idPiloto, idEquipo, q1, q2, q3, posicion, tipo\_clasificacion, condicion\_pista);

**- ResultadoGranPremio**(id, idGranPremo, idPiloto, idEquipo, posicion, tiempo\_total, mejor\_vuelta, vuelta\_mejor\_vuelta, obtuvo\_punto\_vuelta\_rapida, paradas\_box, estado\_final);

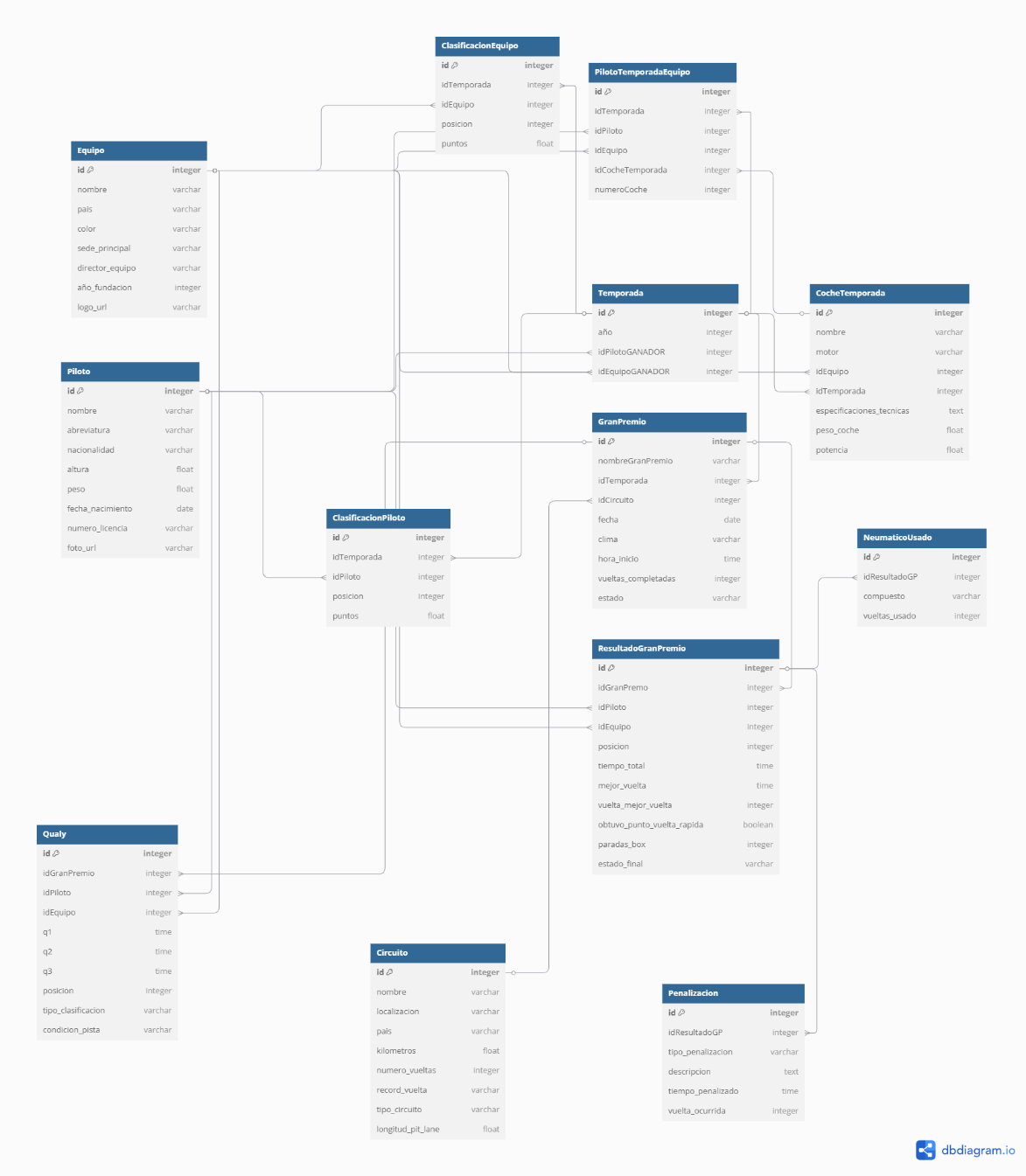
**- NeumaticoUsado**(id, idResultadoGP, compuesto, vueltas\_usado);

**- Penalizacion**(id, idResultadoGP, tipo\_penalizacion, descripcion, tiempo\_penalizado, vuelta\_ocurrida)

# Modelo Conceptual



# Modelo Relacional



# Proceso de Normalización

Paso 1:Primera Forma 1FN

* Todas las tablas cumplen con el 1fn (No hay campos multivaluados)

**Piloto**(id, nombre, abreviatura, edad, nacionalidad, altura, peso, fecha\_nacimiento, numero\_licencia, foto\_url);

**Piloto(id, nombre, abreviatura, nacionalidad, altura, peso, fecha\_nacimiento, numero\_licencia, foto\_url).**

Vamos a eliminar edad ya que puede ser calculada con fecha Nacimiento.

Paso 2:Segunda Forma 2FN

* Todas las tablas cumplen con 2FN (No hay dependencias de claves compuestas)

Paso 3:Tercera Forma 3FN

* No hay dependencias transitivas
* No hay campos redundantes

# Script de Creación de la Base de Datos

drop database if exists Formula1;

create database if not exists Formula1;

use Formula1;

CREATE TABLE Piloto (

id INT AUTO\_INCREMENT PRIMARY KEY,

nombre VARCHAR(255),

abreviatura VARCHAR(50),

nacionalidad VARCHAR(100),

altura FLOAT,

peso FLOAT,

fecha\_nacimiento DATE,

numero\_licencia VARCHAR(50),

foto\_url VARCHAR(255)

);

CREATE TABLE Equipo (

id INT AUTO\_INCREMENT PRIMARY KEY,

nombre VARCHAR(255),

pais VARCHAR(100),

color VARCHAR(50),

sede\_principal VARCHAR(255),

director\_equipo VARCHAR(255),

año\_fundacion INT,

logo\_url VARCHAR(255)

);

CREATE TABLE Temporada (

id INT AUTO\_INCREMENT PRIMARY KEY,

año INT,

idPilotoGANADOR INT,

idEquipoGANADOR INT,

CONSTRAINT fk\_temporada\_piloto FOREIGN KEY (idPilotoGANADOR) REFERENCES Piloto(id),

CONSTRAINT fk\_temporada\_equipo FOREIGN KEY (idEquipoGANADOR) REFERENCES Equipo(id)

);

CREATE TABLE CocheTemporada (

id INT AUTO\_INCREMENT PRIMARY KEY,

nombre VARCHAR(255),

motor VARCHAR(255),

idEquipo INT,

idTemporada INT,

especificaciones\_tecnicas TEXT,

peso\_coche FLOAT,

potencia FLOAT,

CONSTRAINT fk\_coche\_equipo FOREIGN KEY (idEquipo) REFERENCES Equipo(id),

CONSTRAINT fk\_coche\_temporada FOREIGN KEY (idTemporada) REFERENCES Temporada(id)

);

CREATE TABLE PilotoTemporadaEquipo (

id INT AUTO\_INCREMENT PRIMARY KEY,

idTemporada INT,

idPiloto INT,

idEquipo INT,

idCocheTemporada INT,

numeroCoche INT,

CONSTRAINT fk\_pte\_temporada FOREIGN KEY (idTemporada) REFERENCES Temporada(id),

CONSTRAINT fk\_pte\_piloto FOREIGN KEY (idPiloto) REFERENCES Piloto(id),

CONSTRAINT fk\_pte\_equipo FOREIGN KEY (idEquipo) REFERENCES Equipo(id),

CONSTRAINT fk\_pte\_coche FOREIGN KEY (idCocheTemporada) REFERENCES CocheTemporada(id)

);

CREATE TABLE Circuito (

id INT AUTO\_INCREMENT PRIMARY KEY,

nombre VARCHAR(255),

localizacion VARCHAR(255),

pais VARCHAR(100),

kilometros FLOAT,

numero\_vueltas INT,

record\_vuelta VARCHAR(50),

tipo\_circuito VARCHAR(100),

longitud\_pit\_lane FLOAT

);

CREATE TABLE GranPremio (

id INT AUTO\_INCREMENT PRIMARY KEY,

nombreGranPremio VARCHAR(255),

idTemporada INT,

idCircuito INT,

fecha DATE,

clima VARCHAR(100),

hora\_inicio TIME,

vueltas\_completadas INT,

estado VARCHAR(100),

CONSTRAINT fk\_gp\_temporada FOREIGN KEY (idTemporada) REFERENCES Temporada(id),

CONSTRAINT fk\_gp\_circuito FOREIGN KEY (idCircuito) REFERENCES Circuito(id)

);

CREATE TABLE ClasificacionPiloto (

id INT AUTO\_INCREMENT PRIMARY KEY,

idTemporada INT,

idPiloto INT,

posicion INT,

puntos FLOAT,

CONSTRAINT fk\_cp\_temporada FOREIGN KEY (idTemporada) REFERENCES Temporada(id),

CONSTRAINT fk\_cp\_piloto FOREIGN KEY (idPiloto) REFERENCES Piloto(id)

);

CREATE TABLE ClasificacionEquipo (

id INT AUTO\_INCREMENT PRIMARY KEY,

idTemporada INT,

idEquipo INT,

posicion INT,

puntos FLOAT,

CONSTRAINT fk\_ce\_temporada FOREIGN KEY (idTemporada) REFERENCES Temporada(id),

CONSTRAINT fk\_ce\_equipo FOREIGN KEY (idEquipo) REFERENCES Equipo(id)

);

CREATE TABLE Qualy (

id INT AUTO\_INCREMENT PRIMARY KEY,

idGranPremio INT,

idPiloto INT,

idEquipo INT,

q1 TIME,

q2 TIME,

q3 TIME,

posicion INT,

tipo\_clasificacion VARCHAR(100),

condicion\_pista VARCHAR(100),

CONSTRAINT fk\_qualy\_gp FOREIGN KEY (idGranPremio) REFERENCES GranPremio(id),

CONSTRAINT fk\_qualy\_piloto FOREIGN KEY (idPiloto) REFERENCES Piloto(id),

CONSTRAINT fk\_qualy\_equipo FOREIGN KEY (idEquipo) REFERENCES Equipo(id)

);

CREATE TABLE ResultadoGranPremio (

id INT AUTO\_INCREMENT PRIMARY KEY,

idGranPremo INT,

idPiloto INT,

idEquipo INT,

posicion INT,

tiempo\_total TIME,

mejor\_vuelta TIME,

vuelta\_mejor\_vuelta INT,

obtuvo\_punto\_vuelta\_rapida BOOLEAN,

paradas\_box INT,

estado\_final VARCHAR(100),

CONSTRAINT fk\_resultado\_gp FOREIGN KEY (idGranPremio) REFERENCES GranPremio(id),

CONSTRAINT fk\_resultado\_piloto FOREIGN KEY (idPiloto) REFERENCES Piloto(id),

CONSTRAINT fk\_resultado\_equipo FOREIGN KEY (idEquipo) REFERENCES Equipo(id)

);

CREATE TABLE NeumaticoUsado (

id INT AUTO\_INCREMENT PRIMARY KEY,

idResultadoGP INT,

compuesto VARCHAR(100),

vueltas\_usado INT,

CONSTRAINT fk\_neumatico\_resultado FOREIGN KEY (idResultadoGP) REFERENCES ResultadoGranPremio(id)

);

CREATE TABLE Penalizacion (

id INT AUTO\_INCREMENT PRIMARY KEY,

idResultadoGP INT,

tipo\_penalizacion VARCHAR(100),

descripcion TEXT,

tiempo\_penalizado TIME,

vuelta\_ocurrida INT,

CONSTRAINT fk\_penalizacion\_resultado FOREIGN KEY (idResultadoGP) REFERENCES ResultadoGranPremio(id)

);

# Carga de Datos Inicial

-- Equipos

INSERT INTO Equipo (id, nombre, pais, color, sede\_principal, director\_equipo, año\_fundacion, logo\_url) VALUES

(1, 'Red Bull Racing', 'Austria', 'Azul', 'Milton Keynes', 'Christian Horner', 2005, 'https://logo.url/redbull.png'),

(2, 'Mercedes-AMG Petronas', 'Alemania', 'Plateado', 'Brackley', 'Toto Wolff', 2010, 'https://logo.url/mercedes.png'),

(3, 'Scuderia Ferrari', 'Italia', 'Rojo', 'Maranello', 'Frédéric Vasseur', 1950, 'https://logo.url/ferrari.png'),

(4, 'McLaren F1 Team', 'Reino Unido', 'Naranja', 'Woking', 'Andrea Stella', 1963, 'https://logo.url/mclaren.png'),

(5, 'Aston Martin Aramco', 'Reino Unido', 'Verde', 'Silverstone', 'Mike Krack', 2021, 'https://logo.url/astonmartin.png'),

(6, 'Alpine F1 Team', 'Francia', 'Azul', 'Enstone', 'Bruno Famin', 2021, 'https://logo.url/alpine.png'),

(7, 'Williams Racing', 'Reino Unido', 'Azul', 'Grove', 'James Vowles', 1977, 'https://logo.url/williams.png'),

(8, 'Visa Cash App RB', 'Italia', 'Rojo', 'Faenza', 'Laurent Mekies', 2024, 'https://logo.url/vcarb.png'),

(9, 'Haas F1 Team', 'EE.UU.', 'Blanco', 'Kannapolis', 'Ayao Komatsu', 2016, 'https://logo.url/haas.png'),

(10, 'Stake F1 Team Kick Sauber', 'Suiza', 'Negro', 'Hinwil', 'Alessandro Alunni Bravi', 2024, 'https://logo.url/sauber.png');

-- Pilotos

INSERT INTO Piloto (id, nombre, abreviatura, nacionalidad, altura, peso, fecha\_nacimiento, numero\_licencia, foto\_url) VALUES

(1, 'Max Verstappen', 'VER', 'Países Bajos', 1.81, 72, '1997-09-30', 'VER123', 'https://foto.url/verstappen.jpg'),

(2, 'Sergio Pérez', 'PER', 'México', 1.73, 63, '1990-01-26', 'PER123', 'https://foto.url/perez.jpg'),

(3, 'Lewis Hamilton', 'HAM', 'Reino Unido', 1.74, 68, '1985-01-07', 'HAM123', 'https://foto.url/hamilton.jpg'),

(4, 'George Russell', 'RUS', 'Reino Unido', 1.85, 70, '1998-02-15', 'RUS123', 'https://foto.url/russell.jpg'),

(5, 'Charles Leclerc', 'LEC', 'Mónaco', 1.80, 69, '1997-10-16', 'LEC123', 'https://foto.url/leclerc.jpg'),

(6, 'Carlos Sainz', 'SAI', 'España', 1.78, 67, '1994-09-01', 'SAI123', 'https://foto.url/sainz.jpg'),

(7, 'Lando Norris', 'NOR', 'Reino Unido', 1.76, 68, '1999-11-13', 'NOR123', 'https://foto.url/norris.jpg'),

(8, 'Oscar Piastri', 'PIA', 'Australia', 1.78, 68, '2001-04-06', 'PIA123', 'https://foto.url/piastri.jpg'),

(9, 'Fernando Alonso', 'ALO', 'España', 1.71, 68, '1981-07-29', 'ALO123', 'https://foto.url/alonso.jpg'),

(10, 'Lance Stroll', 'STR', 'Canadá', 1.82, 70, '1998-10-29', 'STR123', 'https://foto.url/stroll.jpg'),

(11, 'Esteban Ocon', 'OCO', 'Francia', 1.86, 66, '1996-09-17', 'OCO123', 'https://foto.url/ocon.jpg'),

(12, 'Pierre Gasly', 'GAS', 'Francia', 1.77, 70, '1996-02-07', 'GAS123', 'https://foto.url/gasly.jpg'),

(13, 'Alexander Albon', 'ALB', 'Tailandia', 1.86, 74, '1996-03-23', 'ALB123', 'https://foto.url/albon.jpg'),

(14, 'Logan Sargeant', 'SAR', 'EE.UU.', 1.81, 71, '2000-12-31', 'SAR123', 'https://foto.url/sargeant.jpg'),

(15, 'Yuki Tsunoda', 'TSU', 'Japón', 1.59, 54, '2000-05-11', 'TSU123', 'https://foto.url/tsunoda.jpg'),

(16, 'Daniel Ricciardo', 'RIC', 'Australia', 1.80, 66, '1989-07-01', 'RIC123', 'https://foto.url/ricciardo.jpg'),

(17, 'Kevin Magnussen', 'MAG', 'Dinamarca', 1.74, 68, '1992-10-05', 'MAG123', 'https://foto.url/magnussen.jpg'),

(18, 'Nico Hülkenberg', 'HUL', 'Alemania', 1.84, 74, '1987-08-19', 'HUL123', 'https://foto.url/hulkenberg.jpg'),

(19, 'Valtteri Bottas', 'BOT', 'Finlandia', 1.73, 69, '1989-08-28', 'BOT123', 'https://foto.url/bottas.jpg'),

(20, 'Zhou Guanyu', 'ZHO', 'China', 1.75, 70, '1999-05-30', 'ZHO123', 'https://foto.url/zhou.jpg');

-- Temporada

INSERT INTO Temporada (id, año, idPilotoGANADOR, idEquipoGANADOR) VALUES

(1, 2024, 1, 1);

-- CocheTemporada

INSERT INTO CocheTemporada (id, nombre, motor, idEquipo, idTemporada, especificaciones\_tecnicas, peso\_coche, potencia) VALUES

(1, 'RB20', 'Honda RBPTH001', 1, 1, 'Chasis RB20', 798, 1050),

(2, 'W15', 'Mercedes-AMG F1 M15 E Performance', 2, 1, 'Chasis W15', 798, 1050),

(3, 'SF-24', 'Ferrari 066/10', 3, 1, 'Chasis SF-24', 798, 1050),

(4, 'MCL38', 'Mercedes', 4, 1, 'Chasis MCL38', 798, 1050),

(5, 'AMR24', 'Mercedes', 5, 1, 'Chasis AMR24', 798, 1050),

(6, 'A524', 'Renault', 6, 1, 'Chasis A524', 798, 1050),

(7, 'FW46', 'Mercedes', 7, 1, 'Chasis FW46', 798, 1050),

(8, 'VCARB 01', 'Honda RBPT', 8, 1, 'Chasis VCARB 01', 798, 1050),

(9, 'VF-24', 'Ferrari', 9, 1, 'Chasis VF-24', 798, 1050),

(10, 'C44', 'Ferrari', 10, 1, 'Chasis C44', 798, 1050);

-- PilotoTemporadaEquipo

INSERT INTO PilotoTemporadaEquipo (idTemporada, idPiloto, idEquipo, idCocheTemporada, numeroCoche) VALUES

(1, 1, 1, 1, 1),

(1, 2, 1, 1, 11),

(1, 3, 2, 2, 44),

(1, 4, 2, 2, 63),

(1, 5, 3, 3, 16),

(1, 6, 3, 3, 55),

(1, 7, 4, 4, 4),

(1, 8, 4, 4, 81),

(1, 9, 5, 5, 14),

(1, 10, 5, 5, 18),

(1, 11, 6, 6, 31),

(1, 12, 6, 6, 10),

(1, 13, 7, 7, 23),

(1, 14, 7, 7, 2),

(1, 15, 8, 8, 22),

(1, 16, 8, 8, 3),

(1, 17, 9, 9, 20),

(1, 18, 9, 9, 27),

(1, 19, 10, 10, 77),

(1, 20, 10, 10, 24);

-- Circuitos

INSERT INTO Circuito (id, nombre, localizacion, pais, kilometros, numero\_vueltas, record\_vuelta, tipo\_circuito, longitud\_pit\_lane) VALUES

(1, 'Bahrain International Circuit', 'Sakhir', 'Baréin', 5.412, 57, '1:31.447', 'Permanente', 420.0),

(2, 'Jeddah Street Circuit', 'Jeddah', 'Arabia Saudita', 6.174, 50, '1:30.734', 'Callejero', 430.0),

(3, 'Albert Park Circuit', 'Melbourne', 'Australia', 5.303, 58, '1:20.260', 'Semi-permanente', 410.0);

-- Gran Premio

INSERT INTO GranPremio (id, nombreGranPremio, idTemporada, idCircuito, fecha, clima, hora\_inicio, vueltas\_completadas, estado) VALUES

(1, 'Gran Premio de Baréin', 1, 1, '2024-03-03', 'Soleado', '15:00:00', 57, 'Finalizado'),

(2, 'Gran Premio de Arabia Saudita', 1, 2, '2024-03-10', 'Nublado', '17:00:00', 50, 'Finalizado'),

(3, 'Gran Premio de Australia', 1, 3, '2024-03-17', 'Lluvia', '14:00:00', 58, 'Finalizado');

-- Clasificacion Piloto

INSERT INTO ClasificacionPiloto (id, idTemporada, idPiloto, posicion, puntos) VALUES

(1, 1, 1, 1, 390),

(2, 1, 3, 2, 320),

(3, 1, 5, 3, 270),

(4, 1, 6, 4, 210),

(5, 1, 2, 5, 190),

(6, 1, 4, 6, 160),

(7, 1, 7, 7, 130),

(8, 1, 8, 8, 110),

(9, 1, 9, 9, 90),

(10, 1, 11, 10, 80),

(11, 1, 12, 11, 75),

(12, 1, 13, 12, 60),

(13, 1, 14, 13, 50),

(14, 1, 15, 14, 45),

(15, 1, 16, 15, 40),

(16, 1, 17, 16, 30),

(17, 1, 18, 17, 25),

(18, 1, 19, 18, 20),

(19, 1, 20, 19, 15),

(20, 1, 10, 20, 10);

-- Clasificacion EQuipo

INSERT INTO ClasificacionEquipo (id, idTemporada, idEquipo, posicion, puntos) VALUES

(1, 1, 1, 1, 580),

(2, 1, 2, 2, 530),

(3, 1, 3, 3, 470),

(4, 1, 4, 4, 360),

(5, 1, 5, 5, 300),

(6, 1, 6, 6, 260),

(7, 1, 7, 7, 210),

(8, 1, 8, 8, 180),

(9, 1, 9, 9, 120),

(10, 1, 10, 10, 90);

-- ResultadosGranPRemio

INSERT INTO ResultadoGranPremio

(id, idGranPremio, idPiloto, idEquipo, posicion, tiempo\_total, mejor\_vuelta, vuelta\_mejor\_vuelta, obtuvo\_punto\_vuelta\_rapida, paradas\_box, estado\_final) VALUES

(1, 1, 1, 1, 1, '01:30:00', '01:31.000', 40, TRUE, 2, 'Finalizó'),

(2, 1, 3, 2, 2, '01:30:15', '01:31.200', 42, FALSE, 3, 'Finalizó'),

(3, 1, 5, 3, 3, '01:30:30', '01:31.500', 45, FALSE, 2, 'Finalizó'),

(4, 1, 6, 3, 4, '01:31:10', '01:31.800', 38, FALSE, 2, 'Finalizó'),

(5, 1, 2, 1, 5, '01:31:40', '01:32.000', 44, FALSE, 3, 'Finalizó'),

(6, 1, 4, 2, 6, '01:32:10', '01:32.200', 39, FALSE, 3, 'Finalizó'),

(7, 1, 7, 4, 7, '01:32:45', '01:32.500', 40, FALSE, 3, 'Finalizó'),

(8, 1, 8, 4, 8, '01:33:20', '01:32.700', 41, FALSE, 2, 'Finalizó'),

(9, 1, 9, 5, 9, '01:33:50', '01:33.000', 43, FALSE, 2, 'Finalizó'),

(10, 1, 11, 6, 10, '01:34:15', '01:33.300', 42, FALSE, 2, 'Finalizó'),

(11, 1, 10, 5, 11, '01:34:40', '01:33.600', 46, FALSE, 3, 'Finalizó'),

(12, 1, 12, 6, 12, '01:35:05', '01:33.800', 47, FALSE, 3, 'Finalizó'),

(13, 1, 13, 7, 13, '01:35:30', '01:34.000', 48, FALSE, 2, 'Finalizó'),

(14, 1, 14, 7, 14, '01:35:55', '01:34.200', 49, FALSE, 2, 'Finalizó'),

(15, 1, 15, 8, 15, '01:36:20', '01:34.400', 50, FALSE, 2, 'Finalizó'),

(16, 1, 16, 8, 16, '01:36:45', '01:34.600', 51, FALSE, 2, 'Finalizó'),

(17, 1, 17, 9, 17, '01:37:10', '01:34.800', 52, FALSE, 3, 'Finalizó'),

(18, 1, 18, 9, 18, '01:37:35', '01:35.000', 53, FALSE, 3, 'Finalizó'),

(19, 1, 19, 10, 19, '01:38:00', '01:35.200', 54, FALSE, 2, 'Finalizó'),

(20, 1, 20, 10, 20, '01:38:25', '01:35.400', 55, FALSE, 2, 'Finalizó'),

(21, 2, 3, 2, 1, '01:35:00', '01:30.700', 33, TRUE, 2, 'Finalizó'),

(22, 2, 1, 1, 2, '01:35:10', '01:30.900', 35, FALSE, 3, 'Finalizó'),

(23, 2, 6, 3, 3, '01:35:25', '01:31.100', 32, FALSE, 2, 'Finalizó'),

(24, 2, 5, 3, 4, '01:35:40', '01:31.300', 34, FALSE, 3, 'Finalizó'),

(25, 2, 4, 2, 5, '01:36:00', '01:31.600', 36, FALSE, 2, 'Finalizó'),

(26, 2, 2, 1, 6, '01:36:20', '01:31.900', 37, FALSE, 3, 'Finalizó'),

(27, 2, 7, 4, 7, '01:36:45', '01:32.100', 38, FALSE, 3, 'Finalizó'),

(28, 2, 8, 4, 8, '01:37:05', '01:32.400', 39, FALSE, 2, 'Finalizó'),

(29, 2, 9, 5, 9, '01:37:30', '01:32.700', 40, FALSE, 2, 'Finalizó'),

(30, 2, 11, 6, 10, '01:37:55', '01:33.000', 41, FALSE, 3, 'Finalizó'),

(31, 2, 10, 5, 11, '01:38:20', '01:33.300', 42, FALSE, 2, 'Finalizó'),

(32, 2, 12, 6, 12, '01:38:45', '01:33.600', 43, FALSE, 3, 'Finalizó'),

(33, 2, 13, 7, 13, '01:39:10', '01:33.900', 44, FALSE, 2, 'Finalizó'),

(34, 2, 14, 7, 14, '01:39:35', '01:34.200', 45, FALSE, 2, 'Finalizó'),

(35, 2, 15, 8, 15, '01:40:00', '01:34.500', 46, FALSE, 2, 'Finalizó'),

(36, 2, 16, 8, 16, '01:40:25', '01:34.800', 47, FALSE, 3, 'Finalizó'),

(37, 2, 17, 9, 17, '01:40:50', '01:35.100', 48, FALSE, 2, 'Finalizó'),

(38, 2, 18, 9, 18, '01:41:15', '01:35.400', 49, FALSE, 3, 'Finalizó'),

(39, 2, 19, 10, 19, '01:41:40', '01:35.700', 50, FALSE, 2, 'Finalizó'),

(40, 2, 20, 10, 20, '01:42:05', '01:36.000', 51, FALSE, 3, 'Finalizó'),

(41, 3, 5, 3, 1, '01:25:00', '01:20.100', 20, TRUE, 2, 'Finalizó'),

(42, 3, 1, 1, 2, '01:25:15', '01:20.300', 22, FALSE, 3, 'Finalizó'),

(43, 3, 3, 2, 3, '01:25:30', '01:20.500', 21, FALSE, 2, 'Finalizó'),

(44, 3, 6, 3, 4, '01:25:50', '01:20.700', 23, FALSE, 3, 'Finalizó'),

(45, 3, 4, 2, 5, '01:26:10', '01:21.000', 25, FALSE, 2, 'Finalizó'),

(46, 3, 2, 1, 6, '01:26:30', '01:21.200', 24, FALSE, 3, 'Finalizó'),

(47, 3, 7, 4, 7, '01:26:55', '01:21.400', 26, FALSE, 3, 'Finalizó'),

(48, 3, 8, 4, 8, '01:27:15', '01:21.700', 27, FALSE, 2, 'Finalizó'),

(49, 3, 9, 5, 9, '01:27:40', '01:22.000', 28, FALSE, 2, 'Finalizó'),

(50, 3, 11, 6, 10, '01:28:05', '01:22.300', 29, FALSE, 3, 'Finalizó'),

(51, 3, 10, 5, 11, '01:28:30', '01:22.600', 30, FALSE, 2, 'Finalizó'),

(52, 3, 12, 6, 12, '01:28:55', '01:22.900', 31, FALSE, 2, 'Finalizó'),

(53, 3, 13, 7, 13, '01:29:20', '01:23.200', 32, FALSE, 2, 'Finalizó'),

(54, 3, 14, 7, 14, '01:29:45', '01:23.500', 33, FALSE, 3, 'Finalizó'),

(55, 3, 15, 8, 15, '01:30:10', '01:23.800', 34, FALSE, 2, 'Finalizó'),

(56, 3, 16, 8, 16, '01:30:35', '01:24.100', 35, FALSE, 3, 'Finalizó'),

(57, 3, 17, 9, 17, '01:31:00', '01:24.400', 36, FALSE, 2, 'Finalizó'),

(58, 3, 18, 9, 18, '01:31:25', '01:24.700', 37, FALSE, 3, 'Finalizó'),

(59, 3, 19, 10, 19, '01:31:50', '01:25.000', 38, FALSE, 2, 'Finalizó'),

(60, 3, 20, 10, 20, '01:32:15', '01:25.300', 39, FALSE, 3, 'Finalizó');

-- Qualy

INSERT INTO Qualy (idGranPremio, idPiloto, idEquipo, q1, q2, q3, posicion, tipo\_clasificacion, condicion\_pista) VALUES

(1, 1, 1, '1:30.050', '1:29.800', '1:29.300', 1, 'Clasificación Oficial', 'Seco'),

(1, 2, 1, '1:30.900', '1:30.100', '1:29.900', 2, 'Clasificación Oficial', 'Seco'),

(1, 3, 2, '1:31.000', '1:30.500', '1:30.100', 3, 'Clasificación Oficial', 'Seco'),

(1, 5, 3, '1:31.200', '1:30.700', '1:30.300', 4, 'Clasificación Oficial', 'Seco'),

(1, 4, 2, '1:31.300', '1:30.800', '1:30.400', 5, 'Clasificación Oficial', 'Seco'),

(1, 6, 3, '1:31.400', '1:31.000', '1:30.700', 6, 'Clasificación Oficial', 'Seco'),

(1, 7, 4, '1:31.600', '1:31.200', '1:30.900', 7, 'Clasificación Oficial', 'Seco'),

(1, 8, 4, '1:31.700', '1:31.300', '1:31.000', 8, 'Clasificación Oficial', 'Seco'),

(1, 9, 5, '1:31.800', '1:31.400', '1:31.200', 9, 'Clasificación Oficial', 'Seco'),

(1, 10, 5, '1:31.900', '1:31.500', '1:31.300', 10, 'Clasificación Oficial', 'Seco'),

(1, 11, 6, '1:32.000', '1:31.700', NULL, 11, 'Clasificación Oficial', 'Seco'),

(1, 12, 6, '1:32.100', '1:31.800', NULL, 12, 'Clasificación Oficial', 'Seco'),

(1, 13, 7, '1:32.200', '1:31.900', NULL, 13, 'Clasificación Oficial', 'Seco'),

(1, 14, 7, '1:32.300', '1:32.000', NULL, 14, 'Clasificación Oficial', 'Seco'),

(1, 15, 8, '1:32.400', '1:32.100', NULL, 15, 'Clasificación Oficial', 'Seco'),

(1, 16, 8, '1:32.500', '1:32.200', NULL, 16, 'Clasificación Oficial', 'Seco'),

(1, 17, 9, '1:32.600', NULL, NULL, 17, 'Clasificación Oficial', 'Seco'),

(1, 18, 9, '1:32.700', NULL, NULL, 18, 'Clasificación Oficial', 'Seco'),

(1, 19, 10, '1:32.800', NULL, NULL, 19, 'Clasificación Oficial', 'Seco'),

(1, 20, 10, '1:32.900', NULL, NULL, 20, 'Clasificación Oficial', 'Seco'),

(2, 3, 2, '1:29.700', '1:29.200', '1:28.800', 1, 'Clasificación Oficial', 'Nublado'),

(2, 1, 1, '1:29.800', '1:29.300', '1:28.900', 2, 'Clasificación Oficial', 'Nublado'),

(2, 5, 3, '1:30.100', '1:29.600', '1:29.100', 3, 'Clasificación Oficial', 'Nublado'),

(2, 2, 1, '1:30.400', '1:29.900', '1:29.400', 4, 'Clasificación Oficial', 'Nublado'),

(2, 4, 2, '1:30.600', '1:30.000', '1:29.600', 5, 'Clasificación Oficial', 'Nublado'),

(2, 6, 3, '1:30.800', '1:30.100', '1:29.800', 6, 'Clasificación Oficial', 'Nublado'),

(2, 7, 4, '1:31.000', '1:30.300', '1:30.000', 7, 'Clasificación Oficial', 'Nublado'),

(2, 8, 4, '1:31.100', '1:30.400', '1:30.200', 8, 'Clasificación Oficial', 'Nublado'),

(2, 9, 5, '1:31.300', '1:30.600', '1:30.300', 9, 'Clasificación Oficial', 'Nublado'),

(2, 10, 5, '1:31.500', '1:30.800', '1:30.400', 10, 'Clasificación Oficial', 'Nublado'),

(2, 11, 6, '1:31.700', '1:30.900', NULL, 11, 'Clasificación Oficial', 'Nublado'),

(2, 12, 6, '1:31.800', '1:31.000', NULL, 12, 'Clasificación Oficial', 'Nublado'),

(2, 13, 7, '1:31.900', '1:31.100', NULL, 13, 'Clasificación Oficial', 'Nublado'),

(2, 14, 7, '1:32.000', '1:31.300', NULL, 14, 'Clasificación Oficial', 'Nublado'),

(2, 15, 8, '1:32.200', '1:31.400', NULL, 15, 'Clasificación Oficial', 'Nublado'),

(2, 16, 8, '1:32.300', '1:31.600', NULL, 16, 'Clasificación Oficial', 'Nublado'),

(2, 17, 9, '1:32.500', NULL, NULL, 17, 'Clasificación Oficial', 'Nublado'),

(2, 18, 9, '1:32.600', NULL, NULL, 18, 'Clasificación Oficial', 'Nublado'),

(2, 19, 10, '1:32.700', NULL, NULL, 19, 'Clasificación Oficial', 'Nublado'),

(2, 20, 10, '1:32.800', NULL, NULL, 20, 'Clasificación Oficial', 'Nublado'),

(3, 5, 3, '1:28.900', '1:28.400', '1:28.000', 1, 'Clasificación Oficial', 'Lluvia'),

(3, 6, 3, '1:29.100', '1:28.600', '1:28.200', 2, 'Clasificación Oficial', 'Lluvia'),

(3, 1, 1, '1:29.300', '1:28.800', '1:28.400', 3, 'Clasificación Oficial', 'Lluvia'),

(3, 3, 2, '1:29.400', '1:29.000', '1:28.700', 4, 'Clasificación Oficial', 'Lluvia'),

(3, 4, 2, '1:29.500', '1:29.100', '1:28.800', 5, 'Clasificación Oficial', 'Lluvia'),

(3, 7, 4, '1:29.600', '1:29.200', '1:28.900', 6, 'Clasificación Oficial', 'Lluvia'),

(3, 2, 1, '1:29.700', '1:29.300', '1:29.000', 7, 'Clasificación Oficial', 'Lluvia'),

(3, 8, 4, '1:29.900', '1:29.400', '1:29.100', 8, 'Clasificación Oficial', 'Lluvia'),

(3, 9, 5, '1:30.000', '1:29.500', '1:29.200', 9, 'Clasificación Oficial', 'Lluvia'),

(3, 10, 5, '1:30.100', '1:29.600', '1:29.300', 10, 'Clasificación Oficial', 'Lluvia'),

(3, 11, 6, '1:30.200', '1:29.700', NULL, 11, 'Clasificación Oficial', 'Lluvia'),

(3, 12, 6, '1:30.300', '1:29.800', NULL, 12, 'Clasificación Oficial', 'Lluvia'),

(3, 13, 7, '1:30.400', '1:29.900', NULL, 13, 'Clasificación Oficial', 'Lluvia'),

(3, 14, 7, '1:30.500', '1:30.000', NULL, 14, 'Clasificación Oficial', 'Lluvia'),

(3, 15, 8, '1:30.600', '1:30.100', NULL, 15, 'Clasificación Oficial', 'Lluvia'),

(3, 16, 8, '1:30.700', '1:30.200', NULL, 16, 'Clasificación Oficial', 'Lluvia'),

(3, 17, 9, '1:30.800', NULL, NULL, 17, 'Clasificación Oficial', 'Lluvia'),

(3, 18, 9, '1:30.900', NULL, NULL, 18, 'Clasificación Oficial', 'Lluvia'),

(3, 19, 10, '1:31.000', NULL, NULL, 19, 'Clasificación Oficial', 'Lluvia'),

(3, 20, 10, '1:31.100', NULL, NULL, 20, 'Clasificación Oficial', 'Lluvia');

-- Nuematicos

INSERT INTO NeumaticoUsado (id, idResultadoGP, compuesto, vueltas\_usado) VALUES

(1, 1, 'Blando', 20),

(2, 1, 'Medio', 25),

(3, 2, 'Medio', 30),

(4, 2, 'Duro', 20),

(5, 3, 'Blando', 22),

(6, 3, 'Medio', 28),

(7, 4, 'Medio', 35),

(8, 4, 'Duro', 10),

(9, 5, 'Blando', 25),

(10, 5, 'Medio', 25),

(11, 6, 'Medio', 30),

(12, 6, 'Duro', 20),

(13, 7, 'Blando', 18),

(14, 7, 'Medio', 30),

(15, 8, 'Blando', 25),

(16, 8, 'Medio', 25),

(17, 9, 'Medio', 35),

(18, 9, 'Duro', 15),

(19, 10, 'Blando', 20),

(20, 10, 'Medio', 30),

(21, 11, 'Medio', 40),

(22, 11, 'Duro', 10),

(23, 12, 'Blando', 25),

(24, 12, 'Medio', 25),

(25, 13, 'Medio', 30),

(26, 13, 'Duro', 20),

(27, 14, 'Blando', 18),

(28, 14, 'Medio', 30),

(29, 15, 'Blando', 20),

(30, 15, 'Medio', 30),

(31, 16, 'Medio', 35),

(32, 16, 'Duro', 15),

(33, 17, 'Blando', 20),

(34, 17, 'Medio', 30),

(35, 18, 'Medio', 40),

(36, 18, 'Duro', 10),

(37, 19, 'Blando', 25),

(38, 19, 'Medio', 25),

(39, 20, 'Medio', 30),

(40, 20, 'Duro', 20),

(41, 21, 'Blando', 22),

(42, 21, 'Medio', 28),

(43, 22, 'Medio', 33),

(44, 22, 'Duro', 17),

(45, 23, 'Blando', 20),

(46, 23, 'Medio', 30),

(47, 24, 'Medio', 40),

(48, 24, 'Duro', 10),

(49, 25, 'Blando', 18),

(50, 25, 'Medio', 32),

(51, 26, 'Medio', 30),

(52, 26, 'Duro', 20),

(53, 27, 'Blando', 25),

(54, 27, 'Medio', 25),

(55, 28, 'Medio', 35),

(56, 28, 'Duro', 15),

(57, 29, 'Blando', 20),

(58, 29, 'Medio', 30),

(59, 30, 'Medio', 40),

(60, 30, 'Duro', 10),

(61, 31, 'Blando', 25),

(62, 31, 'Medio', 25),

(63, 32, 'Medio', 30),

(64, 32, 'Duro', 20),

(65, 33, 'Blando', 18),

(66, 33, 'Medio', 32),

(67, 34, 'Blando', 20),

(68, 34, 'Medio', 30),

(69, 35, 'Medio', 35),

(70, 35, 'Duro', 15),

(71, 36, 'Blando', 20),

(72, 36, 'Medio', 30),

(73, 37, 'Medio', 40),

(74, 37, 'Duro', 10),

(75, 38, 'Blando', 25),

(76, 38, 'Medio', 25),

(77, 39, 'Medio', 30),

(78, 39, 'Duro', 20),

(79, 40, 'Blando', 18),

(80, 40, 'Medio', 32),

(81, 41, 'Blando', 20),

(82, 41, 'Medio', 30),

(83, 42, 'Medio', 35),

(84, 42, 'Duro', 15),

(85, 43, 'Blando', 22),

(86, 43, 'Medio', 28),

(87, 44, 'Medio', 40),

(88, 44, 'Duro', 10),

(89, 45, 'Blando', 18),

(90, 45, 'Medio', 32),

(91, 46, 'Medio', 30),

(92, 46, 'Duro', 20),

(93, 47, 'Blando', 25),

(94, 47, 'Medio', 25),

(95, 48, 'Medio', 35),

(96, 48, 'Duro', 15),

(97, 49, 'Blando', 20),

(98, 49, 'Medio', 30),

(99, 50, 'Medio', 40),

(100, 50, 'Duro', 10),

(101, 51, 'Blando', 25),

(102, 51, 'Medio', 25),

(103, 52, 'Medio', 30),

(104, 52, 'Duro', 20),

(105, 53, 'Blando', 18),

(106, 53, 'Medio', 32),

(107, 54, 'Blando', 20),

(108, 54, 'Medio', 30),

(109, 55, 'Medio', 35),

(110, 55, 'Duro', 15),

(111, 56, 'Blando', 20),

(112, 56, 'Medio', 30),

(113, 57, 'Medio', 40),

(114, 57, 'Duro', 10),

(115, 58, 'Blando', 25),

(116, 58, 'Medio', 25),

(117, 59, 'Medio', 30),

(118, 59, 'Duro', 20),

(119, 60, 'Blando', 18),

(120, 60, 'Medio', 32);

-- pena

INSERT INTO Penalizacion (id, idGranPremio, idPiloto, tipo\_penalizacion, tiempo\_penalizacion, descripcion, vuelta) VALUES

(1, 1, 5, 'Drive Through', NULL, 'Exceso de velocidad en el pit lane', 15),

(2, 1, 3, 'Time Penalty', 5, 'Colisión con otro piloto', 30),

(3, 2, 2, 'Stop & Go', NULL, 'Salida en falso', 10),

(4, 2, 8, 'Time Penalty', 10, 'Adelantamiento fuera de pista', 20),

(5, 3, 6, 'Drive Through', NULL, 'Incumplimiento de bandera azul', 25),

(6, 3, 1, 'Time Penalty', 3, 'Exceso de velocidad en el pit lane', 40);

# Funciones y Procedimientos Almacenados

Procedimiento: Calcular y actualizar la Clasificacion de pilotos:

CREATE PROCEDURE ActualizarClasificacionPilotos(IN temporada\_id INT)

BEGIN

-- Vacia la clasificacion sin actualizar

DELETE FROM ClasificacionPiloto WHERE idTemporada = temporada\_id;

-- Insertar nuevas clasificaciones

INSERT INTO ClasificacionPiloto (idTemporada, idPiloto, posicion, puntos)

SELECT

temporada\_id,

idPiloto,

RANK() OVER (ORDER BY SUM(puntos) DESC) AS posicion,

SUM(puntos) AS total\_puntos

FROM (

SELECT

rp.idPiloto,

rp.idGranPremio,

CASE

WHEN rp.posicion = 1 THEN 25

WHEN rp.posicion = 2 THEN 18

WHEN rp.posicion = 3 THEN 15

WHEN rp.posicion = 4 THEN 12

WHEN rp.posicion = 5 THEN 10

WHEN rp.posicion = 6 THEN 8

WHEN rp.posicion = 7 THEN 6

WHEN rp.posicion = 8 THEN 4

WHEN rp.posicion = 9 THEN 2

WHEN rp.posicion = 10 THEN 1

ELSE 0

END +

CASE WHEN rp.obtuvo\_punto\_vuelta\_rapida = TRUE THEN 1 ELSE 0 END AS puntos

FROM ResultadoGranPremio rp

INNER JOIN GranPremio gp ON rp.idGranPremio = gp.id

WHERE gp.idTemporada = temporada\_id

) AS puntosPiloto

GROUP BY idPiloto;

END //

DELIMITER ;

CALL ActualizarClasificacionPilotos(1);

Procedimiento: Calcular y actualizar la Clasificacion de Equipos:

CREATE PROCEDURE ActualizarClasificacionEquipos(IN temporada\_id INT)

BEGIN

-- Elimina clasificación anterior

DELETE FROM ClasificacionEquipo WHERE idTemporada = temporada\_id;

-- Inserta nueva clasificación

INSERT INTO ClasificacionEquipo (idTemporada, idEquipo, posicion, puntos)

SELECT

temporada\_id,

idEquipo,

RANK() OVER (ORDER BY SUM(puntos) DESC) AS posicion,

SUM(puntos) AS total\_puntos

FROM (

SELECT

r.idEquipo,

CASE

WHEN r.posicion = 1 THEN 25

WHEN r.posicion = 2 THEN 18

WHEN r.posicion = 3 THEN 15

WHEN r.posicion = 4 THEN 12

WHEN r.posicion = 5 THEN 10

WHEN r.posicion = 6 THEN 8

WHEN r.posicion = 7 THEN 6

WHEN r.posicion = 8 THEN 4

WHEN r.posicion = 9 THEN 2

WHEN r.posicion = 10 THEN 1

ELSE 0

END +

CASE WHEN r.obtuvo\_punto\_vuelta\_rapida = TRUE THEN 1 ELSE 0 END AS puntos

FROM ResultadoGranPremio r

INNER JOIN GranPremio g ON r.idGranPremio = g.id

WHERE g.idTemporada = temporada\_id

) AS puntosEquipo

GROUP BY idEquipo;

END //

DELIMITER ;

CALL ActualizarClasificacionEquipos(1);

Funcion: Calcular edad de Un piloto:

DELIMITER //

CREATE FUNCTION ObtenerEdadPiloto(pilotoId INT) RETURNS INT

DETERMINISTIC

BEGIN

DECLARE edad INT;

DECLARE fechaNacimiento DATE;

SELECT fecha\_nacimiento INTO fechaNacimiento

FROM Piloto

WHERE id = pilotoId;

IF fechaNacimiento IS NOT NULL THEN

SET edad = TIMESTAMPDIFF(YEAR, fechaNacimiento, CURDATE());

ELSE

SET edad = NULL;

END IF;

RETURN edad;

END //

DELIMITER ;

SELECT ObtenerEdadPiloto(3) AS edad;

Funcion: ObtenerVictoriasPiloto:

DELIMITER //

CREATE FUNCTION ObtenerVictoriasPiloto(pilotoId INT, temporadaId INT)

RETURNS INT

DETERMINISTIC

BEGIN

DECLARE victorias INT;

SELECT COUNT(\*) INTO victorias

FROM ResultadoGranPremio r

INNER JOIN GranPremio g ON r.idGranPremio = g.id

WHERE r.idPiloto = pilotoId

AND g.idTemporada = temporadaId

AND r.posicion = 1;

RETURN victorias;

END //

DELIMITER ;

select ObtenerVictoriasPiloto(2,1) as victorias;

DELIMITER //

# Triggers

Triger: Evita que un piloto se registre dos veces en un equipo en la misma temporada:

DELIMITER //

CREATE TRIGGER evitar\_piloto\_en\_dos\_equipos

BEFORE INSERT ON PilotoTemporadaEquipo

FOR EACH ROW

BEGIN

DECLARE existe INT;

SELECT COUNT(\*) INTO existe

FROM PilotoTemporadaEquipo

WHERE idTemporada = NEW.idTemporada

AND idPiloto = NEW.idPiloto;

IF existe > 0 THEN

SIGNAL SQLSTATE '45000'

SET MESSAGE\_TEXT = 'Error: El piloto ya está registrado en un equipo para esta temporada.';

END IF;

END //

DELIMITER ;

Triger: Pone una fecha cuando la fecha de nacimiento es nula:

DELIMITER //

CREATE TRIGGER actualizar\_fecha\_nacimiento\_nula

BEFORE INSERT ON Piloto

FOR EACH ROW

BEGIN

IF NEW.fecha\_nacimiento IS NULL THEN

SET NEW.fecha\_nacimiento = '1900-01-01'; -- Fecha por defecto

END IF;

END //

DELIMITER ;

-Cada vez que se inserta ClasificacionPiloto guarda en un tabla Log los datos insertados

CREATE TABLE LogClasificacion (

id INT AUTO\_INCREMENT PRIMARY KEY,

mensaje TEXT,

fecha TIMESTAMP DEFAULT CURRENT\_TIMESTAMP

);

DELIMITER //

CREATE TRIGGER registrar\_log\_clasificacion\_piloto

AFTER INSERT ON ClasificacionPiloto

FOR EACH ROW

BEGIN

DECLARE piloto\_nombre VARCHAR(255);

DECLARE temporada\_anio INT;

SELECT p.nombre INTO piloto\_nombre FROM Piloto p WHERE p.id = NEW.idPiloto;

SELECT t.año INTO temporada\_anio FROM Temporada t WHERE t.id = NEW.idTemporada;

INSERT INTO LogClasificacion (mensaje)

VALUES (

CONCAT('Piloto "', piloto\_nombre, '" se clasificó en la posición ', NEW.posicion,

' con ', NEW.puntos, ' puntos en la temporada ', temporada\_anio, '.')

);

END //

DELIMITER ;

# Consultas SQL

-- Rankinfg de pilotos por temportadas:

SELECT

p.nombre AS piloto,

cp.puntos,

cp.posicion

FROM ClasificacionPiloto cp

JOIN Piloto p ON cp.idPiloto = p.id

WHERE cp.idTemporada = 1

ORDER BY cp.puntos DESC;

-- Participacion de pilotos/coche por temporada:

SELECT

t.año,

p.nombre AS piloto,

e.nombre AS equipo,

c.nombre AS coche,

pte.numeroCoche

FROM PilotoTemporadaEquipo pte

JOIN Piloto p ON p.id = pte.idPiloto

JOIN Equipo e ON e.id = pte.idEquipo

JOIN CocheTemporada c ON c.id = pte.idCocheTemporada

JOIN Temporada t ON t.id = pte.idTemporada

WHERE pte.idTemporada = 1;

-- Piloto con mas vueltasRapidas en una temporada:

SELECT

p.nombre AS piloto,

COUNT(\*) AS vueltas\_rapidas

FROM ResultadoGranPremio r

JOIN GranPremio g ON g.id = r.idGranPremio

JOIN Piloto p ON p.id = r.idPiloto

WHERE r.obtuvo\_punto\_vuelta\_rapida = TRUE

AND g.idTemporada = 1

GROUP BY p.id

ORDER BY vueltas\_rapidas DESC;

-- Duracion tipo de neunatico por piloto

SELECT

p.nombre AS piloto,

n.compuesto,

SUM(n.vueltas\_usado) AS total\_vueltas

FROM NeumaticoUsado n

JOIN ResultadoGranPremio r ON n.idResultadoGP = r.id

JOIN GranPremio g ON r.idGranPremio = g.id

JOIN Piloto p ON r.idPiloto = p.id

WHERE g.idTemporada = 1

GROUP BY p.id, n.compuesto

ORDER BY p.nombre, total\_vueltas DESC;

-- Pilotos sancionados:

SELECT

p.nombre AS piloto,

pe.tipo\_penalizacion,

pe.descripcion,

pe.vuelta,

g.nombreGranPremio

FROM Penalizacion pe

JOIN Piloto p ON p.id = pe.idPiloto

JOIN GranPremio g ON g.id = pe.idGranPremio

WHERE g.idTemporada = 1

ORDER BY p.nombre, g.fecha;

-- Mejores coche por peso/potencia

SELECT

e.nombre AS equipo,

c.nombre AS coche,

c.potencia,

c.peso\_coche

FROM CocheTemporada c

JOIN Equipo e ON e.id = c.idEquipo

WHERE c.idTemporada = 1

ORDER BY c.potencia DESC;

-- Promedio de paradas por piloto

SELECT

p.nombre AS piloto,

ROUND(AVG(r.paradas\_box), 2) AS promedio\_paradas

FROM ResultadoGranPremio r

JOIN Piloto p ON p.id = r.idPiloto

JOIN GranPremio g ON g.id = r.idGranPremio

WHERE g.idTemporada = 1

GROUP BY p.id

ORDER BY promedio\_paradas ASC;

-- piloto con mas polePosition

SELECT

p.nombre AS piloto,

COUNT(\*) AS pole\_positions

FROM Qualy q

JOIN Piloto p ON p.id = q.idPiloto

JOIN GranPremio g ON g.id = q.idGranPremio

WHERE q.posicion = 1 AND g.idTemporada = 1

GROUP BY p.id

ORDER BY pole\_positions DESC;

-- Numero de carreras por pais

SELECT

c.pais,

COUNT(\*) AS grandes\_premios

FROM GranPremio gp

JOIN Circuito c ON c.id = gp.idCircuito

WHERE gp.idTemporada = 1

GROUP BY c.pais

ORDER BY grandes\_premios DESC;

-- promedio de posicion por pilotos

SELECT

p.nombre AS piloto,

ROUND(AVG(r.posicion), 2) AS promedio\_posicion

FROM ResultadoGranPremio r

JOIN Piloto p ON p.id = r.idPiloto

JOIN GranPremio g ON g.id = r.idGranPremio

WHERE g.idTemporada = 1

GROUP BY p.id

ORDER BY promedio\_posicion ASC;

# Casos de Prueba y Simulación

INSERCION

-- caso de prueba: Insertar Piloto:

INSERT INTO Piloto (nombre, abreviatura, nacionalidad, altura, peso, fecha\_nacimiento, numero\_licencia, foto\_url)

VALUES ('Max Verstappen', 'VER', 'Países Bajos', 1.81, 72, '1997-09-30', 'MV33F1', 'https://example.com/verstappen.jpg');

-- Insertar Equipo

INSERT INTO Equipo (nombre, pais, color, sede\_principal, director\_equipo, año\_fundacion, logo\_url)

VALUES ('Red Bull Racing', 'Austria', 'Azul y Rojo', 'Milton Keynes', 'Christian Horner', 2005, 'https://example.com/redbull.png');

-- Insertar Temporada

INSERT INTO Equipo (nombre, pais, color, sede\_principal, director\_equipo, año\_fundacion, logo\_url)

VALUES ('Red Bull Racing', 'Austria', 'Azul y Rojo', 'Milton Keynes', 'Christian Horner', 2005, 'https://example.com/redbull.png');

-- Insertar circuito

INSERT INTO Circuito (nombre, localizacion, pais, kilometros, numero\_vueltas, record\_vuelta, tipo\_circuito, longitud\_pit\_lane)

VALUES ('Circuit de Spa-Francorchamps', 'Stavelot', 'Bélgica', 7.004, 44, '1:46.286', 'Permanente', 0.390);

-- insertar Coche-Temporada

INSERT INTO CocheTemporada (nombre, motor, idEquipo, idTemporada, especificaciones\_tecnicas, peso\_coche, potencia)

VALUES ('RB19', 'Honda', 1, 1, 'Chasis aerodinámico optimizado', 798.0, 1050.0);

-- Asociar piloto/equipo

INSERT INTO PilotoTemporadaEquipo (idTemporada, idPiloto, idEquipo, idCocheTemporada, numeroCoche)

VALUES (1, 1, 1, 1, 33);

Verificación Claves Foráneas: Deben dar consultas vacias

-- Verificar claves foraneas

-- Verificar que todos los pilotos en Temporada existen

SELECT t.id, t.año, p.id AS piloto\_id

FROM Temporada t

LEFT JOIN Piloto p ON t.idPilotoGANADOR = p.id

WHERE p.id IS NULL;

-- Verificar que todos los equipos en Temporada existen

SELECT t.id, t.año, e.id AS equipo\_id

FROM Temporada t

LEFT JOIN Equipo e ON t.idEquipoGANADOR = e.id

WHERE e.id IS NULL;

-- Pilotos asignados a temporadas que no existen

SELECT pte.id, pte.idTemporada

FROM PilotoTemporadaEquipo pte

LEFT JOIN Temporada t ON pte.idTemporada = t.id

WHERE t.id IS NULL;

-- Pilotos asignados a equipos que no existen

SELECT pte.id, pte.idEquipo

FROM PilotoTemporadaEquipo pte

LEFT JOIN Equipo e ON pte.idEquipo = e.id

WHERE e.id IS NULL;

-- Pilotos asignados a coches que no existen

SELECT pte.id, pte.idCocheTemporada

FROM PilotoTemporadaEquipo pte

LEFT JOIN CocheTemporada ct ON pte.idCocheTemporada = ct.id

WHERE ct.id IS NULL;

# Resultados y Verificación

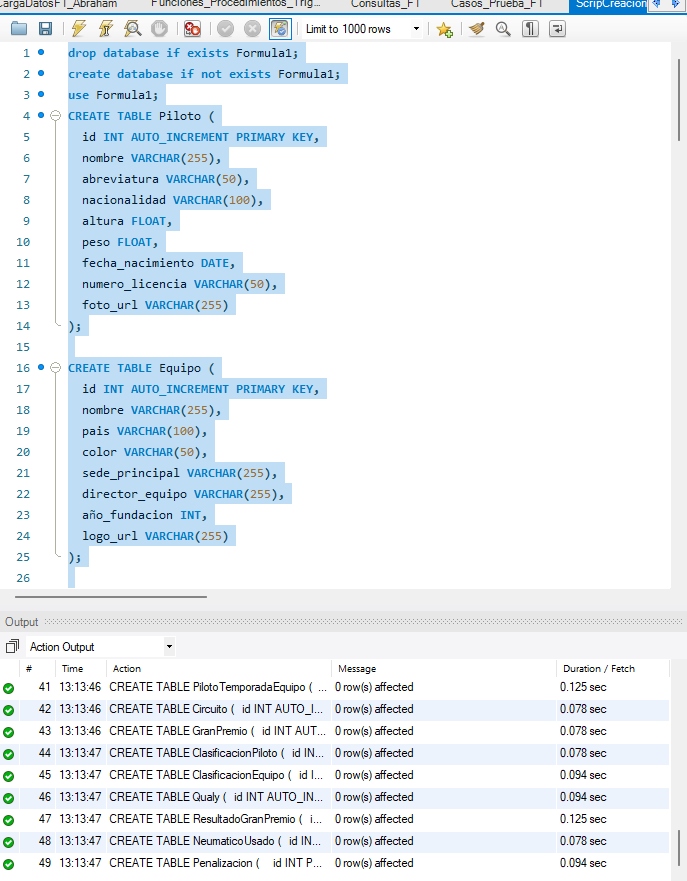
Las pruebas de Inserccion/modificacion se efectúan correctamente.

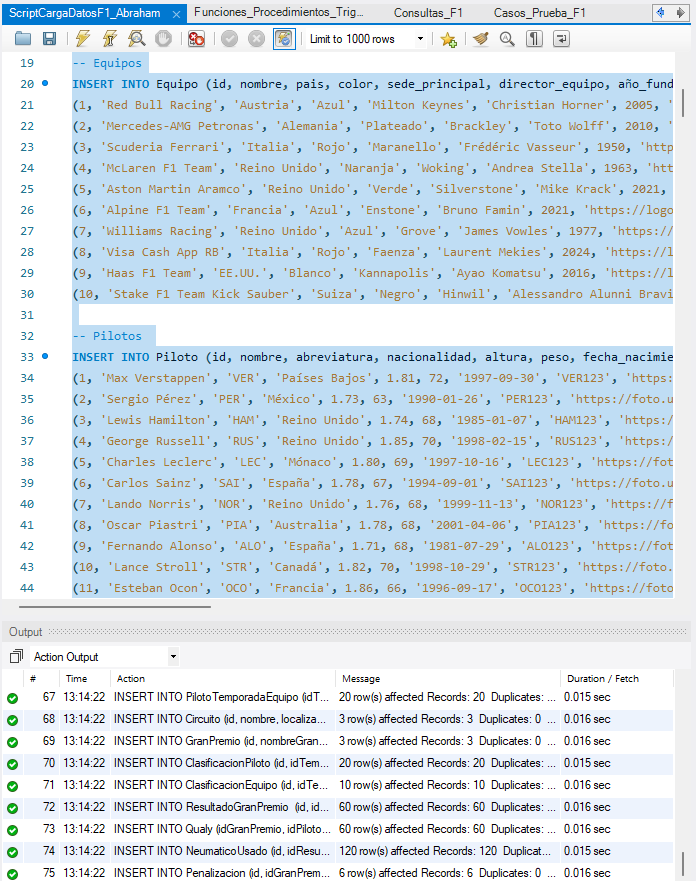
| **Verificación** | **Resultado Esperado** | **Resultado Obtenido** |
| --- | --- | --- |
| Select | Resultado OK | ✅ OK |
| Update | Resultado OK | ✅ OK |
| Insert | Resultado OK | ✅ OK |

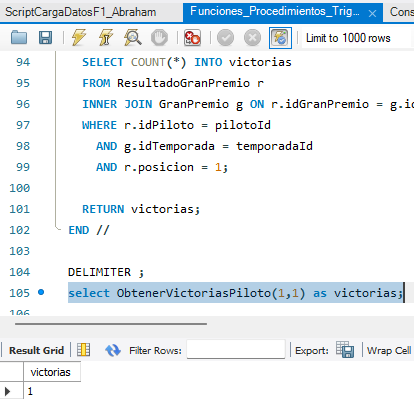
Las pruebas de detección de inconsistencias:

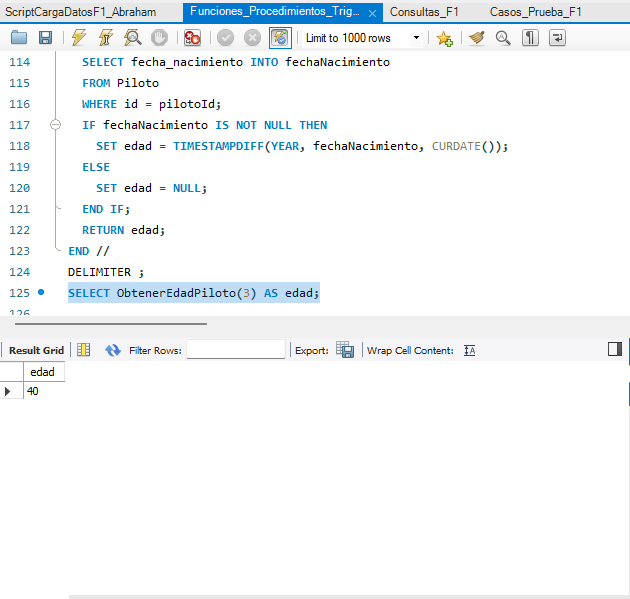
| **Verificación** | **Resultado Esperado** | **Resultado Obtenido** |
| --- | --- | --- |
| Temporadas con piloto ganador inexistente | Ningún resultado | ✅ Ninguno |
| Temporadas con equipo ganador inexistente | Ningún resultado | ✅ Ninguno |
| Pilotos asignados a temporadas que no existen | Ningún resultado | ✅ Ninguno |
| Pilotos asignados a equipos que no existen | Ningún resultado | ✅ Ninguno |
| Pilotos asignados a coches que no existen | Ningún resultado | ✅ Ninguno |

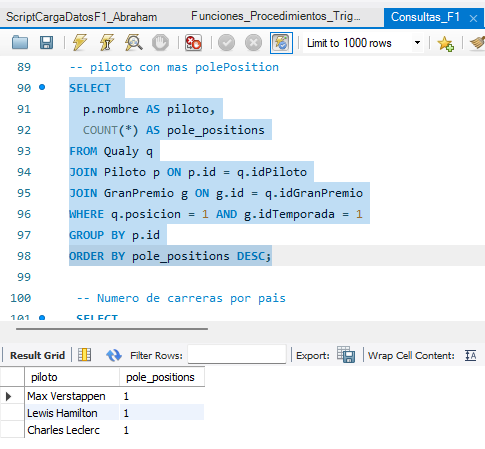
# Capturas de Pantalla (opcional)

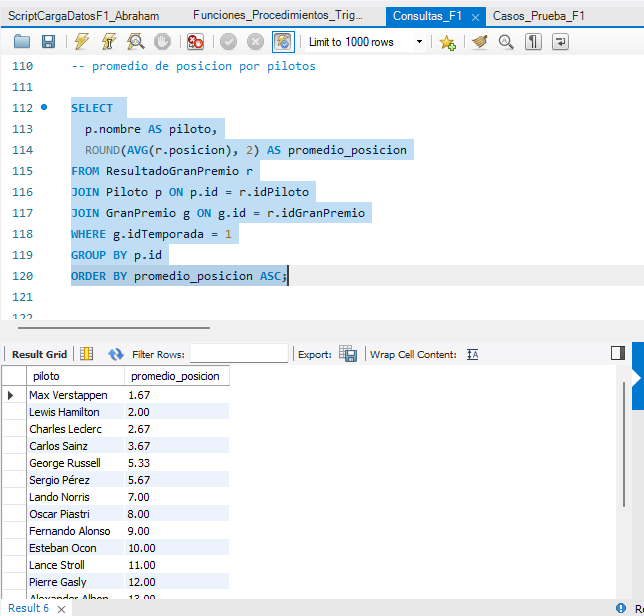


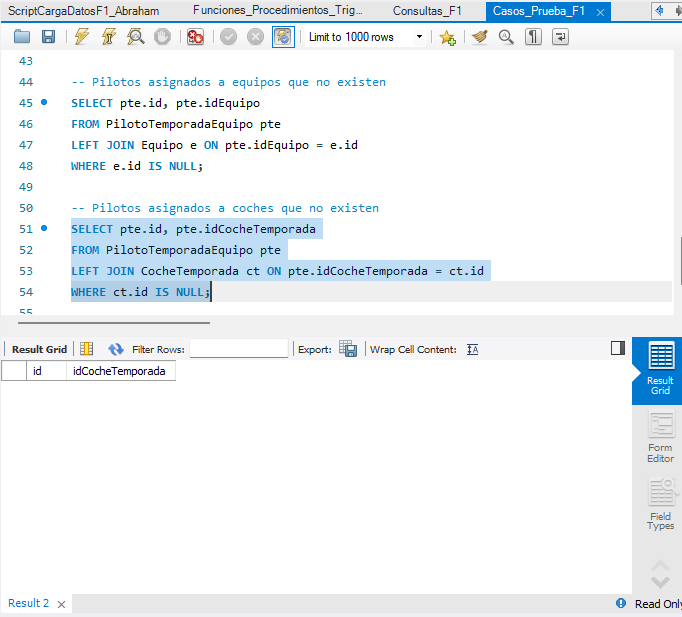


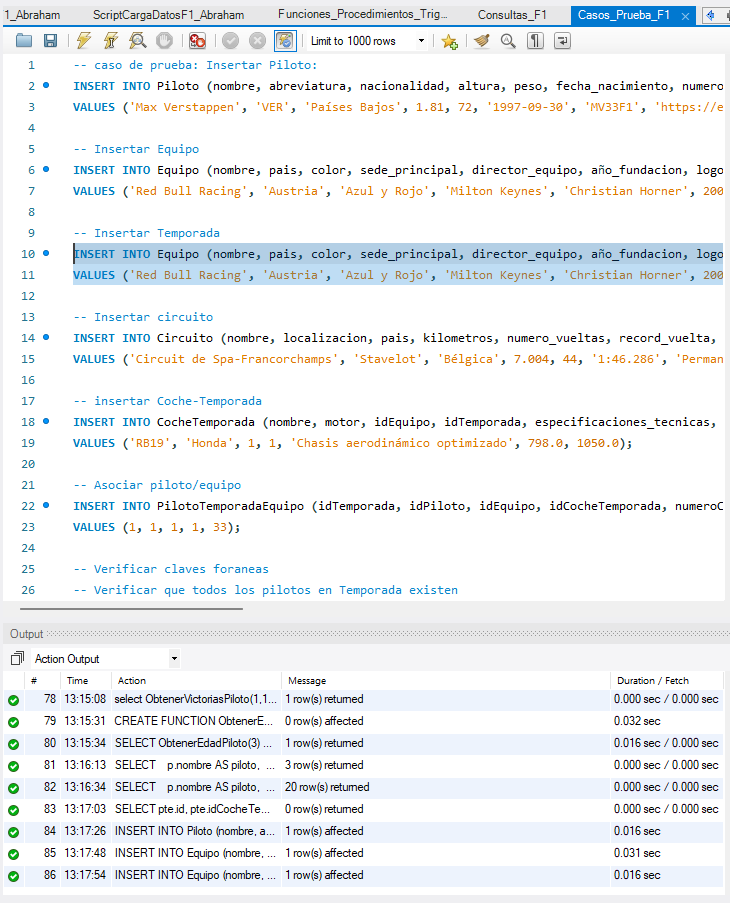


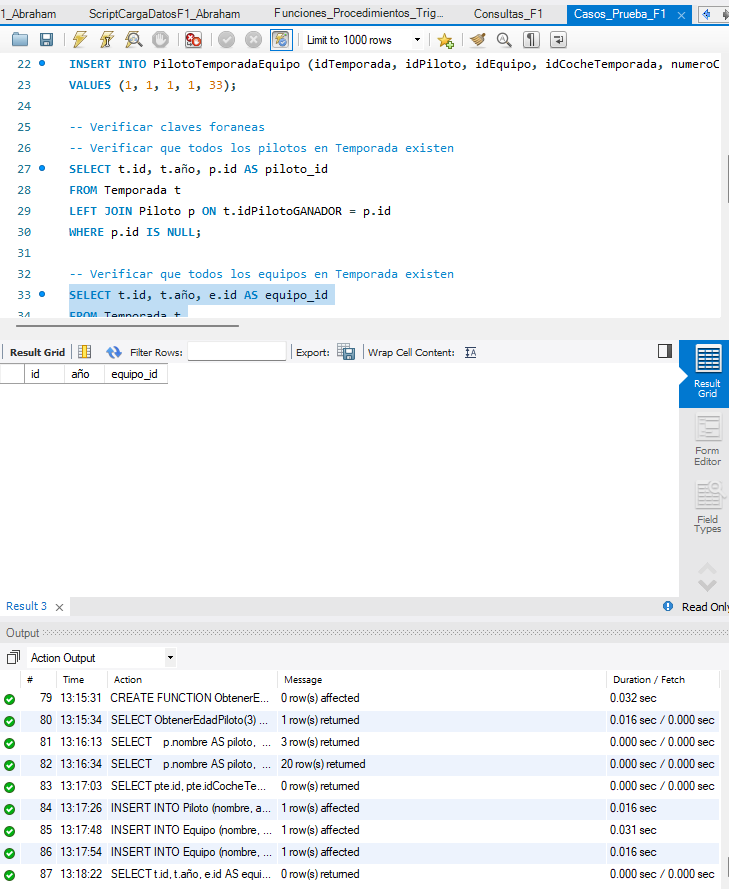












# Conclusiones y Mejoras Futuras

Conclusiones:

El desarrollo e implementación de esta base de datos para la gestión de un campeonato de Fórmula 1 ha permitido estructurar de forma eficiente y relacional toda la información clave del sistema, incluyendo pilotos, equipos, coches, temporadas, circuitos, resultados y penalizaciones.

Mejoras futuras:

Historial y estadisticas avanzadas: Implementar tablas para records historicos, comparativas entre coches y pilotos, estadisticas de motores,etc..

Añadir automatizaciones para detecar penalizaciones al insertar datos a la tablas, o detectar anomalias.

# Enlace al Repositorio en GitHub

https://github.com/Abraham-HUB-777/Tarefa-Final-BD-1-DAM