



FÓRMULA O - FBD

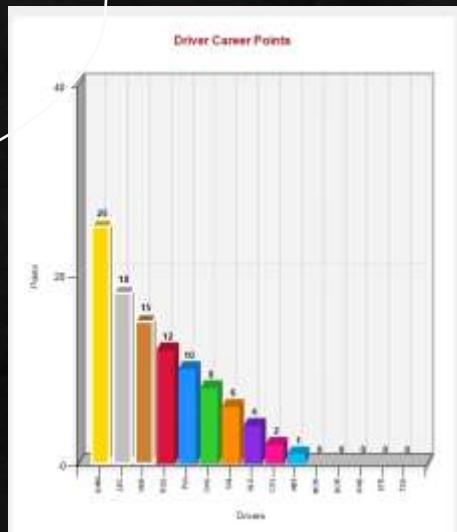
P3G9

Inês Batista, 124877
Maria Quinteiro, 124996



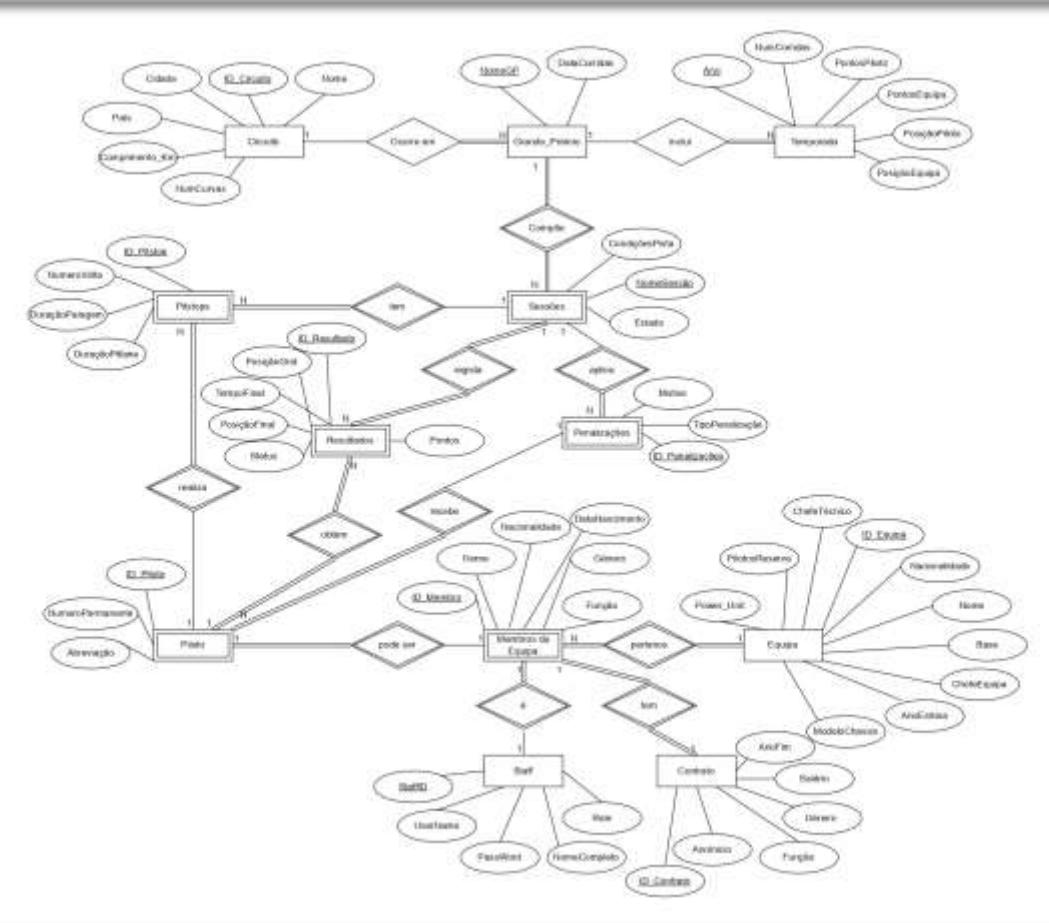
INTRODUÇÃO

- Desenvolvimento de um sistema de base de dados para a gestão de temporadas da Fórmula1;
- Entidades: Circuito, Temporada, Equipa, Membros da Equipa, Staff, Pilotos, Contrato, Grande_Premio, Sessões, Resultados, Penalizações e Pitstops;
- Interação por meios de formulários e apresentação de estatísticas;



A screenshot of a software application window titled "Add New Staff Member". The form contains fields for Username, Password, Full Name, Role, Nationality, Birth Date, Team, and Contract Information (Start Year, End Year, Punctuations, Salary, Gender). Buttons for "Add Staff" and "Cancel" are at the bottom right. The background shows a list of staff members with their names partially visible.

DIAG. ENTIDADE-RELACIONAMETO



ESQUEMA RELACIONAL

A normalização do modelo foi realizada até à BCFN.

Os atributos que dependiam simultaneamente de uma entidade e da Temporada foram removidos e colocados em tabelas associativas, garantindo que todos os determinantes são super-chaves.



STORE PROCEDURES

Add Result - Qualification

Driver:	#57 VET - Sebastian Vettel (Dusty)
Final Position:	1
Final Time:	00:01:00.0000000
Status:	Finished
Points:	0

Add Result **Cancel**

```
CREATE OR ALTER PROCEDURE sp_InsertResult
    @PosicaoOrId INT,
    @TempoFinal TIME = NULL,
    @PosicaoFinal INT,
    @Status NVARCHAR(50),
    @Pontos DECIMAL(10,2),
    @NomeSessao NVARCHAR(100),
    @NomeGP NVARCHAR(100),
    @ID_Piloto INT
AS
BEGIN
    INSERT INTO Resultados (PosicaoOrId, TempoFinal, PosicaoFinal, Status, Pontos, NomeSessao, NomeGP, ID_Piloto)
    VALUES (@PosicaoOrId, @TempoFinal, @PosicaoFinal, @Status, @Pontos, @NomeSessao, @NomeGP, @ID_Piloto)
END
```

```
CREATE PROCEDURE sp_GetDriverStandingBySeason
    @Season INT
AS
BEGIN
    SET NOCOUNT ON;
    SELECT
        ROW_NUMBER() OVER (ORDER BY ISNULL(SUM(r.Pontos), 0) DESC,
                           COUNT(CASE WHEN r.PosicaoFinal = 1 THEN 1 END) DESC) AS Position,
        ISNULL(w.Name, 'Unknown Driver') AS Driver,
        ISNULL(w.Team, 'No Team') AS Team,
        ISNULL(SUM(r.Pontos), 0) AS TotalPoints,
        COUNT(CASE WHEN r.PosicaoFinal = 1 THEN 1 END) AS Wins,
        COUNT(CASE WHEN r.PosicaoFinal <= 3 THEN 1 END) AS Podiums
    FROM Pilotos p
    INNER JOIN Resultados r ON p.ID_Piloto = r.ID_Piloto
    INNER JOIN Temporadas t ON r.ID_Temporada = t.ID_Temporada
    INNER JOIN Sessoes s ON t.ID_Sessao = s.ID_Sessao
    INNER JOIN GP g ON s.ID_GP = g.ID_GP
    INNER JOIN Pilotos w ON r.ID_Piloto = w.ID_Piloto
    WHERE r.ID_Temporada = #Season AND r.Status = 'Race' AND g.NomeGP = '#NomeGP'
    GROUP BY p.ID_Piloto, w.Name, t.Team
    HAVING ISNULL(SUM(r.Pontos), 0) > 0
    ORDER BY TotalPoints DESC, Wins DESC;
```

Standings - 2025 Season

2025 Season Standings

Position	Driver	Team	Points
1	Lewis Hamilton	Toto Wolff Racing	100.00
2	Charles Leclerc	Scuderia Ferrari	98.00
3	Max Verstappen	Red Bull Racing	95.00
4	George Russell	Mercedes	12.00
5	Oscar Piastri	McLaren	10.00
6	Valtteri Bottas	Alpine	8.00
7	Callum Ilott	Williams	6.00
8	Fernando Alonso	Alfa Romeo	4.00
9	François Clérical	AlphaTauri	3.00
10	Kimi Räikkönen	McLaren	1.00

TRIGGERS

Garante a consistência na base de dados sempre que ocorre uma modificação.

```
2 IF OBJECT_ID('trg_UpdateSeasonRaceCount', 'TR') IS NOT NULL
3     DROP TRIGGER trg_UpdateSeasonRaceCount;
4 GO
5
6 CREATE TRIGGER trg_UpdateSeasonRaceCount
7 ON Grande_Premio
8 AFTER INSERT, DELETE
9 AS
10 BEGIN
11     SET NOCOUNT ON;
12
13     UPDATE t
14     SET t.NumCorridas = (
15         SELECT COUNT(*)
16         FROM Grande_Premio gp
17         WHERE gp.Ano_Temporada = t.Ano
18     )
19     FROM Temporada t
20     WHERE t.Ano IN (
21         SELECT DISTINCT Ano_Temporada FROM inserted
22         UNION
23         SELECT DISTINCT Ano_Temporada FROM deleted
24     );
25 END;
26 GO
```

INDEXES

- Utilizados para melhorar significativamente a performance das consultas frequentes.
- Reduz o tempo de resposta e diminui leituras desnecessárias

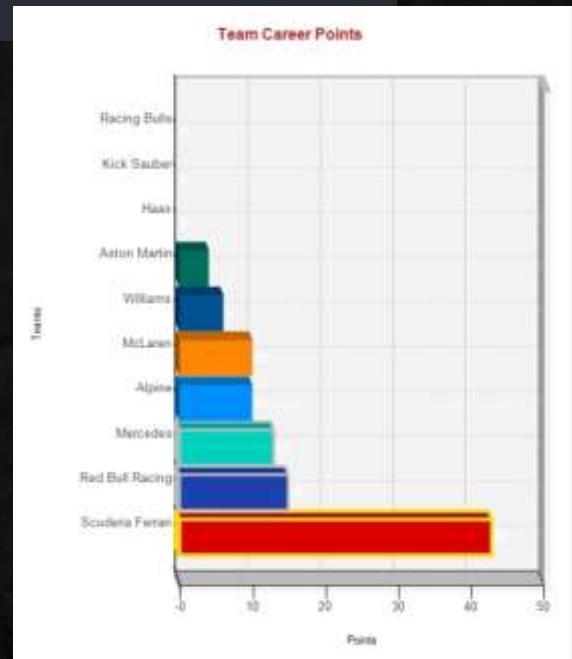
```
12 ✓ IF NOT EXISTS (SELECT * FROM sys.indexes WHERE name = 'IX_Resultados_Session' AND object_id = OBJECT_ID('Resultados'))  
13     CREATE NONCLUSTERED INDEX IX_Resultados_Session ON Resultados(NomeSessão) INCLUDE (Pontos, PosiçãoFinal);  
14     GO
```



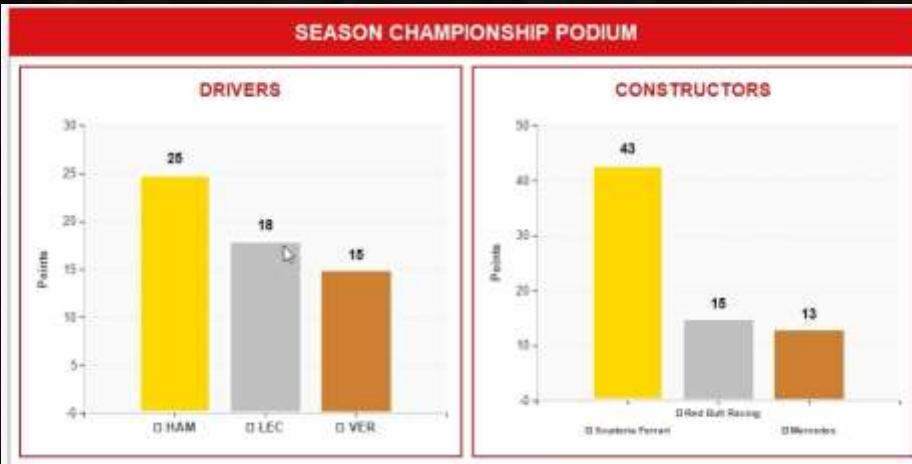
A tabela Resultados é a base das estatísticas.
A coluna NomeSessão é usada como filtro frequente.
O uso de INDICE evita key lookups.

ESTATÍSTICAS

```
1 CREATE OR ALTER PROCEDURE sp_GetTeamCareerPoints  
2 AS  
3 BEGIN  
4     SELECT * FROM vw_Team_CareerPoints ORDER BY TotalPoints DESC, TeamName ASC;  
5 END
```



ESTATÍSTICAS



```
1 IF OBJECT_ID('vw_Saison_DriverPodium', 'V') IS NOT NULL  
2     DROP VIEW vw_Saison_DriverPodium;  
3 GO  
4 CREATE VIEW vw_Saison_DriverPodium AS  
5 SELECT  
6     gp.Ano_Temporada AS Year,  
7     p.Abreviação,  
8     n.Nome AS DriverName,  
9     ISNULL(SUM(r.Pontos), 0) AS TotalPoints  
10    FROM Grande_Premio gp  
11    INNER JOIN Resultados r ON r.NomeGP = gp.NomeGP  
12    INNER JOIN Piloto p ON r.ID_Piloto = p.ID_Piloto  
13    INNER JOIN Membros_da_Equipe m ON p.ID_Número = m.ID_Membro  
14    WHERE r.NomeSessão = 'Race'  
15    GROUP BY gp.Ano_Temporada, p.ID_Piloto, p.Abreviação, n.Nome, p.NumeroPernamente;
```

```
1 IF OBJECT_ID('vw_Saison_TeamPodium', 'V') IS NOT NULL  
2     DROP VIEW vw_Saison_TeamPodium;  
3 GO  
4 CREATE VIEW vw_Saison_TeamPodium AS  
5 SELECT  
6     gp.Ano_Temporada AS Year,  
7     e.Nome AS TeamName,  
8     ISNULL(SUM(r.Pontos), 0) AS TotalPoints  
9     FROM Grande_Premio gp  
10    INNER JOIN Resultados r ON r.NomeGP = gp.NomeGP  
11    INNER JOIN Piloto p ON r.ID_Piloto = p.ID_Piloto  
12    INNER JOIN Equipa e ON p.ID_Equipa = e.ID_Equipa  
13    WHERE r.NomeSessão = 'Race'  
14    GROUP BY gp.Ano_Temporada, e.ID_Equipa, e.Nome;
```



DEMO