

Atividade Prática Supervisionada - APS

Integrantes:

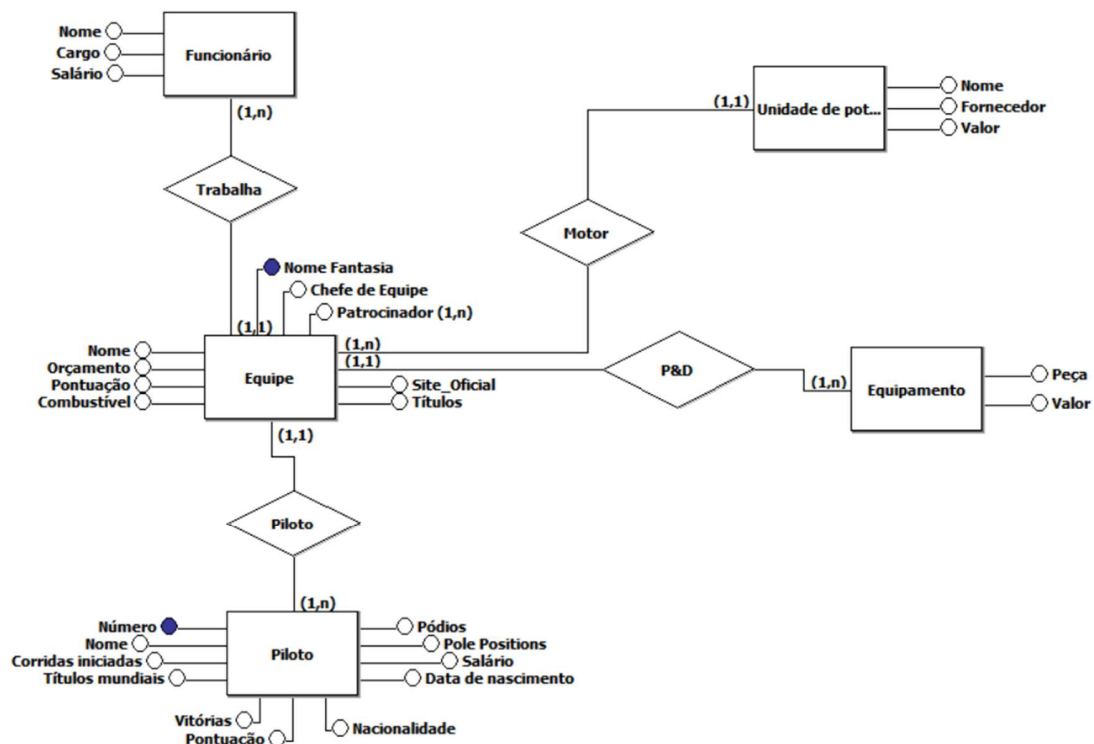
- Gabriel Roberto Pendiuk
- Luan Escudeiro dos Santos Almeida
- Raphael Grechoniak Maria

QUESTÃO 1) Escolha uma aplicação que existam dados correlacionados e a descreva.

A aplicação escolhida foi a de equipes de Fórmula 1 e suas relações dentro do campeonato, onde temos vários dados e várias relações a tratar dos mesmos. Foram escolhidas apenas algumas destas para se fazer essa atividade, com o intuito de tornar a aplicação o mais intuitiva possível perante o objetivo do trabalho. Com isso escolhemos modelar a relação que equipes tem com patrocinadores, pilotos, funcionários, equipamentos e motores.

QUESTÃO 2) Elabore um diagrama Entidade-Relacionamento para esta aplicação.

Pensando da aplicação escolhida pelo grupo, foi elaborado o seguinte diagrama Entidade-Relacionamento utilizando-se o software BrModelo:



QUESTÃO 3) Faça o mapeamento para o modelo relacional, indicando as restrições de integridade.

Para fazer a população da tabela foi mesclado diversos dados reais com dados fictícios criados no mockaroo.com, sendo toda a tabela de piloto com dados reais dos pilotos atuais, motores com todos os motores atuais utilizados pelas equipes, peças com dados reais também utilizadas pelas equipes, a parte de funcionários e patrocinadores, foi feita através de uma mescla de dados reais e dados fictícios, sendo feitos assim para que fossem visíveis as melhoras sobre os usos de índices, tendo aproximadamente 1000 dados nas tabelas funcionários e mais 1000 na tabela patrocinadores.

QUESTÃO 4) Gere o Script de criação das tabelas no PostgreSQL com as restrições.

O software utilizado anteriormente proporciona a geração do código para criação das tabelas com base no MER criado:

-----Criando a base de dados-----

-- Database: F1

-- DROP DATABASE "F1";

--CREATE DATABASE "F1"

-- WITH

-- OWNER = postgres

-- ENCODING = 'UTF8'

-- LC_COLLATE = 'Portuguese_Brazil.1252'

-- LC_CTYPE = 'Portuguese_Brazil.1252'

-- TABLESPACE = pg_default

-- CONNECTION LIMIT = -1;

-----Criando Tabelas-----

CREATE TABLE Unidade_de_potência (

Nome varchar(60),

```
Fornecedor varchar(20) PRIMARY KEY,  
Valor int  
);
```

```
CREATE TABLE Equipe (  
Nome_Fantasia varchar(20) PRIMARY KEY,  
Combustível varchar(20),  
Pontuação smallint,  
Orçamento int,  
Nome varchar(45),  
Títulos smallint,  
Site_Oficial varchar(60),  
FK_Motor varchar(20)  
);
```

```
CREATE TABLE Piloto (  
Nome varchar(60),  
Número smallint PRIMARY KEY,  
Corridas_iniciadas smallint,  
Títulos_mundiais smallint,  
Vitórias smallint,  
Pódios smallserial,  
Pole_Positions smallserial,  
Pontuação smallint,  
Nacionalidade varchar(20),  
Data_de_nascimento date,  
Salário int,
```

```
FK_Equipe varchar(20)
);
```

```
CREATE TABLE Equipamento (
Peça varchar(20),
Valor int,
FK_Equipe varchar(20)
);
```

```
CREATE TABLE Funcionario (
ID serial PRIMARY KEY,
Nome varchar(60),
Idade smallint,
Salario int,
Cargo varchar(30),
FK_Equipe varchar(20)
);
```

```
CREATE TABLE Patrocinador (
ID serial PRIMARY KEY,
Patrocinador varchar(60),
FK_Equipe varchar(20)
);
```

```
ALTER TABLE Equipe ADD FOREIGN KEY(FK_Motor) REFERENCES Unidade_de_potência
(Fornecedor);
```

ALTER TABLE Piloto ADD FOREIGN KEY(FK_Equipe) REFERENCES Equipe (Nome_Fantasia);

ALTER TABLE Equipamento ADD FOREIGN KEY(FK_Equipe) REFERENCES Equipe (Nome_Fantasia);

ALTER TABLE Funcionario ADD FOREIGN KEY(FK_Equipe) REFERENCES Equipe (Nome_Fantasia);

ALTER TABLE Patrocinador ADD FOREIGN KEY(FK_Equipe) REFERENCES Equipe (Nome_Fantasia);

Exemplo tabela equipe:

	nome_fantasia [PK] character varying (20)	combustível character varying (20)	pontuação smallint	orçamento integer	nome character varying (45)
1	Mercedes	Petronas Primax	303	123000000	Mercedes-AMG Petronas Formula One Team
2	RBR	Esso/Mobil Synergy	291	123000000	Red Bull Racing Honda
3	McLaren	BP Ultimate	163	123000000	McLaren F1 Team
4	Aston Martin	Petronas Primax	48	123000000	Aston Martin Cognizant Formula One Team
5	Alpine	BP Ultimate	77	123000000	Alpine F1 Team
6	AlphaTauri	Mobil 1	68	123000000	Scuderia AlphaTauri Honda
7	Ferrari	Shell V-Power	163	123000000	Scuderia Mission Winnow Ferrari
8	Alfa Romeo	Shell V-Power	3	123000000	Alfa Romeo Racing Orlen
9	Haas	Shell V-Power	0	123000000	Uralkali Haas F1 Team
10	Williams	Petronas Primax	10	123000000	Williams Racing

Exemplo tabela piloto:

	nome character varying (60)	número [PK] smallint	corridas_iniciadas smallint	títulos_mundiais smallint	vitórias smallint	pódios smallint	pole_positions smallint	pontuação smallint	nacionalidade character varying (20)	data_de_nascimento date	salário integer	fk_equipe character varying (20)
1	Lewis Hamilton	44	277	7	99	173	101	195	Inglaterra	1985-01-07	27000000	Mercedes
2	Valtteri Bottas	77	168	0	9	62	17	108	Finlândia	1989-08-28	9100000	Mercedes
3	Max Verstappen	33	131	0	15	50	9	187	Bélgica	1997-09-30	22700000	RBR
4	Sergio Pérez	11	204	0	2	12	0	104	México	1990-01-26	7300000	RBR
5	Daniel Ricciardo	3	200	0	7	31	3	50	Austrália	1989-07-01	13600000	McLaren
6	Lando Norris	4	50	0	0	4	0	113	Inglaterra	1999-11-13	4500000	McLaren
7	Lance Stroll	18	90	0	0	3	1	30	Canadá	1998-10-29	9100000	Aston Martin
8	Sebastian Vettel	5	270	4	53	122	57	18	Alemanha	1987-07-03	13600000	Aston Martin
9	Fernando Alonso	14	326	2	32	97	22	38	Espanha	1981-07-29	18700000	Alpine
10	Esteban Ocon	31	79	0	1	2	0	39	Normândia	1996-09-17	2300000	Alpine
11	Yuki Tsunoda	22	12	0	0	0	0	18	Japão	2000-05-11	44000000	AlphaTauri
12	Pierre Gasly	10	76	0	1	3	0	50	França	1996-02-07	4500000	AlphaTauri
13	Charles Leclerc	16	71	0	2	13	9	80	Monaco	1997-10-16	10900000	Ferrari
14	Carlos Sainz Jr.	55	131	0	0	4	0	83	Espanha	1994-09-01	9100000	Ferrari
15	Kimi Räikkönen	7	344	1	21	103	18	2	Finlândia	1979-10-17	9100000	Alfa Romeo
16	Antonio Giovinazzi	99	19	0	0	0	0	1	Itália	1993-12-14	900000	Alfa Romeo
17	Nikita Mazepin	9	12	0	0	0	0	0	Rússia	1999-03-02	90000000	Haas
18	Mick Schumacher	47	12	0	0	0	0	0	Suíça	1999-03-22	90000000	Haas
19	George Russell	63	50	0	0	0	0	4	Inglaterra	1998-02-15	90000000	Williams
20	Nicholas Latifi	6	29	0	0	0	0	6	Canadá	1995-06-29	90000000	Williams

Exemplo tabela equipamento:

	peça character varying (20)	valor integer	fk_equipe character varying (20)
1	Caixa de câmbio	456000	Mercedes
2	ECU	169000	Mercedes
3	Asa dianteira	146000	Mercedes
4	Asa traseira	94000	Mercedes
5	Tanque	135000	Mercedes
6	Volante	67000	Mercedes
7	Halo	25000	Mercedes
8	Jogo de pneus	3000	Mercedes
9	Caixa de câmbio	458000	RBR
10	ECU	165000	RBR
11	Asa dianteira	149000	RBR
12	Asa traseira	93000	RBR
13	Tanque	135000	RBR
14	Volante	68000	RBR
15	Halo	24000	RBR
16	Jogo de pneus	3000	RBR
17	Caixa de câmbio	455000	McLaren
18	ECU	164000	McLaren
19	Asa dianteira	145000	McLaren
20	Asa traseira	90000	McLaren
21	Tanque	132000	McLaren

Exemplo tabela funcionário:

	id [PK] integer	nome character varying (60)	idade smallint	salário integer	cargo character varying (30)	fk_equipe character varying (20)
1	16	Charles Leclerc	27	10900000	Piloto	Ferrari
2	18	Lance Stroll	36	9100000	Piloto	Aston Martin
3	22	Yuki Tsunoda	29	44000000	Piloto	AlphaTauri
4	31	Esteban Ocon	30	2300000	Piloto	Alpine
5	33	Max Verstappen	44	22700000	Piloto	RBR
6	44	Lewis Hamilton	21	27000000	Piloto	Mercedes
7	47	Mick Schumacher	28	90000000	Piloto	Haas
8	55	Carlos Sainz Jr.	31	9100000	Piloto	Ferrari
9	63	George Russell	20	90000000	Piloto	Williams
10	77	Valtteri Bottas	31	9100000	Piloto	Mercedes
11	99	Antonio Giovinazzi	20	900000	Piloto	Alfa Romeo
12	9000	Gabriel	32	10000	Engenheiro	Mercedes
13	1035	AUGUSTINHO	17	1	Testador	Mercedes
14	20000	Iuan	23	1000	Engenheiro	RBR
15	3	Daniel Ricciardo	40	13600000	Piloto	McLaren
16	1	Tarik Riley	34	32310	Engenheiro	Mercedes
17	2	Ahmed McLaughlin	31	44780	Engenheiro	Mercedes
18	8	Anthony Christian	40	49172	Engenheiro	Mercedes
19	12	Brady Hart	32	39355	Engenheiro	Mercedes
20	13	Fitzgerald Koch	27	28844	Engenheiro	Mercedes
21	15	Aaulla Nauven	25	49718	Enoenheiro	Mercedes

Exemplo tabela patrocinador:

	id [PK] integer	patrocinador character varying (60)	fk_equipe character varying (20)
1	1	Petronas	Mercedes
2	2	INEOS	Mercedes
3	3	UBS	Mercedes
4	4	Epson	Mercedes
5	5	Bose	Mercedes
6	6	Tommy Hilfiger	Mercedes
7	7	IWC	Mercedes
8	8	Hewlett Packard	Mercedes
9	9	The Ritz-Carlton	Mercedes
10	10	Monster Energy	Mercedes
11	11	Pure Storage	Mercedes
12	12	CrowdStrike	Mercedes
13	13	TIBCO	Mercedes
14	14	AMD	Mercedes
15	15	Puma	Mercedes
16	16	Police	Mercedes
17	17	OZ Racing	Mercedes
18	18	Endless	Mercedes
19	19	Axalta	Mercedes
20	20	Belstaff	Mercedes
21	21	Marriott Bonvov	Mercedes

Exemplo tabela unidade_de_potencia:

	nome character varying (60)	fornecedor [PK] character varying (20)	valor integer
1	Mercedes-AMG F1 M12 E Performance 1.6 V6 Turbo Híbrido	Mercedes-AMG	19755000
2	RA621H 1.6 V6 Turbo híbrido	Honda	18385000
3	Renault E-Tech 20 1.6 V6 Turbo híbrido	Renault	19132000
4	Ferrari 065 1.6 V6 Turbo híbrido	Ferrari	19507000

QUESTÃO 6) Crie índices adequados as consultas, indicando as consultas que foram beneficiadas, mostrando também o resultado do explain.

a) index para cargo de funcionário:

create index idxCargo on funcionario(Cargo)

Antes do índice funcionario

1	<code>explain</code>
2	<code>analyse</code>
3	<code>select id, fk_equipe, nome, cargo from funcionario</code>
4	<code>where cargo='Diretor'</code>

Data Output	Explain	Messages	Notifications
QUERY PLAN			
text			
1	Seq Scan on funcionario (cost=0.00..37.90 rows=23 width=35) (actual time=0.077..0.080 rows=23 loops=1)		
2	[...] Filter: ((cargo)::text = 'Diretor'::text)		
3	[...] Rows Removed by Filter: 1009		
4	Planning Time: 0.044 ms		
5	Execution Time: 0.089 ms		

Pós-Índice de cargo:

1	create index idxCargo on funcionario(Cargo)
2	
3	explain
4	analyse
5	select id,fk_equipe,nome,cargo from funcionario
6	where cargo='Diretor'

Data Output	Explain	Messages	Notifications
<div> <div>QUERY PLAN</div> <div>text</div> <div> <div>1</div> <div>Index Scan using idxcargo on funcionario (cost=0.15..28.24 rows=23 width=35) (actual time=0.020..0.023 rows=23 loo...</div> </div> <div> <div>2</div> <div>[...] Index Cond: ((cargo)::text = 'Diretor'::text)</div> </div> <div> <div>3</div> <div>Planning Time: 0.499 ms</div> </div> <div> <div>4</div> <div>Execution Time: 0.034 ms</div> </div> </div>			

Como demonstrado, o tempo de pesquisa é reduzido pela metade;

Outra pesquisa sem índice cargo:

1	explain
2	analyse
3	select id,F.fk_equipe,F.nome,cargo,P.salário,P.vitórias from funcionario F,piloto P
4	where cargo='Piloto'

Data Output	Explain	Messages	Notifications
<div> <div>QUERY PLAN</div> <div>text</div> <div> <div>1</div> <div>Nested Loop (cost=0.00..53.10 rows=260 width=41) (actual time=0.010..0.160 rows=400 loops=1)</div> </div> <div> <div>2</div> <div>[...]-> Seq Scan on funcionario f (cost=0.00..37.90 rows=1 width=35) (actual time=0.007..0.073 rows=20 loops=1)</div> </div> <div> <div>3</div> <div>[...] Filter: ((cargo)::text = 'Piloto'::text)</div> </div> <div> <div>4</div> <div>[...] Rows Removed by Filter: 1012</div> </div> <div> <div>5</div> <div>[...]-> Seq Scan on piloto p (cost=0.00..12.60 rows=260 width=6) (actual time=0.000..0.001 rows=20 loops=20)</div> </div> <div> <div>6</div> <div>Planning Time: 0.363 ms</div> </div> <div> <div>7</div> <div>Execution Time: 0.180 ms</div> </div> </div>			

Com índice cargo:

f1/postgres@trabalhof1	
Query Editor Query History	
<pre> 1 explain 2 analyse 3 select id,F.fk_equipe,F.nome,cargo,P.salário,P.vitórias from funcionario F,piloto P 4 where cargo='Piloto'</pre>	
Data Output Explain Messages Notifications	
QUERY PLAN	
text	
1	Nested Loop (cost=0.15..23.37 rows=260 width=41) (actual time=0.035..0.129 rows=400 loops=1)
2	[...] -> Index Scan using idxcargo on funcionario f (cost=0.15..8.17 rows=1 width=35) (actual time=0.024..0.030 rows=2...
3	[...] Index Cond: ((cargo)::text = 'Piloto'::text)
4	[...] -> Seq Scan on piloto p (cost=0.00..12.60 rows=260 width=6) (actual time=0.001..0.001 rows=20 loops=20)
5	Planning Time: 1.092 ms
6	Execution Time: 0.157 ms

Neste caso há uma redução de 23ms no tempo de execução,

b)Index para patrocinadores na tabela patrocinadores:

create index idxPatrocinador on patrocinador(patrocinador)

sem idxPatrocinador:

f1/postgres@trabalhof1	
Query Editor Query History	
<pre> 1 explain analyse 2 select fk_equipe,patrocinador from patrocinador 3 where patrocinador='Sparco'</pre>	
Data Output Explain Messages Notifications	
QUERY PLAN	
text	
1	Seq Scan on patrocinador (cost=0.00..21.50 rows=1 width=16) (actual time=0.020..0.082 rows=2 loops=1)
2	[...] Filter: ((patrocinador)::text = 'Sparco'::text)
3	[...] Rows Removed by Filter: 998
4	Planning Time: 0.068 ms
5	Execution Time: 0.096 ms

Com idxPatrocinador:

1	<code>explain analyse</code>
2	<code>select fk_equipe,patrocinador from patrocinador</code>
3	<code>where patrocinador='Sparco'</code>

Data Output	Explain	Messages	Notifications
QUERY PLAN			
text			
1	Index Scan using idxpatrocinador on patrocinador (cost=0.28..8.29 rows=1 width=16) (actual time=0.030..0.032 ro...		
2	[...] Index Cond: ((patrocinador)::text = 'Sparco'::text)		
3	Planning Time: 0.613 ms		
4	Execution Time: 0.043 ms		

O índice de patrocinador faz com que o tempo de pesquisa no mesmo seja reduzido pela metade.

c)Index para recolher informações sobre funcionário:

create index idxnomeF on funcionario(nome,idade,salário,fk_equipe)

Tempo sem index criado:

f1/postgres@trabalhof1	
Query Editor	Query History
1	<code>explain analyse</code>
2	<code>select nome,idade,salário,fk_equipe from funcionario</code>
3	<code>where nome='Jamal Kramer'</code>
4	

Data Output	Explain	Messages	Notifications
QUERY PLAN			
text			
1	Seq Scan on funcionario (cost=0.00..37.90 rows=1 width=27) (actual time=0.039..0.108 rows=1 loops=1)		
2	[...] Filter: ((nome)::text = 'Jamal Kramer'::text)		
3	[...] Rows Removed by Filter: 1031		
4	Planning Time: 0.487 ms		
5	Execution Time: 0.123 ms		

Tempo com index criado:

f1/postgres@trabalhof1

Query Editor Query History

```

1 explain analyse
2 select nome, idade, salário, fk_equipe from funcionario
3 where nome='Jamal Kramer'
4
5 create index idxnomeF on funcionario(nome, idade, salário, fk_equipe)
6

```

Data Output Explain Messages Notifications

QUERY PLAN

text

1	Index Only Scan using idxnomeF on funcionario (cost=0.28..8.29 rows=1 width=27) (actual time=0.023..0.024 rows=1 loops=1)
2	[...] Index Cond: (nome = 'Jamal Kramer'::text)
3	[...] Heap Fetches: 1
4	Planning Time: 0.076 ms
5	Execution Time: 0.037 ms

7) Crie 3 funções que representem consultas corriqueiras, ou realizem alguma tarefa nas tabelas

a) Gasto total em equipamentos:

Create Function GastoTotal(VARCHAR(20))


returns integer as 'select sum(valor) from equipamento

where fk_equipe=\$1;'



LANGUAGE SQL;

1	<code>select</code>	<code>GastoTotal('Aston Martin')</code>
2		

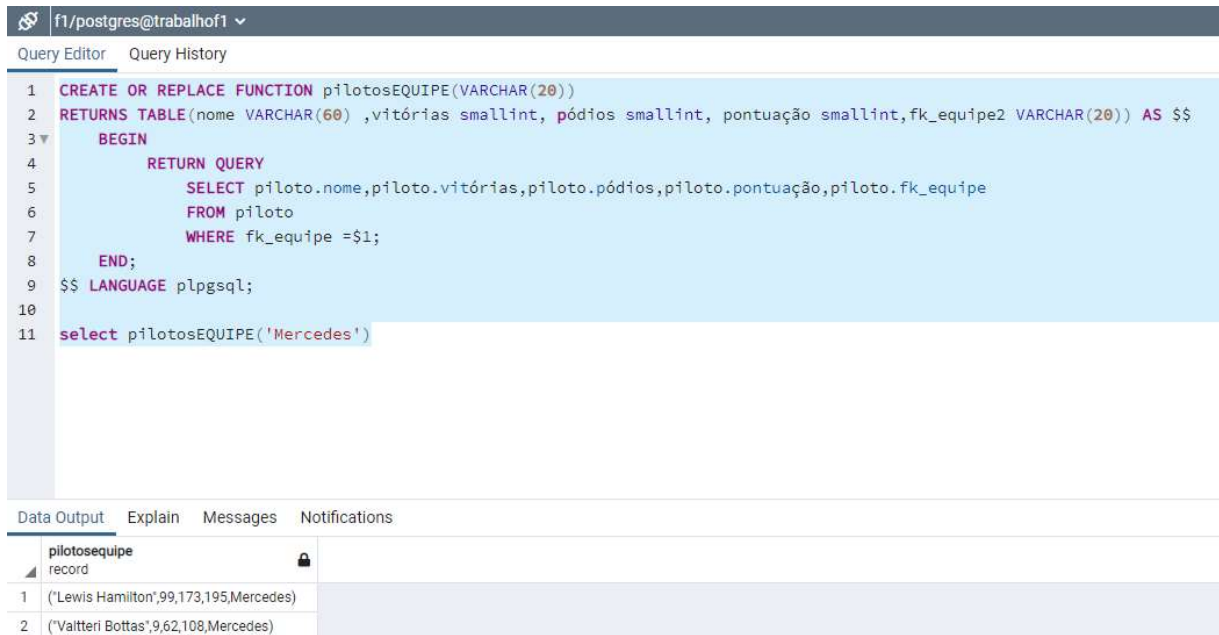
Data Output	Explain	Messages	Notifications
 gastototal integer 			
1	1054700		

	f1/postgres@trabalhof1	▼
Query Editor	Query History	

1	<code>select</code>	<code>GastoTotal('Mercedes')</code>
2		

Data Output	Explain	Messages	Notifications
 gastototal integer 			
1	1095000		

b) Função para retornar pilotos de uma equipe/vitorias/pódios/pontuacao:



The screenshot shows a PostgreSQL query editor interface. The top bar indicates the connection is to 'f1/postgres@trabalhof1'. Below the bar, there are tabs for 'Query Editor' and 'Query History'. The main area contains a SQL script with line numbers 1 through 11. The script defines a function 'pilotosEQUIPE' that returns a table with columns: nome, vitórias, pódios, pontuação, and fk_equipe. It then executes a query to call this function with the parameter 'Mercedes'.

```
1 CREATE OR REPLACE FUNCTION pilotosEQUIPE(VARCHAR(20))
2 RETURNS TABLE(nome VARCHAR(60), vitórias smallint, pódios smallint, pontuação smallint, fk_equipe2 VARCHAR(20)) AS $$
3 BEGIN
4     RETURN QUERY
5     SELECT piloto.nome, piloto.vitórias, piloto.pódios, piloto.pontuação, piloto.fk_equipe
6     FROM piloto
7     WHERE fk_equipe = $1;
8 END;
9 $$ LANGUAGE plpgsql;
10
11 select pilotosEQUIPE('Mercedes')
```

Below the query editor, there are tabs for 'Data Output', 'Explain', 'Messages', and 'Notifications'. The 'Data Output' tab is active, showing the results of the query. The results are displayed in a table with two columns: 'pilotosequipe' and 'record'. There are two rows of data.

	pilotosequipe	record
1	("Lewis Hamilton", 99, 173, 195, Mercedes)	
2	("Valtteri Bottas", 9, 62, 108, Mercedes)	

Exemplo informações RBR:



f1/postgres@trabalhof1 ▾

Query Editor

Query History

```
1 select pilotosEquipe('RBR')
2
```

Data Output

Explain

Messages

Notifications

	pilotosequipe record	
1	("Max Verstappen",15,50,187,RBR)	
2	("Sergio Pérez",2,12,104,RBR)	


c) Função para encontrar patrocinadores de equipe:

Exemplo de patrocinadores da Mercedes:

1	CREATE OR REPLACE FUNCTION patrocinadoresEQUIPE(VARCHAR(20))
2	RETURNS TABLE(patrocinador VARCHAR(60)) AS \$\$
3	BEGIN
4	RETURN QUERY
5	SELECT patrocinador.patrocinador
6	FROM patrocinador
7	WHERE fk_equipe = \$1;
8	END;
9	\$\$ LANGUAGE plpgsql;
10	
11	select patrocinadoresEQUIPE('Mercedes')
12	

Data Output	Explain	Messages	Notifications
patrocinadoresequipe character varying			
1	Petronas		
2	INEOS		
3	UBS		
4	Epson		
5	Bose		
6	Tommy Hilfiger		
7	IWC		
8	Hewlett Packard		
9	The Ritz-Carlton		
10	Monster Energy		
11	Pure Storage		
12	CrowdStrike		
13	TIBCO		
14	AMD		
15	Puma		

Exemplo de patrocinadores da Aston Martin:

 f1/postgres@trabalhof1 ▾

Query Editor

Query History

1

select patrocinadoresEQUIPE('Aston Martin')


2

Data Output

Explain

Messages

Notifications

	patrocinadoresequipe	
	character varying	
1	Cognizant	
2	Crypto.com	
3	NetApp	
4	SentinelOne	
5	Bombardier	
6	Girard-Perregaux	
7	Peroni	
8	BWT	
9	REPLAY	
10	Ravenol	
11	EPOS	
12	JCB	
13	EBB3	
14	UPS Direct	
15	STL	

8) Crie 3 visões comuns e uma tabela de auditoria para uma tabela.

a) Visão de funcionários:

1 CREATE VIEW visaoFuncionario AS

2 SELECT funcionario.nome,funcionario.idade,funcionario.salário,funcionario.cargo

3 FROM funcionario

4

5 select * from visaoFuncionario

6

Data Output

Explain

Messages

Notifications

	<div>nome</div> <div>character varying (60)</div>	<div>idade</div> <div>smallint</div>	<div>salário</div> <div>integer</div>	<div>cargo</div> <div>character varying (30)</div>	
1	Charles Leclerc	27	10900000	Piloto	
2	Lance Stroll	36	9100000	Piloto	
3	Yuki Tsunoda	29	44000000	Piloto	
4	Esteban Ocon	30	2300000	Piloto	
5	Max Verstappen	44	22700000	Piloto	
6	Lewis Hamilton	21	27000000	Piloto	
7	Mick Schumacher	28	90000000	Piloto	
8	Carlos Sainz Jr.	31	9100000	Piloto	
9	George Russell	20	90000000	Piloto	
10	Valtteri Bottas	31	9100000	Piloto	
11	Antonio Giovinazzi	20	900000	Piloto	
12	Daniel Ricciardo	40	13600000	Piloto	
13	Tarik Riley	34	32310	Engenheiro	
14	Ahmed McLaughlin	31	44780	Engenheiro	
15	Anthony Christian	40	49172	Engenheiro	

b) Visão Pontuação Geral:

```
1 CREATE VIEW visaoPontuacao AS
2 SELECT piloto.fk_equipe, sum(pontuação) as pontuacaoEQUIPE
3 FROM piloto
4 group by piloto.fk_equipe
5
6 select * from visaoPontuacao
7
```

Data Output Explain Messages Notifications

	fk_equipe character varying (20)	pontuacaoequipe bigint
1	Alpine	77
2	McLaren	163
3	Haas	0
4	RBR	291
5	Ferrari	163
6	Mercedes	303
7	AlphaTauri	68
8	Aston Martin	48
9	Alfa Romeo	3
10	Williams	10


c) Visão equipamentos:

```
1 CREATE VIEW visaoEquipamentos AS
2 SELECT equipamento.peca, equipamento.valor, equipamento.fk_equipe
3 FROM equipamento
4
5 select * from visaoEquipamentos|
```

Data Output Explain Messages Notifications

	peca character varying (20)	valor integer	fk_equipe character varying (20)	
1	Caixa de câmbio	456000	Mercedes	
2	ECU	169000	Mercedes	
3	Asa dianteira	146000	Mercedes	
4	Asa traseira	94000	Mercedes	
5	Tanque	135000	Mercedes	
6	Volante	67000	Mercedes	
7	Halo	25000	Mercedes	
8	Jogo de pneus	3000	Mercedes	
9	Caixa de câmbio	458000	RBR	
10	ECU	165000	RBR	
11	Asa dianteira	149000	RBR	
12	Asa traseira	93000	RBR	
13	Tanque	135000	RBR	
14	Volante	68000	RBR	
15	Halo	24000	RBR	





d) Tabela de auditoria Funcionário:

 f1/postgres@trabalhof1 ▾

Query Editor Query History

```
1 create table funcionarios_auditoria(  
2     cod serial primary key,  
3     data date,  
4     autor varchar(20),  
5     alteracao varchar(6)  
6 );  
7  
8 CREATE OR REPLACE FUNCTION public.funcionario_audit()  
9     RETURNS trigger  
10    LANGUAGE plpgsql  
11    AS $function$  
12    Begin  
13        insert into funcionarios_auditoria (data, autor, alteracao) values (now(), user, TG_OP);  
14        return new;  
15    end;  
16    $function$;  
17  
18 create trigger tr_gera_log after insert or delete or update  
19 on public.funcionario for each row execute function funcionario_audit();  
20  
21 INSERT INTO public.funcionario(  
22     id, nome, idade, "salário", cargo, fk_equipe)  
23     VALUES (9000, 'Gabriel', 32, 10000, 'Engenheiro', 'Mercedes');  
24  
25 select * from funcionarios_auditoria
```

Data Output Explain Messages Notifications

	 cod [PK] integer	 data date	 autor character varying (20)	 alteracao character varying (6)	
1		1 2021-09-03	postgres	INSERT	
2		2 2021-09-03	postgres	DELETE	
3		6 2021-09-03	postgres	INSERT	
4		8 2021-09-03	postgres	INSERT	

9) Crie 3 triggers que tratem eventos em tabelas com atributos derivados e auditorias acima.

a) Trigger idade de funcionário<18:

```
1 CREATE OR REPLACE FUNCTION public.idade_funcionario()
2 RETURNS trigger
3 LANGUAGE plpgsql
4 AS $function$
5 BEGIN
6     -- Verificar se foi fornecido o nome e o salário do empregado
7     IF NEW.idade<18 THEN
8         RAISE EXCEPTION 'O empregado nao tem idade para trabalhar';
9     END IF;
10    RETURN NEW;
11 END;
12 $function$;
13
14 create trigger validadeidade after
15 insert
16 on
17 public.funcionario for each row execute function idade_funcionario();
18
19 INSERT INTO public.funcionario
20 (id,nome, idade, salário, cargo, fk_equipe)
21 VALUES(1500,'Gabriel', 13, 0, 'Engenheiro', 'Mercedes');
```

Data Output Explain Messages Notifications

ERROR: O empregado nao tem idade para trabalhar
CONTEXT: PL/pgSQL function idade_funcionario() line 5 at RAISE
SQL state: P0001

b) Trigger para número de pilotos não ser maior que 2:

```
1 CREATE OR REPLACE FUNCTION public.numeroPilotos()
2 RETURNS void
3 LANGUAGE plpgsql
4 AS $function$
5 declare
6     QuantidadePilotos Integer;
7 begin
8     QuantidadePilotos := (SELECT COUNT (*) Qtd from piloto where FK_EQUIPE = new.fk_equipe);
9     IF QuantidadePilotos > 2 then
10         RAISE EXCEPTION 'A equipe já possui 2 pilotos';
11     END IF;
12     RETURN NEW;
13 END;
14 $function$;
15
16 create trigger validapilotos after
17 insert
18 on
19     public.piloto for each row execute function numeropilotos();
20
21 INSERT INTO public.piloto
22 (nome, número, corridas_iniciadas, títulos_mundiais, vitórias, pontuação, nacionalidade, data_de_nascimento, salário, fk_equipe)
23 VALUES('GABRIEL', 203, 0, 0, 0, 0, 'TESTE', '1998-11-08', 0, 'Mercedes');
```

Data Output Explain Messages Notifications

ERROR: A equipe já possui 2 pilotos
CONTEXT: PL/pgSQL function numeropilotos() line 7 at RAISE
SQL state: P0001

c) Trigger para não duplicar peça em equipamento:

```
1 CREATE OR REPLACE FUNCTION public.numeropeça()
2 RETURNS trigger
3 LANGUAGE plpgsql
4 AS $function$
5 declare
6     QuantidadePeça Integer;
7 begin
8     QuantidadePeça := (SELECT COUNT(*) Qtd from equipamento
9                        where equipamento.peça= new.peça and equipamento.fk_equipe=new.fk_equipe);
10 IF QuantidadePeça > 0 then
11     RAISE EXCEPTION 'A equipe já comprou esta peça';
12 END IF;
13 RETURN NEW;
14 END;
15 $function$;
16
17 create trigger validacompra after
18 insert
19 on
20     public.equipamento for each row execute function numeropeça();
21
22 INSERT INTO public.equipamento
23 (peça, valor, fk_equipe)
24 VALUES('ECU', 0, 'Mercedes');
25
```

Data Output Explain Messages Notifications

ERROR: A equipe já comprou esta peça
CONTEXT: PL/pgSQL function "numeropeça"() line 8 at RAISE
SQL state: P0001