Grupo 04 – Turma 09

Sprint 5 - entrega

**Treinamento de Aulas Particulares**

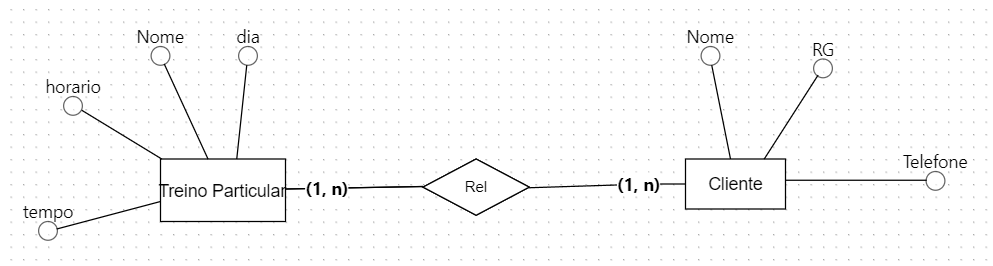
O cliente terá a oportunidade de fazer o seu treino particular na nossa academia.

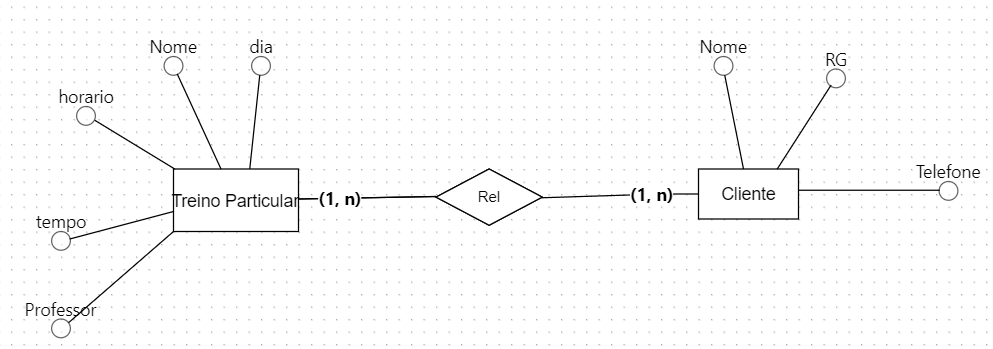
O treino particular precisa ser agendado no dia e horário, e tem a possibilidade de escolher o professor (caso esteja disponível).

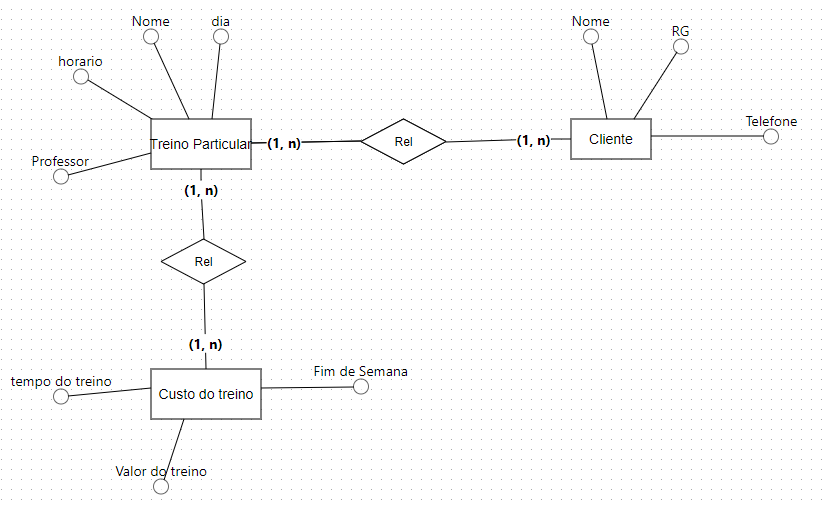
O custo do treino particular depende do tempo de duração do treino, sendo os valores colocados na tabela, os valores padrões para o tempo indicado.

Havendo a escolha ou necessidade de ter o treino particular no fim de semana, logo será acrescentado uma taxa de 20% .

1 - Modelagem conceitual







Necessário otimizar o banco de dados, através da normalização 3FN

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Treino Nome** | **Dia - Data** | **Horário** | **Professor** | **código**  **cliente** | **Cliente** | **Contato - telefone** | **RG** | **Valor do treino** | **Tempo do treino**  **(minutos)** | **Treino - Fim de Semana (Acrescimo de 20%)** |
| Cross | 01/01/24 | 05h00 | Alice | 01 | Paulo | 06454654564 | 353536767 | 100 | 60 | Sim |
| Musculação | 05/02/24 | 07h00 | José | 02 | Tiago | 05184849849 | 8787653 | 100 | 90 | não |
| Fitmess | 11/03/24 | 08h00 | João | 03 | Tadeu | 0215674892 | 535557 | 50 | 50 | não |
| Yoga | 12/04/24 | 10h00 | Allex | 04 | Ronaldo | 02156579448 | 33646 | 150 | 40 | Sim |
| Pilatos | 04/05/24 | 11h00 | Gabi | 05 | Marcelo | 02195555555 | 5353535 | 200 | 120 | não |

3FN

Tabela Cliente

|  |  |  |  |
| --- | --- | --- | --- |
| **ID\_pk**  **código de cliente** | **Cliente** | **Contato** | **RG** |
| 01 | Paulo | 06454654564 | 353536767 |
| 02 | Tiago | 05184849849 | 8787653-SP |
| 03 | Tadeu | 0215674892 | 535557-RJ |
| 04 | Ronaldo | (021)56579448 | 33646 |
| 05 | Marcelo | 02195555555 | 5353535-SP |

Tabela Custo

|  |  |  |
| --- | --- | --- |
| **ID\_pk =>**  **Tempo do treino**  **(minuto)** | **Valor do treino**  **($R Real)** | **Treino - Fim de Semana (Acrescimo de 20%)** |
| 60 | 100 | Sim |
| 90 | 100 | não |
| 50 | 50 | não |
| 40 | 150 | Sim |
| 120 | 200 | não |

Tabela Treino

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **ID\_pk** | **Treino Nome** | **Dia - Data** | **Horário** | **Professor** |
| 01 | Cross | 01/01/24 | 05h00 | Alice |
| 02 | Musculação | 05/02/24 | 07h00 | José |
| 03 | Fitmess | 11/03/24 | 08h00 | João |
| 04 | Yoga | 12/04/24 | 10h00 | Allex |
| 05 | Pilatos | 04/05/24 | 11h00 | Gabi |

**Tabela Criada Exercicio**

Relacionamento da **Tabela Treino** com a **tabela Cliente**

Tabela EXERCICIO

|  |  |  |
| --- | --- | --- |
| **fk\_id\_cliente** | **fk\_id\_treino** | **id\_PK** |
| 01 | 01 | 1 |
| 02 | 02 | 2 |
| 03 | 03 | 3 |
| 04 | 04 | 4 |
| 05 | 05 | 5 |

/\* Criação no SQL \*/

/\* criar a tabela CUSTO com os seus atributos \*/

CREATE TABLE custo (

tempo INT NOT NULL PRIMARY KEY,

valor\_treino FLOAT NOT NULL,

fimSemana BOOLEAN NOT NULL /\* 1=> Fim de semana , 0=> durante a semana \*/

)

;

/\* criar a tabela CLIENTE com os seus atributos \*/

CREATE TABLE cliente (

id SERIAL PRIMARY KEY,

nome VARCHAR(48) NOT NULL,

rg VARCHAR(10) NOT NULL,

telefone VARCHAR (12) NOT NULL

);

/\* criar a tabela Treino com os seus atributos \*/

CREATE TABLE treino (

/\*id SERIAL PRIMARY KEY,\*/

treino VARCHAR(48) NOT NULL,

dia VARCHAR(10) NOT NULL,

horario VARCHAR (12) NOT NULL,

professor VARCHAR (12) NOT NULL

);

/\*Agora vamos inserir valores nas nossas tabelas \*/

/\* Inserir Valores na nossas tabelas TREINO , CLIENTES e CUSTO\*/

**/\*inserir Valores tabela do CLIENTE\*/**

INSERT INTO cliente (nome, rg, telefone) VALUES

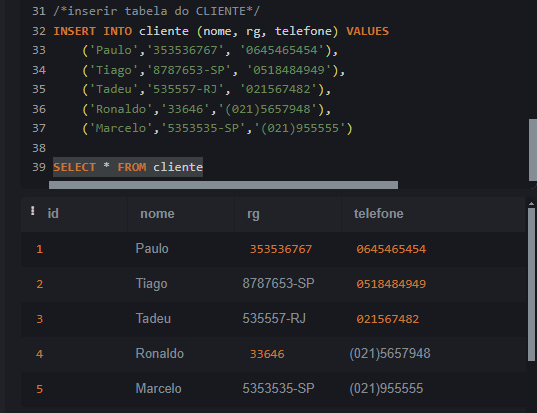
('Paulo','353536767', '0645465454'),

('Tiago','8787653-SP', '0518484949'),

('Tadeu','535557-RJ', '021567482'),

('Ronaldo','33646','(021)5657948'),

('Marcelo','5353535-SP','(021)955555')



**/\* Inserir dados na tabela TREINO \*/**

INSERT INTO treino (treino, dia, horario,professor) VALUES

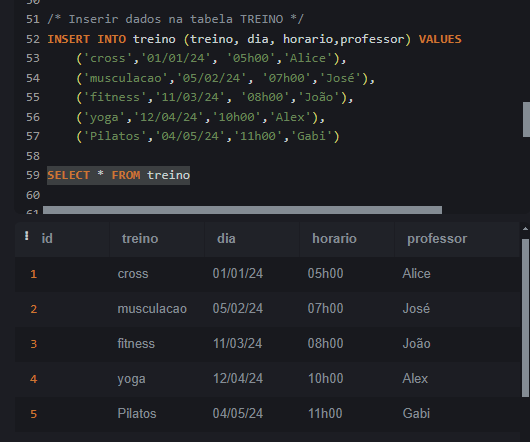
('cross','01/01/24', '05h00','Alice'),

('musculacao','05/02/24', '07h00','José'),

('fitness','11/03/24', '08h00','João'),

('yoga','12/04/24','10h00','Alex'),

('Pilatos','04/05/24','11h00','Gabi')



**/\* Inserir dados na tabela CUSTO\*/**

INSERT INTO custo (tempo,valor\_treino,fimSemana) VALUES

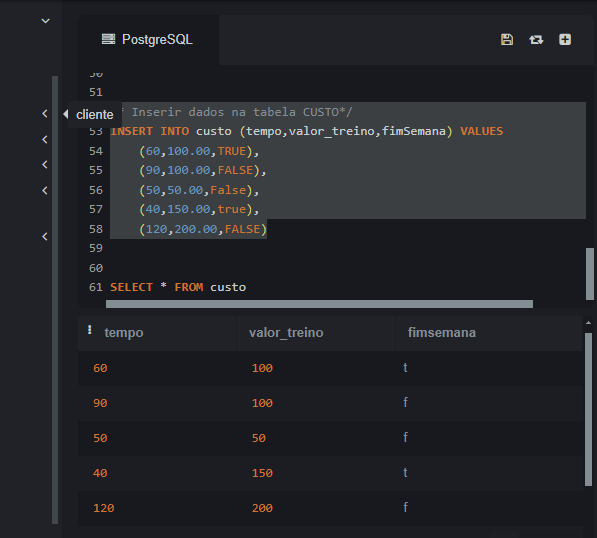
(60,100.00,TRUE),

(90,100.00,FALSE),

(50,50.00,False),

(40,150.00,true),

(120,200.00,FALSE)



/\* Agora como funcionaria a nossa tabela auxiliar \*/

/\* Associar o TREINO com CLIENTE \*/

/\* Criar a tabela EXERCICIO\*/

CREATE TABLE exercicio (

id SERIAL PRIMARY KEY,

fk\_id\_treino INT NOT NULL,

fk\_id\_cliente INT NOT NULL,

CONSTRAINT fk\_treino FOREIGN KEY (fk\_id\_treino) REFERENCES treino(ID),

CONSTRAINT fk\_cliente FOREIGN KEY (fk\_id\_cliente) REFERENCES cliente(ID)

);

/\* Inserindo valores na tabela auxiliar EXERCICIO \*/

INSERT INTO exercicio (fk\_id\_treino,fk\_id\_cliente) VALUES

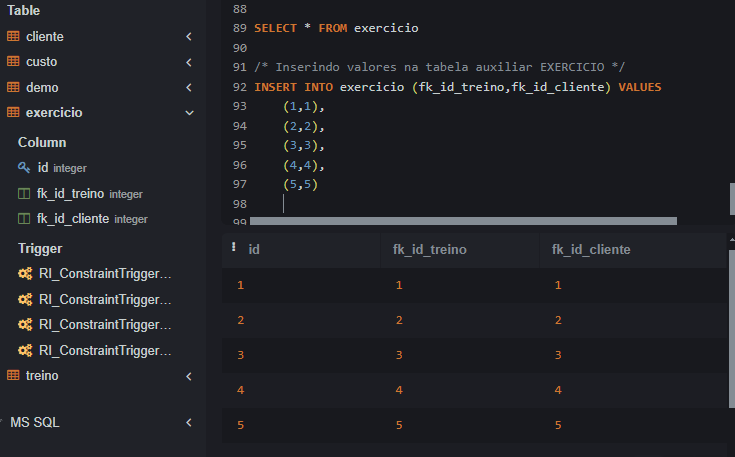
(1,1),

(2,2),

(3,3),

(4,4),

(5,5)



**/\* Fazendo relacionamentos Tabela CLIENTE e tabela TREINO\*/**

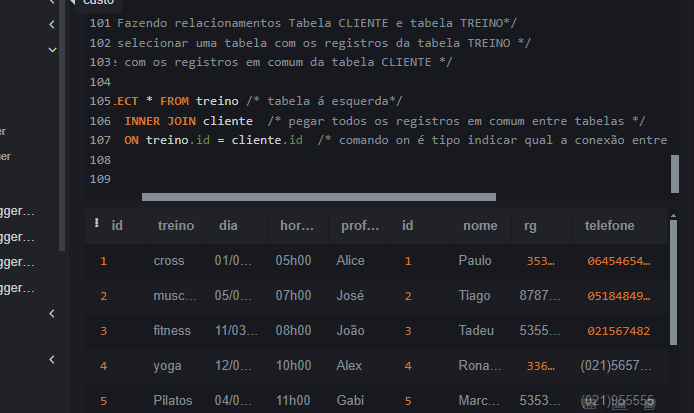
**/\* selecionar uma tabela com os registros da tabela TREINO \*/**

**/\* e com os registros em comum da tabela CLIENTE \*/**

SELECT \* FROM treino /\* tabela á esquerda\*/

inner join cliente /\* pegar todos os registros em comum entre tabelas \*/

on treino.id = cliente.id /\* comando on é tipo indicar qual a conexão entre colunas das tabelas precisam ser iguais \*/



**/\* Agora como funcionaria a nossa tabela auxiliar \*/**

**/\* Associar o TREINO com CUSTO \*/**

**/\* Criar a tabela ORCAMENTO\*/**

CREATE TABLE orcamento (

id SERIAL PRIMARY KEY,

fk\_id\_treino INT NOT NULL,

fk\_id\_custo INT NOT NULL,

CONSTRAINT fk\_treino FOREIGN KEY (fk\_id\_treino) REFERENCES treino(ID),

CONSTRAINT fk\_custo FOREIGN KEY (fk\_id\_custo) REFERENCES custo(tempo)

);

**/\* Inserir valores na tabela ORCAMENTO \*/**

INSERT INTO orcamento (fk\_id\_treino,fk\_id\_custo) VALUES

(1,60),

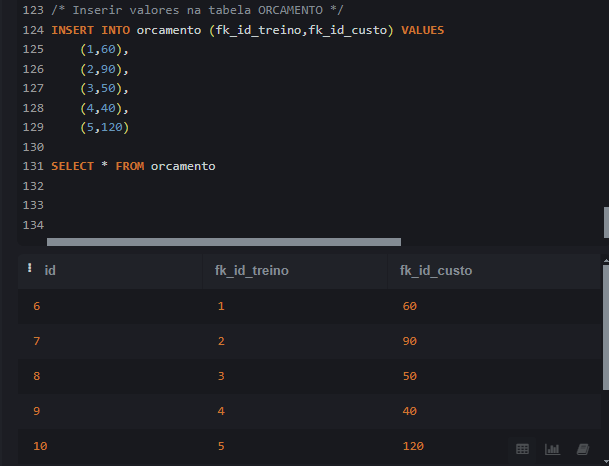
(2,90),

(3,50),

(4,40),

(5,120)

SELECT \* FROM orcamento



**/\*Fazendo relacionamentos Tabela CUSTO e tabela ORCAMENTO \*/**

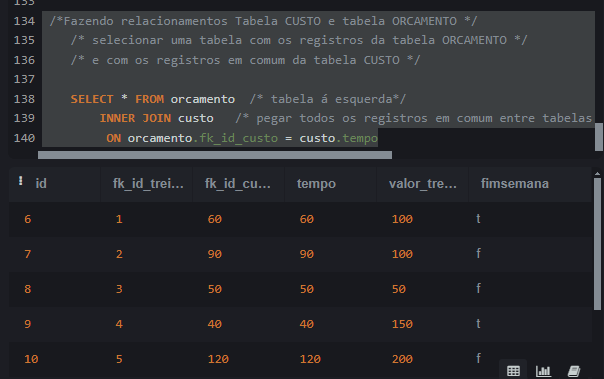
**/\* selecionar uma tabela com os registros da tabela ORCAMENTO \*/**

**/\* e com os registros em comum da tabela CUSTO \*/**

SELECT \* FROM orcamento /\* tabela á esquerda\*/

inner join custo /\* pegar todos os registros em comum entre tabelas \*/

ON orcamento.fk\_id\_custo = custo.tempo



Sendo aula no fim-de-semana, deverá ser aplicado adicional de 20%.

Função para aplicar adicional de 20% na tabela de dados.

**/\* escrever uma função Aplicar\_adicional 20%\*/**

CREATE or REPLACE FUNCTION aplicar\_adicional (preco FLOAT, adicional FLOAT)

RETURNS FLOAT as $$

BEGIN

RETURN preco \* (1 + adicional);

END

$$ LANGUAGE PLPGSQL /\*PROCEDURAL LANGUAGE postgresql \*/

SELECT aplicar\_adicional (10,0.2)

**/\* Função funcionando \*/**

**/\*deverá ser aplicado ORCAMENTO o novo valor\*/**

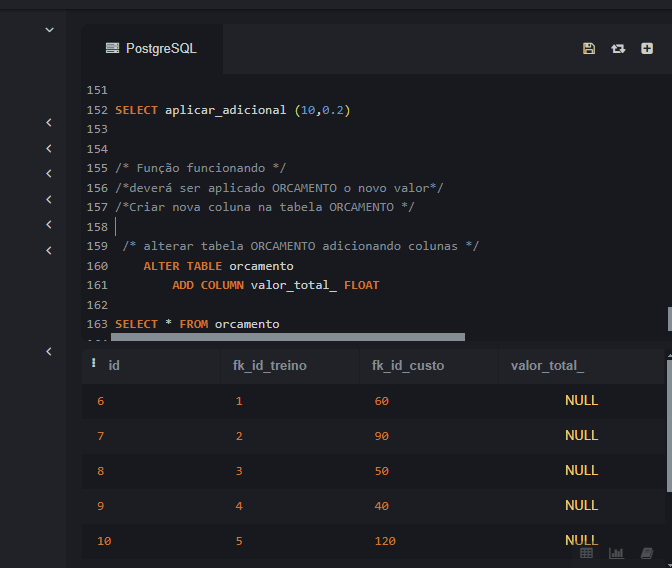
**/\*Criar nova coluna na tabela ORCAMENTO \*/**

/\* alterar tabela ORCAMENTO adicionando coluna\_Valor TOTAL \*/

ALTER TABLE orcamento

ADD COLUMN valor\_total\_ FLOAT

SELECT \* FROM orcamento



**/\* Função funcionando \*/**

**/\*deverá ser aplicado ORCAMENTO o novo valor\*/**

**/\*Criar nova coluna na tabela ORCAMENTO \*/**

/\* alterar tabela ORCAMENTO adicionando coluna de Valor Total \*/

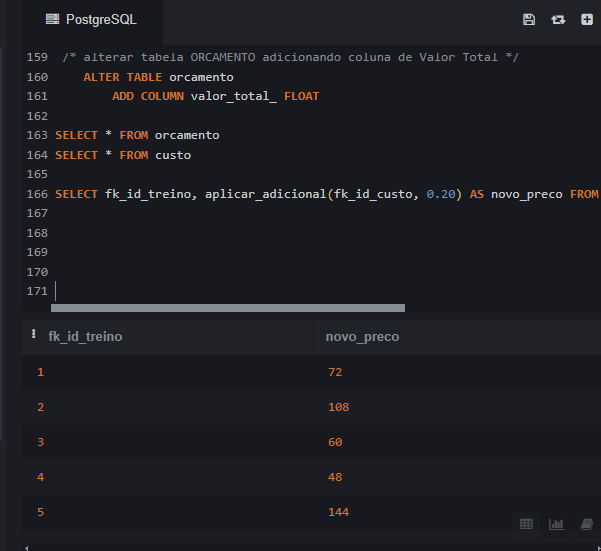
ALTER TABLE orcamento

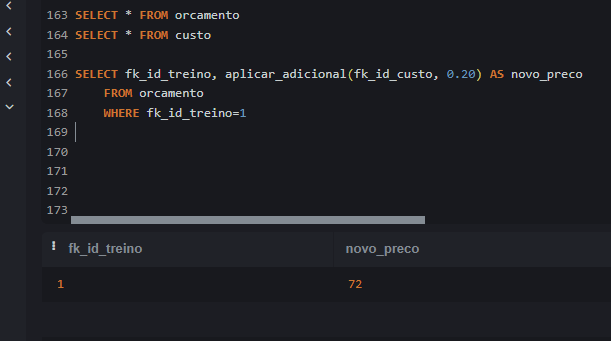
ADD COLUMN valor\_total\_ FLOAT

SELECT \* FROM orcamento

SELECT \* FROM custo

SELECT fk\_id\_treino, aplicar\_adicional(fk\_id\_custo, 0.20) AS novo\_preco FROM orcamento





**/\* Agora, só aplicar o Adicional de 20%, nas linhas onde Haverá treino no Fim de Semana = FimSemana = TRUE \*/**

**UPDATE orcamento**

**SET valor\_total\_ = aplicar\_adicional(100,0.20),**

**fimsemana =TRUE**

**WHERE id=6**

**UPDATE orcamento**

**SET valor\_total\_ = 100,**

**fimsemana =FALSE**

**WHERE id=7**

**UPDATE orcamento**

**SET valor\_total\_ = 50,**

**fimsemana =FALSE**

**WHERE id=8**

**UPDATE orcamento**

**SET valor\_total\_ = aplicar\_adicional(150,0.20),**

**fimsemana =TRUE**

**WHERE id=9**

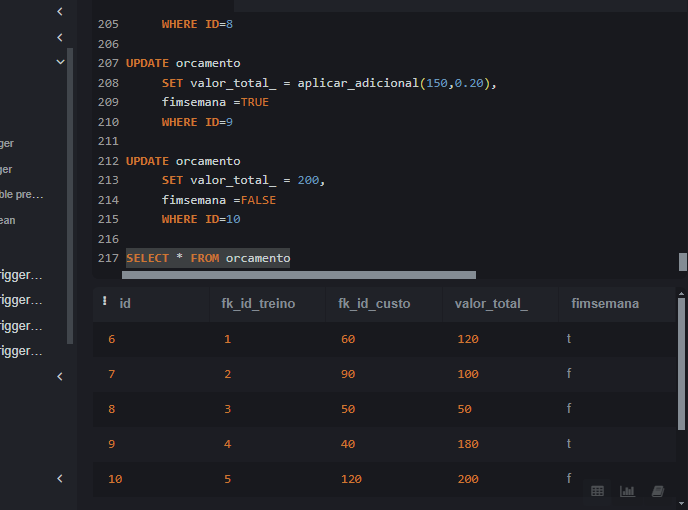
**UPDATE orcamento**

**SET valor\_total\_ = 200,**

**fimsemana =FALSE**

**WHERE id=10**

**SELECT \* FROM orcamento**

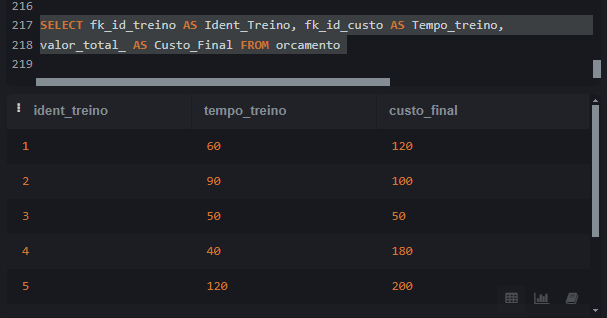
****

**Finalizando,**

**reescrevendo os itens das colunas para melhor entendimento \*/**

SELECT fk\_id\_treino AS Ident\_Treino, fk\_id\_custo as Tempo\_treino,

valor\_total\_ AS Custo\_Final FROM orcamento



**/\* ==================================================================\*/**

**Código total no SQliteonline**

/\* criar a tabela CUSTO com os seus atributos \*/

CREATE TABLE custo (

tempo INT NOT NULL PRIMARY KEY,

valor\_treino FLOAT NOT NULL,

fimSemana BOOLEAN NOT NULL /\* 1=> Fim de semana , 0=> durante a semana \*/

)

;

/\* criar a tabela CLIENTE com os seus atributos \*/

CREATE TABLE cliente (

id SERIAL PRIMARY KEY,

nome VARCHAR(48) NOT NULL,

rg VARCHAR(10) NOT NULL,

telefone VARCHAR (12) NOT NULL

);

CREATE TABLE treino (

/\*id SERIAL PRIMARY KEY,\*/

treino VARCHAR(48) NOT NULL,

dia VARCHAR(10) NOT NULL,

horario VARCHAR (12) NOT NULL,

professor VARCHAR (12) NOT NULL

);

DROP TABLE treino

CREATE TABLE treino (

id SERIAL PRIMARY KEY,

treino VARCHAR(48) NOT NULL,

dia VARCHAR(10) NOT NULL,

horario VARCHAR (12) NOT NULL,

professor VARCHAR (12) NOT NULL

);

/\*inserir tabela do CLIENTE\*/

INSERT INTO cliente (nome, rg, telefone) VALUES

('Paulo','353536767', '0645465454'),

('Tiago','8787653-SP', '0518484949'),

('Tadeu','535557-RJ', '021567482'),

('Ronaldo','33646','(021)5657948'),

('Marcelo','5353535-SP','(021)955555')

SELECT \* from cliente

/\* Inserir dados na tabela TREINO \*/

INSERT INTO treino (treino, dia, horario,professor) VALUES

('cross','01/01/24', '05h00','Alice'),

('musculacao','05/02/24', '07h00','José'),

('fitness','11/03/24', '08h00','João'),

('yoga','12/04/24','10h00','Alex'),

('Pilatos','04/05/24','11h00','Gabi')

SELECT \* FROM treino

/\* Agora como funcionaria a nossa tabela auxiliar \*/

/\* Associar o TREINO com CLIENTE \*/

/\* Criar a tabela EXERCICIO\*/

/\* Inserir dados na tabela CUSTO\*/

INSERT INTO custo (tempo,valor\_treino,fimSemana) VALUES

(60,100.00,TRUE),

(90,100.00,FALSE),

(50,50.00,False),

(40,150.00,true),

(120,200.00,FALSE)

SELECT \* FROM custo

/\* Agora como funcionaria a nossa tabela auxiliar \*/

/\* Associar o TREINO com CLIENTE \*/

/\* Criar a tabela EXERCICIO\*/

CREATE TABLE exercicio (

id SERIAL PRIMARY KEY,

fk\_id\_treino INT NOT NULL,

fk\_id\_cliente INT NOT NULL,

CONSTRAINT fk\_treino FOREIGN KEY (fk\_id\_treino) REFERENCES treino(ID),

CONSTRAINT fk\_cliente FOREIGN KEY (fk\_id\_cliente) REFERENCES cliente(ID)

);

SELECT \* FROM exercicio

/\* Inserindo valores na tabela auxiliar EXERCICIO \*/

INSERT INTO exercicio (fk\_id\_treino,fk\_id\_cliente) VALUES

(1,1),

(2,2),

(3,3),

(4,4),

(5,5)

SELECT \* from exercicio

/\* Fazendo relacionamentos Tabela CLIENTE e tabela TREINO\*/

/\* selecionar uma tabela com os registros da tabela TREINO \*/

/\* e com os registros em comum da tabela CLIENTE \*/

SELECT \* FROM treino /\* tabela á esquerda\*/

inner join cliente /\* pegar todos os registros em comum entre tabelas \*/

on treino.id = cliente.id /\* comando on é tipo indicar qual a conexão entre colunas das tabelas precisam ser iguais \*/

/\* Agora como funcionaria a nossa tabela auxiliar \*/

/\* Associar o TREINO com CUSTO \*/

/\* Criar a tabela ORCAMENTO\*/

CREATE TABLE orcamento (

id SERIAL PRIMARY KEY,

fk\_id\_treino INT NOT NULL,

fk\_id\_custo INT NOT NULL,

CONSTRAINT fk\_id\_treino FOREIGN KEY (fk\_id\_treino) REFERENCES treino(ID),

CONSTRAINT fk\_id\_custo FOREIGN KEY (fk\_id\_custo) REFERENCES custo(tempo)

);

/\*DROP TABLE orcamento\*/

/\* Inserir valores na tabela ORCAMENTO \*/

INSERT INTO orcamento (fk\_id\_treino,fk\_id\_custo) VALUES

(1,60),

(2,90),

(3,50),

(4,40),

(5,120)

SELECT \* FROM orcamento

/\*Fazendo relacionamentos Tabela CUSTO e tabela ORCAMENTO \*/

/\* selecionar uma tabela com os registros da tabela ORCAMENTO \*/

/\* e com os registros em comum da tabela CUSTO \*/

SELECT \* FROM orcamento /\* tabela á esquerda\*/

inner join custo /\* pegar todos os registros em comum entre tabelas \*/

ON orcamento.fk\_id\_custo = custo.tempo

/\* escrever uma função Aplicar\_adicional 20%\*/

CREATE or REPLACE FUNCTION aplicar\_adicional (preco FLOAT, adicional FLOAT)

RETURNS FLOAT as $$

BEGIN

RETURN preco \* (1 + adicional);

END

$$ LANGUAGE PLPGSQL /\*PROCEDURAL LANGUAGE postgresql \*/

SELECT aplicar\_adicional (10,0.2)

/\* Função funcionando \*/

/\*deverá ser aplicado ORCAMENTO o novo valor\*/

/\*Criar nova coluna na tabela ORCAMENTO \*/

/\* alterar tabela ORCAMENTO adicionando coluna de Valor Total \*/

ALTER TABLE orcamento

ADD COLUMN valor\_total\_ FLOAT

SELECT \* FROM orcamento

SELECT \* FROM custo

SELECT fk\_id\_treino, aplicar\_adicional(fk\_id\_custo, 0.20) AS novo\_preco

FROM orcamento

WHERE fk\_id\_treino=1

/\*---------------------------\*/

SELECT \* from custo

/\* Aplicar Adicional somente onde fimSemana=TRUE \*/

ALTER TABLE orcamento

add COLUMN fimsemana BOOLEAN;

SELECT \* FROM orcamento

UPDATE orcamento

SET fimsemana = TRUE

UPDATE orcamento

set fimsemana = FALSE

UPDATE orcamento

SET valor\_total\_ = 100,

fimsemana =TRUE

WHERE id=6

UPDATE orcamento

SET valor\_total\_ = aplicar\_adicional(100,0.20),

fimsemana =TRUE

WHERE id=6

UPDATE orcamento

SET valor\_total\_ = 100,

fimsemana =FALSE

WHERE id=7

UPDATE orcamento

SET valor\_total\_ = 50,

fimsemana =FALSE

WHERE id=8

UPDATE orcamento

SET valor\_total\_ = aplicar\_adicional(150,0.20),

fimsemana =TRUE

WHERE id=9

UPDATE orcamento

SET valor\_total\_ = 200,

fimsemana =FALSE

WHERE id=10

SELECT fk\_id\_treino AS Ident\_Treino, fk\_id\_custo as Tempo\_treino,

valor\_total\_ AS Custo\_Final FROM orcamento

==============================================================