

INSTITUTO FEDERAL
MINAS GERAIS
Campus Formiga

TRABALHO DE BANCO DE DADOS 2

Alunos: Kimberly Lamounier Campos Ferreira - 0016190
Luiz Eduardo Pereira - 0021619

BRIEFING

IDENTIFICAÇÃO

Produto: programa de gerência de finanças

Responsáveis: Kimberly Lamounier e Luiz Eduardo

Data: 02/07/2018

ANUNCIANTE

O produto oferecido é um banco de dados para o gerenciamento de finanças, o mesmo foi denominado Finanças. Esse programa é disponibilizado para download. Sempre que necessário realizar uma consulta de dados sobre um usuário, seus dependentes, suas contas e transações, em forma de relatório o programa disponibiliza menu de consultas.

PÚBLICO-ALVO

O programa está sendo desenvolvido exclusivamente para usuários que desejam organizar suas finanças. Atingindo vários públicos como um empreendedor, controlando as finanças de uma empresa, ou um usuário doméstico controlando seu orçamento.

OBJETIVOS

O objetivo é armazenar todos os dados necessários para o controle do orçamento dos usuários, e manipular os dados cadastrados, além de gerar relatórios para controle de dados, promovendo assim controle de suas receitas e despesas.

PEDIDO DE TRABALHO

Foi solicitado a criação de um programa com conexão ao banco de dados, sendo capaz de armazenar qualquer tipo de dado do usuário, como por exemplo, nome, login, senha, receitas, despesas, e também os dados dos seus dependentes. O programa deve ser dinâmico e de fácil manipulação de seus usuários e banco de dados também deve ser ágil e prático.

PRAZOS PARA A REALIZAÇÃO DO TRABALHO

O prazo para o desenvolvimento foi estipulado para um mês e esse prazo foi cumprido.

MODELO LÓGICO

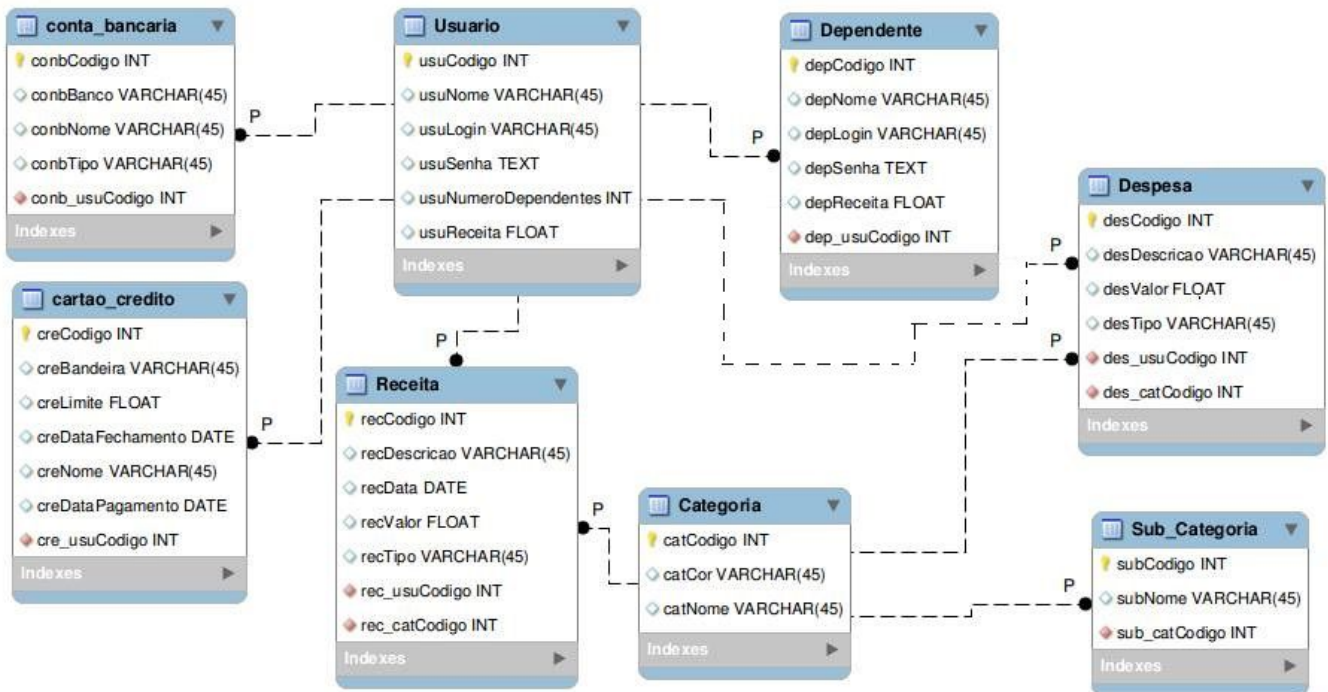
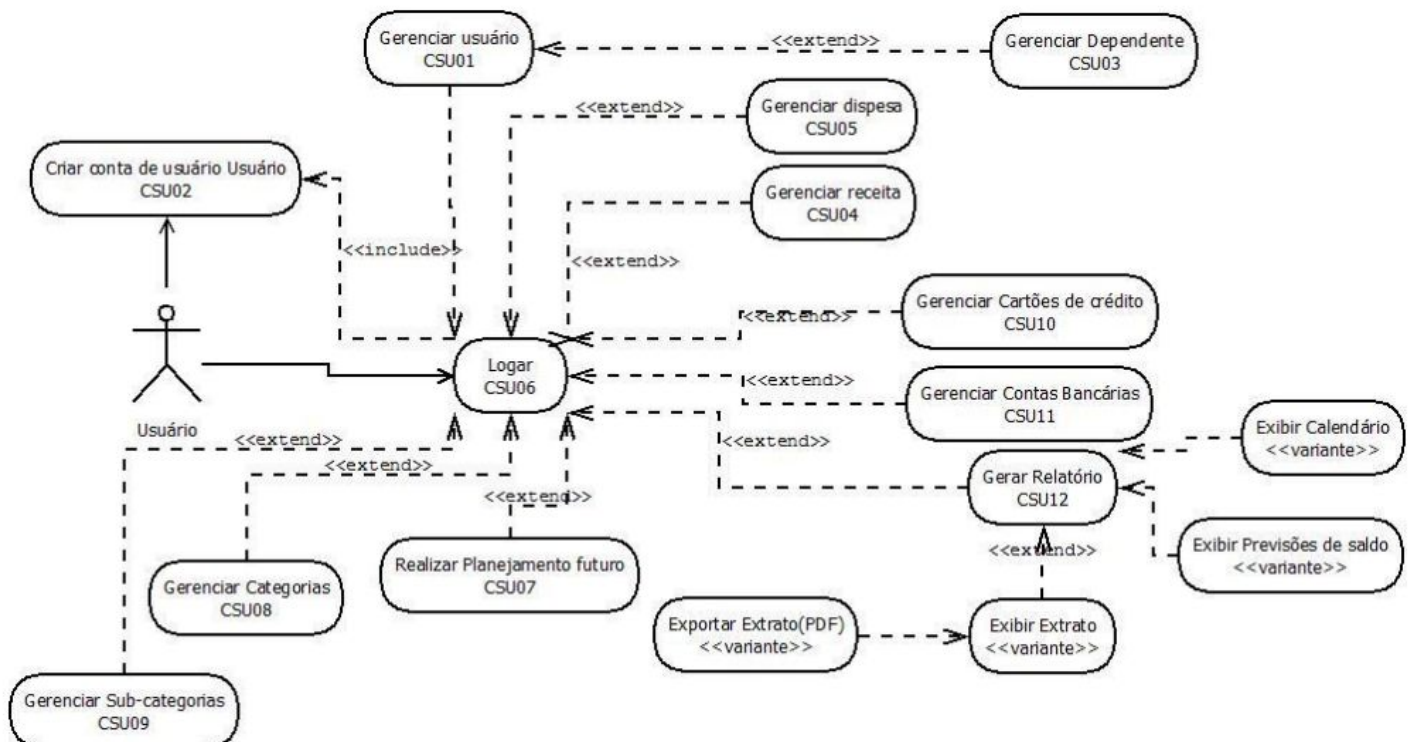
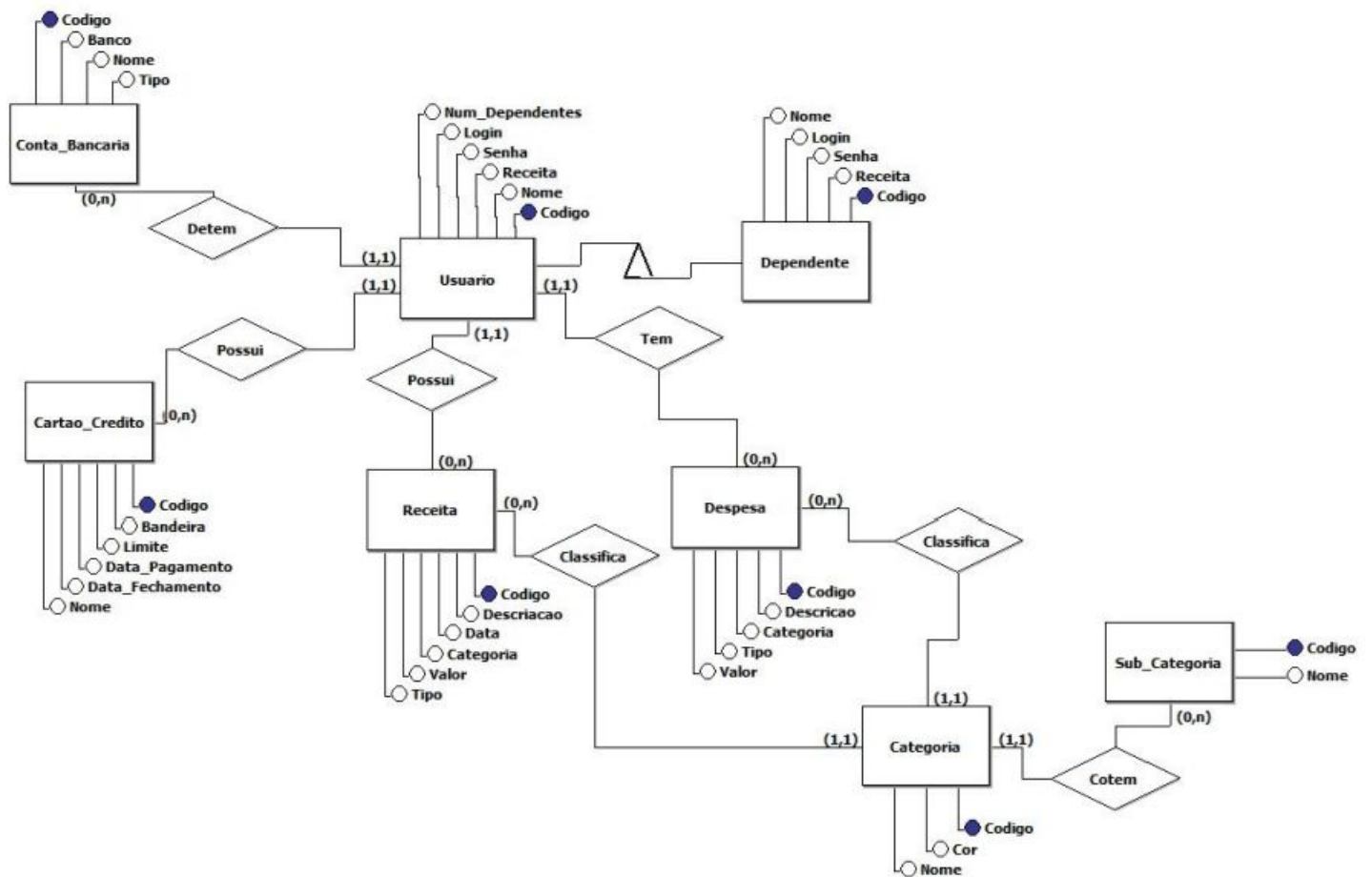


DIAGRAMA DE CASOS DE USO



MODELO CONCEITUAL



DICIONÁRIO DE DADOS

Table: cartao_credito						
Attributes						
Name	Type	Not Null	PK	FK	Default	Comment
creCodigo	INT(11)	Yes	Yes	No		
creBandeira	VARCHAR(45)	No	No	No	NULL	
creLimite	FLOAT	No	No	No	NULL	
creDataFechamento	DATE	No	No	No	NULL	
creNome	VARCHAR(45)	No	No	No	NULL	
creDataPagamento	DATE	No	No	No	NULL	
cre_usuCodigo	INT(11)	Yes	No	Yes		

Table: categoria						
Attributes						
Name	Type	Not Null	PK	FK	Default	Comment
catCodigo	INT(11)	Yes	Yes	No		
catCor	VARCHAR(45)	No	No	No	NULL	
catNome	VARCHAR(45)	No	No	No	NULL	

Table: conta_bancaria						
Attributes						
Name	Type	Not Null	PK	FK	Default	Comment
conbCodigo	INT(11)	Yes	Yes	No		
conbBanco	VARCHAR(45)	No	No	No	NULL	
conbNome	VARCHAR(45)	No	No	No	NULL	
conbTipo	VARCHAR(45)	No	No	No	NULL	
conb_usuCodigo	INT(11)	Yes	No	Yes		

Table: conta_bancaria						
Attributes						
Name	Type	Not Null	PK	FK	Default	Comment
conbCodigo	INT(11)	Yes	Yes	No		
conbBanco	VARCHAR(45)	No	No	No	NULL	
conbNome	VARCHAR(45)	No	No	No	NULL	
conbTipo	VARCHAR(45)	No	No	No	NULL	
conb_usuCodigo	INT(11)	Yes	No	Yes		

Table: dependente						
Attributes						
Name	Type	Not Null	PK	FK	Default	Comment
depCodigo	INT(11)	Yes	Yes	No		
depNome	VARCHAR(45)	No	No	No	NULL	
depLogin	VARCHAR(45)	No	No	No	NULL	
depSenha	TEXT	No	No	No	NULL	
depReceita	FLOAT	No	No	No	NULL	
dep_usuCodigo	INT(11)	Yes	No	Yes		

Table: despesa

Attributes						
Name	Type	Not Null	PK	FK	Default	Comment
desCodigo	INT(11)	Yes	Yes	No		
desDescricao	VARCHAR(45)	No	No	No	NULL	
desValor	FLOAT	No	No	No	NULL	
desTipo	VARCHAR(45)	No	No	No	NULL	
des_usuCodigo	INT(11)	Yes	No	Yes		
des_catCodigo	INT(11)	Yes	No	Yes		

Table: receita

Attributes						
Name	Type	Not Null	PK	FK	Default	Comment
recCodigo	INT(11)	Yes	Yes	No		
recDescricao	VARCHAR(45)	No	No	No	NULL	
recData	DATE	No	No	No	NULL	
recValor	FLOAT	No	No	No	NULL	
recTipo	VARCHAR(45)	No	No	No	NULL	
rec_usuCodigo	INT(11)	Yes	No	Yes		
rec_catCodigo	INT(11)	Yes	No	Yes		

Table: sub_categoria

Attributes						
Name	Type	Not Null	PK	FK	Default	Comment
subCodigo	INT(11)	Yes	Yes	No		
subNome	VARCHAR(45)	No	No	No	NULL	
sub_catCodigo	INT(11)	Yes	No	Yes		

Table: usuario

Attributes						
Name	Type	Not Null	PK	FK	Default	Comment
usuCodigo	INT(11)	Yes	Yes	No		
usuNome	VARCHAR(45)	No	No	No	NULL	
usuLogin	VARCHAR(45)	No	No	No	NULL	
usuSenha	TEXT	No	No	No	NULL	
usuNumeroDependentes	INT(10)	No	No	No	NULL	
usuReceita	FLOAT	No	No	No	NULL	

CRIAÇÃO DO BANCO DE DADOS

```
SET @OLD_UNIQUE_CHECKS=@@UNIQUE_CHECKS, UNIQUE_CHECKS=0;
SET @OLD_FOREIGN_KEY_CHECKS=@@FOREIGN_KEY_CHECKS,
FOREIGN_KEY_CHECKS=0;
SET @OLD_SQL_MODE=@@SQL_MODE,
SQL_MODE='TRADITIONAL,ALLOW_INVALID_DATES';
```

```
CREATE SCHEMA IF NOT EXISTS `financas_db` DEFAULT CHARACTER SET
utf8 COLLATE utf8_general_ci ;
```

```
CREATE TABLE IF NOT EXISTS `financas_db`.`conta_bancaria` (
  `conbCodigo` INT(11) NOT NULL AUTO_INCREMENT,
  `conbBanco` VARCHAR(45) NULL DEFAULT NULL,
  `conbNome` VARCHAR(45) NULL DEFAULT NULL,
  `conbTipo` VARCHAR(45) NULL DEFAULT NULL,
  `conb_usuCodigo` INT(11) NOT NULL,
  PRIMARY KEY (`conbCodigo`),
  INDEX `fk_conta_bancaria_usuario_idx` (`conb_usuCodigo` ASC),
  CONSTRAINT `fk_conta_bancaria_usuario`
    FOREIGN KEY (`conb_usuCodigo`)
    REFERENCES `financas_db`.`Usuario` (`usuCodigo`)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION)
ENGINE = InnoDB
DEFAULT CHARACTER SET = utf8
COLLATE = utf8_general_ci;
```

```
CREATE TABLE IF NOT EXISTS `financas_db`.`Usuario` (
  `usuCodigo` INT(11) NOT NULL AUTO_INCREMENT,
  `usuNome` VARCHAR(45) NULL DEFAULT NULL,
  `usuLogin` VARCHAR(45) NULL DEFAULT NULL,
  `usuSenha` TEXT NULL DEFAULT NULL,
  `usuNumeroDependentes` INT(10) UNSIGNED NULL DEFAULT NULL,
  `usuReceita` FLOAT(11) NULL DEFAULT NULL,
  PRIMARY KEY (`usuCodigo`))
ENGINE = InnoDB
DEFAULT CHARACTER SET = utf8
COLLATE = utf8_general_ci;
```

```
CREATE TABLE IF NOT EXISTS `financas_db`.`Dependente` (
  `depCodigo` INT(11) NOT NULL AUTO_INCREMENT,
  `depNome` VARCHAR(45) NULL DEFAULT NULL,
  `depLogin` VARCHAR(45) NULL DEFAULT NULL,
  `depSenha` TEXT NULL DEFAULT NULL,
  `depReceita` FLOAT(11) NULL DEFAULT NULL,
```

```

`dep_usuCodigo` INT(11) NOT NULL,
PRIMARY KEY (`depCodigo`),
INDEX `fk_Dependente_Usuario1_idx` (`dep_usuCodigo` ASC),
CONSTRAINT `fk_Dependente_Usuario1`
    FOREIGN KEY (`dep_usuCodigo`)
    REFERENCES `financas_db`.`Usuario` (`usuCodigo`)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION)
ENGINE = InnoDB
DEFAULT CHARACTER SET = utf8
COLLATE = utf8_general_ci;

```

```

CREATE TABLE IF NOT EXISTS `financas_db`.`Despesa` (
    `desCodigo` INT(11) NOT NULL AUTO_INCREMENT,
    `desDescricao` VARCHAR(45) NULL DEFAULT NULL,
    `desValor` FLOAT(11) NULL DEFAULT NULL,
    `desTipo` VARCHAR(45) NULL DEFAULT NULL,
    `des_usuCodigo` INT(11) NOT NULL,
    `des_catCodigo` INT(11) NOT NULL,
    PRIMARY KEY (`desCodigo`),
    INDEX `fk_Despesa_Usuario1_idx` (`des_usuCodigo` ASC),
    INDEX `fk_Despesa_Categorial_idx` (`des_catCodigo` ASC),
    CONSTRAINT `fk_Despesa_Usuario1`
        FOREIGN KEY (`des_usuCodigo`)
        REFERENCES `financas_db`.`Usuario` (`usuCodigo`)
        ON DELETE NO ACTION
        ON UPDATE NO ACTION,
    CONSTRAINT `fk_Despesa_Categorial`
        FOREIGN KEY (`des_catCodigo`)
        REFERENCES `financas_db`.`Categoria` (`catCodigo`)
        ON DELETE NO ACTION
        ON UPDATE NO ACTION)
ENGINE = InnoDB
DEFAULT CHARACTER SET = utf8
COLLATE = utf8_general_ci;

```

```

CREATE TABLE IF NOT EXISTS `financas_db`.`Receita` (
    `recCodigo` INT(11) NOT NULL AUTO_INCREMENT,
    `recDescricao` VARCHAR(45) NULL DEFAULT NULL,
    `recData` DATE NULL DEFAULT NULL,
    `recValor` FLOAT(11) NULL DEFAULT NULL,
    `recTipo` VARCHAR(45) NULL DEFAULT NULL,
    `rec_usuCodigo` INT(11) NOT NULL,
    `rec_catCodigo` INT(11) NOT NULL,
    PRIMARY KEY (`recCodigo`),
    INDEX `fk_Receita_Usuario1_idx` (`rec_usuCodigo` ASC),

```



```

INDEX `fk_Receita_Categorial_idx` (`rec_catCodigo` ASC),
CONSTRAINT `fk_Receita_Usuario1`
  FOREIGN KEY (`rec_usuCodigo`)
  REFERENCES `financas_db`.`Usuario` (`usuCodigo`)
  ON DELETE NO ACTION
  ON UPDATE NO ACTION,
CONSTRAINT `fk_Receita_Categorial`
  FOREIGN KEY (`rec_catCodigo`)
  REFERENCES `financas_db`.`Categoria` (`catCodigo`)
  ON DELETE NO ACTION
  ON UPDATE NO ACTION)
ENGINE = InnoDB
DEFAULT CHARACTER SET = utf8
COLLATE = utf8_general_ci;

CREATE TABLE IF NOT EXISTS `financas_db`.`Categoria` (
  `catCodigo` INT(11) NOT NULL AUTO_INCREMENT,
  `catCor` VARCHAR(45) NULL DEFAULT NULL,
  `catNome` VARCHAR(45) NULL DEFAULT NULL,
  PRIMARY KEY (`catCodigo`))
ENGINE = InnoDB
DEFAULT CHARACTER SET = utf8
COLLATE = utf8_general_ci;

CREATE TABLE IF NOT EXISTS `financas_db`.`Sub_Categoria` (
  `subCodigo` INT(11) NOT NULL AUTO_INCREMENT,
  `subNome` VARCHAR(45) NULL DEFAULT NULL,
  `sub_catCodigo` INT(11) NOT NULL,
  PRIMARY KEY (`subCodigo`),
  INDEX `fk_Sub_Categoria_Categorial_idx` (`sub_catCodigo` ASC),
  CONSTRAINT `fk_Sub_Categoria_Categorial`
    FOREIGN KEY (`sub_catCodigo`)
    REFERENCES `financas_db`.`Categoria` (`catCodigo`)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION)
ENGINE = InnoDB
DEFAULT CHARACTER SET = utf8
COLLATE = utf8_general_ci;

CREATE TABLE IF NOT EXISTS `financas_db`.`cartao_credito` (
  `creCodigo` INT(11) NOT NULL AUTO_INCREMENT,
  `creBandeira` VARCHAR(45) NULL DEFAULT NULL,
  `creLimite` FLOAT(11) NULL DEFAULT NULL,
  `creDataFechamento` DATE NULL DEFAULT NULL,
  `creNome` VARCHAR(45) NULL DEFAULT NULL,
  `creDataPagamento` DATE NULL DEFAULT NULL,

```

```

`cre_usuCodigo` INT(11) NOT NULL,
PRIMARY KEY (`creCodigo`),
INDEX `fk_cartao_credito_Usuariol_idx` (`cre_usuCodigo` ASC),
CONSTRAINT `fk_cartao_credito_Usuariol`
    FOREIGN KEY (`cre_usuCodigo`)
    REFERENCES `financas_db`.`Usuario` (`usuCodigo`)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION)
ENGINE = InnoDB
DEFAULT CHARACTER SET = utf8
COLLATE = utf8_general_ci;

SET SQL_MODE=@OLD_SQL_MODE;
SET FOREIGN_KEY_CHECKS=@OLD_FOREIGN_KEY_CHECKS;
SET UNIQUE_CHECKS=@OLD_UNIQUE_CHECKS;
SET GLOBAL sql_mode=(SELECT
REPLACE(@@sql_mode, 'ONLY_FULL_GROUP_BY', ''));

```

INSERÇÃO DE DADOS

```

#usuarios
insert into Usuario(UsuNome, UsuReceita, UsuLogin,
UsuSenha,UsuNumeroDependentes)
values ('Bruna Duarte', '300.00', 'bruna123', 'overw1234', 2);

insert into Usuario(UsuNome, UsuReceita, UsuLogin,
UsuSenha,UsuNumeroDependentes)
values ('Kimberly Ferreira', '700.00', 'kimberly12','crystal123',
1);

insert into Usuario(UsuNome, UsuReceita, UsuLogin,
UsuSenha,UsuNumeroDependentes)
values ('Luiz Eduardo', '200.00', 'LuizMagic30', 'jogo3322', 0);

#contas
insert into conta_bancaria(conbBanco, conbNome, conbTipo,
conb_usuCodigo)
values ('Banco do Brasil', 'Principal', 'Conta Corrente', 1);

insert into conta_bancaria(conbBanco, conbNome, conbTipo,
conb_usuCodigo)
values ('Santander', 'Emergencia', 'Conta Corrente', 2);

insert into conta_bancaria(conbBanco, conbNome, conbTipo,

```

```
conb_usuCodigo)
values ('Banco do Brasil', 'Principal', 'Conta Corrente', 2);
```

```
#cartao_credito
insert into cartao_credito(creBandeira, creLimite,
creDataPagamento,
creDataFechamento, creNome, cre_UsuCodigo)
values ('MasterCard', '1000.00', '2018-09-10', '2018-09-20',
'emergencias',1);
```

```
insert into cartao_credito(creBandeira, creLimite,
creDataPagamento,
creDataFechamento, creNome, cre_UsuCodigo)
values ('Elo', '280.00', '2017-11-10', '2017-11-20',
'alimentacao',1);
```

```
insert into cartao_credito(creBandeira, creLimite,
creDataPagamento,
creDataFechamento, creNome, cre_UsuCodigo)
values ('Visa', '3000.00', '2018-10-10', '2018-11-20',
'universitario', 2);
```

```
#categoria
insert into Categoria(catCor, catNome)
values ('Vermelho', 'Lazer');
```

```
insert into Categoria(catCor, catNome)
values ('Azul', 'Alimentação');
```

```
insert into Categoria(catCor, catNome)
values ('Rosa', 'Transporte');
```

```
insert into Categoria(catCor, catNome)
values ('Amarelo', 'Escola');
```

```
insert into Categoria(catCor, catNome)
values ('Preto', 'Faculdade');
```

```
insert into Categoria(catCor, catNome)
values ('Roxo', 'Casa');
```

```
#Receita
```

```
insert into Receita(recDescricao, recData, recValor, recTipo,
rec_UsuCodigo,
rec_CatCodigo)
values ('Auxilio', '2018-03-10', '100.00', 'Mensal', 1, 1);

insert into Receita(recDescricao, recData, recValor, recTipo,
rec_UsuCodigo,
Rec_CatCodigo)
values ('Auxilio', '2018-01-10', '80.00', 'Mensal', 2, 2);

insert into Receita(recDescricao, recData, recValor, recTipo,
rec_UsuCodigo,
Rec_CatCodigo)
values ('Economias', '2018-04-29', '200.00', 'Bimestral', 3, 3);

insert into Receita(recDescricao, recData, recValor, recTipo,
rec_UsuCodigo,
rec_CatCodigo)
values ('Salario', '2018-05-24', '25.00', 'Mensal', 1, 1);

insert into Receita(recDescricao, recData, recValor, recTipo,
rec_UsuCodigo,
rec_CatCodigo)
values ('Salario', '2018-02-17', '700.00', 'Mensal', 2, 1);

insert into Receita(recDescricao, recData, recValor, recTipo,
rec_UsuCodigo,
Rec_CatCodigo)
values ('Salario', '2018-01-30', '4350.00', 'Mensal', 3, 2);

insert into Receita(recDescricao, recData, recValor, recTipo,
rec_UsuCodigo,
Rec_CatCodigo)
values ('Salario', '2018-06-29', '2690.00', 'Mensal', 1, 2);

insert into Receita(recDescricao, recData, recValor, recTipo,
rec_UsuCodigo,
rec_CatCodigo)
values ('Salario', '2018-05-22', '1500.00', 'Mensal', 2, 1);

insert into Receita(recDescricao, recData, recValor, recTipo,
rec_UsuCodigo,
Rec_CatCodigo)
values ('Salario', '2018-03-29', '2999.00', 'Mensal', 3, 3);
```

```
insert into Receita(recDescricao, recData, recValor, recTipo,
rec_UsuCodigo,
rec_CatCodigo)
values ('Salario', '2018-03-12', '12800.00', 'Mensal', 1, 1);
```

#Despesa

```
insert into Despesa(desDescricao, desValor, desTipo,
des_UsuCodigo,
des_CatCodigo)
values ('Viagem para Bahia', '1500.00', 'Anual', 2, 1);
```

```
insert into Despesa(desDescricao, desValor, desTipo,
des_UsuCodigo,
Des_CatCodigo)
values ('Almoço no IFMG', '1300.00', 'Diario', 3, 2);
```

```
insert into Despesa(desDescricao, desValor, desTipo,
des_UsuCodigo,
des_CatCodigo)
values ('Ida para Divinópolis', '100.00', 'Quinzenal', 1, 5);
```

```
insert into Despesa(desDescricao, desValor, desTipo,
des_UsuCodigo,
des_CatCodigo)
values ('Agua', '100.00', 'Mensal', 1, 3);
```

```
insert into Despesa(desDescricao, desValor, desTipo,
des_UsuCodigo,
des_CatCodigo)
values ('Faculdade', '3000.00', 'Mensal', 1, 2);
```

```
insert into Despesa(desDescricao, desValor, desTipo,
des_UsuCodigo,
Des_CatCodigo)
values ('Plano de Saude', '60.00', 'Anual', 2, 2);
```

```
insert into Despesa(desDescricao, desValor, desTipo,
des_UsuCodigo,
des_CatCodigo)
values ('Academia', '100.00', 'Mensal', 2, 3);
```

```
insert into Despesa(desDescricao, desValor, desTipo,
des_UsuCodigo,
des_CatCodigo)
values ('Jogo de basquete', '250.00', 'Mensal', 3, 3);
```



```

#Sub-Categoria
insert into Sub_Categoria(subNome, sub_CatCodigo)
values ('Supermercado', 2);

insert into Sub_Categoria(subNome, sub_CatCodigo)
values ('Livros', 5);

insert into Sub_Categoria(subNome, sub_CatCodigo)
values ('Viagem', 1);

insert into Sub_Categoria(subNome, sub_CatCodigo)
values ('Material Escolar', 4);

insert into Sub_Categoria(subNome, sub_CatCodigo)
values ('Cozinha', 6);

insert into Sub_Categoria(subNome, sub_CatCodigo)
values ('Limpeza', 6);

#Dependente
insert into Dependente(depNome, depLogin, depSenha, depReceita,
dep_UsuCodigo)
values ('Sonia Duarte', 'scDuarte', 'Senha123', 2000.00, 1);

insert into Dependente(depNome, depLogin, depSenha, depReceita,
dep_UsuCodigo)
values ('Fatima Faria', 'fatimafaria', 'Senha000', 5000.00, 1);

insert into Dependente(depNome, depLogin, depSenha, depReceita,
dep_UsuCodigo)
values ('Aldair Ferreira', 'aldairf', 'Senha284', 150.00, 2);

insert into Dependente(depNome, depLogin, depSenha, depReceita,
dep_UsuCodigo)
values ('Joana Rodrigues', 'jrodrigues', 'senha999', 564.00, 3);

insert into Dependente(depNome, depLogin, depSenha, depReceita,
dep_UsuCodigo)
values ('Itamar Barbosa', 'itabarbosa', 'senha555', 1452.00, 2);

```

CONSULTAS

#Pelo menos 3 Consultas SQL relevantes(para relatórios) entre duas tabelas ou mais, com a utilização de pelo menos uma função de data em cada consulta;

#1

Mostrar os usuarios e nome do cartao que tem data de pagamento de cartao vencida

```
select usuNome, creNome
```

```
from Usuario join cartao_credito
```

```
where cre_usuCodigo = usuCodigo and creDataPagamento < current_date()
```

#2

Mostrar usuario com alguma receita entre dia 10/01/2018 e 23/03/2018

```
select distinct usuNome
```

```
from Usuario join Receita
```

```
where rec_usuCodigo = usuCodigo and recData between '2018-01-10' and '2018-03-23';
```

#3

Mostrar usuario com alguma receita no mês de março

```
select distinct usuNome
```

```
from Usuario join Receita
```

```
where rec_usuCodigo = usuCodigo and MONTH(recData) = '03';
```

Pelo menos 5 Consultas SQL relevantes(para relatórios) usando agrupamentos e funções agregadas diversas;

#4

#Mostrar o somatorio das despesas de cada cliente que possui

```
select usuNome, sum(desValor)
```

```
from Usuario join Despesa
```

```
where usuCodigo = des_usuCodigo
```

```
group by usuCodigo;
```

#5

#Mostrar o numero de cartões de crédito que cada usuario tem

```
select usuNome, count(creCodigo)
```

```
from Usuario join cartao_credito
```

```
where usuCodigo = cre_usuCodigo
```

```
group by usuCodigo;
```

#6

#Mostrar o numero de dependentes de cada usuario

```
select usuNome, count(depCodigo)
```

```
from Usuario join Dependente
```

```
where usuCodigo = dep_usuCodigo
```

```
group by usuCodigo;
```

#7

#Mostrar o numero de tipos de categorias das receita de cada usuario

```
select usuNome, count(catCodigo)
```

```
from Usuario join Receita join Categoria
where usuCodigo = rec_usuCodigo and rec_catCodigo = catCodigo
group by usuCodigo;
```

#8

```
#Mostrar o numero de tipos de categorias das despesas de cada usuario
select usuNome, count(catCodigo)
from Usuario join Despesa join Categoria
where des_usuCodigo = usuCodigo and des_catCodigo = catCodigo
group by usuCodigo;
```

#Pelo menos 5 Consultas SQL relevantes(para relatórios) usando corretamente a cláusula Having;

#9

```
#Mostrar usuarios que possuem mais de 2 contas bancarias
select usuNome, count(conb_usuCodigo)
from Usuario join conta_bancaria
where usuCodigo = conb_usuCodigo
group by usuCodigo
having count(conb_usuCodigo) >= 2;
```

#10

```
#mostrar usuarios que tem receita menor que a soma das receitas de seus dependentes
select usuNome, usuReceita as receitaUsuario
from Usuario join Dependente
where usuCodigo = dep_usuCodigo
group by usuCodigo
having sum(depReceita) > receitaUsuario;
```

#11

```
#Mostrar usuarios que tenham Receitas maiores 200.00 reais
select usuNome, recValor
from Usuario join Receita
where usuCodigo = rec_usuCodigo
group by usuCodigo
having recValor >= 200.00;
```

#12

```
#Mostrar usuarios que tenham dependentes com receita maior que 500.00 reais
select usuNome, depReceita as 'Receita do dependente'
from Usuario join Dependente
where usuCodigo = dep_usuCodigo
group by usuCodigo
having depReceita >= 500.00;
```

#13

#Mostrar usuarios que tenham mais de 3 receitas

```
select usuNome, count(recCodigo)
from Usuario join Receita
where usuCodigo = rec_usuCodigo
group by usuCodigo
having count(recCodigo) > 3;
```

#Pelo menos 2 Consultas SQL relevantes(para relatórios) aninhadas(na cláusula From);

#14

#Mostra dependentes tem que despesa na sub-categoria Livros

```
select depNome
from Dependente join Usuario join Despesa join Categoria join (
select sub_catCodigo
from Sub_Categoria
where subNome = 'Livros' )selecionaSubCat
where usuCodigo = dep_usuCodigo and usuCodigo = des_usuCodigo and catCodigo =
des_catCodigo and catCodigo = sub_catCodigo;
```

#15

#Mostra dependentes e usuarios que possuem contas no banco do brasil

```
select depNome
from Dependente join ( select usuCodigo
                        from Usuario join conta_bancaria
                        where (usuCodigo = conb_usuCodigo and conbBanco = 'Banco do Brasil'))ContaBB
where usuCodigo = dep_usuCodigo;
```

#Pelo menos 4 Consultas SQL relevantes(para relatórios) aninhadas(na cláusula Where);

#16

#Mostra usuarios e sua receita que a receita seja maior que a despesa mensal

```
select usuNome, sum(recValor) as 'total'
from Usuario join Receita
where recValor > (select sum(desValor) from Despesa
where desTipo = 'Mensal');
```

#17

#Mostra usuario e seu limite de crédito em que o maior valor de receita seja abaixo de 1000 reais

```
select usuNome, creLimite as 'Limite'
from Usuario join cartao_credito
where usuCodigo = cre_usuCodigo and creLimite < (select max(recValor) from Receita
where recValor < 1000.00);
```

#18

#Mostra usuario e da despesa mais cara mensal

```
select usuNome, desDescricao as 'Despesa'
from Usuario join Despesa
where usuCodigo = des_usuCodigo and desValor = (select max(desValor) from Despesa
where desTipo = 'Mensal');
```

#19

#Mostra usuario e a receita do tipo salario mais barata

```
select usuNome, recValor
from Usuario join Receita
where usuCodigo = rec_usuCodigo and recValor = (select min(recValor) from Receita
where recDescricao = 'Salario')
group by usuCodigo;
```

#Pelo menos 1 Consultas SQL relevantes(para relatórios) aninhadas (envolvendo 3 ou mais tabelas);

#20

#Mostra usuario e sua receita, seu credito e seu banco do valor de receita mais alto

```
select usuNome, recValor, creLimite, conbBanco
from Usuario join Receita join cartao_credito join conta_bancaria
where usuCodigo = rec_usuCodigo and usuCodigo = cre_usuCodigo
and usuCodigo = conb_usuCodigo and recValor = (select max(recValor) from Receita)
group by usuCodigo;
```

Escolha 2 de suas consultas acima solicitadas, reescreva-as e escreva uma outra forma(para cada consulta) de obter o mesmo resultado, explique qual e porque uma das consultas será mais customizada que a outra.

#21

#Mostrar o numero de cartões de crédito que cada usuario tem

```
select usuNome, count(creCodigo)
from Usuario inner join cartao_credito on usuCodigo = cre_usuCodigo
group by usuCodigo;
```

#22

#Mostrar o numero de dependentes de cada usuario

```
select usuNome, count(depCodigo)
from Usuario inner join Dependente on usuCodigo = dep_usuCodigo
group by usuCodigo;
```


Em questão de resultados, não há diferenças entre os 2 tipos de consultas. Sobre desempenho, a condição de junção INNER JOIN tem um ganho discreto comparado com o WHERE, pelo fato de ser interpretado antes. Mas INNER JOIN é melhor por ser Padronizado e flexível.

CRIAÇÃO DE VIEW:

```
CREATE VIEW listarDepedente
AS
SELECT *
FROM Dependente;
```

A view cria uma tabela virtual contendo alguma operação específica, esse view é relevante pois usamos a seleção de dependentes algumas vezes como em relatórios e em exclusão de dependente, então é só chamar a view ao invés de fazer a seleção da tabela.

Como usar a view :

```
SELECT * FROM listarDepedente;
```

CRIAÇÃO DE PROCEDURE:

```
DELIMITER $$
CREATE PROCEDURE alteraReceita
(IN codUsuario INT, valor FLOAT)
BEGIN
UPDATE Usuario
SET usuReceita = usuReceita * valor
WHERE usuCodigo = codUsuario;
END $$
DELIMITER ;
```

Procedure é utilizado para criação de alguma sub-operação para fazer o reuso de alguma operação para otimização. Nesse procedimento recebemos o usuário e o valor para aumento e esse aumento é acrescentado ao valor de receita do usuário informado.

Como usar:

```
CALL alteraReceita(1,1.20);
```

CRIAÇÃO DE TRIGGER:

```
DELIMITER $$
CREATE TRIGGER atulizaNumeroDependente AFTER INSERT
BEGIN
    UPDATE Usuario
        SET usuNumeroDependentes = usuNumeroDependentes + 1
    WHERE usuCodigo = NEW.dep_usuCodigo;
END;
DELIMITER;
```

O Trigger significa que ao ser realizado certa operação será disparado imediatamente a realização de outra determinada operação. Essa trigger dispara uma operação que imediatamente aumenta o número de dependentes em um usuário após a inserção de um dependente.

CRIAÇÃO DE TABELA TEMPORÁRIA

```
CREATE TEMPORARY TABLE Usuario_temporaria
SELECT usuNome,usuNumeroDependentes
FROM Usuario;
```

A tabela permanente fica “escondida”, durante a sessão ou enquanto a temporária ainda existir. É útil para criar e testar *queries*, usando o nome de uma tabela existente, sem afetá-la.