

Disciplina Banco de Dados - [Semestre Letivo: 2022/2]	
Nome dos(as) acadêmicos(as) 1 – Keven Lucas Vieira Gondim 2 – Luca Mascarenhas Plaster 3 – Matheus Leonel de Andrade	Números de matrícula 1 – 202000181 2 – 202014610 3 – 202109822
Turma: INF0022	Professor(a): Leonardo Andrade Ribeiro

TEMA: Trabalho Final de Banco de Dados

I – Mini Mundo

O intuito deste trabalho é de colocar em prática as ideias aprendidas no curso de Banco de Dados, o mini mundo escolhido para implantar as ideias foi de uma instituição financeira, segue-se a seguir a descrição do mesmo:

O nosso mini mundo vai representar uma instituição financeira na qual é constituída de contas, conta possui número da conta, saldo, data de cadastro, toda conta deve ser do tipo conta poupança, possuindo taxa de juros, ou conta-corrente. Também possui o cliente, que tem como atributos nome, endereço, telefone e id da conta, cliente sempre é do tipo físico possuindo cpf, ou jurídico possuindo cnpj.

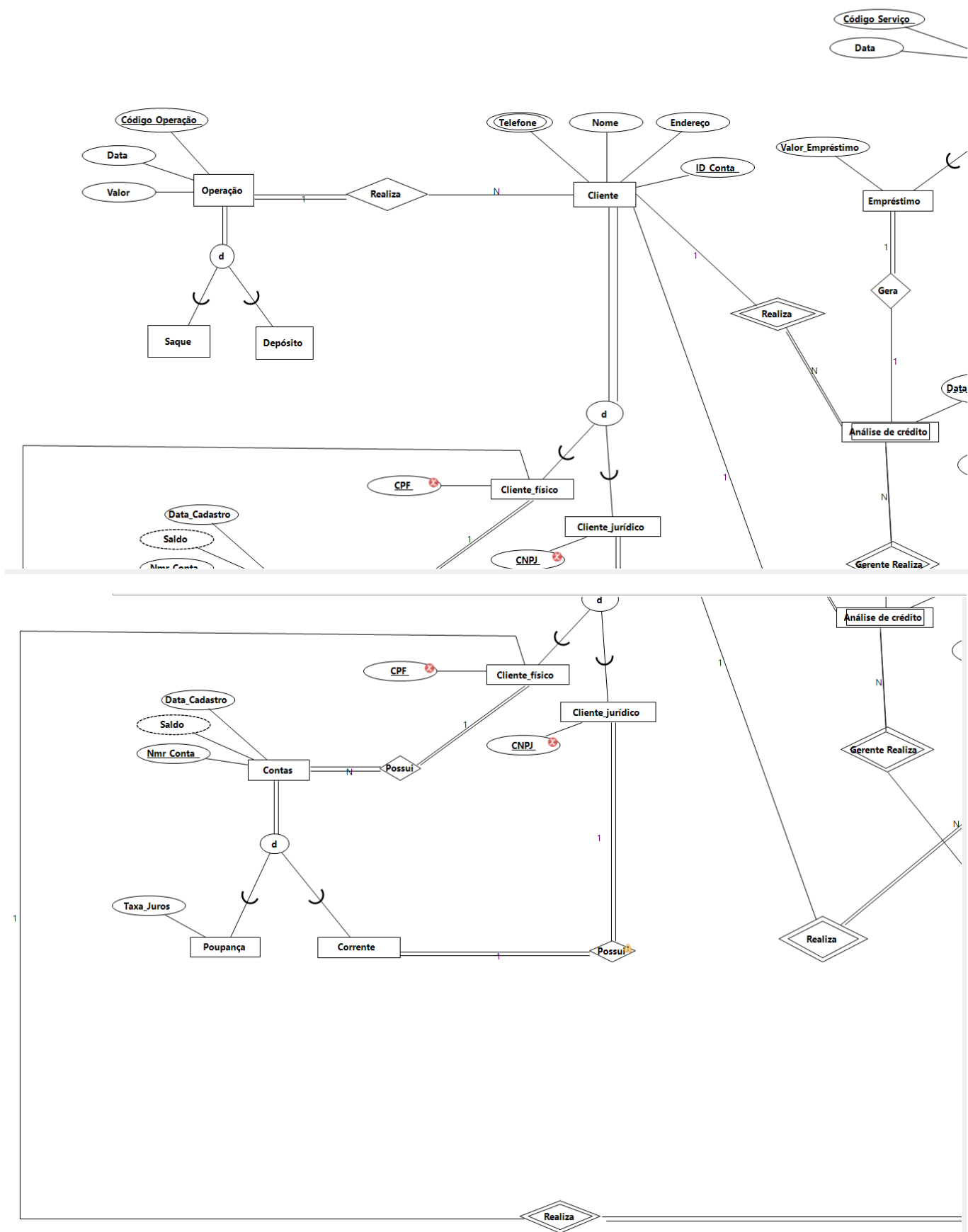
Também é constituído de agências, a qual possui como atributos, número da agência, endereço e telefone. Também possui serviços que tem como atributo código do serviço, data, e sempre são do tipo empréstimo, com valor do empréstimo, ou seguro com valor mensal. Também possui funcionário, que tem como atributos, cpf, nome e salário, são sempre do tipo gerente com bonificação, ou atendente com comissão, ou consultor de investimento com comissão. E por último temos a operação que tem como atributos, código da operação data, valor, sempre são do tipo saque ou depósito.

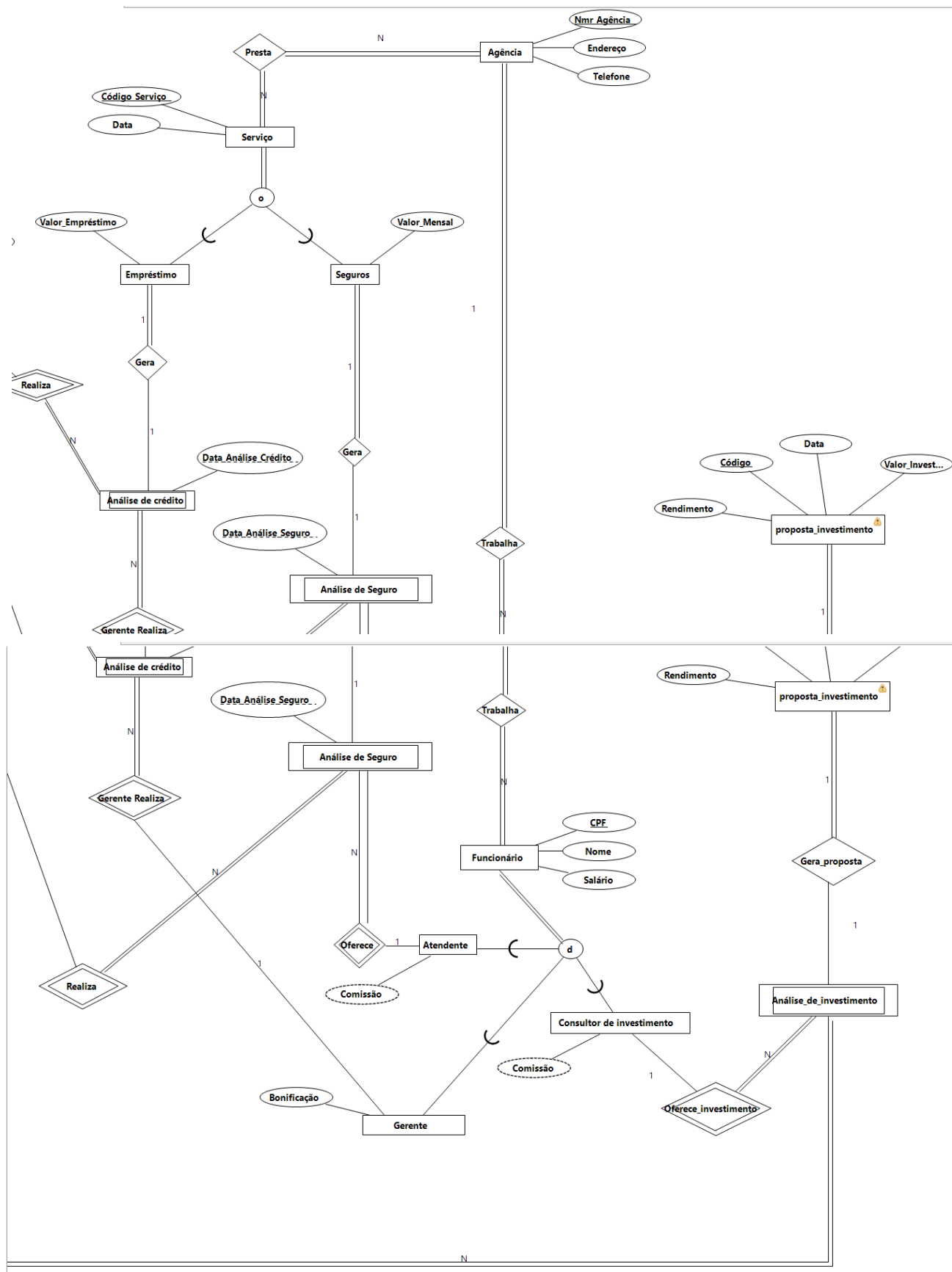
Cliente pode realizar nenhuma ou várias operações, operação sempre é realizada por um cliente. Cliente físico sempre possui 1 ou mais contas, cliente jurídico sempre possui uma conta-corrente. O cliente realiza análise de crédito com um gerente a qual gera um empréstimo, podendo fazer mais de uma vez em datas diferentes.

O cliente realiza análise de seguro com um atendente a qual gera um seguro, podendo fazer mais de uma vez em datas diferentes. Cliente físico realiza análise de investimento com um analista de investimento que gera um investimento, podendo fazer mais de uma vez em datas diferentes. Funcionário trabalha em apenas uma agência. A Agência presta serviço através de seus funcionários que são do tipo: gerente, analista e atendente.

I – Modelo Entidade

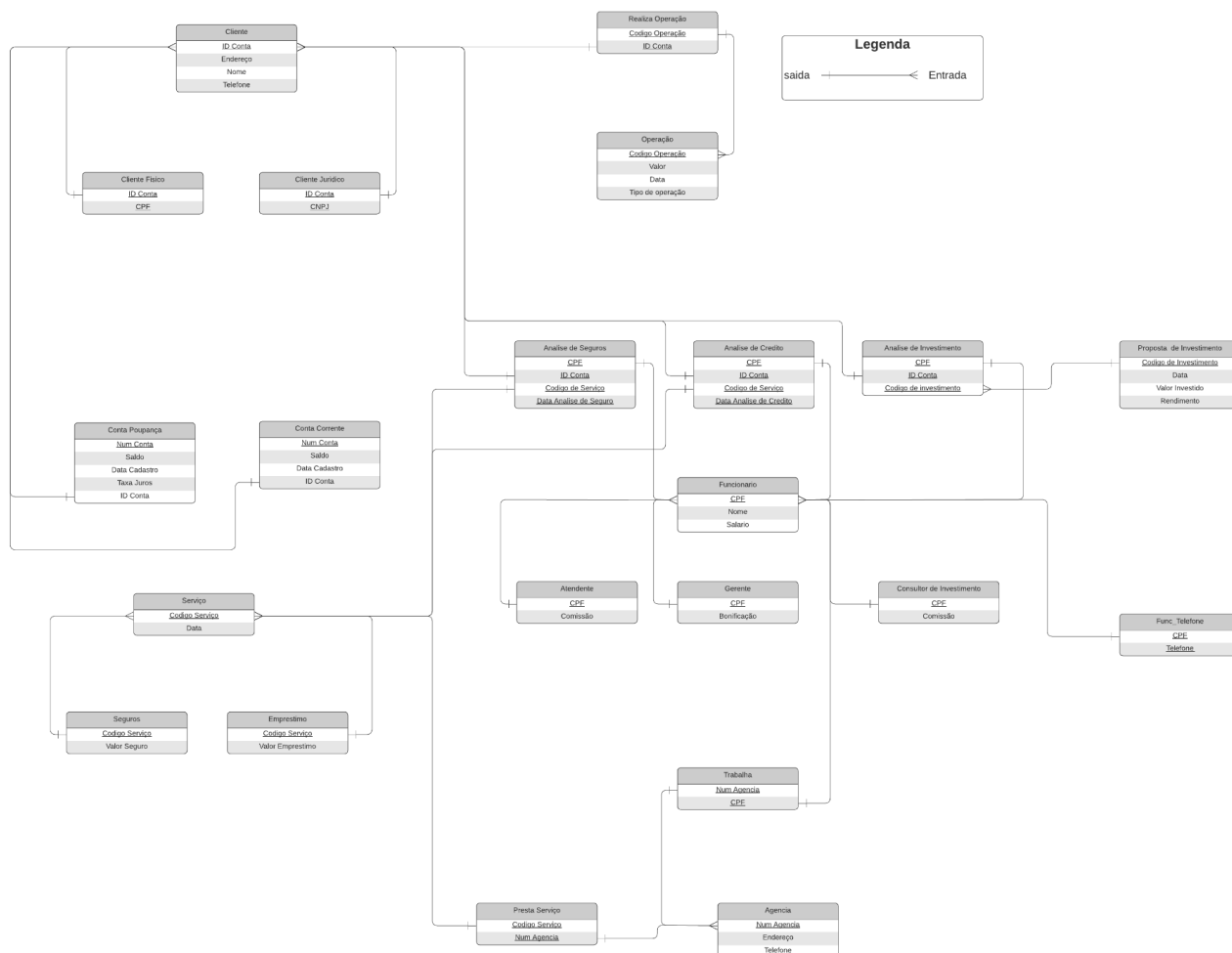
O modelo entidade a partir do mini mundo proposto foi criado pelo software EERCASE, e sera apresentado da maneira a seguir:





III – Modelo Relacional

O modelo relacional foi desenvolvido a partir do modelo entidade relacionamento, seguindo as regras de Mapeamento ER-Relacional apresentadas no curso, e normalizadas segundo a 3 forma normal, a criação do diagrama foi utilizando a plataforma digital lucidchart, sendo apresentado a seguir:



IV – Esquema em SQL

```
create table cliente (  
    id_conta integer primary key,  
    endereco character varying(60),  
    nome character varying(30) not null,  
    telefone character(14)  
);  
  
create table cliente_fisico (  
    id_conta integer not null,  
    cpf character(9) not null,  
    unique(cpf), --> Chave Secundaria  
    foreign key (id_conta) references cliente(id_conta) on delete set null on update cascade  
);  
  
create table cliente_juridico (  
    id_conta integer not null,  
    cnpj character(9) not null,  
    unique(cnpj), --> Chave Secundaria  
    foreign key (id_conta) references cliente(id_conta) on delete set null on update cascade  
);  
  
create table conta_poupanca (  
    num_conta integer not null,  
    saldo numeric(10,2),  
    data_cadastro date,  
    taxa_juros numeric(10,2),  
    id_conta integer not null,  
    primary key (num_conta),  
    foreign key (id_conta) references cliente(id_conta) on delete set null on update cascade  
);  
  
create table conta_corrente (  
    num_conta integer not null,  
    saldo numeric(10,2),  
    data_cadastro date,  
    id_conta integer not null,  
    primary key (num_conta),  
    foreign key (id_conta) references cliente(id_conta) on delete set null on update cascade  
);  
  
create table operação (  
    codigo_operação integer primary key,  
    valor numeric(10,2),  
    data_operação date,  
    tipo_operação character (3) not null  
    /*      saq ou dep      */  
);  
  
create table realiza_operação (  
    codigo_operação integer not null references operação(codigo_operação) on delete cascade on update  
cascade,  
    id_conta integer not null references cliente(id_conta) on delete cascade on update cascade,  
    primary key (codigo_operação, id_conta)  
);  
  
create table funcionario (  
    id_funcionario integer primary key,  
    nome character varying(30) not null,  
    data_nascimento date,  
    data_admissao date,  
    cargo character varying(30) not null,  
    salario numeric(10,2),  
    foreign key (id_funcionario) references funcionario(id_funcionario) on delete set null on update cascade  
);
```

```
nome character varying(30) not null,
cpf character(9) not null,
salario numeric(10,2),
primary key (cpf)
);

create table atendente (
    cpf character(9) not null,
    comissão numeric(10,2),
    foreign key (cpf) references funcionario(cpf) on delete cascade on update cascade
);

create table gerente (
    cpf character(9) not null,
    bonificação numeric(10,2),
    foreign key (cpf) references funcionario(cpf) on delete cascade on update cascade
);

create table consultor_investimento (
    cpf character(9) not null,
    comissão numeric(10,2),
    foreign key (cpf) references funcionario(cpf) on delete cascade on update cascade
);

create table serviço (
    codigo_serviço integer not null,
    data_serviço date not null,
    primary key (codigo_serviço)
);

create table seguros (
    codigo_serviço integer not null,
    valor_seguro numeric(10,2),
    foreign key (codigo_serviço) references serviço(codigo_serviço) on delete cascade on update cascade
);

create table emprestimo (
    codigo_serviço integer not null,
    valor_emprestimo numeric(10,2),
    foreign key (codigo_serviço) references serviço(codigo_serviço) on delete cascade on update cascade
);

create table proposta_de_investimento (
    codigo_investimento integer not null,
    data_investimento date not null,
    valor_investido numeric(10,2),
    rendimento numeric(10,2),
    primary key (codigo_investimento)
);

create table analise_seguros (
    cpf character(9) not null references funcionario(cpf) on delete cascade on update cascade,
    id_conta integer not null references cliente(id_conta) on delete cascade on update cascade,
    codigo_serviço integer not null references serviço(codigo_serviço) on delete cascade on update cascade,
    data_analise date not null,
    primary key (cpf, id_conta, codigo_serviço, data_analise)
);

create table analise_credito (
```

```
cpf character(9) not null references funcionario(cpf) on delete cascade on update cascade,
id_conta integer not null references cliente(id_conta) on delete cascade on update cascade,
codigo_servico integer not null references servico(codigo_servico) on delete cascade on update cascade,
data_analise date not null,
primary key (cpf, id_conta, codigo_servico, data_analise)
);

create table analise_investimento (
    cpf character(9) not null references funcionario(cpf) on delete cascade on update cascade,
    id_conta integer not null references cliente(id_conta) on delete cascade on update cascade,
    codigo_investimento integer not null references proposta_de_investimento(codigo_investimento) on delete
cascade on update cascade,
    primary key(cpf, id_conta, codigo_investimento)
);

create table agencia (
    num_agencia integer primary key,
    endereco character varying(60),
    telefone character(14)
);

create table trabalha_em (
    num_agencia integer not null references agencia(num_agencia) on delete cascade on update cascade,
    cpf character(9) not null references funcionario(cpf) on delete cascade on update cascade,
    primary key(cpf, num_agencia)
);

create table presta_servico (
    num_agencia integer not null references agencia(num_agencia) on delete cascade on update cascade,
    codigo_servico integer not null references servico(codigo_servico) on delete cascade on update cascade,
    primary key(num_agencia, codigo_servico)
);

create table func_telefone (
    telefone character(14),
    cpf character(9) not null references funcionario(cpf) on delete cascade on update cascade,
    primary key(telefone,cpf)
);
```

V – Inserções em SQL

```
INSERT INTO cliente VALUES (67, 'Rua dos laranjais', 'Matheus', '(62)988884444');
INSERT INTO cliente VALUES (68, 'Rua do Governador', 'Luca', '(62)98000-5555');
INSERT INTO cliente VALUES (69, 'Rua do aviador', 'Keven', '(62)94445-5555');
INSERT INTO cliente VALUES (66, 'Alameda Principal', 'João', '(62)98445-5555');
INSERT INTO cliente VALUES (65, 'Rua do amanhecer', 'Isabela', '(62)98985-5555');
INSERT INTO cliente VALUES (62, 'Rua H-76', 'Raquel', '(62)97745-5555');
INSERT INTO cliente VALUES (56, 'Rua do anoitecer', 'Isabel', '(62)93335-5555');
INSERT INTO cliente VALUES (55, 'Rua H-90', 'Ana Laura', '(62)90045-5555');
INSERT INTO cliente VALUES (51, 'Av V5', 'Amanda', '(62)96665-5555');
INSERT INTO cliente VALUES (58, 'Rua JK', 'Jonas', '(62)94445-7777');
INSERT INTO cliente VALUES (30, 'Rua dos lírios', 'Arthur', '(62)93995-5555');
INSERT INTO cliente VALUES (31, 'Rua H-99', 'Angêla', '(62)92045-5555');
INSERT INTO cliente VALUES (32, 'Av V8', 'Vanessa', '(62)96665-5555');
INSERT INTO cliente VALUES (33, 'Rua dona iris', 'Marta', '(62)94445-7337');
INSERT INTO cliente VALUES (20, 'Rua do Marechal', 'Leonardo', '(62)93995-5225');
INSERT INTO cliente VALUES (21, 'Rua H-09', 'Sara', '(62)92945-5215');
INSERT INTO cliente VALUES (22, 'Av V8', 'Vanessa', '(62)93365-5555');
```

```
INSERT INTO cliente VALUES (23, 'Rua dona iris', 'Marta', '(62)98845-7337');
```

```
INSERT INTO cliente_fisico VALUES (67, '111111111');  
INSERT INTO cliente_fisico VALUES (68, '222222222');  
INSERT INTO cliente_fisico VALUES (69, '333333333');  
INSERT INTO cliente_fisico VALUES (66, '444444444');  
INSERT INTO cliente_fisico VALUES (65, '555555555');  
INSERT INTO cliente_fisico VALUES (62, '666666666');  
INSERT INTO cliente_fisico VALUES (56, '777777777');  
INSERT INTO cliente_fisico VALUES (55, '888888888');  
INSERT INTO cliente_fisico VALUES (51, '999999999');  
INSERT INTO cliente_fisico VALUES (58, '400000000');
```

```
INSERT INTO cliente_juridico VALUES (30, '999995555');  
INSERT INTO cliente_juridico VALUES (31, '920045555');  
INSERT INTO cliente_juridico VALUES (32, '966665555');  
INSERT INTO cliente_juridico VALUES (33, '944457337');  
INSERT INTO cliente_juridico VALUES (20, '939985225');  
INSERT INTO cliente_juridico VALUES (21, '929465215');  
INSERT INTO cliente_juridico VALUES (22, '933675555');  
INSERT INTO cliente_juridico VALUES (23, '988437337');
```

```
INSERT INTO conta_poupanca VALUES (155, 100000.00, '2000-12-20', 0.01,67);  
INSERT INTO conta_poupanca VALUES (120, 10000.00, '2020-10-20', 0.01,68);  
INSERT INTO conta_poupanca VALUES (125, 1000.00, '2005-12-10', 0.01,69);  
INSERT INTO conta_poupanca VALUES (225, 20000.00, '2015-06-10', 0.01,66);  
INSERT INTO conta_poupanca VALUES (245, 20000.00, '2011-07-25', 0.01,65);  
INSERT INTO conta_poupanca VALUES (195, 2050.00, '2011-08-11', 0.01,62);  
INSERT INTO conta_poupanca VALUES (198, 20500.00, '2012-09-05', 0.01,56);  
INSERT INTO conta_poupanca VALUES (188, 2050.00, '2011-08-11', 0.01,55);  
INSERT INTO conta_poupanca VALUES (128, 20500.00, '2012-11-05', 0.01,51);
```

```
INSERT INTO conta_corrente VALUES (350, 100000.00, '2000-12-20',30);  
INSERT INTO conta_corrente VALUES (320, 10000.00, '2020-10-20',31);  
INSERT INTO conta_corrente VALUES (325, 1000.00, '2005-12-10',32);  
INSERT INTO conta_corrente VALUES (425, 20000.00,'2015-06-10',33);  
INSERT INTO conta_corrente VALUES (345, 20000.00, '2011-07-25',20);  
INSERT INTO conta_corrente VALUES (395, 2050.00, '2011-08-11',21);  
INSERT INTO conta_corrente VALUES (445, 2800.00, '2012-09-05',22);  
INSERT INTO conta_corrente VALUES (455, 28500.00, '2022-09-15',23);  
INSERT INTO conta_corrente VALUES (388, 2380.00, '2012-09-28',55);  
INSERT INTO conta_corrente VALUES (385, 8870.00, '2017-01-18',51);  
INSERT INTO conta_corrente VALUES (311, 10870.00, '2015-01-18',58);
```

```
INSERT INTO operação VALUES (292, 500.00, '2023-01-01','saq');  
INSERT INTO operação VALUES (293, 250.00, '2023-01-02','dep');  
INSERT INTO operação VALUES (294, 850.00, '2023-01-03','dep');  
INSERT INTO operação VALUES (295, 580.00, '2023-01-04','saq');  
INSERT INTO operação VALUES (296, 280.00, '2023-01-05','dep');  
INSERT INTO operação VALUES (297, 882.00, '2023-01-06','dep');  
INSERT INTO operação VALUES (298, 500.00, '2023-01-07','saq');  
INSERT INTO operação VALUES (299, 250.00, '2023-01-08','dep');  
INSERT INTO operação VALUES (300, 850.00, '2023-01-09','dep');  
INSERT INTO operação VALUES (301, 500.00, '2023-01-10','saq');  
INSERT INTO operação VALUES (302, 250.00, '2023-01-11','dep');  
INSERT INTO operação VALUES (303, 850.00, '2023-01-12','dep');  
INSERT INTO operação VALUES (304, 580.00, '2023-01-14','saq');  
INSERT INTO operação VALUES (305, 280.00, '2023-01-15','dep');  
INSERT INTO operação VALUES (306, 882.00, '2023-01-16','dep');
```



```
INSERT INTO operação VALUES (307, 500.00, '2023-01-17','saq');
INSERT INTO operação VALUES (308, 250.00, '2023-01-18','dep');
INSERT INTO operação VALUES (309, 850.00, '2023-01-19','dep');
```

```
INSERT INTO realiza_operação VALUES (292,67);
INSERT INTO realiza_operação VALUES (293,68);
INSERT INTO realiza_operação VALUES (294,69);
INSERT INTO realiza_operação VALUES (295,66);
INSERT INTO realiza_operação VALUES (296,65);
INSERT INTO realiza_operação VALUES (297,62);
INSERT INTO realiza_operação VALUES (298,56);
INSERT INTO realiza_operação VALUES (299,55);
INSERT INTO realiza_operação VALUES (300,51);
INSERT INTO realiza_operação VALUES (301,58);
INSERT INTO realiza_operação VALUES (302,30);
INSERT INTO realiza_operação VALUES (303,31);
INSERT INTO realiza_operação VALUES (304,32);
INSERT INTO realiza_operação VALUES (305,33);
INSERT INTO realiza_operação VALUES (306,20);
INSERT INTO realiza_operação VALUES (307,21);
INSERT INTO realiza_operação VALUES (308,22);
INSERT INTO realiza_operação VALUES (309,23);
```

```
INSERT INTO funcionario VALUES ('Ana', '123456789', 5000.00);
INSERT INTO funcionario VALUES ('Ane', '123452333', 6000.00);
INSERT INTO funcionario VALUES ('Analise', '123450000', 5500.00);
INSERT INTO funcionario VALUES ('Cris', '123451111', 5250.00);
INSERT INTO funcionario VALUES ('Anabele', '123452222', 6000.00);
INSERT INTO funcionario VALUES ('Carl', '123453333', 7000.00);
INSERT INTO funcionario VALUES ('Jonh', '123454444', 5000.00);
INSERT INTO funcionario VALUES ('Guilherme', '123455555', 5500.00);
INSERT INTO funcionario VALUES ('João', '123456666', 8000.00);
INSERT INTO funcionario VALUES ('Madu', '123457777', 8000.00);
INSERT INTO funcionario VALUES ('Flávia', '123458888', 9000.00);
INSERT INTO funcionario VALUES ('Alice', '123459999', 7000.00);
INSERT INTO funcionario VALUES ('Amanda', '123451234', 5000.00);
INSERT INTO funcionario VALUES ('Lucas', '123451235', 20000.00);
INSERT INTO funcionario VALUES ('Carl', '123451236', 20000.00);
INSERT INTO funcionario VALUES ('Livia', '123451237', 20000.00);
```

```
INSERT INTO atendente VALUES ('123456789', 0.1);
INSERT INTO atendente VALUES ('123452333', 0.1);
INSERT INTO atendente VALUES ('123450000', 0.1);
INSERT INTO atendente VALUES ('123451111', 0.1);
INSERT INTO atendente VALUES ('123452222', 0.1);
INSERT INTO atendente VALUES ('123453333', 0.1);
INSERT INTO atendente VALUES ('123454444', 0.1);
INSERT INTO atendente VALUES ('123455555', 0.1);
```

```
INSERT INTO consultor_investimento VALUES('123456666',0.15);
INSERT INTO consultor_investimento VALUES('123457777',0.15);
INSERT INTO consultor_investimento VALUES('123458888',0.15);
INSERT INTO consultor_investimento VALUES('123459999',0.15);
INSERT INTO consultor_investimento VALUES('123451234',0.15);
```

```
INSERT INTO gerente VALUES ('123451235',0.12);
INSERT INTO gerente VALUES ('123451236',0.12);
INSERT INTO gerente VALUES ('123451237',0.12);
```

```
INSERT INTO serviço VALUES (1, '2023-01-02');
INSERT INTO serviço VALUES (2, '2023-01-03');
INSERT INTO serviço VALUES (3, '2023-01-04');
INSERT INTO serviço VALUES (4, '2023-01-05');
INSERT INTO serviço VALUES (5, '2023-01-06');
INSERT INTO serviço VALUES (6, '2023-01-07');
INSERT INTO serviço VALUES (7, '2023-01-08');
INSERT INTO serviço VALUES (8, '2023-01-02');
INSERT INTO serviço VALUES (9, '2023-01-03');
INSERT INTO serviço VALUES (10, '2023-01-04');
INSERT INTO serviço VALUES (11, '2023-01-05');
INSERT INTO serviço VALUES (12, '2023-01-06');
INSERT INTO serviço VALUES (13, '2023-01-07');
INSERT INTO serviço VALUES (14, '2023-01-08');
INSERT INTO serviço VALUES (15, '2023-01-02');
INSERT INTO serviço VALUES (16, '2023-01-02');
INSERT INTO serviço VALUES (17, '2023-01-03');
INSERT INTO serviço VALUES (18, '2023-01-04');
INSERT INTO serviço VALUES (19, '2023-01-05');
INSERT INTO serviço VALUES (20, '2023-01-06');
```

```
INSERT INTO seguros VALUES (1,2000.00);
INSERT INTO seguros VALUES (2,3000.00);
INSERT INTO seguros VALUES (3,4000.00);
INSERT INTO seguros VALUES (4,5000.00);
INSERT INTO seguros VALUES (5,6000.00);
INSERT INTO seguros VALUES (16,1000.00);
INSERT INTO seguros VALUES (17,1500.00);
INSERT INTO seguros VALUES (18,4500.00);
INSERT INTO seguros VALUES (19,5200.00);
INSERT INTO seguros VALUES (20,5250.00);
```

```
INSERT INTO emprestimo VALUES (6,2000.00);
INSERT INTO emprestimo VALUES (7,3000.00);
INSERT INTO emprestimo VALUES (8,4000.00);
INSERT INTO emprestimo VALUES (9,5000.00);
INSERT INTO emprestimo VALUES (10,6000.00);
INSERT INTO emprestimo VALUES (11,8000.00);
INSERT INTO emprestimo VALUES (12,1000.00);
INSERT INTO emprestimo VALUES (13,2000.00);
INSERT INTO emprestimo VALUES (14,1250.00);
INSERT INTO emprestimo VALUES (15,1350.00);
```

```
INSERT INTO proposta_de_investimento VALUES (1,'2023-01-01',500.00, 0.1);
INSERT INTO proposta_de_investimento VALUES (2,'2023-01-02',800.00, 0.12);
INSERT INTO proposta_de_investimento VALUES (3,'2023-01-03',600.00, 0.1);
INSERT INTO proposta_de_investimento VALUES (4,'2023-01-04',700.00, 0.14);
INSERT INTO proposta_de_investimento VALUES (5,'2023-01-05',550.00, 0.15);
INSERT INTO proposta_de_investimento VALUES (6,'2023-01-06',800.00, 0.13);
INSERT INTO proposta_de_investimento VALUES (7,'2023-01-07',500.00, 0.17);
INSERT INTO proposta_de_investimento VALUES (8,'2023-01-09',1000.00, 0.14);
INSERT INTO proposta_de_investimento VALUES (9,'2023-01-10',500.00, 0.1);
INSERT INTO proposta_de_investimento VALUES (10,'2023-01-11',700.00, 0.17);
```

```
INSERT INTO analise_seguros VALUES ('123456789',67,1, '2023-02-01');
INSERT INTO analise_seguros VALUES ('123452333',68,2,'2023-02-02');
INSERT INTO analise_seguros VALUES ('123452333',69,3,'2023-02-03');
INSERT INTO analise_seguros VALUES ('123450000',66,4,'2023-02-04');
```

```
INSERT INTO analise_seguros VALUES ('123451111',65,5,'2023-02-05');
INSERT INTO analise_seguros VALUES ('123456789',58,16,'2023-02-06');
INSERT INTO analise_seguros VALUES ('123452333',51,17,'2023-02-07');
INSERT INTO analise_seguros VALUES ('123452333',30,18,'2023-02-08');
INSERT INTO analise_seguros VALUES ('123450000',21,19,'2023-02-09');
INSERT INTO analise_seguros VALUES ('123451111',23,20,'2023-02-10');
```

```
INSERT INTO analise_credito VALUES ('123451235',62,6,'2023-02-01');
INSERT INTO analise_credito VALUES ('123451236',56,7,'2023-02-02');
INSERT INTO analise_credito VALUES ('123451237',55,8,'2023-02-03');
INSERT INTO analise_credito VALUES ('123451235',31,9,'2023-02-04');
INSERT INTO analise_credito VALUES ('123451237',32,10,'2023-02-05');
INSERT INTO analise_credito VALUES ('123451236',33,11,'2023-02-06');
INSERT INTO analise_credito VALUES ('123451235',51,12,'2023-02-07');
INSERT INTO analise_credito VALUES ('123451236',30,13,'2023-02-08');
INSERT INTO analise_credito VALUES ('123451237',21,14,'2023-02-09');
INSERT INTO analise_credito VALUES ('123451237',23,15,'2023-02-10');
```

```
INSERT INTO analise_investimento VALUES ('123456666',67,1);
INSERT INTO analise_investimento VALUES ('123457777',68,2);
INSERT INTO analise_investimento VALUES ('123458888',69,3);
INSERT INTO analise_investimento VALUES ('123459999',66,4);
INSERT INTO analise_investimento VALUES ('123451234',65,5);
INSERT INTO analise_investimento VALUES ('123456666',62,6);
INSERT INTO analise_investimento VALUES ('123457777',56,7);
INSERT INTO analise_investimento VALUES ('123458888',55,8);
INSERT INTO analise_investimento VALUES ('123459999',51,9);
INSERT INTO analise_investimento VALUES ('123451234',58,10);
```

```
INSERT INTO agencia VALUES (1, 'Rua Central, segundo andar, shopping Central', '08004555');
INSERT INTO agencia VALUES (2, 'Rua Oeste, segundo andar, shopping Oeste', '08004556');
INSERT INTO agencia VALUES (3, 'Rua Leste, segundo andar, shopping leste', '08004557');
```

```
INSERT INTO trabalha_em VALUES (1,'123456789');
INSERT INTO trabalha_em VALUES (1,'123452333');
INSERT INTO trabalha_em VALUES (1,'123450000');
INSERT INTO trabalha_em VALUES (1,'123456666');
INSERT INTO trabalha_em VALUES (1,'123457777');
INSERT INTO trabalha_em VALUES (1,'123451235');
INSERT INTO trabalha_em VALUES (2,'123451111');
INSERT INTO trabalha_em VALUES (2,'123452222');
INSERT INTO trabalha_em VALUES (2,'123453333');
INSERT INTO trabalha_em VALUES (2,'123458888');
INSERT INTO trabalha_em VALUES (2,'123459999');
INSERT INTO trabalha_em VALUES (2,'123451236');
INSERT INTO trabalha_em VALUES (3,'123454444');
INSERT INTO trabalha_em VALUES (3,'123455555');
INSERT INTO trabalha_em VALUES (3,'123451234');
INSERT INTO trabalha_em VALUES (3,'123451237');
```

```
INSERT INTO presta_servico VALUES (1,1);
INSERT INTO presta_servico VALUES (1,2);
INSERT INTO presta_servico VALUES (1,3);
INSERT INTO presta_servico VALUES (1,4);
INSERT INTO presta_servico VALUES (2,5);
INSERT INTO presta_servico VALUES (1,16);
INSERT INTO presta_servico VALUES (1,17);
INSERT INTO presta_servico VALUES (1,18);
```

```
INSERT INTO presta_serviço VALUES (1,19);
INSERT INTO presta_serviço VALUES (2,20);
INSERT INTO presta_serviço VALUES (1,6);
INSERT INTO presta_serviço VALUES (2,7);
INSERT INTO presta_serviço VALUES (3,8);
INSERT INTO presta_serviço VALUES (1,9);
INSERT INTO presta_serviço VALUES (2,10);
INSERT INTO presta_serviço VALUES (3,11);
INSERT INTO presta_serviço VALUES (1,12);
INSERT INTO presta_serviço VALUES (2,13);
INSERT INTO presta_serviço VALUES (3,14);
INSERT INTO presta_serviço VALUES (1,15);
```

```
INSERT INTO func_telefone VALUES ('6211111111','123456789');
INSERT INTO func_telefone VALUES ('6222222222','123452333');
INSERT INTO func_telefone VALUES ('6233333333','123450000');
INSERT INTO func_telefone VALUES ('6244444444','123451111');
INSERT INTO func_telefone VALUES ('6255555555','123452222');
INSERT INTO func_telefone VALUES ('6266666666','123453333');
INSERT INTO func_telefone VALUES ('6277777777','123454444');
INSERT INTO func_telefone VALUES ('6288888888','123455555');
INSERT INTO func_telefone VALUES ('6299999999','123456666');
INSERT INTO func_telefone VALUES ('6284444444','123457777');
INSERT INTO func_telefone VALUES ('6285555555','123458888');
INSERT INTO func_telefone VALUES ('6286666666','123459999');
INSERT INTO func_telefone VALUES ('6287777777','123451234');
INSERT INTO func_telefone VALUES ('6288888888','123451235');
INSERT INTO func_telefone VALUES ('6289999999','123451236');
INSERT INTO func_telefone VALUES ('6280000000','123451237');
```

```
/*
select * from cliente order by id_conta;
select * from cliente_fisico;
select * from cliente_juridico;
select * from conta_poupanca;
select * from conta_corrente;
select * from operação;
select * from realiza_operação;
select * from funcionario;
select * from atendente;
select * from gerente;
select * from consultor_investimento;
select * from serviço;
select * from seguros;
select * from emprestimo;
select * from proposta_de_investimento;
select * from analise_seguros;
select * from analise_credito;
select * from analise_investimento;
select * from agencia;
select * from trabalha_em;
select * from presta_serviço;
select * from func_telefone;
*/
```

VI – Consultas em SQL

-- Retorne todos os atributos de cliente (Consulta Teste)

```
select *  
from cliente;
```

-- Selecione todas as contas correntes e contas poupanças que tenham saldo maior ou igual de R\$10.000 e retorne o

-- nome do cliente, id da conta e o saldo. (Union - 1)

```
select cli.nome, cp.id_conta, cp.saldo  
from conta_poupanca as cp inner join cliente as cli on cli.id_conta = cp.id_conta  
where cp.saldo >= 10000  
union  
      (select cli.nome, cc.id_conta, cc.saldo  
        from conta_corrente as cc inner join cliente as cli on cli.id_conta = cc.id_conta  
        where cc.saldo >= 10000)  
order by id_conta asc;
```

-- Retorne o nome e cpf caso o cliente tenha uma conta poupança e uma conta corrente.) (Intersect - 2)

```
select nome, cpf, id_conta  
from cliente natural join cliente_fisico natural join (select id_conta  
  
from (select id_conta  
  
      from conta_corrente  
  
      intersect  
  
      select id_conta  
  
      from conta_poupanca) as subquery_1) as subquery_2;
```

-- Retorne nome, cpf, salario e comissão dos atendentes (Aninhadas pela cláusula FROM - 1)

```
select nome, cpf, salario, comissão  
from (select *  
      from atendente  
      ) as subquery_1  
natural join  
(select *  
  from funcionario  
  --where salario >= 500  
  ) as subquery_2;
```

-- Retorne o nome, endereço e saldo da conta poupança do cliente com o maior saldo que mora especificamente em alguma alameda (Aninhadas pela cláusula FROM - 2)

```
select cliente.nome, cliente.endereco, conta_poupanca.saldo  
from cliente  
inner join conta_poupanca on cliente.id_conta = conta_poupanca.id_conta  
where conta_poupanca.saldo = (  
  select max(saldo)  
  from conta_poupanca  
  where conta_poupanca.id_conta = cliente.id_conta  
) and endereco like 'Alameda%';
```

-- Retorne o nome de todos os funcionarios que possuem salario = 20.000, 5.000 e 8.000. (in - 1)

```
select nome, salario
```

from funcionario where salario in (20000, 5000, 8000)
order by salario asc;

-- Retorne todos os atributos em que os funcionarios recebem pelo menos mais que R\$8.000 da tabela funcionario. (some - 2)

```
select *  
from funcionario where salario > some (select 8000);
```

-- Retorne os atributos dos funcionario que não ganham mais que R\$6.000 (exists - 3)

```
select *  
from funcionario f1 where not exists (select 1  
                                     from funcionario f2 where f1.nome =  
f2.nome and f2.salario > 6000);
```

-- Retorne o rendimento da tabela proposta de investimento quando há valor investido entre 500 a 700 (any - 4)

```
select rendimento  
from proposta_de_investimento  
where valor_investido = any (select valor_investido from proposta_de_investimento where valor_investido between  
500 and 700);
```

-- Selecione o nome e endereço dos clientes que possuem conta poupança com saldo maior que R\$5000 e taxa de juros acima de 1%. (Consultas aninhadas e correlacionadas - 1)

```
select nome, endereco  
from cliente  
where id_conta in (  
    select id_conta  
    from conta_poupanca  
    where saldo > 5000 and taxa_juros = 0.01  
);
```

-- Selecione o nome dos clientes que possuem empréstimos e o valor total dos empréstimos contratados por cada um deles. (Consultas aninhadas e correlacionadas - 2)

```
select c.nome, sum(e.valor_emprestimo) as total_emprestimos  
from cliente c, emprestimo e, servico s, analise_credito ac  
where c.id_conta = ac.id_conta and  
    ac.codigo_servico = s.codigo_servico and  
    s.codigo_servico = e.codigo_servico and  
    e.codigo_servico = ac.codigo_servico  
group by c.nome;
```

-- Retorne quantas contas poupanças existem (Consulta Escalar - 1)

```
select count(*) from conta_poupanca;
```

-- Retorne o saldo da conta numero 311 (Consulta Escalar - 2)

```
select saldo from conta_corrente where num_conta = 311;
```

-- Retorne uma consulta para obter a soma dos valores de todas as operações de saque realizadas no mês de janeiro de 2023 (Consulta Escalar - 3)

```
select sum(valor)  
from operacao where tipo_operacao = 'saq'  
and extract(month from data_operacao) = 1  
and extract(year from data_operacao) = 2023;
```


-- Retorne as informações do nome, telefone e endereço dos clientes que possuem uma conta corrente (consulta envolvendo a operação de junção definida na cláusula FROM - 1)

```
select c.nome, c.telefone, c.endereco  
from cliente c  
join conta_corrente cc on c.id_conta = cc.id_conta;
```

-- Obter informações de todos os serviços e informações de seguro, incluindo serviços que não têm seguro associado. (Consultas envolvendo outer joins - 1)

```
select *  
from serviço  
left outer join seguros on serviço.codigo_serviço = seguros.codigo_serviço;
```

-- Obter o nome de todos os clientes que não possuem nenhum tipo de conta (poupança ou corrente) e também não possuem nenhum registro na tabela

-- de cliente_fisico ou cliente_juridico. (Consultas envolvendo outer joins - 2)

```
select c.nome  
from cliente c  
left join conta_poupanca cp on cp.id_conta = c.id_conta  
left join conta_corrente cc on cc.id_conta = c.id_conta  
left join cliente_fisico cf on cf.id_conta = c.id_conta  
left join cliente_juridico cj on cj.id_conta = c.id_conta  
where cp.num_conta is null and cc.num_conta is null  
and cf.cpf is null and cj.cnpj is null;
```

-- Obter o número de análises de investimento realizadas por cada consultor de investimento. (Consultas envolvendo agrupamentos e agregações - 1)

```
select ai.cpf, count(*) as num_analises  
from analise_investimento ai  
group by ai.cpf;
```

-- Consulta para obter o valor total em empréstimos por agência: (Consultas envolvendo agrupamentos e agregações - 2)

```
select t.num_agencia, sum(e.valor_emprestimo) as valor_total_emprestimo  
from emprestimo e  
inner join serviço s on e.codigo_serviço = s.codigo_serviço  
inner join presta_serviço t on s.codigo_serviço = t.codigo_serviço  
group by t.num_agencia;
```

--Retorne as contas poupança com saldo médio acima de R\$ 500, agrupadas pelo cliente que as possui. (Consulta envolvendo a cláusula HAVING - 1)

```
select cp.id_conta, round(avg(cp.saldo), 2) as saldo_medio  
from conta_poupanca as cp  
group by cp.id_conta  
having avg(cp.saldo) > 500;
```

--Retorne os serviços que possuem valor de seguro acima de R\$ 100, agrupados pelo código do serviço e ordenados em ordem crescente pelo valor. (Consulta envolvendo a cláusula HAVING - 2)

```
select s.codigo_serviço, sum(seg.valor_seguro) as valor_total_seguro  
from serviço as s  
inner join seguros as seg on s.codigo_serviço = seg.codigo_serviço  
group by s.codigo_serviço  
having sum(seg.valor_seguro) > 100
```

order by valor_total_seguro asc;

-- (Operações de inserção, com restrições de integridade - 1)

INSERT INTO cliente VALUES (100, 'Alameda Pedro I', 'jubileu', '(61)988884444');

INSERT INTO cliente_fisico VALUES (100, '321456987');

-- (Operações de inserção, com restrições de integridade - 2)

INSERT INTO funcionario VALUES ('Josepha', '321456987', 5400.00);

INSERT INTO gerente VALUES ('321456987', 0.14);

--(Operações de deleção, com restrições de integridade - 1)

delete from cliente

where id_conta = 100

--(Operações de deleção, com restrições de integridade - 2)

delete from funcionario

where cpf = '321456987'

--(Operações de modificação, com restrições de integridade - 1)

ALTER TABLE analise_investimento

ADD aprovado boolean default null;

--(Operações de modificação, com restrições de integridade - 2)

UPDATE cliente

SET endereco = 'Alameda Jose da Silva'

WHERE id_conta = 100;

--(Operações de modificação, com restrições de integridade - 3)

UPDATE cliente_fisico

SET cpf = '121456989'

WHERE id_conta = 100;

-- Lista de clientes que possuem contas correntes, mostrando o número da conta, o saldo atual e a data de cadastro da conta. (Visão - 1)

create view clientes_com_contas_correntes as

select cc.num_conta, cc.saldo, cc.data_cadastro, c.nome,
case when cf.cpf is not null then cf.cpf else cj.cnpj end as documento

from conta_corrente cc

inner join cliente c on cc.id_conta = c.id_conta

left join cliente_fisico cf on c.id_conta = cf.id_conta

left join cliente_juridico cj on c.id_conta = cj.id_conta;

-- Usada para exibir as contas de poupança com saldos acima da média das contas de poupança. (Visão - 2)

CREATE VIEW poupanca_saldos_acima_media AS

SELECT num_conta, saldo, id_conta

FROM conta_poupanca

WHERE saldo > (SELECT AVG(saldo) FROM conta_poupanca);

VII – Metodologia de Trabalho

O trabalho foi realizado a partir de reuniões virtuais e presenciais no instituto de informática, na qual foram divididos as tarefas da seguinte maneira:

A parte de criação do mini mundo e criação do modelo entidade relação, foi feita em conjunto, via reuniões pelo Google meet.

Keven → Responsável pelas inserções do banco de dados em SQL

Luca → Responsável pelas consultas do banco de dados em SQL

Matheus → Responsável pelo mapeamento ER-Relacional e Criação dos esquemas do banco de dados em SQL