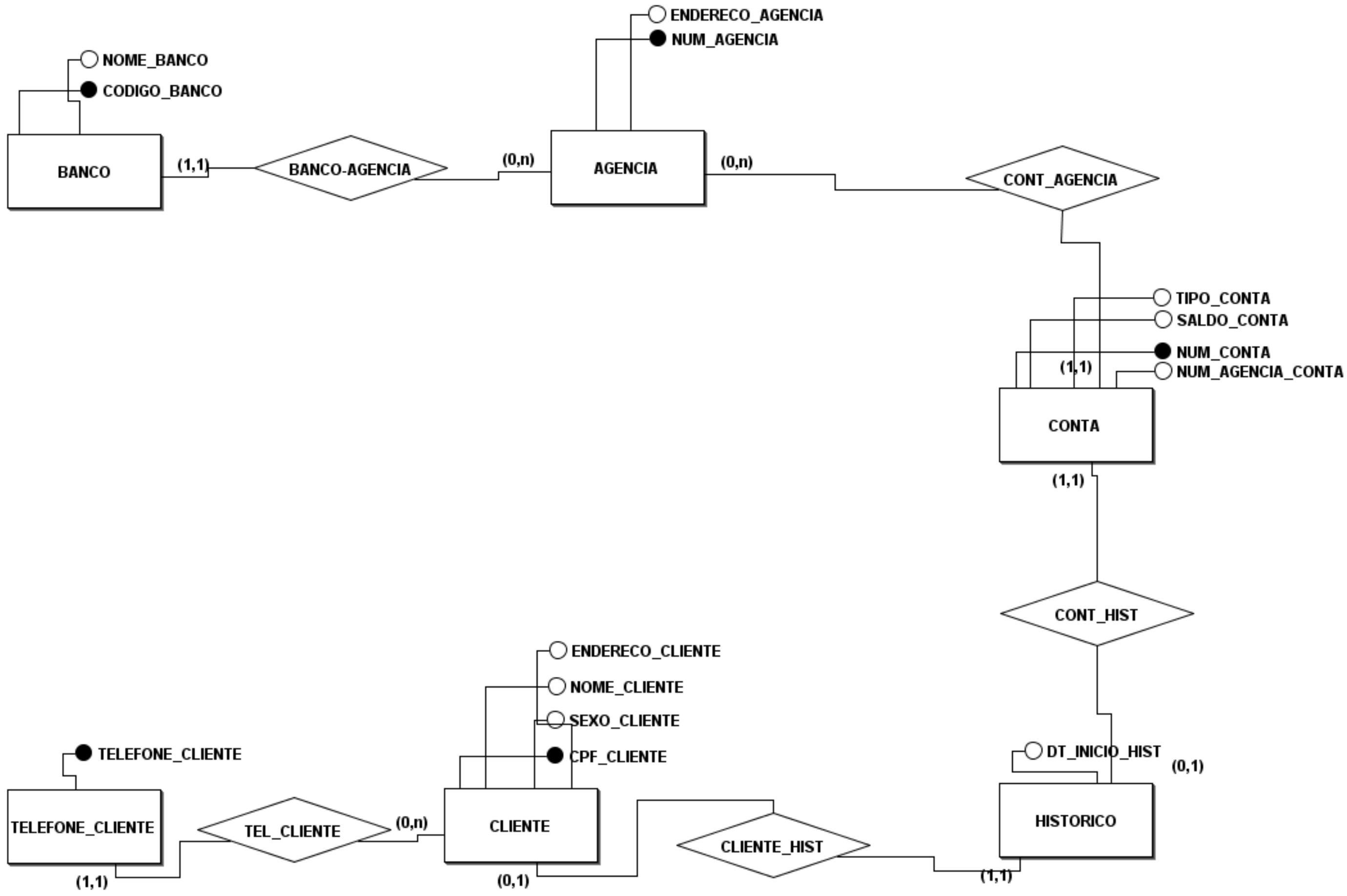
TRABALHO 1 - BANCO DE DADOS 2

ANDERSON SOARES DA SILVA	02150875
MARIA EDUARDA PEDROSO	02150336
MARIANA GONÇALVES RODRIGUES	02151014

PARTE A

Modelo conceitual



Modelo lógico relacional

Banco (**Código**, Nome)

Agência (**Cód_Banco, Número_Agencia**, Endereço)

Cód_Banco referencia Banco

Cliente(**CPF**, Nome, Sexo, Endereço)

Conta(**Número_conta**, Saldo, Tipo_Conta, Num_Agencia)

Num_Agencia referencia Agência.

Histórico(**Cpf_cliente**, **Num_Conta**, Data_Inicio)

Cpf_cliente referencia Cliente

Num_Conta referencia Conta

Telefone_cliente(**Cpf_cli**, **Telefone**)

Cpf_cli referencia Cliente

Construindo o banco de dados

```
create database instituicaoFinanceira;

use instituicaoFinanceira;

-- instituicaoFinanceira.banco

CREATE TABLE
    `banco` (
        `codBanco` BIGINT NOT NULL AUTO_INCREMENT,
        `nome` varchar (150) NOT NULL,
        -- DataCriacao date NOT NULL,
        -- UsuarioCriacao varchar (50) NOT NULL,
        --  DataAlteracao date NOT NULL,
        -- UsuarioAlteracao varchar (50) NOT NULL,
        PRIMARY KEY (`codBanco`)
    );

-- instituicaoFinanceira.cliente

CREATE TABLE
    cliente (
        cpf varchar(14) NOT NULL,
        nome varchar(150) NOT NULL,
        sexo varchar(1) NOT NULL,
        endereco VARCHAR(600) NOT NULL,
        -- DataCriacao date NOT NULL,
        -- UsuarioCriacao varchar (50) NOT NULL,
        --  DataAlteracao date NOT NULL,
        -- UsuarioAlteracao varchar (50) NOT NULL,
        PRIMARY KEY (cpf)
    );

-- instituicaoFinanceira.telefoneCliente

CREATE TABLE
    telefoneCliente (
        telefone varchar(20) NOT NULL,
        cpf varchar(14) NOT NULL,
        -- DataCriacao date NOT NULL,
        -- UsuarioCriacao varchar (50) NOT NULL,
        --  DataAlteracao date NOT NULL,
        -- UsuarioAlteracao varchar (50) NOT NULL,
        PRIMARY KEY (telefone),
        FOREIGN KEY (cpf) REFERENCES cliente (cpf)
    );

-- instituicaoFinanceira.agencia

CREATE TABLE
```

```
    agencia (
        codAgencia serial NOT NULL,
        endereco VARCHAR(600) NOT NULL,
        codBanco bigint NOT NULL,
        -- DataCriacao date NOT NULL,
        -- UsuarioCriacao varchar (50) NOT NULL,
        --  DataAlteracao date NOT NULL,
        -- UsuarioAlteracao varchar (50) NOT NULL,
        PRIMARY KEY (codAgencia),
        FOREIGN KEY (codBanco) REFERENCES banco (codBanco)
    );

-- instituicaoFinanceira.conta

CREATE TABLE
    conta (
        codConta varchar(7) NOT NULL,
        saldo NUMERIC NOT NULL,
        tipoConta int NOT NULL,
        codAgencia bigint(20) NOT NULL,
        -- DataCriacao date NOT NULL,
        -- UsuarioCriacao varchar (50) NOT NULL,
        --  DataAlteracao date NOT NULL,
        -- UsuarioAlteracao varchar (50) NOT NULL,
        PRIMARY KEY (codConta)
    );

-- instituicaoFinanceira.historico

CREATE TABLE
    historico (
        cpf varchar(14) NOT NULL,
        codConta varchar(7) NOT NULL,
        dataInicio DATE NOT NULL,
        -- DataCriacao date NOT NULL,
        -- UsuarioCriacao varchar (50) NOT NULL,
        --  DataAlteracao date NOT NULL,
        -- UsuarioAlteracao varchar (50) NOT NULL,
        PRIMARY KEY (codConta),
        FOREIGN KEY (cpf) REFERENCES cliente(cpf),
        FOREIGN KEY (codConta) REFERENCES conta(codConta)
    );
```

Inserções no banco de dados

```
INSERT INTO banco (codBanco,Nome)
VALUES (1,'Banco do Brasil');

INSERT INTO BANCO (codBanco,Nome)
VALUES (4,'CEF');
```

```
INSERT INTO AGENCIA ( codAgencia, Endereco, codBanco)
VALUES ('0562', 'Rua Joaquim Teixeira Alves, 1555', 4);

INSERT INTO AGENCIA ( codAgencia, Endereco, codBanco)
VALUES ('3153', 'Av. Marcelino Pires, 1960', 1);

INSERT INTO CLIENTE (Cpf, Nome, Sexo, Endereco)
VALUES ('111.222.333-44', 'Jennifer B Souza', 'F', 'Rua Cuiabá, 1050');

INSERT INTO CLIENTE (Cpf, Nome, Sexo, Endereco)
VALUES ('666.777.888-99', 'Caetano k Lima', 'M', 'Rua Ivinhema, 879');

INSERT INTO CLIENTE (Cpf, Nome, Sexo, Endereco)
VALUES ('555.444.777-33', 'Silvia Macedo', 'F', 'Rua Estados Unidos,
735');

INSERT INTO CONTA (codConta, Saldo, tipoConta, codAgencia)
VALUES ('86340-2', '763.05', 2, 3153);

INSERT INTO CONTA (codConta, Saldo, tipoConta, codAgencia)
VALUES ('23584-7', '3879.12', 1, 0562);

INSERT INTO HISTORICO (Cpf, codConta, dataInicio)
VALUES ('111.222.333-44', '23584-7', '17-12-1997');

INSERT INTO HISTORICO (Cpf, codConta, dataInicio)
VALUES ('666.777.888-99', '23584-7', '17-12-1997');

INSERT INTO HISTORICO (Cpf, codConta, dataInicio)
VALUES ('555.444.777-33', '86340-2', '29-11-2010');

INSERT INTO telefoneCliente (cpf, telefone)
VALUES ('111.222.333-44', '(67)3422-7788');

INSERT INTO telefoneCliente (cpf, telefone)
VALUES ('666.777.888-99', '(67)3423-9900');

INSERT INTO telefoneCliente (cpf, telefone)
VALUES ('666.777.888-99', '(67)8121-8833');
```

PARTE B

```
////////////////////////////////////////////////////////////////////////
-- 1
-- select * from uscensus where id=9000;
-- 16:37:41 select * from uscensus where id=9000 LIMIT 0, 1000  1 row(s)
returned    0,031 sec / 0,0000069 sec

-- explain select * from uscensus where id=9000;
-- 16:38:59 explain select * from uscensus where id=9000  1 row(s)
returned    0,00027 sec / 0,0000060 sec
--
+----+-------------+----------+------------+------+---------------+----
--+---------+------+-------+----------+-------------+
-- | id | select_type | table    | partitions | type | possible_keys |
key  | key_len | ref  | rows  | filtered | Extra       |
--
+----+-------------+----------+------------+------+---------------+----
--+---------+------+-------+----------+-------------+
-- |  1 | SIMPLE      | uscensus | NULL       | ALL  | NULL          |
NULL | NULL    | NULL | 32391 |    10.00 | Using where |
--
+----+-------------+----------+------------+------+---------------+----
--+---------+------+-------+----------+-------------+

-- explain analyze select * from uscensus where id=9000;
-- '-> Filter: (uscensus.id = 9000)  (cost=3328.75 rows=3241) (actual
time=6.779..23.467 rows=1 loops=1)\n    -> Table scan on uscensus
(cost=3328.75 rows=32405) (actual time=0.018..21.582 rows=32561
loops=1)\n'
-- O CUSTO FOI DE 0.3328.75

////////////////////////////////////////////////////////////////////////
-- 2
-- ALTER TABLE uscensus add primary key (id);

--  explain select * from uscensus where id=9000;
/*
+----+-------------+----------+------------+-------+---------------+---
------+---------+-------+------+----------+-------+
| id | select_type | table    | partitions | type  | possible_keys |
key     | key_len | ref   | rows | filtered | Extra |
+----+-------------+----------+------------+-------+---------------+---
------+---------+-------+------+----------+-------+
|  1 | SIMPLE      | uscensus | NULL       | const | PRIMARY       |
PRIMARY | 4       | const |    1 |   100.00 | NULL  |
+----+-------------+----------+------------+-------+---------------+---
------+---------+-------+------+----------+-------+
*/
-- '1', 'SIMPLE', 'uscensus', NULL, 'const', 'PRIMARY', 'PRIMARY', '4',
'const', '1', '100.00', NULL
-- 16:53:15 explain select * from uscensus where id=9000  1 row(s)
returned    0,00029 sec / 0,0000060 sec

 -- explain analyze select * from uscensus where id=9000;
 -- '-> Rows fetched before execution  (cost=0.00..0.00 rows=1) (actual
time=0.000..0.000 rows=1 loops=1)\n'

-- O CUSTO FOI DE 0.000
-- Que conclusoes pode-se tirar comparando os custos com indices e sem
indices?
-- O custo sem a chave primária foi muito maior que o custo com a chave
primária.

///////////////////////////////////////////////////////////////////////
-- 3
-- sem indice
-- explain analyze select * from uscensus where education =
'Preschool';
-- '-> Filter: (uscensus.education = \'Preschool\')  (cost=3583.00
rows=3230) (actual time=0.246..19.868 rows=51 loops=1)\n    -> Table
scan on uscensus  (cost=3583.00 rows=32300) (actual time=0.096..18.059
rows=32561 loops=1)\n'

-- CUSTO SEM INDICE = 3583.00

-- com indice
-- CREATE INDEX preescola ON uscensus(education);

 -- explain analyze select * from uscensus where education =
'Preschool';
-- '-> Index lookup on uscensus using preescola
(education=\'Preschool\')  (cost=55.96 rows=51) (actual
time=0.321..0.349 rows=51 loops=1)\n'

-- CUSTO COM INDICE = 55.96

///////////////////////////////////////////////////////////////////////
-- 4
-- sem indice
--  explain analyze select AVG(AGE) from uscensus where Race = 'White'
group by workclass ;
-- '-> Table scan on <temporary>  (actual time=30.134..30.135 rows=9
loops=1)\n    -> Aggregate using temporary table  (actual
time=30.133..30.133 rows=9 loops=1)\n        -> Filter: (uscensus.race
= \'White\')  (cost=3582.06 rows=3230) (actual time=0.126..15.077
rows=27816 loops=1)\n            -> Table scan on uscensus
(cost=3582.06 rows=32300) (actual time=0.121..10.609 rows=32561
loops=1)\n'
-- CUSTO SEM INDICE = 3582.06

-- com indice
-- CREATE INDEX idade ON uscensus(age);
-- explain analyze select AVG(AGE) from uscensus where Race = 'White'
group by workclass ;
-- '-> Table scan on <temporary>  (actual time=22.677..22.677 rows=9
loops=1)\n    -> Aggregate using temporary table  (actual
time=22.676..22.676 rows=9 loops=1)\n        -> Filter: (uscensus.race
= \'White\')  (cost=3582.06 rows=3230) (actual time=0.035..11.469
rows=27816 loops=1)\n            -> Table scan on uscensus
```

```
(cost=3582.06 rows=32300) (actual time=0.033..8.112 rows=32561
loops=1)\n'
-- CUSTO COM INDICE = 3582.06
-- os dois tem a mesma performance devido a media ele precisa passar em
todas as linhas da tabela ele não utiliza o indice secundário

////////////////////////////////////////////////////////////////////////
-- 5
-- sem indice
-- drop index idade on uscensus;
-- explain analyze select AVG(AGE) from uscensus where Race = 'White'
and educationnum>3 and relationship='Own-child' group by workclass ;
-- '-> Table scan on <temporary>  (actual time=26.126..26.127 rows=9
loops=1)\n    -> Aggregate using temporary table  (actual
time=26.125..26.125 rows=9 loops=1)\n        -> Filter:
((uscensus.relationship = \'Own-child\') and (uscensus.race =
\'White\') and (uscensus.educationnum > 3))  (cost=3582.06 rows=108)
(actual time=0.065..22.741 rows=4231 loops=1)\n            -> Table
scan on uscensus  (cost=3582.06 rows=32300) (actual time=0.058..17.216
rows=32561 loops=1)\n'

-- explain analyze select AVG(AGE) from uscensus where Race = 'White'
and educationnum>3 and relationship='Not-in-family' group by workclass
;
-- '-> Table scan on <temporary>  (actual time=23.347..23.348 rows=8
loops=1)\n    -> Aggregate using temporary table  (actual
time=23.346..23.346 rows=8 loops=1)\n        -> Filter:
((uscensus.relationship = \'Not-in-family\') and (uscensus.race =
\'White\') and (uscensus.educationnum > 3))  (cost=3582.06 rows=108)
(actual time=0.040..18.838 rows=7009 loops=1)\n            -> Table
scan on uscensus  (cost=3582.06 rows=32300) (actual time=0.037..12.833
rows=32561 loops=1)\n'

-- explain analyze select AVG(AGE) from uscensus where Race = 'White'
and educationnum>3 and (relationship='Own-child' or
relationship='Husband') group by workclass ;
-- '-> Table scan on <temporary>  (actual time=27.722..27.723 rows=9
loops=1)\n    -> Aggregate using temporary table  (actual
time=27.721..27.721 rows=9 loops=1)\n        -> Filter: ((uscensus.race
= \'White\') and (uscensus.educationnum > 3) and
((uscensus.relationship = \'Own-child\') or (uscensus.relationship =
\'Husband\')))  (cost=3582.06 rows=205) (actual time=0.155..20.057
rows=15975 loops=1)\n            -> Table scan on uscensus
(cost=3582.06 rows=32300) (actual time=0.140..11.375 rows=32561
loops=1)\n'
-- mesmo custo de 3582.06

-- com indice
-- CREATE INDEX idade ON uscensus(age);
-- explain analyze select AVG(AGE) from uscensus where Race = 'White'
and educationnum>3 and relationship='Own-child' group by workclass ;
```

```
-- '-> Table scan on <temporary>  (actual time=31.064..31.065 rows=9
loops=1)\n    -> Aggregate using temporary table  (actual
time=31.063..31.063 rows=9 loops=1)\n        -> Filter:
((uscensus.relationship = \'Own-child\') and (uscensus.race =
\'White\') and (uscensus.educationnum > 3))  (cost=3582.06 rows=108)
(actual time=0.240..27.113 rows=4231 loops=1)\n            -> Table
scan on uscensus  (cost=3582.06 rows=32300) (actual time=0.213..20.674
rows=32561 loops=1)\n'


-- explain analyze select AVG(AGE) from uscensus where Race = 'White'
and educationnum>3 and relationship='Not-in-family' group by workclass
;
-- '-> Table scan on <temporary>  (actual time=17.017..17.018 rows=8
loops=1)\n    -> Aggregate using temporary table  (actual
time=17.017..17.017 rows=8 loops=1)\n        -> Filter:
((uscensus.relationship = \'Not-in-family\') and (uscensus.race =
\'White\') and (uscensus.educationnum > 3))  (cost=3582.06 rows=108)
(actual time=0.040..13.937 rows=7009 loops=1)\n            -> Table
scan on uscensus  (cost=3582.06 rows=32300) (actual time=0.036..10.026
rows=32561 loops=1)\n'


-- explain analyze select AVG(AGE) from uscensus where Race = 'White'
and educationnum>3 and (relationship='Own-child' or
relationship='Husband') group by workclass ;
-- '-> Table scan on <temporary>  (actual time=42.192..42.193 rows=9
loops=1)\n    -> Aggregate using temporary table  (actual
time=42.191..42.191 rows=9 loops=1)\n        -> Filter: ((uscensus.race
= \'White\') and (uscensus.educationnum > 3) and
((uscensus.relationship = \'Own-child\') or (uscensus.relationship =
\'Husband\')))  (cost=3582.06 rows=205) (actual time=0.063..30.481
rows=15975 loops=1)\n            -> Table scan on uscensus
(cost=3582.06 rows=32300) (actual time=0.057..17.241 rows=32561
loops=1)\n'
-- mesmo custo de 3582.06
-- os tres tem a mesma performance devido a media ele precisa passar em
todas as linhas da tabela ele não utiliza o indice secundário
```