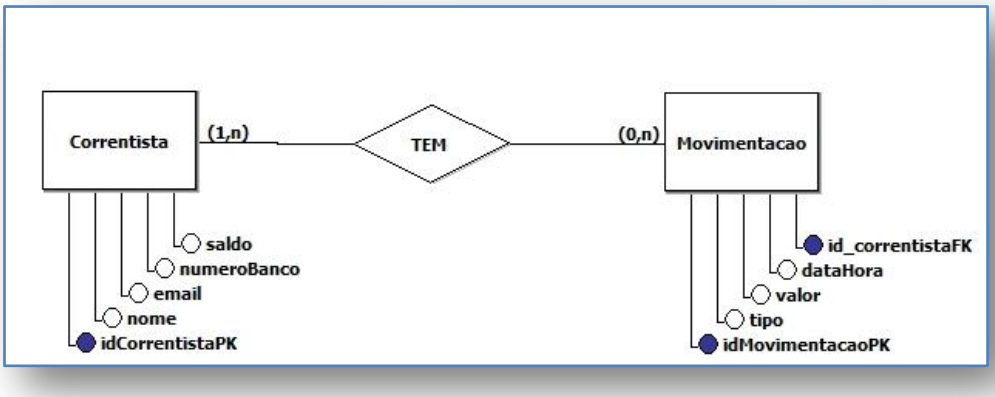
Cardinalidade:

Uma das principais funções de cardinalidade, é manter a integridade do banco de dados, em associação com as regras de negócio, não permitindo que essas regras sejam quebradas causando anomalias no SGBD, dados repetidos ou fora de normalização. Essas associações são ligadas através de chaves (chave estrangeira e chave primária) que são registro de indexação que não se repetem e que podem ser usadas como um índice para os demais campos da tabela do banco de dados. Em chaves primárias, não pode haver valores nulos nem repetição.

Em modelos de dados complexos, o relacionamento poderá ocorrer centenas de vezes, envolvendo dezenas de tabelas. O renomado cientista da computação, C.J. Date, criou um método sistemático para essa organização do modelos de banco de dados. Esse modelo é o que conhecemos como *normalização de dados*. A normalização de dados permite-nos entre outras coisas, evitar anomalias em comando de delete e updates. Na vida real o processo de normalização de banco de dados consiste em dividir uma grande tabela com diversas colunas em tabelas menores. Progamação é melhor que Banco de Dados.

```
Correntista   1_  *      Movimentação
idConta                 idMovimentacao
idBanco                  tipo - (Retirada_ deposito)
nome                     valor -
email                     DataHora
saldo                        id_Correntista
```

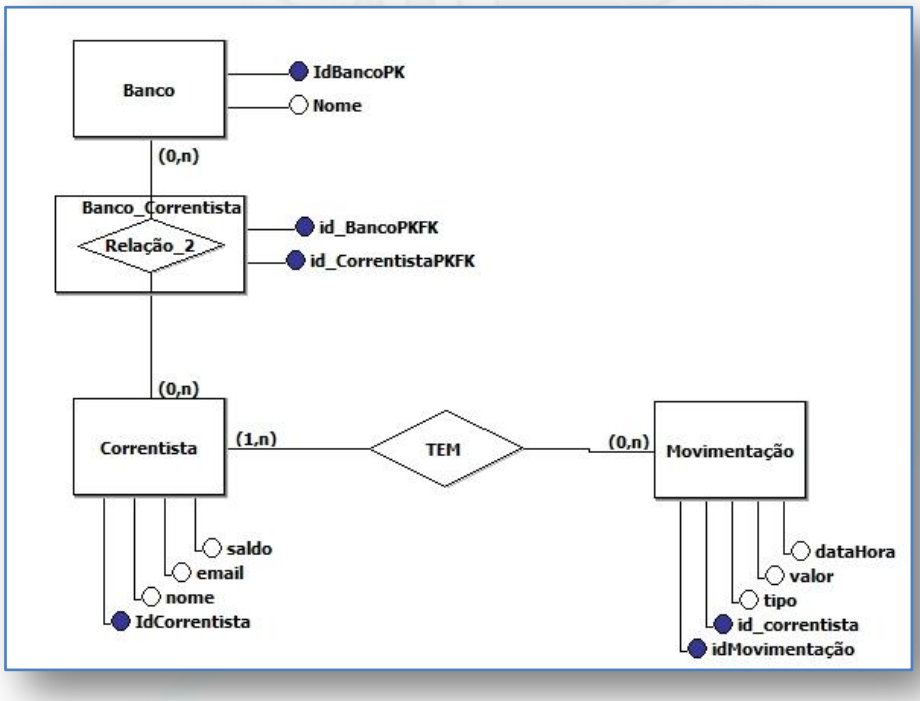


script.sql

```
create database modelagem3;
use modelagem3;

create table correntista(
    idCorrentista int primary key,
    numeroBanco int,
    nome varchar (50),
     email varchar (50) unique,
     saldo double);

 create table movimentacao(idMovimentacao int primary key,
   tipo enum ('deposito','retirada','transferencia') not null,
   valor double,
   datahora datetime,
   id_correntista int,
   foreign key(id_correntista) references
Correntista(idCorrentista)
               );
```



```
#nao faça esse comando
drop database if exists modelagem3;

#Cuidado com o comando acima

create database modelagem3;
use modelagem3;

create table banco(
      idBanco int primary key,
      nome varchar (50)
);

create table correntista(
      idCorrentista int primary key,
      nome varchar (50),
      email varchar (50) unique,
      saldo Double
);

#Chave Composta (porque eu tenho 2 chaves Primárias
#ENTIDADE ASSOCIATIVA _ PORQUE ELE VEM DE UM NPM

#Eles são chaves Primárias e Estrangeiras ...

create table banco_correntista(
      id_Banco int,
      id_correntista int,
      primary key(id_banco, id_correntista),
      foreign key(id_banco) references banco(idBanco),
      foreign key(id_Correntista) references
correntista(idCorrentista)
   );

create table movimentacao(
      idMovimentacao int primary key,
      tipo enum ('deposito','retirada','transferencia') not null,
      valor double,
      datahora datetime,
      id_correntista int,
      foreign key(id_correntista) references
Correntista(idCorrentista)
);
```

```
#NPM

insert into banco values (1000,'Itau');
insert into banco values (1001,'Bradesco');

insert into correntista values (5000,'lu','luciana@gmail.com',
15000);
insert into correntista values
(5001,'sergio','sergio@gmail.com', 55000);
insert into correntista values (5002,'digao','digao@gmail.com',
2000);


insert into banco_correntista values (1000, 5000);
insert into banco_correntista values (1000, 5001);
insert into banco_correntista values (1001, 5002);


select * from banco;
+---------+----------+
| idBanco | nome     |
+---------+----------+
|    1000 | Itau     |
|    1001 | Bradesco |
+---------+----------+

select * from correntista;
+---------------+--------+-------------------+-------+
| idCorrentista | nome   | email             | saldo |
+---------------+--------+-------------------+-------+
|          5000 | lu     | luciana@gmail.com | 15000 |
|          5001 | sergio | sergio@gmail.com  | 55000 |
|          5002 | digao  | digao@gmail.com   |  2000 |
+---------------+--------+-------------------+-------+

select * from banco_correntista;
+----------+----------------+
| id_Banco | id_correntista |
+----------+----------------+
|     1000 |           5000 |
|     1000 |           5001 |
|     1001 |           5002 |
+----------+----------------+
```

```
#Para Juntar as 3 tabelas devo utilizar

select b.idbanco, b.nome, c.nome "correntista",  c.email,
c.saldo, c.idCorrentista from  banco_correntista   as bc
     inner join correntista c  on c.idCorrentista =
bc.id_correntista
     inner join banco b     on    b.idbanco = bc.id_banco;
```

```
+---------+----------+-------------+--------------------+-------
|idbanco | nome | correntista | email     | saldo |idCorrentista
+---------+----------+-------------+--------------------+-------
|1000 | Itau     | lu     | luciana@gmail.com | 15000 |   5000 |
|1000 | Itau     | sergio | sergio@gmail.com  | 55000 |   5001 |
|1001 | Bradesco | digao  | digao@gmail.com   |  2000 |   5002 |
+---------+----------+-------------+--------------------+-------
```