Trabalho da disciplina banco de dados I - UERJ 2019/2

Integrantes

- Dennis Ribeiro Paiva - 201610050611
- Paulo Victor Coelho - 201610049711
- Vinicius Sathler - 201610051611

Introdução

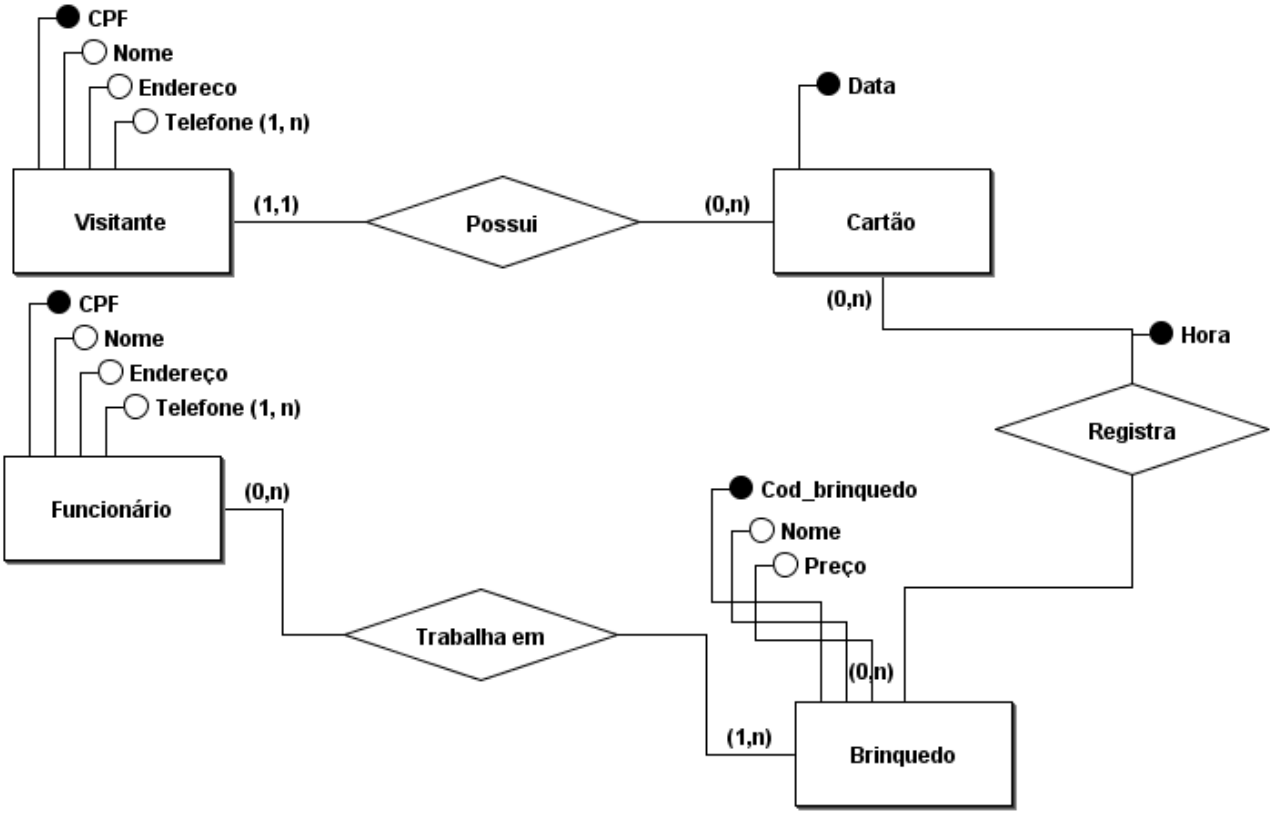
Neste trabalho serão apresentados todos os passos de modelagem de um projeto básico de banco de dados, da descrição do minimundo até sua implementação funcional. O tema abordado será o gerenciamento do cartão de visita de um visitante em um parque de diversões.

Minimundo

O parque de diversões 'SmashLand' é um parque moderno mas muito ganancioso. Seus visitantes recebem na entrada um cartão digital que deve ser apresentado na entrada de cada brinquedo. Cada brinquedo possui um nome e código de identificação. Sendo a gerência do parque muito gananciosa, cada cartão de visitante deve registrar a cobrança de entrada nos brinquedos cada vez que for utilizado, ou seja, o visitante paga cada vez que for usar um brinquedo. Os funcionários deste parque também são muito ocupados, tendo muitas vezes que trabalhar em mais de um brinquedo, sendo que cada brinquedo pode precisar de um ou mais funcionários. Para estimular uma concorrência saudável entre seus funcionários, a gerência do parque paga um adicional de dois por cento do dinheiro arrecadado em cada brinquedo para cada funcionario responsável por ele. A fim de evitar fraudes, tanto os clientes quanto os funcionários devem ser registrados de acordo com o seu nome completo, CPF, endereço e telefone(s) para contato. Para disfarçar sua ganância o parque permite que cada cartão sejá válido por um dia inteiro. Ao final do dia o visitante deve pagar o valor acumulado de todos os brinquedos que visitou.

Modelo conceitual

Diagrama entidade-relacionamento



Restrições de Domínio

- Visitante:
  - CPF: Número inteiro de onze dígitos.
  - Nome: String de no máximo 45 caracteres.
  - Endereço: String de no máximo 100 caracteres.
  - Telefone: Numero inteiro formado por oito ou nove dígitos.
- Cartão:
  - Data: Data no formato aaaa-mm-dd de acordo com o tipo DATE da linguagem mySQL.
- Brinquedo:
  - Cod_brinquedo: Numero inteiro de cinco dígitos.
  - Nome: String de no máximo 45 caracteres.
  - Preço: Número real positivo com duas casas decimais de precisão.
- Registra (Cartão-Brinquedo):
  - Hora: registro de hora no formato hh-mm-ss de acordo com o tipo TIME da linguagem mySQL.
- Funcionário:
  - CPF: Número inteiro de onze dígitos.
  - Nome: String de no máximo 45 caracteres.
  - Endereço: String de no máximo 100 caracteres.
  - Telefone: Numero inteiro formado por oito ou nove dígitos.

Modelo Relacional

Descrição

- **Chave primária**
- *Chave estrangeira*

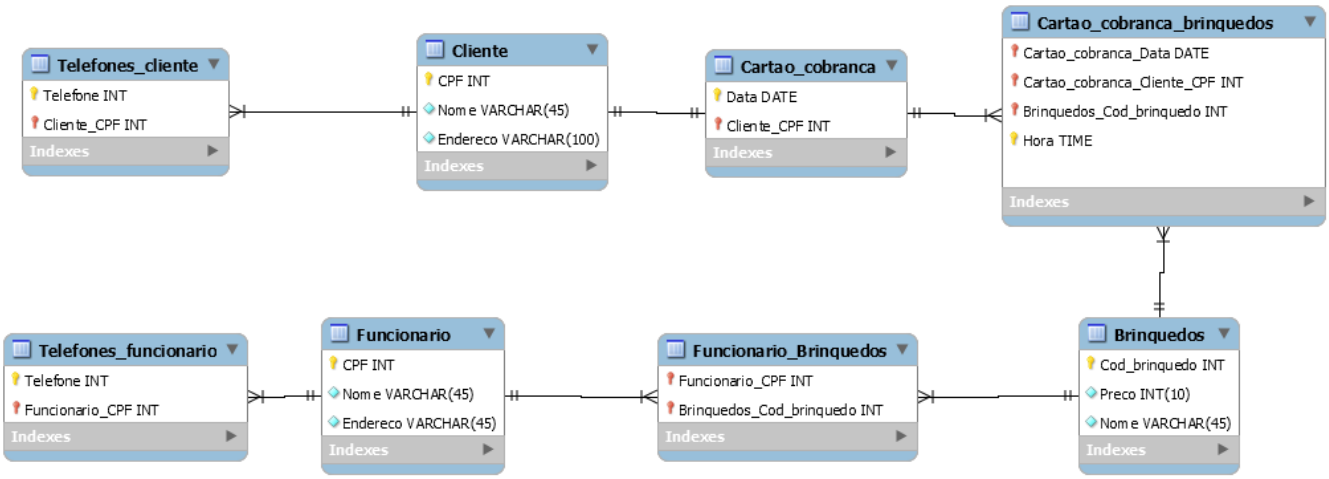
- Cliente (**CPF**, Nome, Endereço)
- Telefones_cliente (**Telefone**, ***Cliente_CPF***)
- Cartao_Cobrança (**Data**, ***Cliente_CPF***)
- Brinquedo (**Cod_brinquedo**, Nome, Preço)
- Cartao_cobranca_brinquedo (**Hora**, ***Cartao_Cobrança_Data***, ***Cartao_Cobranca_Cliente_CPF***, ***Brinquedo_Cod_brinquedo***)
- Funcionario (**CPF**, Nome, Endereço)
- Telefones_funcionario (**Telefone**, ***Funcionario_CPF***)
- Funcionario_brinquedo (***Funcionario_CPF***, ***Brinquedo_Cod_brinquedo***)

Sabendo-se que cada Cliente possui nenhum ou vários cartões, mas que cada cartão está vinculado a obrigatoriamente um e apenas um cliente, foi adicionada uma coluna extra na tabela cartão para referenciar o cliente ao qual ele está asssociado

Como cada cartão de cobrança pode registrar nenhum ou vários brinquedos, e cada brinquedo pode estar registrado em nenhum ou vários cartões, criou-se uma nova tabela para relação

Cada funcionário pode trabalhar em nenhum ou mais de um brinquedo, e cada brinquedo possui entre um e varios funcionários. De maneira análoga a relação brinquedo-cartão foi criada uma nova tabela

Diagrama



Restrições Semânticas

1. Cada funcionário não pode receber menos do que o salário minímo, estipulado em $sc 500,00
2. O visitante paga por cada brinquedo visitado, mas não pode pagar menos de $sc 100,00

Algebra Relacional

Cobrança do visitante Paulo Coelho no dia 01 de Junho de 2019

$$cartaoPaulo \leftarrow Cliente \bowtie_{CPF=Cartao\_Cobranca\_Cliente\_CPF} Cartao\_cobranca\_brinquedo$$

$$brinquedosPaulo \leftarrow \Pi_{Brinquedo\_cod\_brinquedo,hora}(\sigma_{nome='PauloCoelho' AND data='01/06/2019'}(cartaoPaulo))$$

$$precoBrinquedo \leftarrow brinquedosPaulo \bowtie_{Brinquedo\_cod\_brinquedo=Cod\_brinquedo} Brinquedos$$

$$cobranca \leftarrow g_{max(100,sum(preco))}(precoBrinquedo)$$

Calculo do salário do funcionário Dennis Ribeiro no mês de junho

$$Dennis \leftarrow \sigma_{nome='DennisRibeiro'}(Funcionarios)$$

$$brinquedosDennis \leftarrow \Pi_{cod\_brinquedo}(Dennis \bowtie_{CPF=Funcionarios\_CPF} Funcionarios\_brinquedos)$$

$$visitasBrinquedos \leftarrow \sigma_{Cartao\_Cobranca\_Data>'31/05/2019' AND Cartao\_Cobranca\_Data<'01/07/2019'}(brinquedosDennis \bowtie_{cod\_brinquedo=Brinquedos\_cod\_brinquedo} Cartao\_cobranca\_brinquedo)$$

$$precoBrinquedos \leftarrow visitasBrinquedos \bowtie Brinquedos$$

$$salarioDennis \leftarrow \Pi_{preco+500}(g_{sum(preco)}(\Pi_{cod_bringuedo,hora,data,preco*0.02}))$$

Clientes que visitaram todos os brinquedos em qualquer intervalo de tempo

$$clienteBrinquedo \leftarrow \delta_{Brinquedos\_cod\_brinquedo \rightarrow cod_bringuedo}(\Pi_{cartao\_cobranca\_cliente\_CPF,Brinquedos\_cod\_brinquedo}(Cartao\_cobranca\_brinquedos))$$

$$listaBrinquedos \leftarrow \Pi_{cod\_brinquedo} Brinquedos$$

$$clientesQueVisitaramTodosOsBrinquedos \leftarrow clienteBrinquedo \div listaBrinquedos$$

Funcionários que visitaram os brinquedos em que trabalham como clientes

$$FC \leftarrow Funcionarios \cap Clientes$$

$$brinquedosV \leftarrow \Pi_{Brinquedo\_cod\_brinquedo,Cartao\_cobranca\_cliente\_CPF}(Cartao\_cobranca\_brinquedo)$$

$$brinquedosT \leftarrow \Pi_{Brinquedo\_cod\_brinquedo,Funcionario\_CPF}(Funcionario\_brinquedo)$$

$$\delta_{Brinquedos\_cod\_brinquedo \rightarrow cod\_brinquedo,Cartao\_cobranca\_cliente\_CPF \rightarrow CPF}(brinquedosV)$$

$$\delta_{Brinquedos\_cod\_brinquedo \rightarrow cod\_brinquedo,Funcionario\_CPF \rightarrow CPF}(brinquedosT)$$

$$brinquedosVT \leftarrow \Pi_{CPF}(brinquedosV \cap brinquedosT)$$

$$Nomes \leftarrow \Pi_{Nome}(FC \bowtie_{FC.cpf=brinquedosVT.cpf} BrinquedosVT)$$

Consultas em SQL

Listar valor a ser cobrado de cada cliente por cada dia de visita

```
SELECT A.Cartao_cobranca_data as Data,
C.nome as Nome, sum(B.preco) as Valor
FROM Cartao_cobranca_brinquedos A, Brinquedos B, Cliente C
WHERE A.Brinquedos_Cod_brinquedo = B.cod_brinquedo
AND A.Cartao_cobranca_Cliente_CPF = C.CPF
GROUP BY A.Cartao_cobranca_data,A.Cartao_cobranca_Cliente_CPF
```

Salario de cada funcionario para o mês de Junho

```
select C.nome, 500+sum(0.02*A.visitas*D.preco) as Salario
from (select Brinquedos_Cod_brinquedo as cod_brinquedo,
count(Cartao_cobranca_cliente_cpf) as visitas
from cartao_cobranca_brinquedos
where Cartao_cobranca_data <"2019-07-01"
AND Cartao_cobranca_data > "2019-05-31"
group by Brinquedos_Cod_brinquedo) A, funcionario_brinquedos B,
Funcionario C, Brinquedos D
where C.cpf = B.Funcionario_CPF
AND B.Brinquedos_cod_brinquedo = A.cod_brinquedo
AND B.Brinquedos_cod_brinquedo = D.cod_brinquedo
group by nome
order by nome
```

Funcionários que também visitaram o parque como clientes

```
SELECT nome FROM Cliente INTERSECT SELECT nome FROM Funcionario
ORDER BY nome
```

Clientes e funcionarios que possuem mais de um telefone para contato ou possuem telefone igual á outro cliente

```
(select C.Cliente_CPF
from
(select * from telefones_cliente
UNION
select * from telefones_Funcionario) C,
(select * from telefones_cliente
UNION
select * from telefones_Funcionario) D
where C.Telefone = D.Telefone
AND C.Cliente_cpf <> D.cliente_cpf)
UNION
(select B.cliente_CPF
from (select A.Cliente_CPF, count(A.Telefone) as cnt
from (select * from telefones_cliente UNION select *
from telefones_Funcionario) A group by Cliente_CPF) B
where B.cnt > 1)
```

Script DDL

```
create database bd1;

use bd1;

CREATE TABLE Cliente (
  CPF BIGINT NOT NULL PRIMARY KEY,
  Nome VARCHAR(45) NOT NULL,
  Endereco VARCHAR(100) NOT NULL
);

CREATE TABLE Brinquedos (
  Cod_brinquedo INT NOT NULL PRIMARY KEY,
  Preco INT(10) NOT NULL,
  Nome VARCHAR(45) NOT NULL
);

CREATE TABLE Funcionario (
  CPF BIGINT NOT NULL PRIMARY KEY,
  Nome VARCHAR(45) NOT NULL,
  Endereco VARCHAR(45) NOT NULL
);

CREATE TABLE Funcionario_Brinquedos (
  Funcionario_CPF BIGINT NOT NULL,
  Brinquedos_Cod_brinquedo INT NOT NULL,
  CONSTRAINT pk_Funcionario_Brinquedos PRIMARY KEY
  (Funcionario_CPF, Brinquedos_Cod_brinquedo),
  CONSTRAINT fk_Funcionario_CPF
  FOREIGN KEY (Funcionario_CPF)
  REFERENCES Funcionario(CPF),
  CONSTRAINT fk_Brinquedos_Cod_brinquedo_funcionarios
  FOREIGN KEY (Brinquedos_Cod_brinquedo)
  REFERENCES Brinquedos(Cod_brinquedo)
);

CREATE TABLE Cartao_cobranca (
  Data DATE NOT NULL,
  Cliente_CPF BIGINT NOT NULL,
  CONSTRAINT pk_Cartao_cobranca PRIMARY KEY (Data, Cliente_CPF),
  CONSTRAINT fk_Cliente_CPF
  FOREIGN KEY (Cliente_CPF)
  REFERENCES Cliente(CPF)
);

CREATE TABLE Cartao_cobranca_brinquedos (
  Cartao_cobranca_Data DATE NOT NULL,
  Cartao_cobranca_Cliente_CPF BIGINT NOT NULL,
  Brinquedos_Cod_brinquedo INT NOT NULL,
  Hora TIME NOT NULL,
  CONSTRAINT pk_Cartao_cobranca_brinquedos
  PRIMARY KEY (Cartao_cobranca_Data, Cartao_cobranca_Cliente_CPF,
Brinquedos_Cod_brinquedo, Hora),
  CONSTRAINT fk_Cartao_cobranca_Data
  FOREIGN KEY (Cartao_cobranca_Data)
  REFERENCES Cartao_cobranca(Data),
  CONSTRAINT fk_Cartao_cobranca_Cliente_CPF
  FOREIGN KEY (Cartao_cobranca_Cliente_CPF)
  REFERENCES Cartao_cobranca(Cliente_CPF),
```

```
  CONSTRAINT fk_Brinquedos_Cod_brinquedo_cartao_cobranca
  FOREIGN KEY (Brinquedos_Cod_brinquedo)
  REFERENCES Brinquedos(Cod_brinquedo)
);

CREATE TABLE Telefones_funcionario (
  Telefone INT NOT NULL,
  Funcionario_CPF BIGINT NOT NULL,
  CONSTRAINT pk_Telefones_funcionario
  PRIMARY KEY (Telefone, Funcionario_CPF),
  CONSTRAINT fk_Funcionario_CPF_tel
  FOREIGN KEY (Funcionario_CPF)
  REFERENCES Funcionario(CPF)
);

CREATE TABLE Telefones_cliente (
  Telefone INT NOT NULL,
  Cliente_CPF BIGINT NOT NULL,
  CONSTRAINT pk_Telefones_cliente PRIMARY KEY (Telefone, Cliente_CPF),
  CONSTRAINT fk_cliente_cpf_tel
  FOREIGN KEY (Cliente_CPF)
  REFERENCES Cliente(CPF)
);
```

Avaliação de qualidade das tabelas

Telefone

O atributo telefone se encontra na tabela de Funcionário e Cliente. Por ser multivalorado se fez necessária, de acordo com a primeira forma normal, a criação de tabelas a parte. Como clientes diferentes podem cadastrar o mesmo telefone (marido e mulher por exemplo), foi definido o campo telefone como chave primária.

Cartao_Cobrança

A tabela cartão_cobrança se relaciona n para n. O cartão contém o campo Data que especifica o dia em que o cliente foi ao parque, como cada cliente pode visitar o parque em diferentes dias, se fez necessário tornar o atributo data chave primária junto do cpf do cliente, que também é chave estrangeira que relaciona o cartão ao cliente. Dessa forma garantiu-se que a tabela satisfaz todas as condições da terceira forma normal.

Cartao_Cobranca_Brinquedos

A tabela relaciona cada cartão de cobrança ao brinquedos visitados pelo cliente, dessa forma fez-se necessário a inclusão de chaves estrangeiras para cada chave primaria em ambas as tabelas, Como cada cliente pode visitar o mesmo brinquedo várias vezes durante o dia foi adicionada uma nova chave primária, hora, que especifica a hora em que o brinquedo foi visitado. A tabela está de acordo com todas as formas normais pois todos os campos são chaves primárias da tabela e não existe relação de transitividade.