

Megadados

Modelagem relacional, DDL

Maciel C. Vidal
Marcio Stabile

Na aula de hoje

- Modelagem relacional (continuação)
- CREATE

Leituras:

- SQL: Curso Prático, cap. 2, 4
- Testing:
<http://jasonpolites.github.io/tao-of-testing/intro-1.1.html>

Modelagem de dados

Objetivo: “desenvolver um **modelo de dados** que seja capaz de **representar** os **requerimentos de informações** de negócios” (Oliveira)

Modelagem **entidade-relacionamento**: o modelo é desenvolvido em termos de entidades e relacionamentos, que podem ser visualizados no **diagrama entidade-relacionamento**.

Modelagem **relacional**: o modelo é desenvolvido em termos de **relações** (tabelas)

Entidades

Agrupamento lógico de informações **inter-relacionadas**. Representa (normalmente) um objeto do mundo real.

(**Entidades associativas**: entidades especiais que existem apenas para relacionar duas entidades fundamentais.)

No modelo de banco de dados **relacional**, **entidades** são **tabelas**.

Atributos

São **características** das entidades.

Não confunda com entidades!

No modelo de banco de dados relacional, atributos são **colunas**.

Atributos obrigatórios e opcionais

Nem todo atributo é obrigatório.

Quando um atributo não é obrigatório, dizemos que o atributo pode ser **nulo** (em SQL: NULL).

Dicionário de dados

É a **documentação** de cada **atributo**:

- Qual o tipo de dados
- Quais os valores usuais
- Pode ser NULL?
- É unico?
- É chave primaria?
- É chave estrangeira? Com qual atributo se relaciona?

Chaves



Primárias: São atributos (ou grupos de atributos) que identificam unicamente cada instância de uma entidade

ATENÇÃO: CHAVES PRIMÁRIAS SÃO ÚNICAS E IMUTÁVEIS!

Estrangeiras: São atributos de uma entidade que são chaves primárias de alguma outra entidade. Servem para conectar entidades.

Exemplo

Vamos construir o diagrama do modelo relacional para a seguinte situação:

- Funcionários tem um id funcional, um nome e estão alocados em alguma sala.
- As salas tem um id, ficam em um certo prédio e num certo andar, e tem um número na porta.

Atividade

Desenhe no MySQL Workbench o diagrama correspondente à seguinte especificação:

Pessoa (nome, apelido)

TipoVeiculo (id, marca, modelo, ano, peso, potencia, torque)

Veiculo (id, dono, tipo, cor)

Nota: estão faltando nesta especificação

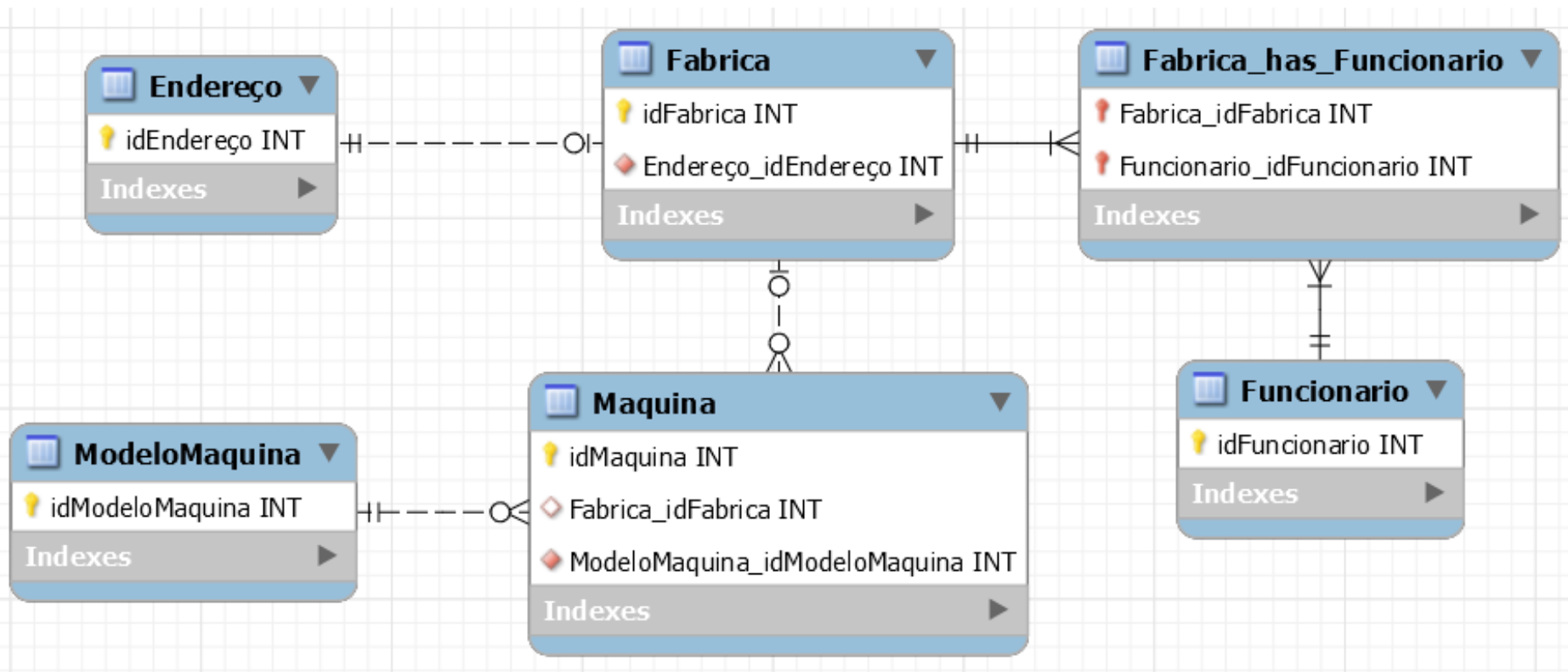
- Quem seriam as chaves primárias e estrangeiras?
- Quais os tipos de cada atributo?
- Quais atributos são opcionais (ou seja, podem ser não-nulos)?

Responda ainda:

- Este *design* está bom? Que modificações você faria? Existe o risco de informação duplicada e inconsistente?

Relacionamentos

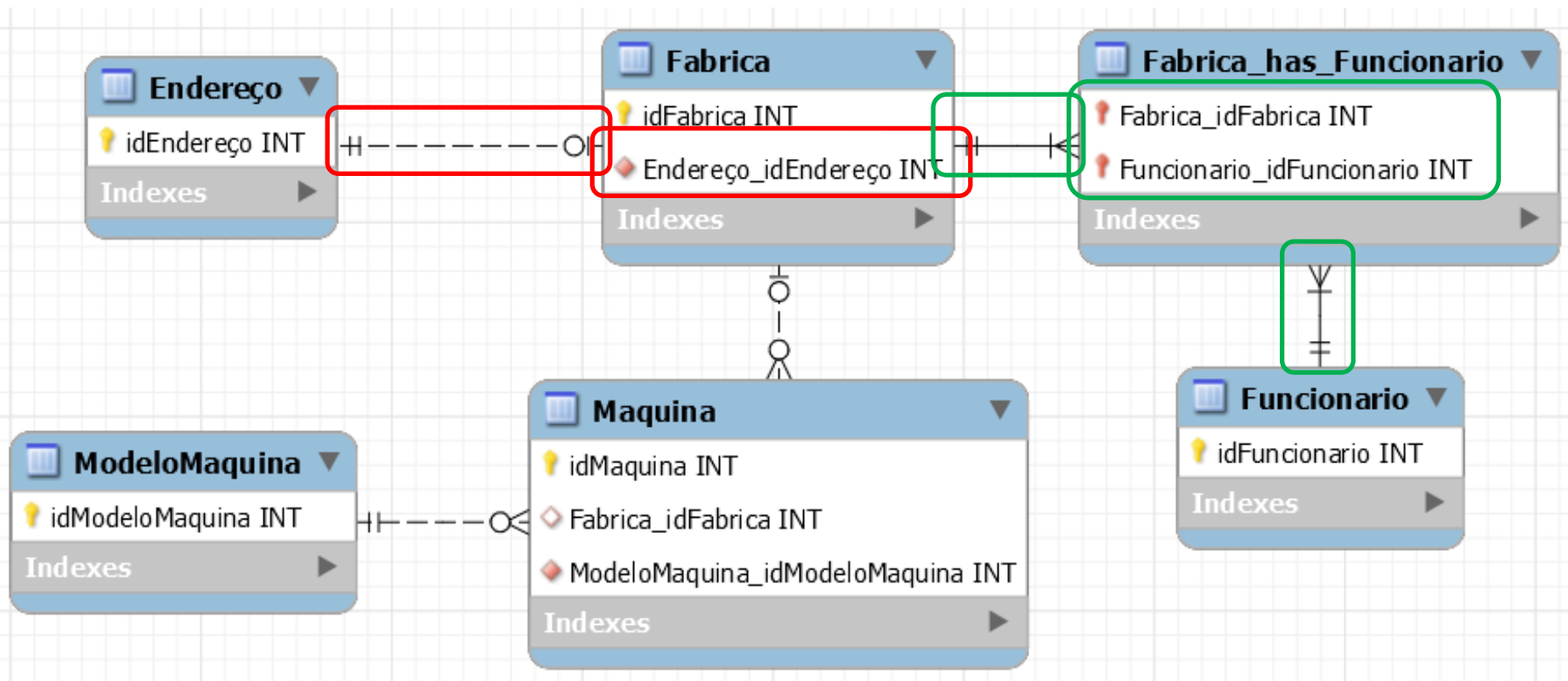
Conexões entre entidades



Relacionamentos identificadores e não-identificadores

Não-identificador

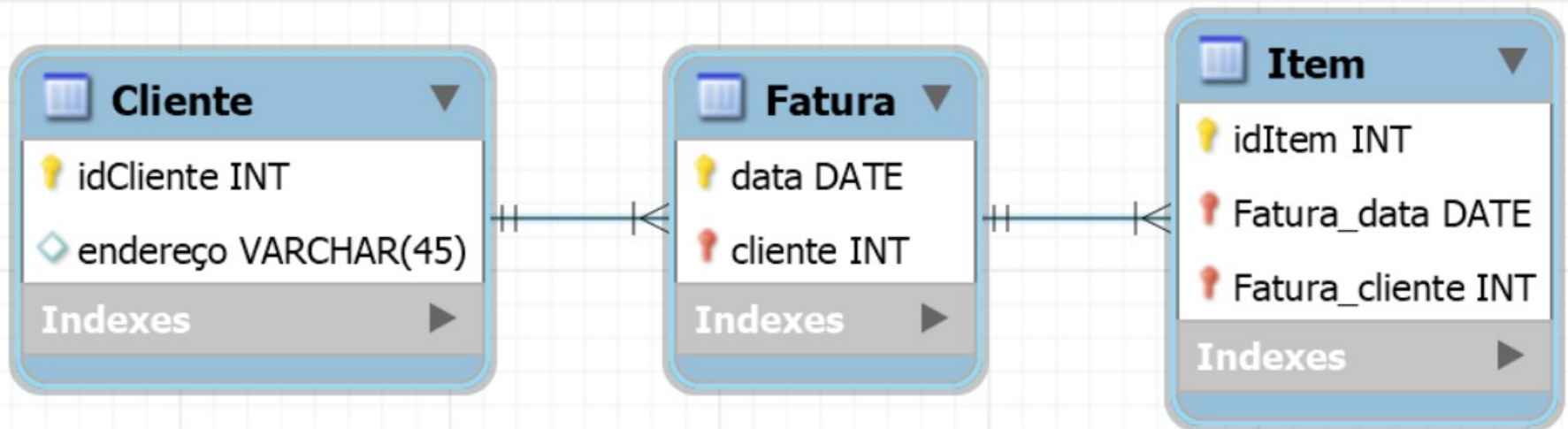
Identificador



Relacionamento identificador

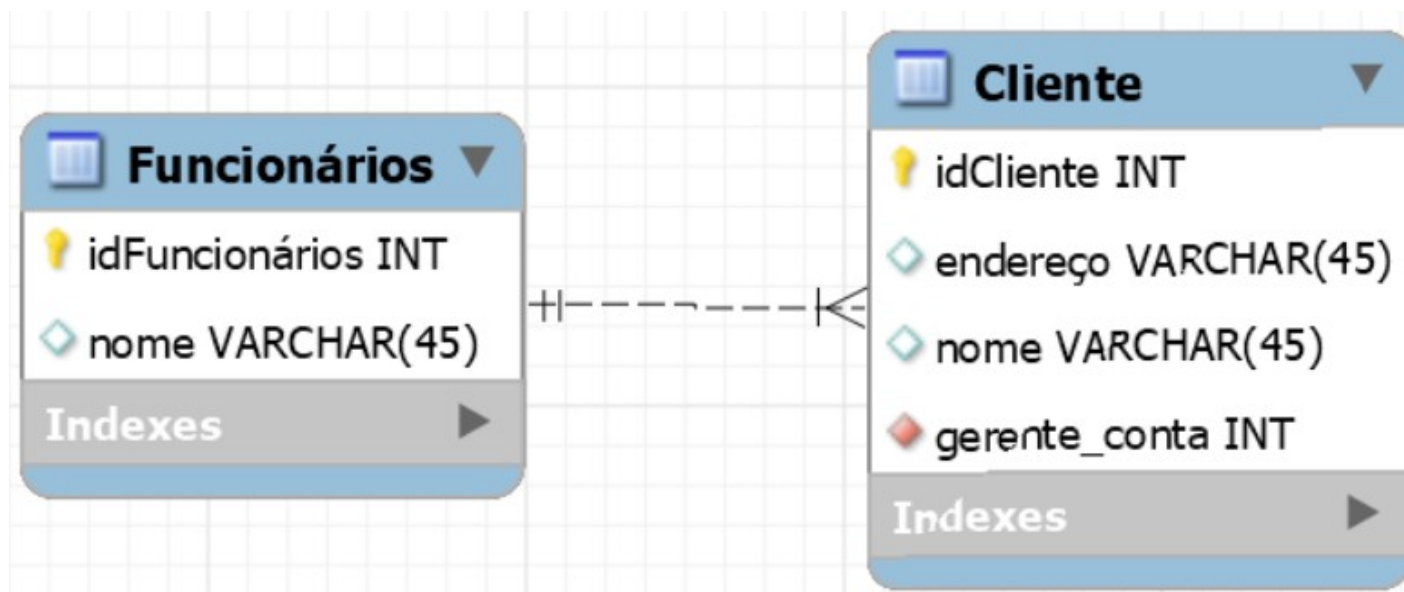
Relacionamento em que a identidade de uma instância de uma entidade está atrelada à identidade de outra instância em outra entidade.

Basicamente: quando a chave primária de uma entidade deveria **em princípio** conter uma chave estrangeira para outra entidade.



Relacionamento não-identificador

Quando a identidade de uma instância não depende intrinsecamente da identidade da outra instância



Exercícios

Construa o diagrama do modelo relacional nos seguintes cenários:

1. O banco de dados de uma empresa armazena informação sobre os funcionários (RG, nome, salário, telefone) e departamentos (dep_id, nome). Armazena também informação sobre em qual departamento um funcionário trabalha.
2. A empresa do problema 1 decidiu adicionar a lista de dependentes de cada funcionário. Que relacionamento temos aqui?

Exercícios

3. A empresa decidiu que cada funcionário deve indicar outro funcionário como seu “mentor”.
4. A empresa também quer armazenar a informação a respeito dos gerentes de departamento.
5. A empresa decidiu que cada funcionário pode trabalhar em vários departamentos.

A linguagem SQL

- Structured Query Language
 - Originalmente chamada de SEQUEL (Structured-English Query Language)
- É uma “domain-specific language” para trabalhar com bancos de dados relacionais
 - Hoje usada também (de forma adaptada) em bancos NoSQL
- Vendor-dependent, com alguma padronização

SQL

<https://dev.mysql.com/doc/refman/5.7/en/sql-syntax.html>

- Data Definition Language (DDL)
- Data Manipulation Language (DML)
- Data Query Language (DQL)
- Data Control Language (DCL)

SQL

Data Definition Language (DDL)

- CREATE TABLE
- ALTER TABLE
- DROP TABLE
- CREATE INDEX
- ALTER INDEX
- DROP INDEX
- e muitos outros!

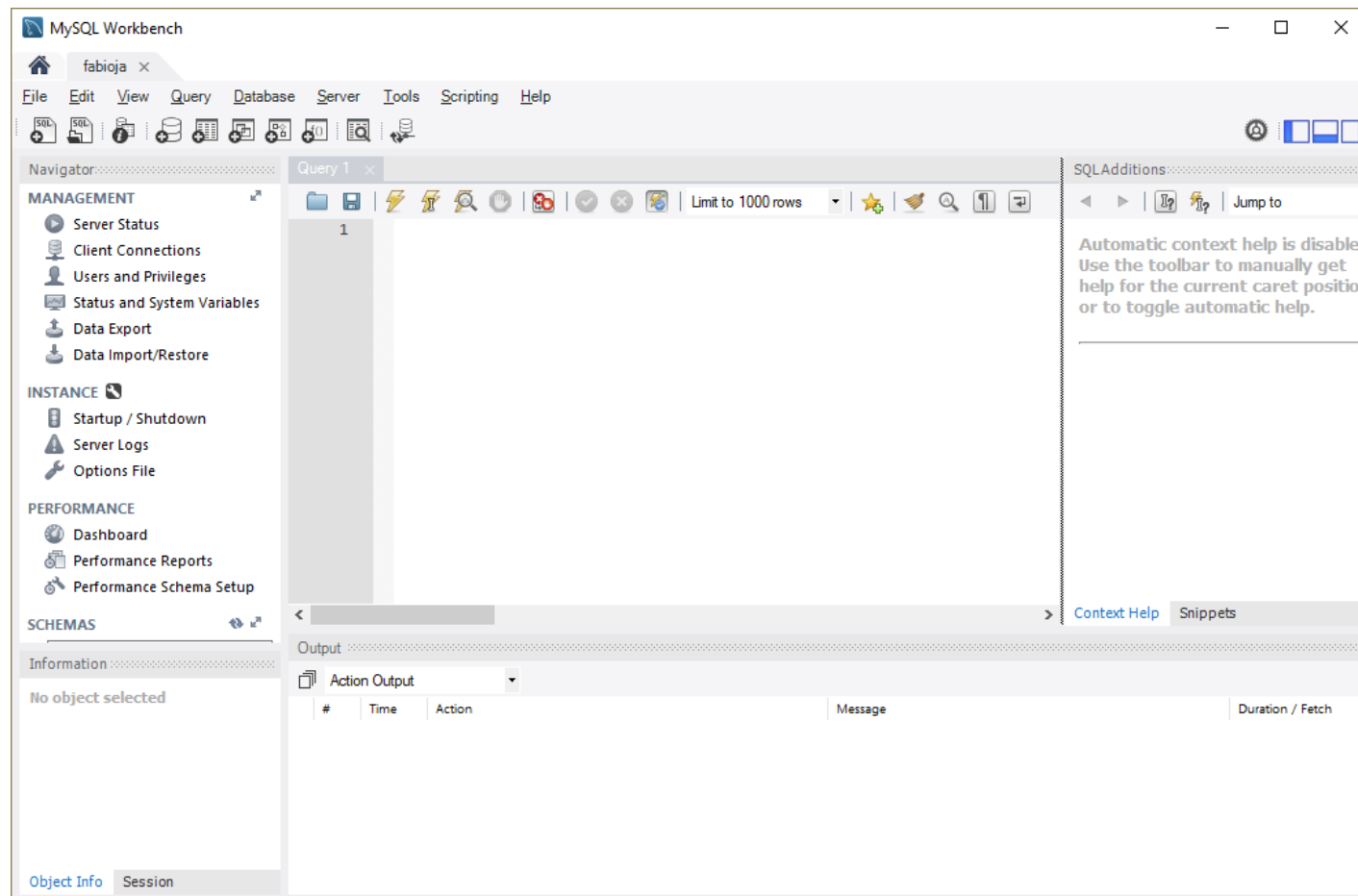
Tipos de dados

- INTEGER ou INT
- DECIMAL
- BIT
- DATE
- DATETIME
- TIMESTAMP
- VARCHAR
- e muitos outros!

<https://dev.mysql.com/doc/refman/8.0/en/data-types.html>

Vamos explorar um pouco

Abra o “MySQL Workbench”



CREATE DATABASE

```
CREATE DATABASE megadados; Teste você mesm  
USE megadados;
```

- O que acontece se você rodar duas vezes este script? **Teste você mesm**
- Boas práticas: torne seus scripts idempotentes, ou seja, devemos poder rodar o mesmo script duas vezes sem alterar o resultado.
 - Porque?

DROP

`DROP DATABASE megadados;` **Teste você mesmo**

- Mas e se a base não existisse em primeiro lugar?
Rode de novo o comando para ver.

`DROP DATABASE IF EXISTS megadados;`

`CREATE DATABASE megadados;`

`USE megadados;`

Teste você mesmo

`script_01.sql`

**Acostume-se a consultar
documentação por conta própria!!**

CREATE TABLE

```
USE megadados;  
DROP TABLE IF EXISTS Funcionario;  
CREATE TABLE Funcionario (  
    RG INT NOT NULL,  
    orgao VARCHAR(10) NOT NULL,  
    nome VARCHAR(80) NOT NULL UNIQUE,  
    salario FLOAT NOT NULL DEFAULT 500.0,  
    telefone VARCHAR(30),  
    idDepartamento INT NOT NULL,  
    PRIMARY KEY (RG, orgao),  
    FOREIGN KEY (idDepartamento)  
        REFERENCES Departamento(idDepartamento),  
    CHECK (salario >= 0)  
);
```

Problema: tabela Departamento ainda não existe!

CREATE TABLE

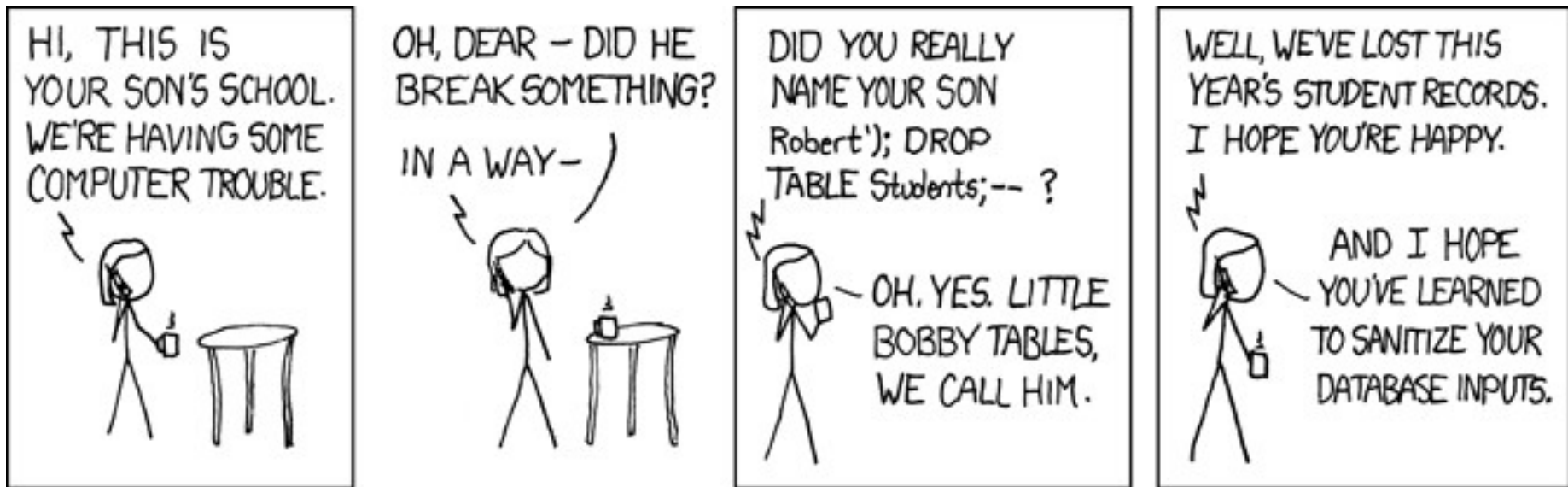
```
USE megadados;  
DROP TABLE IF EXISTS Departamento;  
CREATE TABLE Departamento (  
    idDepartamento INT NOT NULL,  
    nome VARCHAR(80) NOT NULL,  
    PRIMARY KEY (idDepartamento)  
);
```

Teste você mesmo:
`script_02.sql`

DROP TABLE

DROP TABLE IF EXISTS Funcionario;

- Um clássico: “Little Bobby Tables”



<https://xkcd.com/327/>

ALTER TABLE

USE megadados;

```
ALTER TABLE Funcionario
  ADD COLUMN (
    RG_mentor INT,
    orgao_mentor INT
  );
```

```
ALTER TABLE Funcionario
  MODIFY COLUMN
    orgao_mentor VARCHAR(10);
```

```
ALTER TABLE Funcionario
  ADD FOREIGN KEY (RG_mentor,orgao_mentor)
  REFERENCES Funcionario(RG, orgao);
```

E o tal “IF EXISTS”? Eis uma bela gamb... quer dizer... solução engenhosa... : <https://stackoverflow.com/questions/24571611/mysql-alter-table-if-column-not-exists>

Tente você mesmo: `script_03.sql`

CREATE INDEX

```
USE megadados;  
CREATE INDEX idxTelefone  
    ON Funcionario (telefone);
```

Quando criar?

- Chave primária
- Chave estrangeira
- Colunas usadas frequentemente para busca

Quando não criar?

- Colunas com muitos NULL
- Colunas com muitos valores iguais
- Tabelas muito pequenas

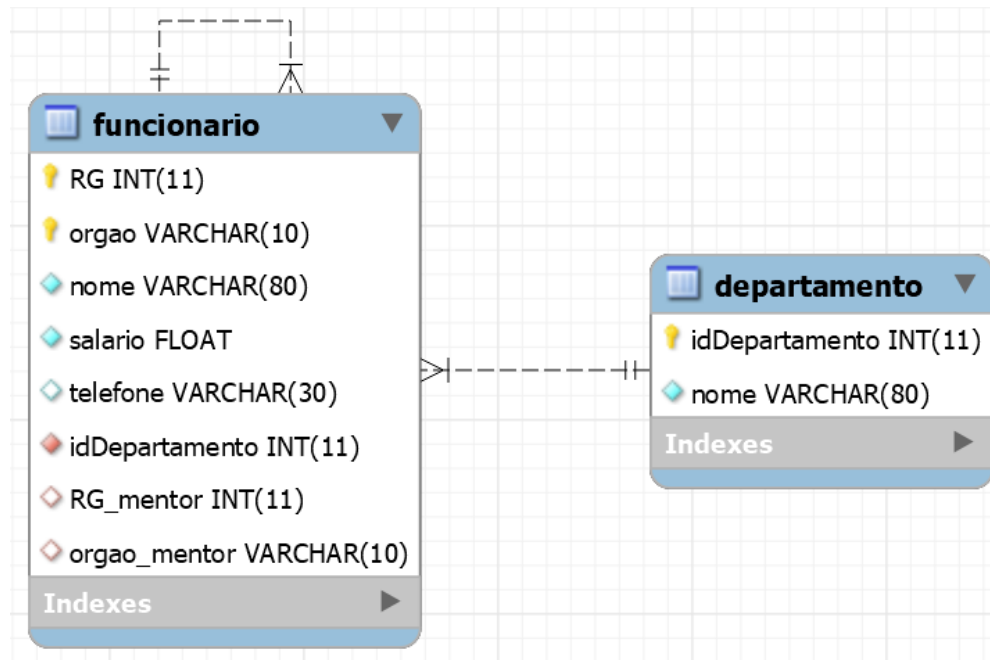
Não crie mais do que 6 índices por tabela

Tente você mesmo:

[script_04.sql](#)

Verifique os índices existentes antes e depois

Geração de código DDL com MySQL Workbench



Gerando scripts com MySQL Workbench

Forward Engineer to Database

Connection Options

Options

Select Objects

Review SQL Script

Commit Progress

Review the SQL Script to be Executed

This script will now be executed on the DB server to create your databases.
You may make changes before executing.

```
1  -- MySQL Workbench Forward Engineering
2
3  SET @OLD_UNIQUE_CHECKS=@@UNIQUE_CHECKS, UNIQUE_CHECKS=0;
4  SET @OLD_FOREIGN_KEY_CHECKS=@@FOREIGN_KEY_CHECKS, FOREIGN_KEY_CHECKS=0;
5  SET @OLD_SQL_MODE=@@SQL_MODE, SQL_MODE='TRADITIONAL,ALLOW_INVALID_DATES';
6
7  -----
8  -- Schema mydb
9  -----
10 -----
11 -- Schema megadados
12 -----
13 -----
14 -- Schema megadados
15 -----
16 -----
17 CREATE SCHEMA IF NOT EXISTS `megadados` DEFAULT CHARACTER SET utf8mb4 COLLATE utf8mb4_0900_ai_ci ;
18 USE `megadados` ;
19 -----
20 -----
21 -- Table `megadados`.`departamento`
22 -----
23 DROP TABLE IF EXISTS `megadados`.`departamento` ;
24
25 CREATE TABLE IF NOT EXISTS `megadados`.`departamento` (
26   `idDepartamento` INT(11) NOT NULL,
27   `nome` VARCHAR(80) NOT NULL,
28   PRIMARY KEY (`idDepartamento`))
29 ENGINE = InnoDB
30 DEFAULT CHARACTER SET = utf8mb4
31 COLLATE = utf8mb4_0900_ai_ci;
32
33 -----
34 -----
35 -- Table `megadados`.`funcionario`
```

PERIGO!
Não rode
este exemplo!

Save to File...

Copy to Clipboard

Back

Next

Cancel

Atividade

Vamos implementar a base de dados dos exercícios anteriores EM SEQUENCIA (como se cada exercício tivesse acontecido em momentos diferentes) só para praticar. O exercício 5 é um desafio!

Para cada exercício:

- Implemente a base de dados dos exercícios anteriores (Faça o script)
- Em um script separado insira dados usando o comando INSERT (Desafio)

Próxima aula

- Exercícios do capítulo 4 do livro texto.
- SQL: Curso Prático, capítulo 5

Insper

www.insper.edu.br