

# Megadados

## Aula 2

Sistema de Gerenciamento de Bancos de Dados,  
Modelagem relacional

Fábio Ayres <[fabioja@insper.edu.br](mailto:fabioja@insper.edu.br)>

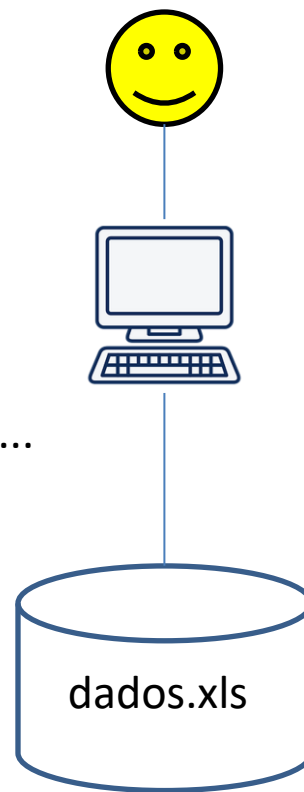
# **Sistemas de gerenciamento de bancos de dados**

# Banco de dados: porque?

- Você não precisa sempre de um banco de dados para armazenar dados! Você pode usar:
  - Um guardanapo de papel!
  - Um arquivo de texto no seu laptop!
  - Uma planilha Excel!
  - etc...
- Quando será que precisamos de um banco de dados?

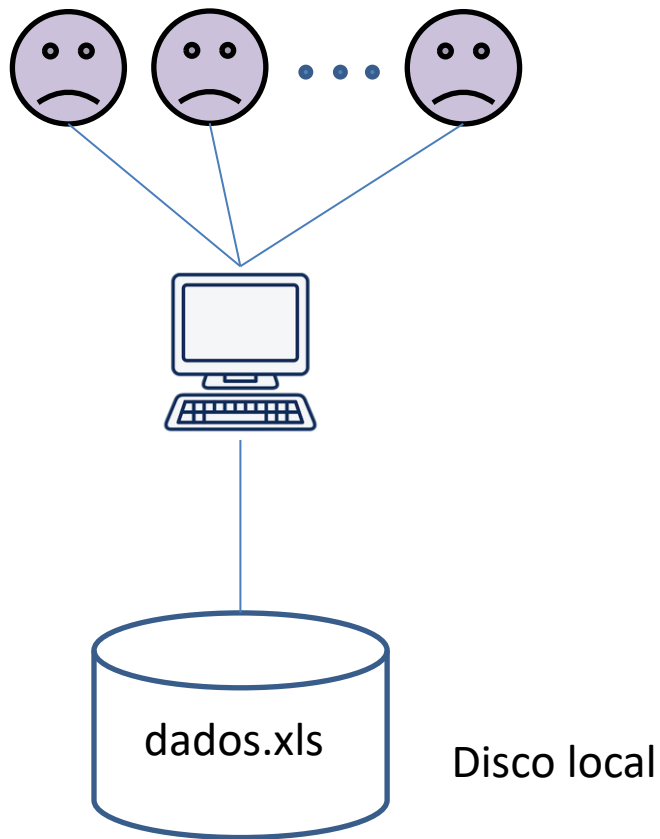
# Exemplo

Só um usuário, arquivo pequeno...  
Não precisa de banco de dados!

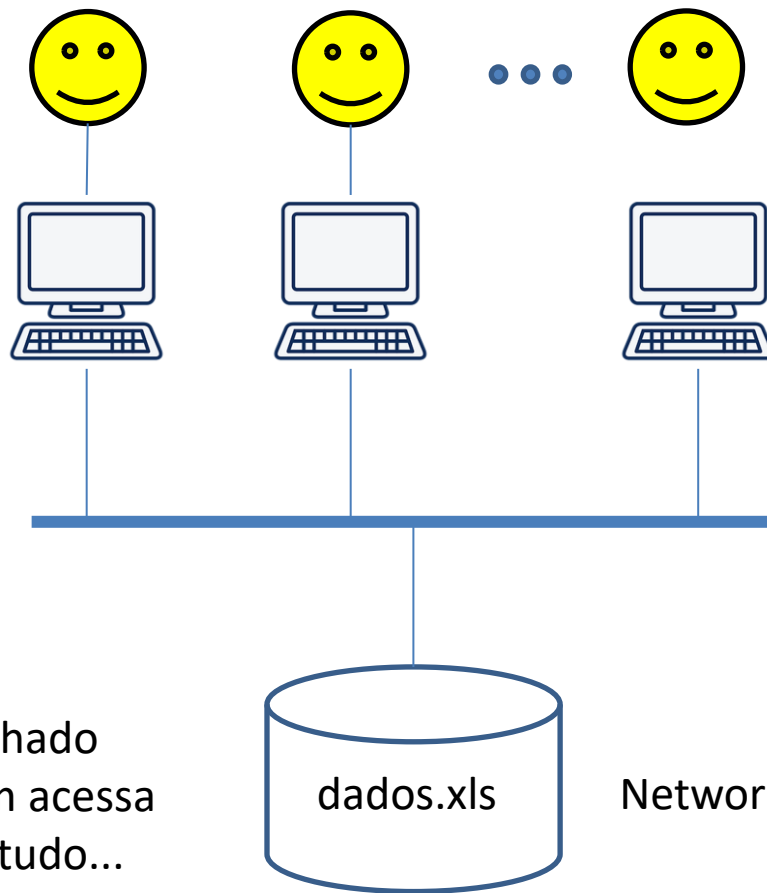


# Exemplo

Muitos usuários formando fila para conseguir acessar o terminal!

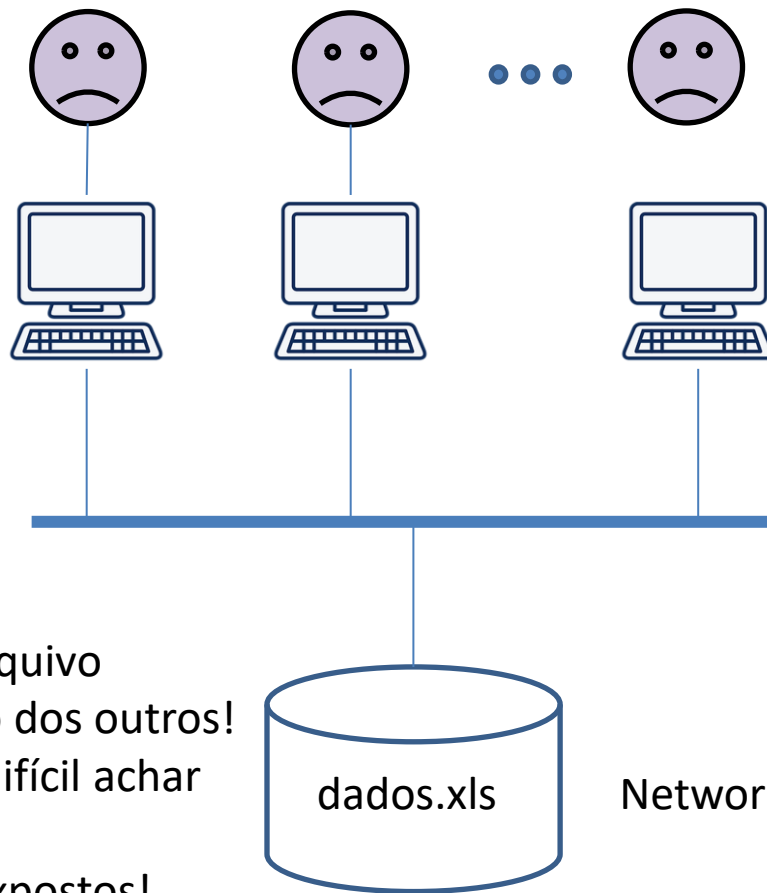


# Exemplo



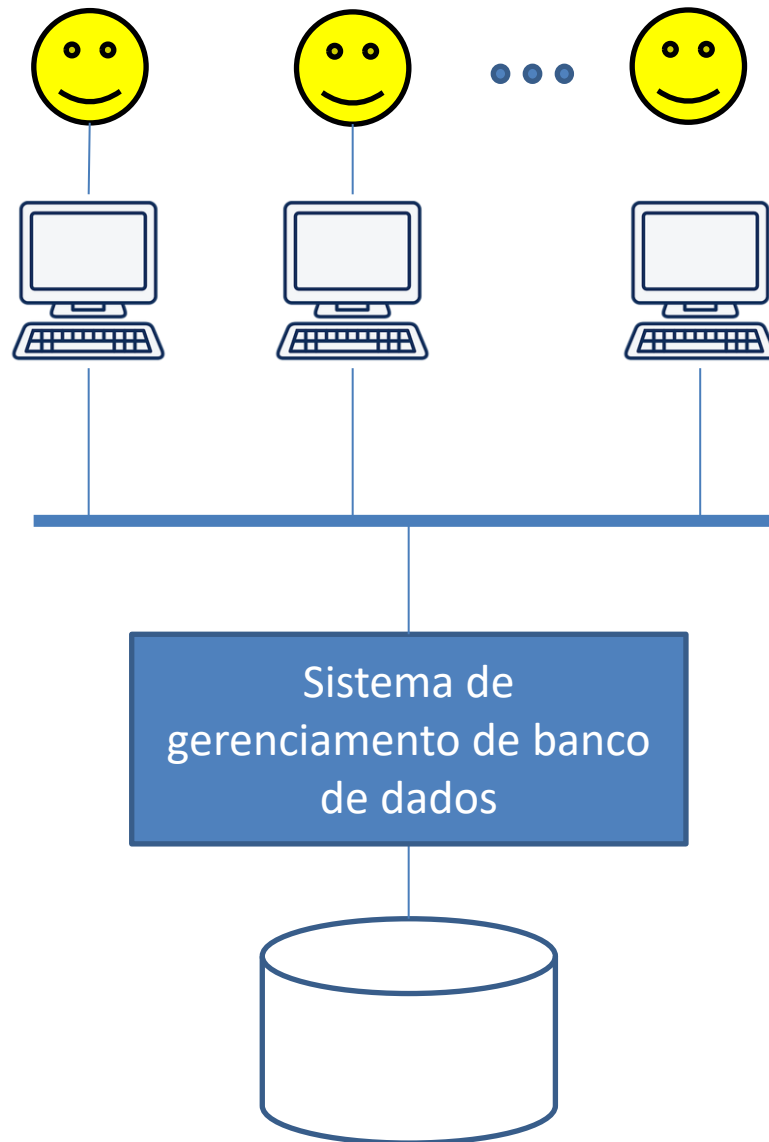
Arquivo está compartilhado na rede agora, cada um acessa com seu terminal. Contudo...

# Exemplo



- Não dá para trabalhar simultaneamente no arquivo sem estragar o trabalho dos outros!
- Dados muito grandes, difícil achar resultados
- Dados sigilosos estão expostos!

# Exemplo





# Motivos para ter um sistema de gerenciamento de banco de dados

- Tamanho

Pode não caber na RAM!

- Velocidade

Existem maneiras eficientes de armazenar e recuperar dados!

- Conveniência

O SGBD já vem com mecanismos sofisticados de consulta!

- Precisão

Um bom projeto evita redundâncias gerencia transações e mantém consistência!

- Proteção

Controle de acesso e registro de atividade!

- Robustez

Proteção contra falhas!

# Aplicações

- Vendas e estoque
- Recursos humanos e informações empresariais
- Dados científicos
- Informações geográficas
- Multimídia
- Jogos online
- Etc!



Sistema de informação

Aplicação/Serviço de consulta



Sistema de Gerenciamento de Banco de Dados

Processamento de consultas

Acesso aos dados



Armazenamento

# Tipos de banco de dados

- Relacional (também chamado de bancos de dados SQL): representa os dados usando o modelo relacional, onde dados são representados através de tabelas bidimensionais.
  - Este é o modelo mais usado em bancos de dados atualmente

# Tipos de banco de dados

- NoSQL: bancos de dados não-relacionais, dentre os quais destacam-se:
  - Key-value stores (e.g. Redis)
  - Document stores (e.g. MongoDB)
  - Column-oriented (e.g. Cassandra)

(Artigo interessante:

<http://www.dataversity.net/review-pros-cons-different-databases-relational-versus-non-relational/>)

# Tipos de banco de dados

- NewSQL: Nova geração de bancos de dados que mesclam as vantagens de alguns tipos de bancos NoSQL (como escalabilidade e disponibilidade) com garantias de consistência transacional do SQL.
  - Exemplo: Google Spanner

(Artigo interessante:

<http://www.odbms.org/blog/2018/03/on-rdbms-nosql-and-newsql-databases-interview-with-john-ryan/>)

# Modelagem relacional

# Como representar os dados?

- Estrutura
- Operações
- Restrições
- Abstração versus implementação



# Abstração

## Estrutura

- Como representar os dados de um modo que faça sentido para nós?

## Operações

- Como especificar nossa interação com os dados?
  - Consultas (em inglês: *query*, plural *queries*)
  - Modificações: criar, remover, alterar
  - CRUD: Create, Read, Update, Delete

## Restrições

- Como especificar invariantes e restrições nos dados
  - Garantir integridade e segurança

# Implementação

## Estrutura

- Como armazenar **grandes volumes de dados** de maneira eficiente?
  - Armazenamento de massa
  - Armazenamento em RAM

## Operações

- Como realizar as operações definidas na abstração de modo **eficiente**?
  - Otimização de query
  - Caching
  - Indexação

## Restrições

- Como **garantir** as restrições?
  - Chaves primárias e estrangeiras
  - Transações
  - Controle de acesso

# Modelos de dados

- Modelo não-estruturado
  - Exemplo: documentos de texto em um sistema de arquivos de larga escala
  - Hadoop Distributed File System (HDFS)
- Modelo semi-estruturado
  - Exemplo: documentos hierárquicos como XML ou JSON
  - Bancos de dados NoSQL
- Modelo estruturado
  - Exemplo: **modelo relacional**
  - Bancos de dados relacionais (“SQL”)

# Bancos de dados relacionais

Vamos focar em bancos de dados relacionais neste início de curso.

Estes bancos implementam o **modelo relacional**

Análise semântica:

**Modelo Entidade-Relacionamento**

Entidades, atributos, ...

Fundamentos matemáticos:

**Álgebra relacional**

Tuplas, conjuntos, operações, ...

Linguagem de programação:

**Structured Query Language (SQL)**

Tabelas, colunas, linhas, ...

# Modelo relacional

## *Information Retrieval*

---

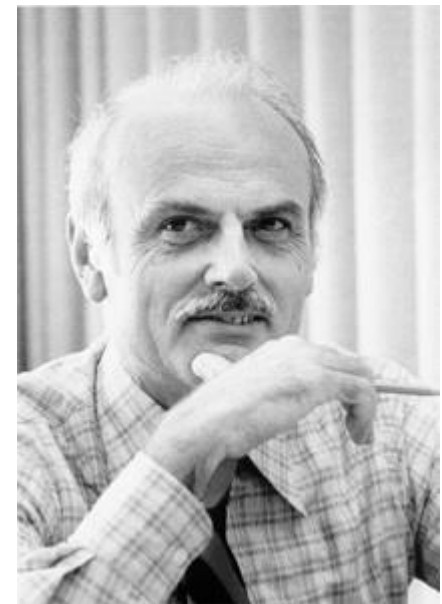
### A Relational Model of Data for Large Shared Data Banks

E. F. Codd

*IBM Research Laboratory, San Jose, California*

Future users of large data banks must be protected from having to know how the data is organized in the machine (the internal representation). A prompting service which supplies such information is not a satisfactory solution. Activities of users at terminals and most application programs should remain unaffected when the internal representation of data is changed and even when some aspects of the external representation are changed. Changes in data representation will often be needed as a result of changes in query, update, and report traffic and natural growth in the types of stored information.

Communications of the ACM,  
v. 13(6), pp. 377-387, Junho  
1970



Edgar Frank Codd  
Recebeu o Turing Award em 1981

# Modelo relacional

- Relação: basicamente tabelas onde a ordem das linhas não importa

Atributo : tipo

Nome da relação

Tabela “comidas”

<u>Comida</u> :VARCHAR(20)	Preço:DECIMAL(10,2)	Pais:VARCHAR(20)
Feijoada	25,00	Brasil
Macarrão	15,00	China
Surströmming	NULL	Suécia

Em linguagem de bancos de dados relacionais: linha  
Em linguagem de teoria do modelo relacional: tupla

# Modelo relacional

Formalmente:

- **Relação**: conjunto de tuplas
- **Tupla**: lista de atributos
- **Atributos**: valores, associados a tipos
- **Tipos**: conjuntos de valores possíveis para um atributo
- **Chave**: atributo identificador da tupla
  
- **Esquema de relação**: nome da relação e lista de nomes de atributos

chave



Comidas (Comida, Preço, País)

- **Esquema do banco de dados**: conjunto de esquemas de relação

# Exercícios

Escreva os esquemas de relação para as seguintes situações:

1. Bicho de estimação
2. Alunos do curso
3. Cardápio do restaurante



# Chaves

A chave é um atributo usado para identificar dados

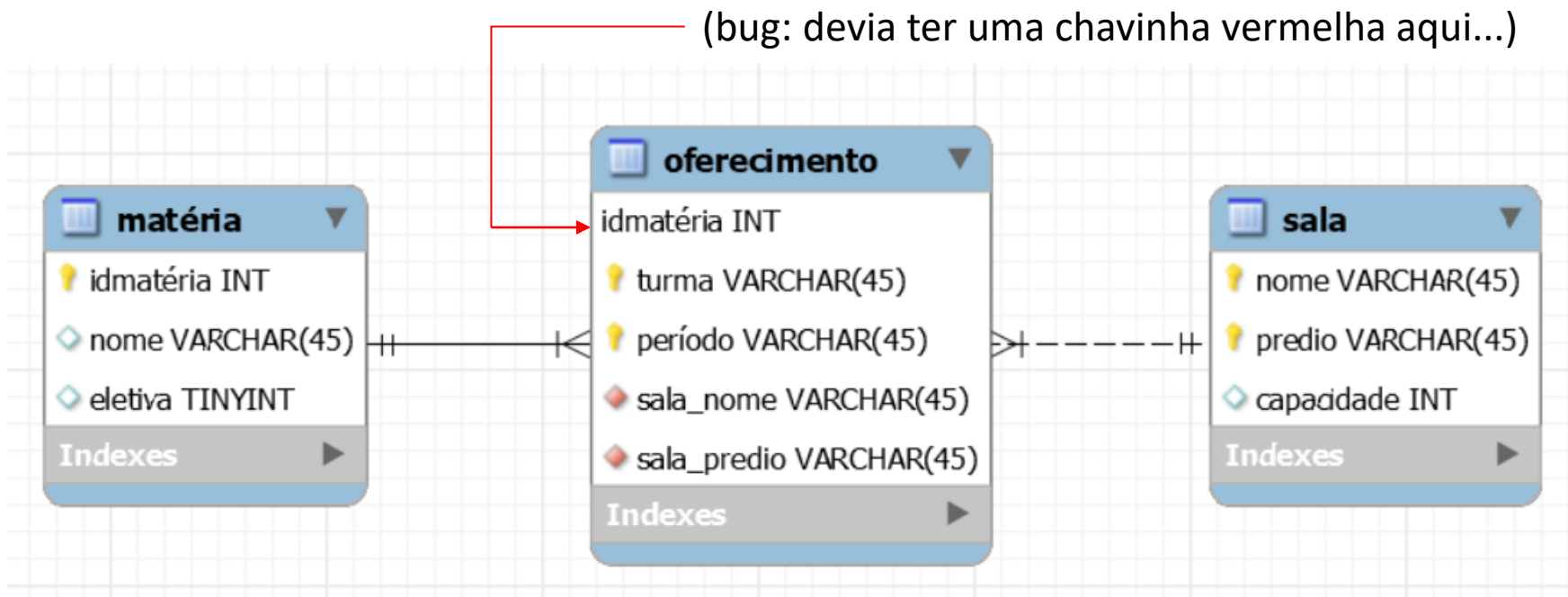
Chave primária: o R.G. da tupla!

- Atributo (ou conjunto de atributos) que permite **identificar unicamente** uma ocorrência de uma tupla em uma tabela.
  - Deve ser **único** para cada linha
  - Deve ser **imutável**
  - Os demais atributos acabam por depender deste
- Caso nenhum atributo (ou conjunto de atributos) satisfaça estes requisitos, devemos criar um atributo artificial (geralmente um id inteiro).

## Exercício

Nas entidades obtidas anteriormente, quais são as chaves primárias?

# Diagrama do modelo relacional



Schema:

matéria (idmatéria, nome, eletiva)

oferecimento (idmatéria, turma, período, sala\_nome, sala\_prédio)

sala (nome, prédio, capacidade)

# (bug no MySQL Workbench...)

MySQL Bugs: #92141: MySQL Wo x +

bugs.mysql.com/bug.php?id=92141

MySQL.

Developer Zone	Bugs Home	Report a bug	Statistics	Advanced search	Saved searches	Tags
<b>Bug #92141</b>		<b>MySQL Workbench Model does not display Primary Foreign Keys icon (red key)</b>				
Submitted:		23 Aug 2018 4:45				
Reporter:		<a href="#">Dave Eccles</a>				
Status:		Verified				
Category:		MySQL Workbench: Modeling				
Version:		8.0.12				
Assigned to:						
Tags:		<a href="#">Windows Mac OSX Modeling Workbench PFK</a>				

# Chaves

Chave estrangeira: um atributo de uma tabela que é chave primária de outra tabela, permitindo assim relacionar itens entre tabelas.

Uma chave estrangeira pode também ser chave primária da tabela: relacionamento identificador

Mas isso não é necessário, e na maioria dos casos não ocorre: relacionamento não-identificador

## Exercício

Pense em um par de entidades que se relacionam. Indique as chaves primárias e estrangeiras

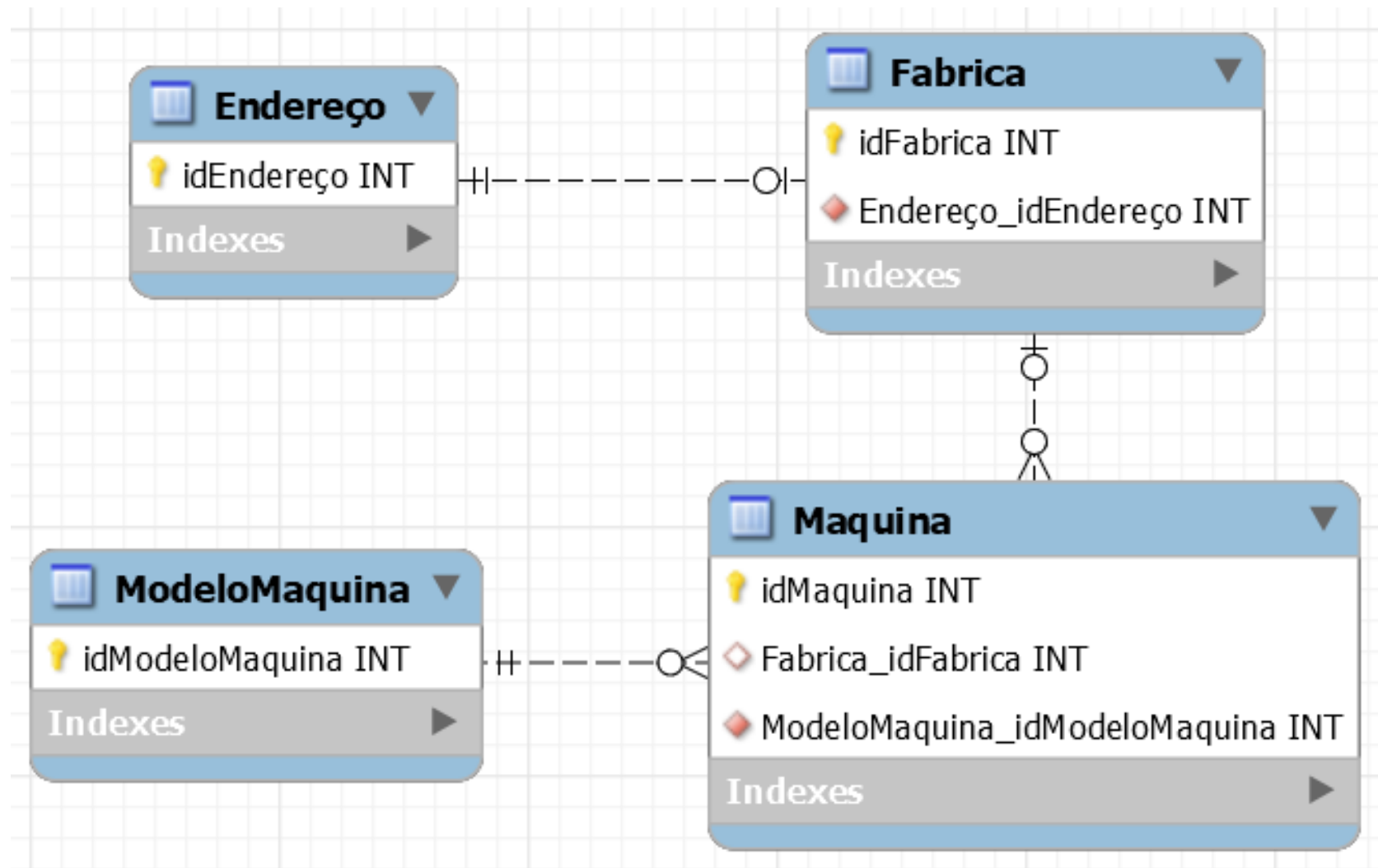
# Relacionamento

## Conexão entre entidades

- 1:1 – Entidades que se relacionam diretamente. Se você tem relações 1:1 no seu diagrama, talvez valha a pena integrar as entidades.
- 1:n – Cada linha em uma entidade A se relaciona com n linhas na outra entidade B, mas cada linha de B se relaciona apenas com uma entidade de A

Ambos os casos são implementados simplesmente com o uso de chave estrangeira

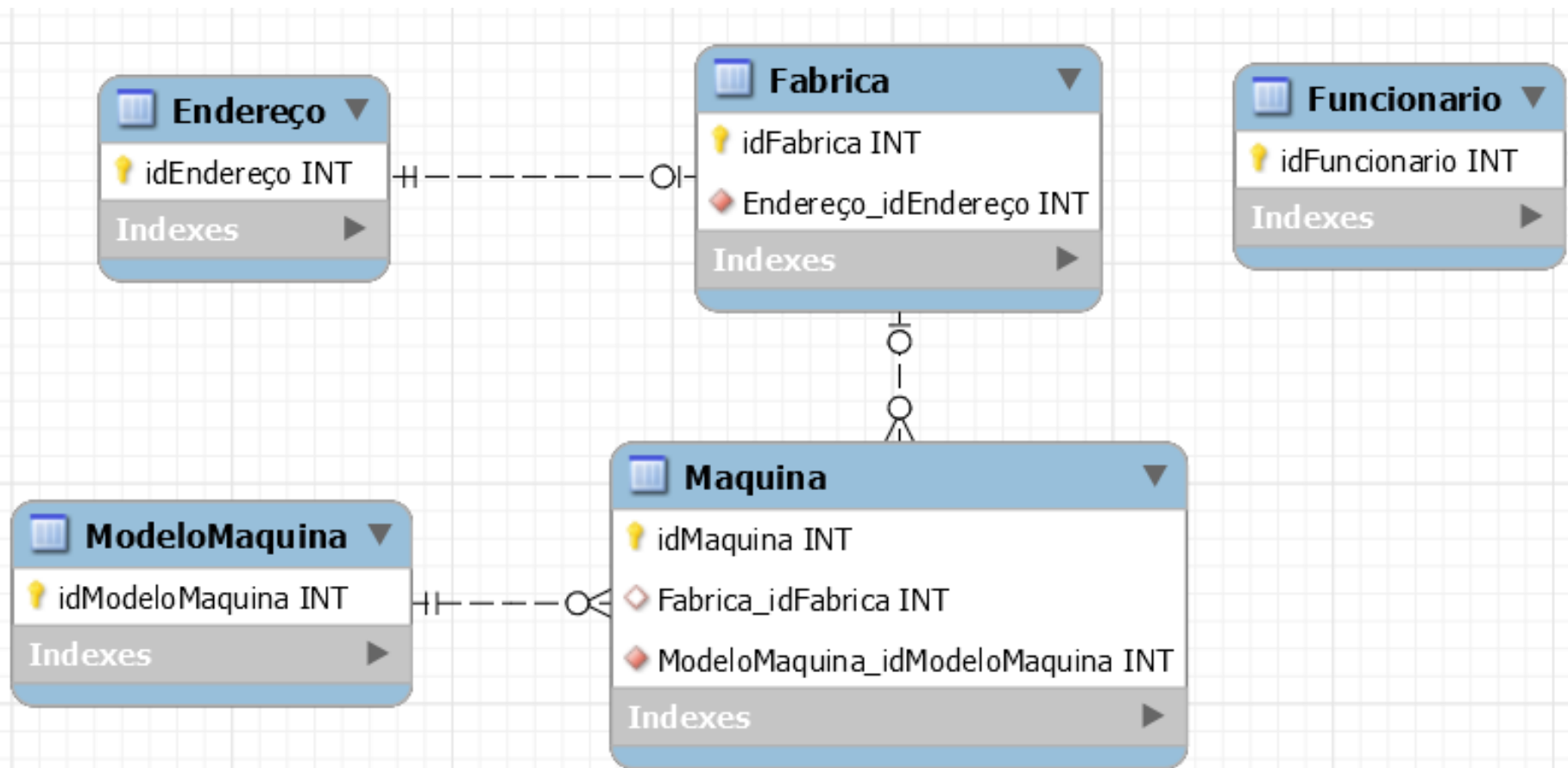
# Relacionamento





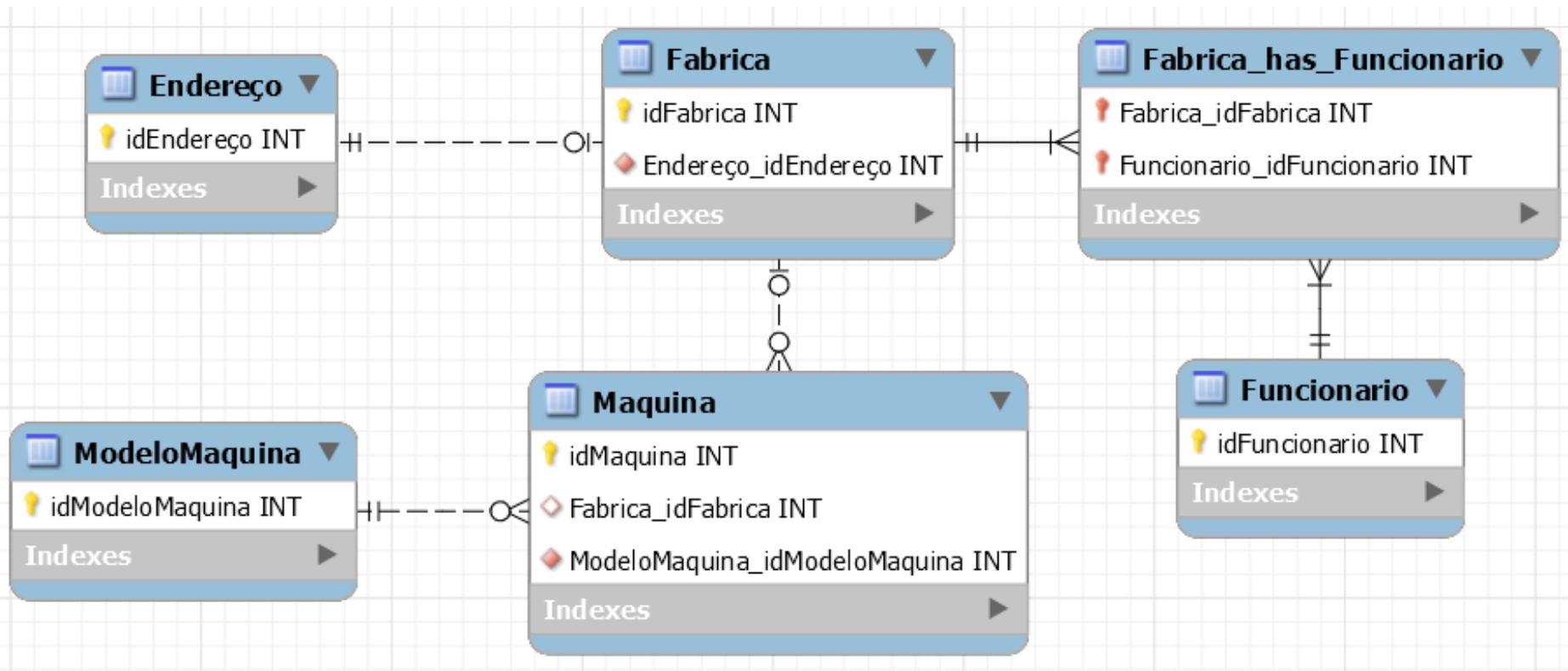
# Relacionamento m:n

Como fazer um relacionamento m:n?



# Relacionamento m:n

Solução: usar uma entidade-relacionamento!



# Exercício

Construa a relação entre as seguintes entidades:

- Disciplinas e alunos
- Prato de comida e ingredientes
- Prato de comida e adjetivos (vegano ou não, com ou sem glúten, dietético ou não, etc) –  
Pense se precisamos realmente de outras tabelas aqui.

# Atividade

Vamos criar um banco de dados para um *food truck*.

O analista de dados entrevistou o cliente e preparou o seguinte schema:

Ingrediente (nome, tipo, calorias, preço\_unitário, quantidade)

Prato (nome, preço)

Ingrediente\_Prato (nome\_prato, nome\_ingrediente, quantidade)

1. Desenhe o diagrama do modelo relacional no MySQL Workbench
2. Pesquise como criar e popular essas tabelas usando a linguagem SQL – vamos abordar esse assunto em detalhes na próxima aula

# Insper

[www.insper.edu.br](http://www.insper.edu.br)