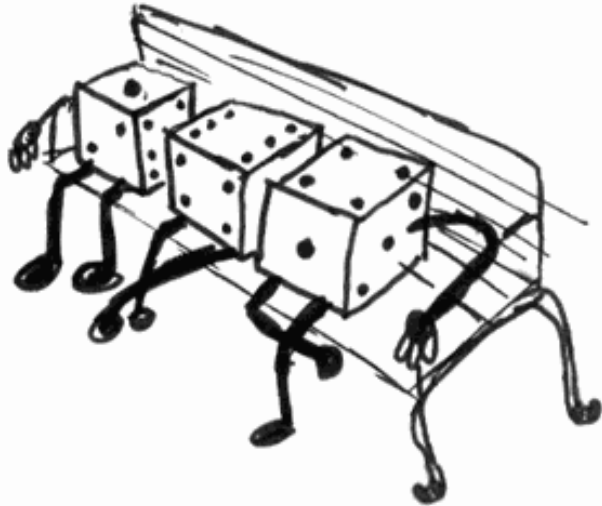
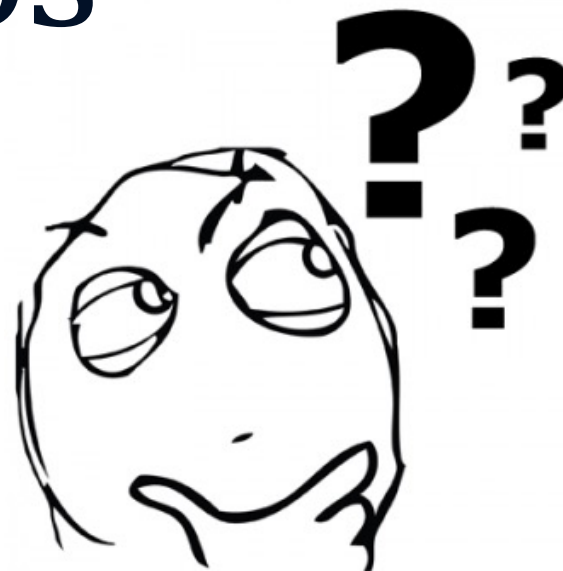


O BANCO DE DADOS

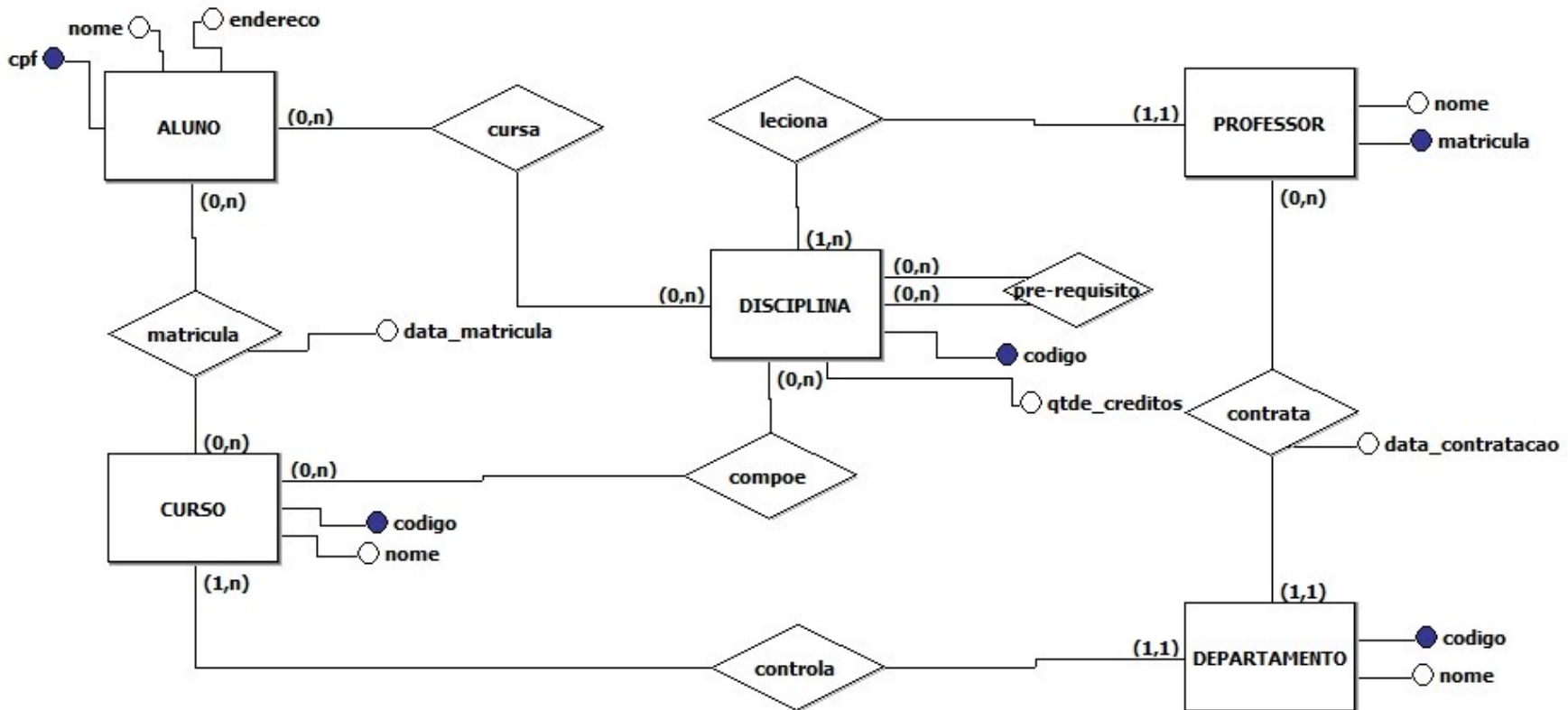


Banco de Dados

Modelo Relacional



CORREÇÃO DO DESAFIO



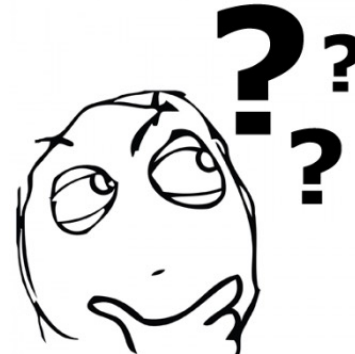
CORREÇÃO DO DESAFIO

ENTIDADES:

- **ALUNO**
 - o CPF- Chave Primária
 - o NOME
 - o ENDERECO
- **CURSO**
 - o CODIGO- Chave Primária
 - o NOME
- **DISCIPLINA**
 - o CODIGO- Chave Primária
 - o QTDE_CREDITOS
- **PROFESSOR**
 - o NOME
 - o MATRICULA- Chave Primária
- **DEPARTAMENTO**
 - o CODIGO- Chave Primária
 - o NOME

RELACIONAMENTOS:

1. MATRICULA
 - a. DATA_MATRICULA
2. CURSA
3. LECIONA
4. COMPOE
5. CONTROLA
6. CONTRATA
 - a. DATA_CONTRATACAO
7. PRE-REQUISITO



CORREÇÃO DO DESAFIO

Uma escola decide criar uma base de dados de seus departamentos, alunos, professores, disciplinas e cursos. Para qualquer aluno sabemos CPF, nome e endereço. Para qualquer curso temos o código e nome. Qualquer disciplina temos código e quantidade de credito. Qualquer professor temos nome e matricula. Por último, para qualquer departamento temos código e nome.

- Um aluno pode se matricular em vários cursos assim como em um curso podem ser matriculados vários alunos, sendo necessário especificar a data da matricula.
- Um departamento pode contratar vários cursos como também vários professores sendo necessário especificar a data de contratação desses professores. Um curso ou um professor só podem ser contratados por um único departamento.
- Um professor pode lecionar várias disciplinas para os vários alunos que cursam essas disciplinas e algumas disciplinas podem apresentar pré-requisitos de outras várias disciplinas.
- Uma disciplina só pode ser lecionada por apenas um professor.
- Um curso pode ser composto com várias disciplinas assim como uma disciplina pode compor vários cursos.

DESAFIO

```
ALUNO { cpf, nome, endereco}

Chave Primaria: cpf

CURSO { codigo, nome, codigo_depto }

Chave Primaria: codigo
Chave Estrangeira: codigo_depto referencia DEPARTAMENTO


MATRICULA { cpf, codigo, data_matricula }

Chave Primaria: cpf, codigo
Chave Estrangeira: cpf referencia ALUNO
Chave Estrangeira: codigo referencia CURSO


DISCIPLINA { codigo, qtde_creditos, matricula }

Chave Primaria: codigo
Chave Estrangeira: matricula referencia PROFESSOR


CURSA {cpf, codigo }

Chave Primaria: cpf, codigo
Chave Estrangeira: cpf referencia ALUNO
Chave Estrangeira: codigo referencia DISCIPLINA


COMPOE { codigo_curso, codigo_disciplina }

Chave Primaria: codigo_curso, codigo_disciplina
Chave Estrangeira: codigo_disciplina referencia DISCIPLINA
Chave Estrangeira: codigo_curso referencia CURSO


PROFESSOR { matricula, nome, codigo_departamento, data_contratacao }

Chave Primaria: matricula
Chave Estrangeira: codigo_departamento referencia DEPARTAMENTO


DEPARTAMENTO { codigo, nome }

Chave Primaria: codigo


PRE_REQUISITO {codigo, codigo_disciplina_dependencia}

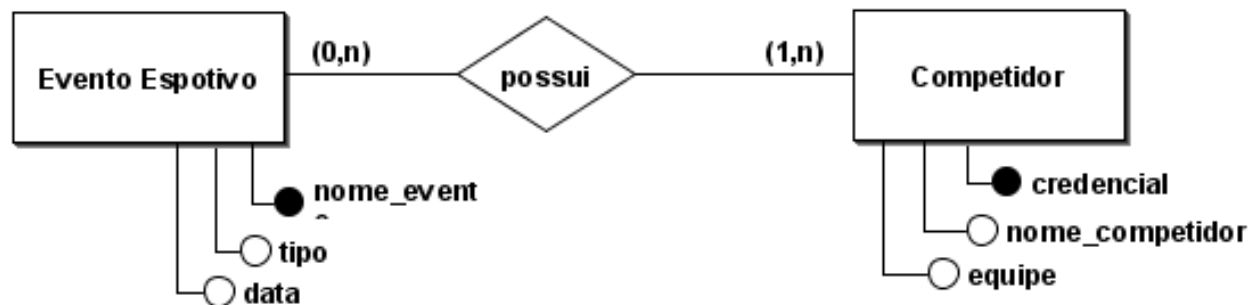
Chave Primaria: codigo, codigo_disciplina_dependencia}
Chave Estrangeira: codigo referencia DISCIPLINA
Chave Estrangeira: codigo_disciplina_dependencia} referencia DISCIPLINA
```

CORREÇÃO DO EXERCÍCIO DE DER

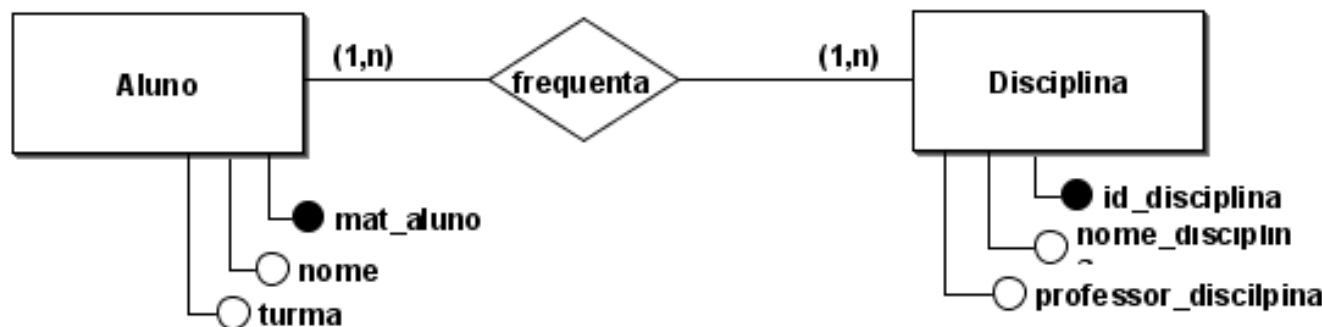
1. Uma universidade tem muitos estudantes e um estudante pode se dedicar a no máximo uma universidade.
2. Uma aeronave pode ter muitos passageiros, mas um passageiro só pode estar em um voo de cada vez.
3. Uma nação possui vários estados, e um estado, muitas cidades. Um estado só poderá estar vinculado a uma nação e uma cidade só poderá estar vinculado a um estado.
4. Um encontro de eventos esportivos pode ter muitos competidores e um competidor pode participar de mais de um evento.
5. Um paciente pode ter muitos médicos e um médico muitos pacientes.
6. Um aluno pode frequentar mais de uma disciplina e uma mesma disciplina pode ter muitos alunos.

CORREÇÃO DO EXERCÍCIO DE DER

Um encontro de eventos esportivos pode ter muitos competidores e um competidor pode participar de mais de um evento



Um aluno pode frequentar mais de uma disciplina e uma mesma disciplina pode ter muitos alunos



O que veremos hoje ?

5. Modelo Relacional

Tuplas e Tabelas

Tipos de dados suportados

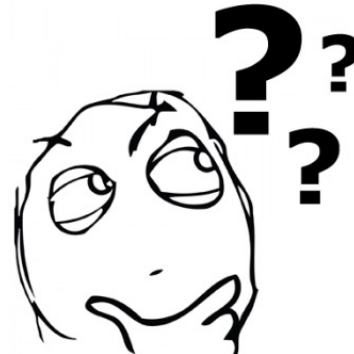
Chave estrangeira



UM POUCO DE HISTÓRIA

O Modelo Relacional foi introduzido por Edgar Frank Codd (1970) e tornou-se um padrão para aplicações comerciais, devido a sua simplicidade e desempenho.

É um modelo formal, bastante representativo e ao mesmo tempo bastante simples, foi o primeiro modelo de dados descrito teoricamente.



MODELO RELACIONAL

- O modelo Relacional é um modelo lógico, utilizado em bancos de dados relacionais.
- Ele tem por finalidade representar os dados como uma coleção de tabelas e cada linha (**tupla**) de uma tabela representa uma coleção de dados relacionados.
- Neste modelo, começamos a nos preocupar em como os dados devem ser armazenados e em como criaremos os relacionamentos do modelo conceitual.
- É também nessa etapa que definimos o SGBD que será utilizado, bem como os tipos de dados para cada atributo.
- O modelo Relacional é definido usando como base o MER.

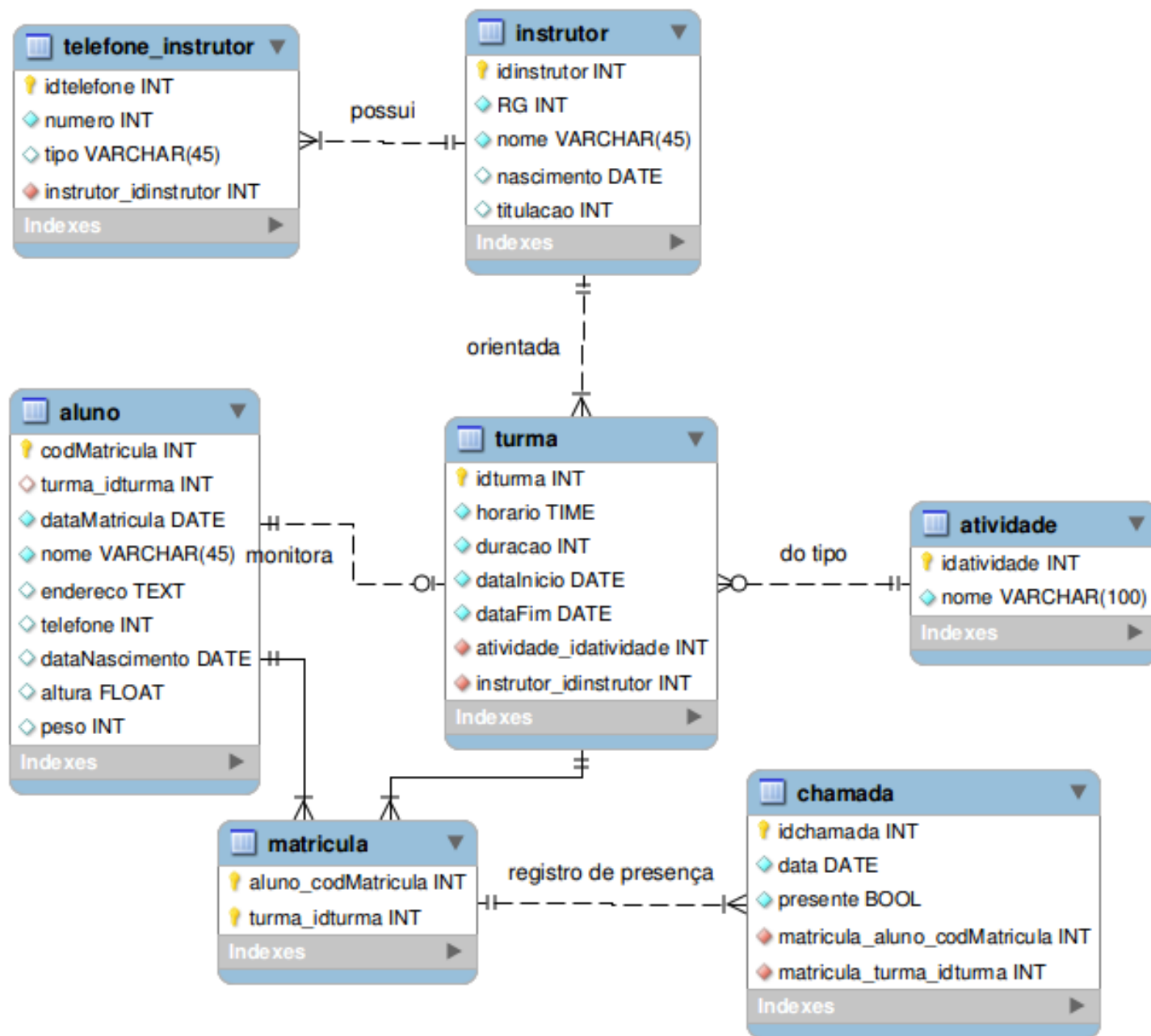
MODELO RELACIONAL

O Modelo Relacional representa os dados num Banco de Dados como uma **coleção de tabelas** e seus **relacionamentos**.

Cada tabela contém um nome e um conjunto de **atributos** com seus respectivos nomes.

No modelo relacional os atributos precisam ter um domínio definido, ou seja, precisamos especificar todos os valores possíveis que um atributo pode receber.

O domínio de um atributo define qual o tipo de dado e o formato que o dado pode ser armazenado por aquele atributo



TABELAS

- Para descrever uma tabela no modelo relacional, usamos o nome da tabela seguida dos atributos entre parênteses.
- Para identificar a chave primária, devem-se sublinhar o(s) atributo(s) correspondente(s) a ela.
- O tipo de cada atributo também deve aparecer no modelo relacional, como mostra o exemplo abaixo:

```
tbAluno (matricula_aluno: inteiro, nome_aluno: caracter(100),  
data_nasc_aluno: data)
```

TUPLAS

- As tuplas representam os valores de uma tabela.
- Note que as colunas da tabela representam os atributos, enquanto as linhas representam os registros. Se uma tabela não tiver tuplas, ela estará vazia, ou seja, sem dados.

matricula_aluno	nome_aluno	data_nasc_aluno
100	Ana	12/05/1997
101	João	15/04/1996
102	Maria	22/06/1998
103	José	03/01/1997
104	Marcos	19/03/1996

TIPOS DE DADOS

- Todo campo/valor é de um tipo específico
- SGBD trabalham basicamente com 3 categorias*:
 - Tipos numéricos
 - Tipos de data e hora
 - Tipos literais

NOME: texto	IDADE: inteiro
João	12
Carlos	20
Renata	17

TIPOS DE DADOS - Convenções

Vamos usar as seguintes convenções para facilitar:

- M - indicará a largura que pode ser armazenada de determinado tipo (exemplo: `INTEGER(M)`), armazenará um inteiro de M dígitos)
- D - indicará o número de dígitos da parte decimal de um valor (exemplo: `DECIMAL(M , D)`), armazenará um decimal de M dígitos na parte inteira e D dígitos na parte decimal)
- $[]$ - indicará uma parte opcional (exemplo: `INTEGER[(M)]`, armazenará um inteiro de M dígitos, mas (M) é opcional na definição)

TIPOS DE DADOS - Numéricos

Tipo	Descrição	Padrão
BIT[(M)]	Armazena bits, onde M é a quantidade de bits	1
TINYINT[(M)]	De -128 a 127	3
BOOL, BOOLEAN	Sinônimo de TINYINT(1), onde zero é falso e não-zero é verdadeiro	
SMALLINT[(M)]	De -32768 a 32767	5
MEDIUMINT[(M)]	De -8388608 a 8288607	7
INT[(M)]	De -2147483648 a 2147483647	10
INTEGER[(M)]	Sinônimo de INT	10
BIGINT[(M)]	De -92233720368544775808 a 92233720368544775807	20
FLOAT[(M,D)]	Até 7 casas decimais	Limite
DOUBLE[(M,D)]	Até 15 casas decimais	Limite
DECIMAL[(M[,D])]	Até 30 casas decimais	10,0
DEC[(M[,D])]	Sinônimo de DECIMAL	10,0

TIPOS DE DADOS - Numéricos

- Idade TINYINT UNSIGNED
 - ✓ Idade é do tipo TINYINT, apresenta apenas números positivos de 0 a 255
 - $[\text{valor abs}(-128-127) = 255]$
- Salario DECIMAL(5,2) ZEROFILL
 - ✓ Salario é do tipo DECIMAL, apresenta até 5 dígitos na parte inteira e 2 na parte decimal, completa o restante com zeros (exemplo, 00700.54)
- Demitido BOOLEAN
 - ✓ Demitido é do tipo BOOLEAN, apresenta 0 (false) enquanto estiver trabalhando e não-zero (true) quando for demitido

TIPOS DE DADOS – Data e Hora

Tipo	Descrição
DATE	Guarda data no formato YYYY-MM-DD, de 1001-01-01 até 9999-12-31
DATETIME	Combinação de data e hora no mesmo campo, no formato YYYY-MM-DD HH:MM:SS, de 1001-01-01 00:00:00 até 9999-12-31 23:59:59
TIMESTAMP	Guarda número de segundos desde o EPOCH, 1970-01-01 até 2038-01-09
TIME	Armazena tempo no formato HH:MM:SS, de -838:59:59 até 838:59:59
YEAR[(2 4)]	O ano em 2 ou 4 (padrão) dígitos, de 1901 a 2155 para 4 dígitos, ou de 70 a 69 (1970 a 2069) para 2 dígitos

TIPOS DE DADOS – Literais

Tipo	Descrição	Padrão
CHAR[(M)]	Texto de tamanho fixo, de 0 a 255, preenche o que sobrar com espaços em branco	1
VARCHAR(M)	Texto de tamanho variável, de 0 a 65535	Usuário
BINARY(M)	Mesmo que CHAR, mas armazena texto binário em byte, onde M é o tamanho em bytes (geralmente um texto em memória)	Usuário
VARBINARY(M)	Mesmo que CHAR, mas armazena texto binário em byte, onde M é o tamanho em bytes (geralmente um texto em memória)	Usuário

TIPOS DE DADOS – Literais

- CPF CHAR(14)
 - ✓ Um texto fixo de 14 posições
- Nome VARCHAR(255)
 - ✓ Um texto variável de até 255 posições
- Texto TEXT
 - ✓ Um texto de até $65.535 * (2^{16} - 1)$ caracteres
- Sexo ENUM('M', 'F')
 - ✓ Um texto que deverá aceitar APENAS 'M' ou 'F'

RESUMINDO ...

Basicamente utilizamos de modo geral os seguintes tipos:

Tipo	Descrição
VARCHAR	Valores no campo VARCHAR são strings de tamanho variável. Você pode declarar um campo VARCHAR para ter qualquer tamanho entre 1 e 255, assim como para campo CHAR. No entanto, diferente de CHAR, valores VARCHAR são armazenados usando apenas quantos caracteres forem necessários, mais 1 byte para gravar o tamanho.
INT	Valores inteiros de -2147483648 a 2147483647.
DECIMAL	O tipo DECIMAL é usado por valores para os quais é importante preservar a exatidão como, por exemplo, dados monetários.
DATA	O tipo DATA é usado quando se necessita apenas do valor da data, sem a parte da hora. MySQL recupera e mostra valores do tipo DATA no formato 'ano-mm-dd'

EXERCÍCIO DE 5 min

Adicione o tipo
de dado em cada
campo/coluna
apropriadamente

ORDEM_SERVICO

NUMERO
DATA_ABERTURA
HORA_ABERTURA
CLIENTE_CPF (FK)
TECNICO_MATRICULA (FK)
PROBLEMA
SOLUCAO
DATA_FECHAMENTO
HORA_FECHAMENTO

CLIENTE

CPF
NOME
ENDERECO
TELEFONE

TECNICO

MATRICULA
NOME
ENDERECO
TELEFONE
NIVEL_ESCOLARIDADE

SOLUÇÃO

ORDEM_SERVICO

NUMERO **INT(10)**

DATA_ABERTURA **DATE(10)**

HORA_ABERTURA **TIME(9)**

CLIENTE_CPF (FK) **VARCHAR(14)**

TECNICO_MATRICULA (FK) **INT(9)**

PROBLEMA **VARCHAR(1000)**

SOLUCAO **VARCHAR(1000)**

DATA_FECHAMENTO **DATE(10)**

HORA_FECHAMENTO **TIME(9)**

CLIENTE

VARCHAR(14)

CPF

VARCHAR(45)

NOME

VARCHAR(45)

ENDERECO

VARCHAR(15)

TELEFONE

TECNICO

MATRICULA **INT(9)**

NOME **VARCHAR(45)**

ENDERECO **VARCHAR(45)**

TELEFONE **VARCHAR(15)**

NIVEL_ESCOLARIDADE **VARCHAR(15)**

CHAVE ESTRANGEIRA

- (Foreign Key – FK) - é um atributo da tabela, que faz referencia a uma chave primaria de outra tabela ou da própria tabela.
- O atributo que é chave estrangeira deve ser do mesmo tipo e do mesmo tamanho que sua primária correspondente. É importante deixar explicito a qual tabela a chave estrangeira está fazendo referência.
- Uma chave estrangeira sempre faz referencia a uma chave primária. A chave estrangeira nunca fará referencia a um atributo que não seja uma chave primária.

CHAVE ESTRANGEIRA

Suponha que tenhamos as tabelas “Turma” e “Aluno”, observe que a tabela **tbAluno** possui um atributo **código_turma**. Esse atributo é chave primária na tabela **tbTurma** e, portanto, é uma chave estrangeira na tabela **tbAluno**.

```
tbTurma (código_turma: inteiro, nome_turma: caracter(5))
```

```
tbAluno (matricula_aluno: inteiro, nome_aluno: caracter(100),  
data_nasc_aluno: data, código_turma: inteiro)
```

```
código_turma referencia tbTurma
```

CHAVE ESTRANGEIRA

No Modelo Relacional é a chave estrangeira que especifica o relacionamento entre as tabelas. É através da chave estrangeira que conseguimos descobrir, por exemplo, que o aluno João pertence a turma do 2º ano do Curso Técnico em Informática.

matricula_aluno	nome_aluno	data_nasc_aluno	código_turma
100	Ana	12/05/1997	1
101	João	15/04/1996	3
102	Maria	22/06/1998	2
103	José	03/01/1997	1
104	Marcos	19/03/1996	3

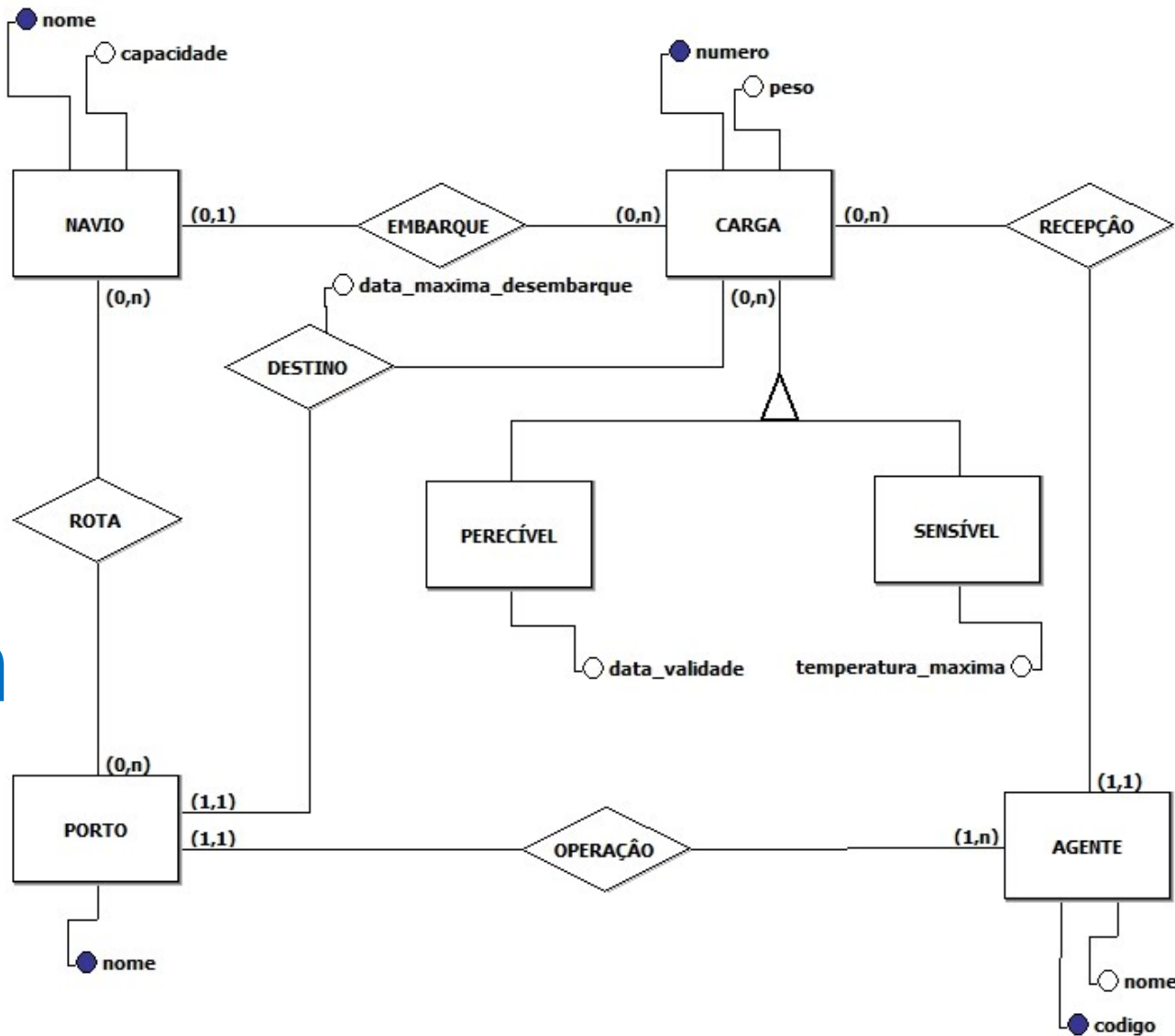
codigo_turma	nome_turma
1	1º Informática
2	2º Informática
3	3º Informática
4	1º Enfermagem

EXEMPLO

MINI MUNDO 4 - EMPRESA DE TRANSPORTE MARÍTIMO

Construa o
modelo
relacional
correspondente.

10min



EXEMPLO - Solução

NAVIO {nome, capacidade}

CHAVE PRIMARIA: nome

CARGA {numero, peso, nome_navio, codigo_agente, nome_porto, data_maxima_desembarque}

chave primária: numero

CHAVE ESTRANGEIRA: nome_navio referencia NAVIO

CHAVE ESTRANGEIRAa: codigo_agente referencia AGENTE

CHAVE ESTRANGEIRA: nome_porto referencia PORTO

PERECIVEL {numero, data_validade}

CHAVE PRIMÁRIA: numero

CHAVE ESTRANGEIRA: numero referencia CARGA

SENSIVEL {numero, tempertatura}

CHAVE PRIMÁRIA: numero

CHAVE ESTRANGEIRA: numero referencia CARGA

PORTO {nome}

CHAVE PRIMÁRIA: nome

AGENTE {codigo, nome, nome_porto}

CHAVE PRIMARIA: codigo

CHAVE ESTRANGEIRA: nome_porto referencia PORTO

ROTA {nome_navio, nome_porto}

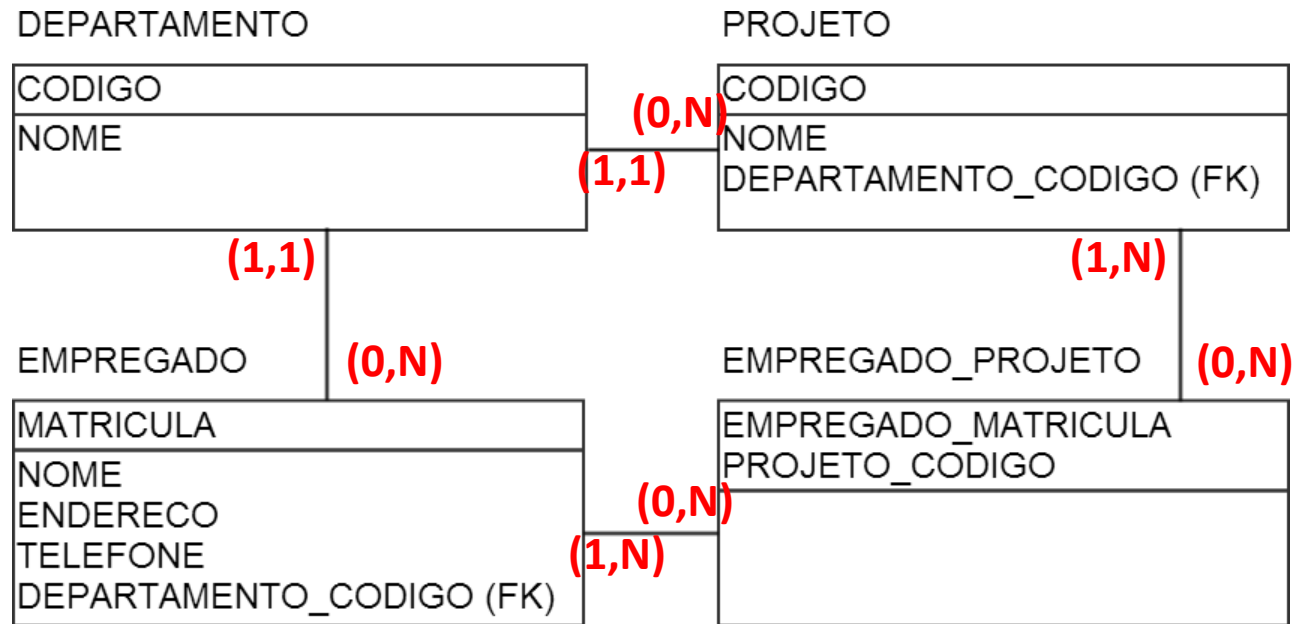
CHAVE PRIMARIA: nome_navio, nome_porto

CHAVE ESTRANGEIRA: nome_navio referencia NAVIO

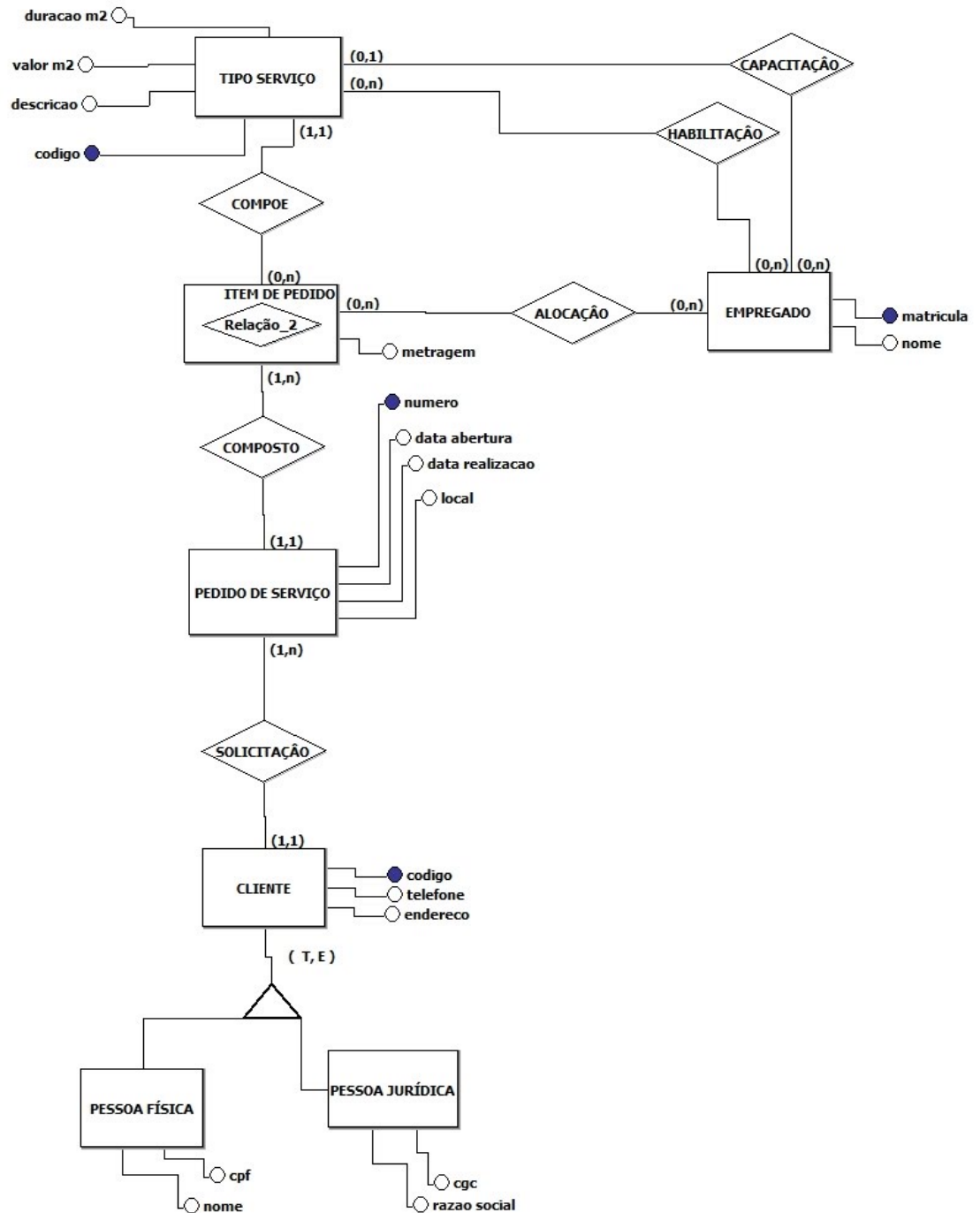
CHAVE ESTRANGEIRA: nome_porto referencia PORTO

EXERCICIO

Adicione o tipo de dado em cada campo/coluna apropriadamente



DESAFIO



PARA PRÓXIMA AULA ...

Na nossa próxima aula contaremos com uma oficina prática do MySQL Workbench. Quem tiver computador já pode baixar o software. Aos que tem apenas celulares, vamos procurar apps que similares ao funcionamento do computador.

