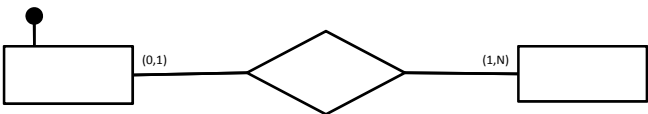


Ministério
FIRJAN INFORMA, FORMA, TRANSFORMA.



Banco de Dados

Fabício Curvello Gomes

Ministério
FIRJAN INFORMA, FORMA, TRANSFORMA.






Chaves, Relacionamentos e
Cardinalidades no MySQL

INFORMA, FORMA, TRANSFORMA.


Chaves

Em um banco de dados, podemos trabalhar com chaves, que podem ser:



- Chave Primária
- Chave Estrangeira
- Chave Alternativa (que também pode ser chamada de Chave Candidata)



Já vimos estes conceitos em **Modelagem de Banco de Dados**. Caso queira relembrar, consulte o arquivo “**BD – TI – 03 – Abordagem Relacional**”.



3


INFORMA, FORMA, TRANSFORMA.

Chave Primária



No **MySQL**, já estamos trabalhando com vários exercícios contendo tabelas com chaves primárias.

A cláusula `PRIMARY KEY` já foi utilizada diversas vezes, sempre que foi criada uma nova tabela. Este assunto já está explicado perfeitamente nos arquivos anteriores.

Qualquer dúvida, reveja as tabelas que foram criadas desde o arquivo “**BD - TI - 05 - Criando um Banco de Dados no MySQL**”.



4

  INFORMA, FORMA, TRANSFORMA.


Chave Estrangeira



No **MySQL**, ainda não trabalhamos com Chave Estrangeira.

A cláusula que faz referência à uma chave estrangeira é a `FOREIGN KEY`.

Veremos mais adiante como se trabalha no **MySQL** com as chaves estrangeiras.

Vamos, antes disso, concluir toda esta teoria conceitual.

 5

  INFORMA, FORMA, TRANSFORMA.


Chave Alternativa



Também não estudamos ainda no **MySQL** como se trabalha com as chaves alternativas.

A chave alternativa também pode ser chamada de **chave candidata**.

Para se criar uma chave candidata no **MySQL**, utiliza-se a cláusula `UNIQUE`.

Conforme dito anteriormente, veremos após esta etapa conceitual, como se trabalha no **MySQL** com as chaves alternativas.

 6

  INFORMA, FORMA, TRANSFORMA.


Cardinalidades



Conforme também vimos na modelagem de BD, as **cardinalidades máximas** entre os relacionamentos podem ser:

- Relacionamento de 1 para 1 – (1:1)
- Relacionamento de 1 para N (ou de N para 1) – (1:N)
- Relacionamento de N para N – (N:N)

As cardinalidades serão importantíssimas no momento de criação de relacionamentos no **MySQL**.

Caso tenha esquecido, considere o fato de dar uma revisada no arquivo “**BD - TI - 02 - Abordagem Entidade-Relacionamento**”

7


  INFORMA, FORMA, TRANSFORMA.

Até aqui...

Percebe-se que a modelagem de BD é fundamental no processo de elaboração de projetos de BD.

Toda a parte conceitual necessária à criação dos modelos será utilizada no momento de criação das tabelas e seus relacionamentos dentro do MySQL.

Vamos estudar como tudo acontece, com exemplos passo a passo, conforme estamos fazendo desde que entramos no estudo do MySQL.

8

FIRJAN


INFORMA, FORMA, TRANSFORMA.

Novo BD no MySQL para Treino

Vamos criar e ativar um novo banco de dados, chamado ESCOLA:

```
CREATE DATABASE escola;  
USE escola;
```

No BD escola, serão criadas duas tabelas para demonstrar o uso de chaves, sendo a tabela de **cadastro de professores (cadpro)** e a tabela de **cadastro de disciplinas (caddis)**. As estruturas estão demonstradas a seguir:



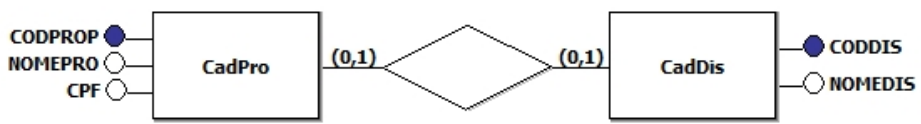
9


FIRJAN



INFORMA, FORMA, TRANSFORMA.


Novo BD para Treino (Cont.)

Eis a modelagem:





CadPro	
	CODPROP: INT(3)
	NOMEPRO: VARCHAR(40)
	CPF: CHAR(11)

CadDis	
	CODDIS: CHAR(6)
	NOMEDIS: VARCHAR(40)
	CODPROF: INT(3)




10



  INFORMA, FORMA, TRANSFORMA.

Novo BD para Treino (Cont.)

Tabela: cadpro - Cadastro de Professores		
Campo	Tipo	Descrição
CODPRO	INTEGER (3)	Código do Professor (Não Nulo)
NOMEPRO	VARCHAR (40)	Nome do Professor (Não Nulo)
CPF	CHAR (11)	CPF do Professor (Não Nulo)
Chave Primária		CODPROP
Chave Alternativa		CPF

Tabela: caddis - Cadastro de Disciplinas		
Campo	Tipo	Descrição
CODDIS	CHAR (6)	Código da Disciplina (Não Nulo)
NOMEDIS	VARCHAR (40)	Nome da Disciplina (Não Nulo)
CODPRO	INTEGER (3)	Código do Professor
Chave Primária		CODDIS
Chave Estrangeira		CODPRO (referência à CODPRO de cadpro)

 11


  INFORMA, FORMA, TRANSFORMA.

Novo BD para Treino (Cont.)

Criando primeiramente só as tabelas:

```
CREATE TABLE cadpro (CODPRO INT(3) NOT NULL  
AUTO_INCREMENT PRIMARY KEY, NOMEPRO  
VARCHAR(40) NOT NULL, CPF CHAR(11) NOT NULL  
UNIQUE) ;
```

```
CREATE TABLE caddis (CODDIS CHAR(6) NOT NULL  
PRIMARY KEY, NOMEDIS VARCHAR(40) NOT NULL,  
CODPRO INTEGER(3)) ;
```

 12

Preparação para o Relacionamento

Após a criação das tabelas **cadpro** e **caddis**, é necessário preparar o relacionamento propriamente dito que existirá entre estas tabelas.

É preciso definir uma restrição de **chave estrangeira** na tabela **caddis**, que indique seu vínculo de relação entre o campo CODPRO da tabela **caddis** e o campo CODPRO da tabela **cadpro**. É necessário fazer uma pequena alteração na estrutura da tabela **caddis** com a instrução:

```
ALTER TABLE caddis ADD CONSTRAINT fkCODPRO FOREIGN KEY  
(CODPRO) REFERENCES cadpro (CODPRO)  
ON UPDATE RESTRICT ON DELETE RESTRICT;
```

Observe a definição, após o comando **ALTER TABLE** da tabela **caddis**, de uma restrição (**ADD CONSTRAINT**) de nome **fkCODPRO** como sendo a *chave estrangeira* (**FOREIGN KEY**) do campo **CODPRO**, definido como referência (**REFERENCES**) direta ao campo **CODPRO** da tabela **cadpro**. O comando **CONSTRAINT** é o responsável por criar de forma explícita o relacionamento do campo **CODPRO** da tabela **caddis** com o campo **CODPRO** da tabela **cadpro**.

Também é estabelecido por **ON UPDATE RESTRICT** que não pode ser alterado o campo **CODPRO** da tabela **cadpro** que tenha um vínculo direto com algum registro da tabela **caddis**, pois se isso for permitido, pode gerar erros graves de integridade entre os dados das tabelas vinculadas. Já **ON DELETE RESTRICT** estabelece que a tabela **cadpro** não pode ter nenhum registro apagado, se existir para esse registro um vínculo direto com algum registro da tabela **caddis**.



13

Novo BD para Treino (Cont.)

OBSERVAÇÃO:

A tabela **caddis** poderia ser criada já contendo com o relacionamento que foi exemplificado anteriormente. O comando seria:

```
CREATE TABLE caddis (CODDIS CHAR(6) NOT NULL  
PRIMARY KEY, NOMEDIS VARCHAR(40) NOT NULL, CODPRO  
INTEGER(3), CONSTRAINT fkCODPRO FOREIGN  
KEY(CODPRO) REFERENCES cadpro(CODPRO) ON UPDATE  
RESTRICT ON DELETE RESTRICT);
```



14

Testando o Relacionamento

Para demonstrar o uso de relacionamento com chave primária e chave estrangeira, será necessário popular as tabelas **cadpro** e **caddis** com dados compatíveis com suas estruturas.

Vamos inserir 3 registros na tabela **cadpro**:

```
INSERT INTO cadpro VALUES (100, 'SILVANA', '11122233399');  
INSERT INTO cadpro VALUES (110, 'ROBERTO', '11122244488');  
INSERT INTO cadpro VALUES (120, 'RENATO', '11122244477');
```

Vamos inserir 6 registros na tabela **caddis**, sendo 2 por professor cadastrado em **cadpro**:

```
INSERT INTO caddis VALUES ('CG-100', 'MATEMATICA', 100);  
INSERT INTO caddis VALUES ('CG-110', 'FISICA', 100);  
INSERT INTO caddis VALUES ('CG-200', 'PORTUGUES', 110);  
INSERT INTO caddis VALUES ('CG-300', 'FILOSOFIA', 110);  
INSERT INTO caddis VALUES ('TI-100', 'ALGORITMOS', 120);  
INSERT INTO caddis VALUES ('TI-200', 'BD', 120);
```

Agora visualize o conteúdo das duas tabelas.



15

 **FIRJAN**



INFORMA. FORMA. TRANSFORMA.

Testando o Relacionamento (Cont.)

Agora tente inserir os próximos 3 registros e observe bem os motivos de dar erro nestas inserções:

```
INSERT INTO cadpro VALUES (120, 'CARLOS', '11122277744');  
INSERT INTO cadpro VALUES (300, 'JULIA', '11122244488');  
INSERT INTO caddis VALUES ('CG-400', 'BIOLOGIA', 380);
```



No primeiro exemplo existe um erro de restrição por duplicidade de dados, por conta do CODPRO 120 já existir cadastrado na tabela **cadpro**. CODPRO é chave primária.

No segundo exemplo ocorreu um erro no campo CPF, pois já existe na tabela **cadpro** um campo CPF idêntico ao que se tentou inserir agora. O campo CPF é chave alternativa.

No terceiro exemplo ocorreu erro devido ao fato de não existir nenhum professor com código 380. Restrição aplicada pela chave estrangeira.



16

  INFORMA, FORMA, TRANSFORMA.

Exibir Chaves e Índices


Para exibir as chaves e índices de uma tabela, o comando é:

```
SHOW INDEX FROM <tabela>;
```

Faça testes comparativos entre as tabelas **cadpro** e **caddis**:


```
SHOW INDEX FROM cadpro;  
SHOW INDEX FROM caddis;
```



A definição de cada coluna exibida por este comando é explicada a seguir.

 17

Exibir Chaves e Índices (Cont.)


Coluna	Descrição
Table	Nome da tabela.
Non_unique	Valor 0 (zero) se o índice não puder conter duplicidades, 1(um) se o índice puder conter duplicidades.
Key_name	Nome do índice.
Seq_in_index	Número da sequência da coluna no índice a partir de 1.
Column_name	Nome da coluna.
Collation	Informa como a coluna é ordenada no índice. Se for apresentado 'A', indica ordenação ascendente. Se NULL, está sem ordenação.
Cardinality	Número de valores únicos no índice.
Sub_part	Número de caracteres indexados se a coluna for indexada parcialmente ou NULL se a chave inteira for indexada.
Null	Possui valor 'YES' se a coluna puder conter NULL
Index_type	Método de índice utilizado.
Comment	Pode conter alguns comentários.

 18

  INFORMA, FORMA, TRANSFORMA.

Índices

Este assunto será explicado no item
BD - TI - 14 - Indexação de Tabelas.

 19


Cardinalidades no MySQL

Já percebemos que as cardinalidades máximas influenciam diretamente na definição das chaves estrangeiras e relacionamentos.

As cardinalidades mínimas vão influenciar se estas chaves estrangeiras vão receber ou não o parâmetro ***not null*** em suas definições.

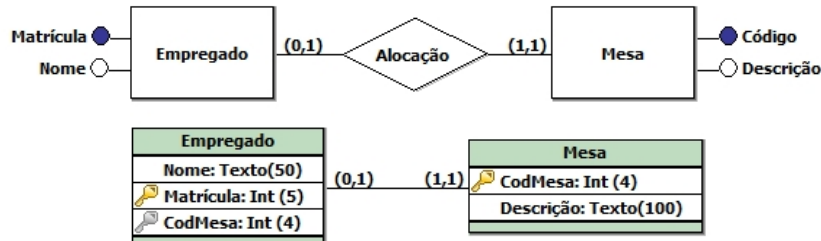
Quando a cardinalidade mínima é 1, entende-se que o relacionamento é obrigatório, e por conta disso, a chave estrangeira deste relacionamento precisa ter o parâmetro ***not null***.

Quando a cardinalidade mínima é 0, o relacionamento não é obrigatório, então a chave estrangeira deste relacionamento não poderá receber o parâmetro ***not null***.

 20

Cardinalidades no MySQL (Cont.)

Observe novamente esta modelagem:



Interpretando as cardinalidades, entende-se que:

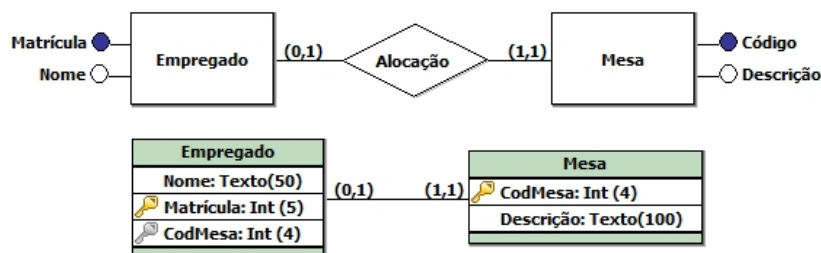
- Um empregado está alocado em no mínimo 1 e no máximo 1 mesa.
- Uma mesa está alocada para no mínimo 0 empregados e no máximo 1 empregado.

Como a chave estrangeira está na tabela *Empregado*, e todo empregado precisa ter no mínimo 1 mesa (cardinalidade mínima 1), então esta chave estrangeira precisa ter o parâmetro **not null**, pois seu preenchimento é obrigatório.

21

Cardinalidades no MySQL (Cont.)

Código de criação da tabela *Empregado* (A tabela *Mesa* precisa ser criada antes da tabela *Empregado*, pois a chave estrangeira em *Empregado* está relacionada à *Mesa*):



```
CREATE TABLE empregado (MATRICULA INT(5) NOT NULL  
PRIMARY KEY, NOME VARCHAR(50) NOT NULL, CODMESA  
INT(4) NOT NULL, CONSTRAINT fkCODMESA FOREIGN  
KEY(CODMESA) REFERENCES mesa(CODMESA) ON UPDATE  
RESTRICT ON DELETE RESTRICT);
```

22

Ministério
FIRJAN INFORMA, FORMA, TRANSFORMA.


Dúvidas?



23

Ministério
FIRJAN INFORMA, FORMA, TRANSFORMA.

Bibliografia



MySQL 5.1 Interativo (3ª Edição)
José Augusto N.G. Manzano
Ed. Érica

24