



Chaves estrangeiras, Relacionamentos

Prof. Dr. Nazareno de Oliveira Pacheco
nazareno.pacheco@prof.sc.senac.br

Um exemplo atual

- . Existem 10960 atletas participando das olimpíadas de Londres
- . Cada atleta tem nome e sexo, pertence a um país e atua em algum esporte
- . Cada país tem nome oficial, um nome resumido para o COI e um código específico do COI
 - . COI – Comitê Olímpico Internacional
- . Cada esporte tem um nome e um código do COI

Um exemplo atual

- Para armazenar estes dados, foi definida a seguinte tabela:

	Data Output	Explain	Messages	History				
	codigo_atleta integer	nome_atleta character varying(100)	sexo character(1)	nome_esporte character varying(100)	sigla_coi_esporte character(2)	nome_oficial_pais character varying(100)	nome_coi_pais character varying(100)	sigla_coi_pais character(3)
1	1	A Lamusi	M	Judo	JU	China	China	CHN
2	2	AARRASS Jamale	M	Athletics	AT	France	France	FRA
3	3	AATAKNI Abdelhak	M	Boxing	BX	Morocco	Morocco	MAR
4	4	ABAKUMOVA Maria	F	Athletics	AT	Russian Federation	Russia	RUS
5	5	ABALO Luc	M	Handball	HB	France	France	FRA
6	6	ABALO Maria Laura	F	Rowing	RO	Argentina	Argentina	ARG
7	7	ABARHOUN Mohamed	M	Football	FB	Morocco	Morocco	MAR
8	8	ABATE Emanuele	M	Athletics	AT	Italy	Italy	ITA
9	9	ABBADI Ilyas	M	Boxing	BX	Algeria	Algeria	ALG
10	10	ABBAS Sohail	M	Hockey	HO	Pakistan	Pakistan	PAK
11	11	ABBASI Shakeel	M	Hockey	HO	Pakistan	Pakistan	PAK
12	12	ABBASIAZAD Soulmaz	F	Rowing	RO	Iran, Islamic Repub	Iran	IRI
13	13	ABBATE Simona	F	Water Polo	WP	Italy	Italy	ITA
14	14	ABDALAZEM Reem	F	Synchronised swimmi	SY	Egypt	Egypt	EGY

Fonte: <http://www.guardian.co.uk/sport/datablog/2012/jul/27/london-olympic-athletes-full-list#data>

Alguns problemas

- Como utilizamos uma só tabela, acabamos com muitos valores duplicados nas colunas fabricante e categoria
- Esta redundância de dados causa alguns problemas
 - **Desperdício**: precisamos repetir o mesmo nome várias vezes, desperdiçando espaço
 - **Risco de inconsistência**: podemos ter um atleta do Brazil, e outro do Brasil. Isto pode dificultar as nossas consultas

Ligando “tabelas”

- A melhor forma de lidarmos com este tipo de situação é utilizar várias tabelas
 - Uma tabela para atletas
 - Uma tabela para esportes
 - Uma tabela para países
- Porém, é necessário que seja associar cada atleta a um país e a um esporte
 - Solução: chaves estrangeiras (foreign keys, ou FK)

Chaves estrangeiras

- Uma chave estrangeira é um conjunto de um ou mais campos de uma tabela que referência a chave primária de outra tabela

Exemplo (2 chaves estrangeiras)

Tabela atleta

	codigo_atleta integer	nome_atleta character varying(100)	sexo character(1)	codigo_esporte integer	codigo_pais integer
1	1	A Lamusi	M	22	40
2	2	AARRASS Jamale	M	2	65
3	3	AATAKNI Abdelhak	M	6	122
4	4	ABAKUMOVA Maria	F	2	149
5	5	ABALO Luc	M	20	65
6	6	ABALO Maria Laura	F	24	8

	codigo_esporte integer	nome_esporte character varying(100)	sigla_coi_esporte character(2)
1	1	Archery	AR
2	2	Athletics	AT
3	3	Badminton	BD
4	4	Basketball	BK
5	5	Beach Volleyball	BV
6	6	Boxing	BX

Tabela esporte

	codigo_pais integer	nome_oficial_pais character varying(100)	nome_coi_pais character varying(100)	sigla_coi_pais character(3)
1	1	Afghanistan	Afghanistan	AFG
2	2	Albania	Albania	ALB
3	3	Algeria	Algeria	ALG
4	4	American Samoa	American Samoa	ASA
5	5	Andorra	Andorra	AND
6	6	Angola	Angola	ANG

Tabela pais

Chaves estrangeiras

- Não é possível adicionar um registro em uma tabela cujo valor da chave estrangeira não corresponda ao valor da chave primária na tabela referenciada
 - Por exemplo: não é possível adicionar um registro na tabela *atleta* cujo `codigo_esporte` não corresponda a chave primária de algum registro já existente na tabela *esporte*
- Isto garante **integridade referencial**
 - Se um atleta está associado a um código de esporte, este esporte EXISTE na tabela de esportes

Definindo chaves estrangeiras

- Chaves estrangeiras podem ser definidas no momento de criação das tabelas
- Os tipos das colunas devem ser o mesmo
 - Existe uma exceção: quando a tabela que vai ser referenciada utiliza SERIAL como chave primária, a coluna da chave estrangeira deve ser INT)
 - Por exemplo: se a tabela esporte tem a coluna codigo_esporte SERIAL, a tabela atleta deve ter uma coluna codigo_esporte INT

Definindo chaves estrangeiras

--Criando as tabelas necessarias para as FKs

```
CREATE TABLE esporte (  
    codigo_esporte INT NOT NULL PRIMARY KEY,  
    nome_esporte VARCHAR(100) NOT NULL,  
    sigla_coi_esporte CHAR(2) NOT NULL  
);  
  
CREATE TABLE pais (  
    codigo_pais integer NOT NULL PRIMARY KEY,  
    nome_oficial_pais VARCHAR(100) NOT NULL,  
    nome_coi_pais VARCHAR(100) NOT NULL,  
    sigla_coi_pais CHAR(3) NOT NULL  
);  
...
```

Definindo chaves estrangeiras

...

--criando a tabela e definindo as FKs

```
CREATE TABLE atleta (  
    codigo_atleta INT NOT NULL PRIMARY KEY,  
    nome_atleta VARCHAR(100) NOT NULL,  
    sexo CHAR(1) NOT NULL,  
    codigo_esporte INT NOT NULL,  
    codigo_pais INT NOT NULL,  
    FOREIGN KEY (codigo_esporte) REFERENCES esporte(codigo_esporte) ,  
    FOREIGN KEY (codigo_pais) REFERENCES pais(codigo_pais)  
);
```

Sintaxe alternativa

...

--também é possível definir as FKs da seguinte forma:

```
CREATE TABLE atleta (  
    codigo_atleta INT NOT NULL PRIMARY KEY,  
    nome_atleta VARCHAR(100) NOT NULL,  
    sexo CHAR(1) NOT NULL,  
    codigo_esporte INT NOT NULL REFERENCES esporte(codigo_esporte),  
    codigo_pais INT NOT NULL REFERENCES pais(codigo_pais)  
);
```

Exercícios (16)

- 1 – Cada país pertence a um continente. Cada continente tem um nome.
 - Crie uma tabela de continentes e uma chave estrangeira correspondente na tabela de países
 - Não é necessário reinserir os dados na tabela de países
- 2 – Para cada esporte olímpico existe uma série de eventos (por exemplo: na natação temos 50m livre masculino, 50m livre feminino, 100m borboleta masculino, etc...). Crie uma tabela de eventos
 - Cada evento tem um nome e um esporte ao qual está associado.

Senac

Exercícios (17)

- 1 – Reorganize a tabela produto dos exercícios anteriores de forma a utilizar chaves estrangeiras onde apropriado
 - Não é necessário inserir os dados, apenas criar/recrutar as tabelas necessárias
- 2 – Cada evento olímpico é composto por uma série de provas (um jogo de futebol, uma prova de natação, etc). Cada prova tem um nome e um horário de início. Crie uma tabela de provas

Relacionamentos

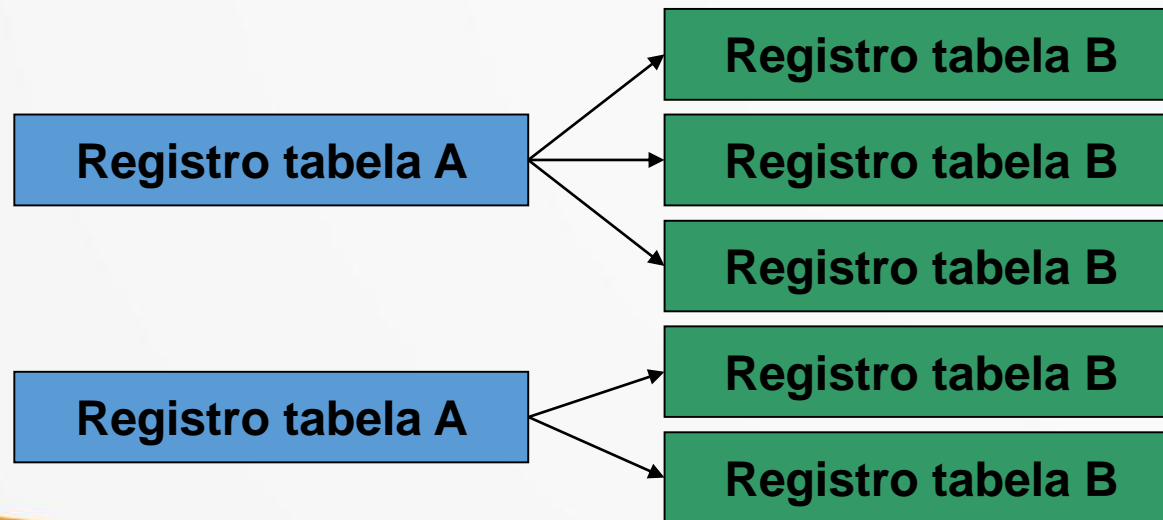
- Chaves estrangeiras servem para expressar **relacionamentos** entre tabelas
- Existem vários tipos de relacionamentos possíveis
- Vamos focar nos dois tipos de relacionamentos mais comuns:
 - Relacionamentos *1 para Muitos* $1 - n$
 - Relacionamentos *Muitos para Muitos* $m-n$
- Existem também os relacionamentos 1 para 1, mas sua utilização é menos comum, e não vamos falar dele neste curso

Relacionamentos 1 para Muitos

- . É mais fácil explicar por exemplos:
 - **Atleta X País:** Um país pode ter vários atletas competindo, mas um atleta só pode competir por um país
 - **Cliente X Pedido:** Um cliente pode fazer vários pedidos, mas um pedido só é feito por um cliente
 - **Funcionário X Departamento:** Um departamento pode ter vários funcionários, mas um funcionário só pode estar locado em um departamento

Relacionamentos 1 para Muitos

- . Ou seja, nestes relacionamentos, uma *tabela A* está associada a vários registros de uma *tabela B*, mas um registro da tabela B está associado a apenas um registro da tabela A

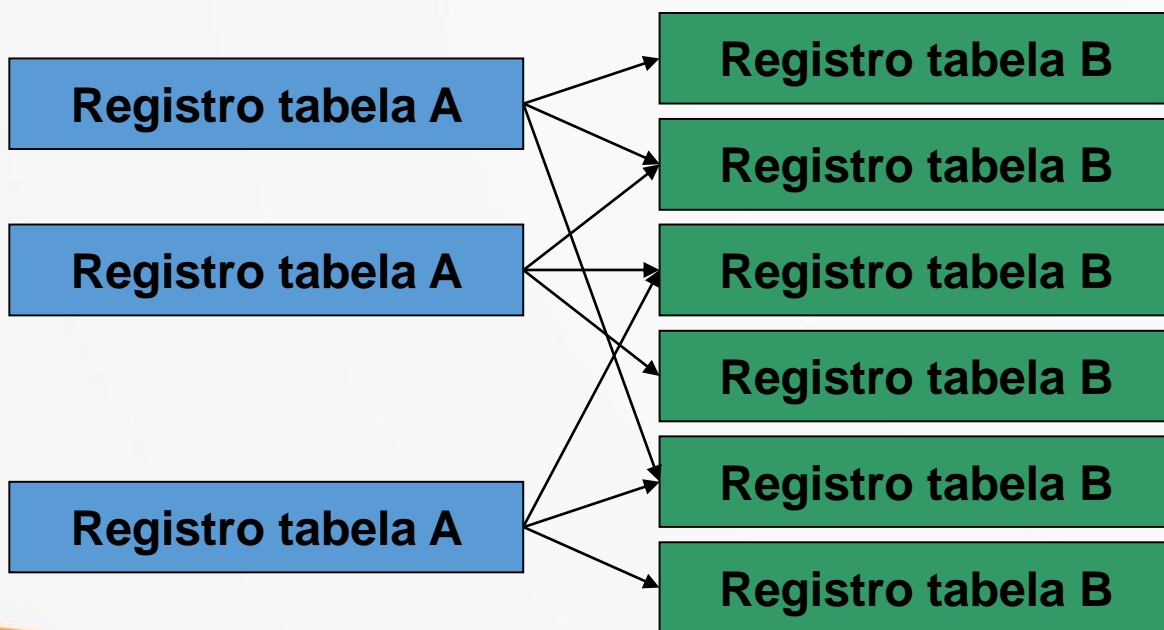


Relacionamentos 1 para Muitos

- Para definir um relacionamento deste tipo, basta criar uma chave estrangeira na tabela B referenciando a tabela A
 - Criar uma chave estrangeira na tabela *atleta* referenciando a tabela *país*
 - Criar uma chave estrangeira na tabela *pedido* referenciando a tabela *cliente*
 - Criar uma chave estrangeira na tabela *funcionario* referenciando a tabela *departamento*

Relacionamentos Muitos para Muitos

- Nestes casos, um registro de uma tabela A pode estar associado a diversos registros de uma tabela B, e VICE-VERSA



Relacionamentos 1 para Muitos

- Para definir um relacionamento deste tipo, basta criar uma chave estrangeira na tabela B referenciando a tabela A
 - Criar uma chave estrangeira na tabela *atleta* referenciando a tabela *país*
 - Criar uma chave estrangeira na tabela *pedido* referenciando a tabela *cliente*
 - Criar uma chave estrangeira na tabela *funcionario* referenciando a tabela *departamento*

Relacionamentos Muitos para Muitos

- . Exemplos:
 - **Atleta X Idioma:** um atleta pode falar diversos idiomas, e um idioma pode ser falado por diversos atletas
 - **Pedido X Produto:** um pedido pode conter diversos produtos, e um produto pode estar associado a diversos pedidos
 - **Aluno X Disciplina:** um aluno pode estar cursando diversas disciplinas, e uma disciplina pode estar associada a diversos alunos



Relacionamentos Muitos para Muitos

- Estes tipos de relacionamentos são um pouco mais complicados de serem representados
- Neste caso, não adianta apenas definir uma chave estrangeira
- Suponha que tentássemos criar uma chave estrangeira no caso Atleta X Idioma
 - Se criarmos um campo `codigo_atleta` na tabela de idiomas, cada idioma só poderá ser falado por um atleta
 - Se criarmos um campo `codigo_idioma` na tabela de atletas, cada atleta só poderia falar um idioma
 - Criar várias chaves estrangeiras (`codigo_idioma1`, `codigo_idioma2`, etc...) também não é uma boa solução

Solução – tabela junção

- Para resolver este problema, é necessário criar uma tabela junção
- Esta tabela será composta com as seguintes colunas
 - As colunas da chave primária da tabela A
 - As colunas da chave primária da tabela B
- Além disso, a chave primária da tabela junção será composta com todas as colunas das duas chaves estrangeiras
 - Desta forma, cada coluna será, ao mesmo tempo, parte da chave primária da tabela junção e chave estrangeira de outra tabela

Exemplo

--eh a mesma tabela criada no exemplo anterior

```
CREATE TABLE atleta (  
    codigo_atleta INT NOT NULL PRIMARY KEY,  
    nome_atleta VARCHAR(100) NOT NULL,  
    sexo CHAR(1) NOT NULL,  
    codigo_esporte INT NOT NULL REFERENCES esporte(codigo_esporte),  
    codigo_pais INT NOT NULL REFERENCES pais(codigo_pais)  
);
```

--nova tabela de idiomas

```
CREATE TABLE idioma (  
    codigo_idioma INT NOT NULL PRIMARY KEY,  
    nome_idioma VARCHAR(50) NOT NULL  
);
```

Exemplo

--tabela junção

```
CREATE TABLE atleta_idioma (  
    codigo_atleta INT NOT NULL,  
    codigo_idioma INT NOT NULL,  
    PRIMARY KEY (codigo_atleta, codigo_idioma),  
    FOREIGN KEY (codigo_atleta) REFERENCES  
atleta(codigo_atleta),  
    FOREIGN KEY (codigo_idioma) REFERENCES  
idioma(codigo_idioma)  
);
```

Exemplo

Tabela *atleta*

	codigo_atleta integer	nome_atleta character varying(100)	sexo character(1)	codigo_esporte integer	codigo_pais integer
1	1	A Lamusi	M	22	40
2	2	AARRASS Jamale	M	2	65
3	3	AATAKNI Abdelhak	M	6	122
4	4	ABAKUMOVA Maria	F	2	149
5	5	ABALO Luc	M	20	65
6	6	ABALO Maria Laura	F	24	8
7	7	ABERNATHY M...	M	45	100

Tabela *idioma*

	codigo_idioma integer	nome_idioma character varying(50)
1	1	French
2	2	English
3	3	Mandarin
4	4	Spanish
5	5	Arabic
6	6	Bengali
7	7	Portuguese

	codigo_atleta integer	codigo_idioma integer
1	46	6
2	96	6
3	107	6
4	108	6
5	244	6
6	248	6
7

Tabela junção *atleta_idioma*

Senac

Exercícios (18)

- 1 – Nosso modelo original dos atletas olímpicos não está 100% correto. É possível que um atleta participe de mais de uma esporte (isto é comum nas diversas modalidades de ciclismo). Ajuste nosso modelo de dados de forma a permitir isto
 - Não é necessário reinserir os dados no sistema, apenas criar / recriar as tabelas necessárias
- 2 – Cada evento de uma olimpíada tem diversos atletas participantes. Ajuste nosso modelo de dados para que esta informação possa ser armazenada

Exercícios (19)

- 1 – Cada país pode ter um conjunto de idiomas oficiais. Ajuste nosso banco de dados de forma a permitir o armazenamento desta informação
- 2 – Cada prova olímpica é disputada por uma série de atletas participantes. Ajuste nosso modelo para permitir o armazenamento destas informações

Exercícios (20)

- Crie um modelo de dados que permita armazenar as seguintes informações:
 - Uma universidade tem diversos alunos e professores, todos identificados por um número de matrícula.
 - A universidade também presta diversos cursos, cada um deles chefiados por um professor.
 - Cada curso é composto de várias disciplinas, que são ministradas a vários alunos por um professor. Cada disciplina tem uma ementa e uma carga horária
 - A universidade também dispõe de diversas salas identificadas por um número, que podem ser utilizadas para as disciplinas. Cada disciplina é dada em uma ou mais salas (nos casos de disciplinas com aulas de laboratório)