

Fundamentos de Base de Dados

Modelo Relacional

Docente: Fátima Leal

DCT DEPARTAMENTO CIÊNCIA
E TECNOLOGIA

Modelo Relacional

- Relembrando...
- O modelo conceptual, como vimos para o modelo ER, define um modelo para a BD independente do tipo de base de dados.
- **Modelo Relacional** - baseado no conceito de tabela, também chamado de tabela.
- **Entidades-tipo e relacionamentos** no modelo ER podem ser mapeados em **tabelas no modelo relacional**.
- Um **modelo relacional** pode ser depois **implementado num SGBD** baseado na linguagem SQL (como veremos também depois).

Modelo Relacional – Conceitos base

- **Modelo relacional** representa uma base de dados **como um conjunto de tabelas** em que uma **relação** é uma **tabela**. No modelo relacional encontramos os seguintes conceitos:

- **tabela**
- **linhas, registos ou linhas**
- **Atributos**

The diagram shows a table titled 'ALUNO' with four columns: 'Num', 'NumBI', 'Nome', and 'Curso'. An arrow points from the text 'Nome da tabela' to the table title. Another arrow points from the text 'Atributos (colunas)' to the column headers. A bracket on the right side of the table rows is labeled 'linhas Registos Linhas'.

ALUNO			
Num	NumBI	Nome	Curso
41950	13555678	Joel Tavares	Lic. Informática
41479	12444678	José Leitão	Lic. Informática
41216	16777890	Fernanda Chaves	Lic. Informática
41367	25666789	Tiago Sá	Lic. Informática

Modelo Relacional – Conceitos base

FUNCIONARIO				
<u>F_ident</u>	F_nome	F_morada	F_salario	F_dt_nasc
1163	Fatima Leal	Porto	750 €	12/08/1974
1164	Manuela Nunes	Lisboa	1760 €	28/07/1987
1165	Antonio Pacheco	Coimbra	750 €	06/03/1980
1166	Luís Borges	Faro	635 €	04/04/2000
1167	Tiago Sousa	Bragança	750 €	06/10/1990

Modelo Relacional – Conceitos

- Uma **tabela** é um **conjunto de linhas não ordenados** e que são representados em forma de **tabela**. A **tabela tem um esquema** associado definido por um nome e sequência de atributos.
- Contudo, vamos utilizar o termo **tabela** para a noção de tabela:
 - Evita a dúvida entre o relacionamento no modelo ER e tabela no modelo Relacional
 - Termo utilizado para designar/implementar uma tabela no contexto concreto de um de SGBD relacional baseados em SQL.
- **Atributo** identifica uma característica/propriedade de uma tabela.
- **Domínio** é o tipo de dados.

Modelo Relacional – Conceitos

- tabela \longrightarrow tabela
- linha linha \longrightarrow atributos
- Atributo \longrightarrow coluna
- Domínio \longrightarrow tipo de dados

- **Um esquema de tabela/tabela** $R(A_1, A_2, \dots, A_n)$ é composto por uma tabela de nome R e com uma lista de atributos A_1, A_2, \dots, A_n .

- Cada **atributo** A_i é o nome do **papel desempenhado** num domínio D no esquema de tabela R .

- O **grau de uma tabela** é o **número n de atributos** que o esquema que define a tabela possui.

Modelo Relacional – Conceitos

FUNCIONARIO				
<u>ident</u>	nome	morada	salario	dt_nasc
1163	Fatima Leal	Porto	750 €	12/08/1974
1164	Manuela Nunes	Lisboa	1760 €	28/07/1987
1165	Antonio Pacheco	Coimbra	750 €	06/03/1980
1166	Luís Borges	Faro	635 €	04/04/2000
1167	Tiago Sousa	Bragança	750 €	06/10/1990

$t_4[\text{nome}] = ?$

Qual o valor correspondente?

Modelo Relacional: Ordenação

- **Ordenação de linhas numa tabela**
- Uma **tabela** é definida por um **conjunto de linhas**.
- **As linhas de uma tabela não possuem nenhuma ordenação**

tabela A	
<u>ident</u>	nome
1163	Fatima Leal
1164	Manuela Nunes
1165	Antonio Pacheco
1166	Luís Borges
1167	Tiago Sousa

=

tabela B	
<u>ident</u>	nome
1167	Tiago Sousa
1165	Antonio Pacheco
1163	Fatima Leal
1164	Manuela Nunes
1166	Luís Borges

Modelo Relacional: Ordenação

- **Ordenação dos valores dentro de uma linha**
- Uma **linha é uma lista ordenada de valores**. Então a ordem dos valores é importante.

tabela A	
<u>ident</u>	nome
1163	Fatima Leal
1164	Manuela Nunes
1165	Antonio Pacheco
1166	Luís Borges
1167	Tiago Sousa

!=

tabela N	
nome	<u>ident</u>
Fatima Leal	1163
Manuela Nunes	1164
Antonio Pacheco	1165
Luís Borges	1166
Tiago Sousa	1167

Modelo Relacional: Valores

- A cada **atributo** está associado a um **domínio de valores**.
- Os **valores de um atributo são atômicos** e **podem incluir** o valor especial **NULL** para denotar a ausência de valor definido.


Modelo Relacional: Restrições

- **Restrição de domínio** especifica que o **valor de cada atributo** tem que ser um valor **atômico** dentro de um domínio em todas as linhas da tabela
- **Restrição de chave** especifica que todos os valores de um conjunto são distintos. **Todas as linhas de uma tabela devem ser distintas.**
- Duas linhas não podem ter a mesma combinação de valores para os seus atributos.

Modelo Relacional: Chaves e Superchaves

- **Superchave:** subconjunto de atributos de uma tabela para a qual todos os linhas são diferentes. Permite identificar de forma única os linhas de uma tabela. Todas as tabelas têm por defeito uma superchave – o conjunto de todos os atributos da tabela.

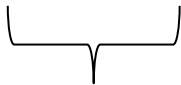
FUNCIONARIO				
<u>ident</u>	nome	morada	salario	dt_nasc



Modelo Relacional: Chaves e Superchaves

- **Chave:** conjunto de atributos mínimo capaz de garantir que cada linha é único.

FUNCIONARIO				
<u>ident</u>	nome	morada	salario	dt_nasc



Modelo Relacional: Chaves e Superchaves

- Um esquema R pode **ter mais que uma chave**. Cada uma delas é intitulada de **chave candidata**.



- Assim, uma tabela pode ter várias chaves, mas apenas uma deve ser designada como a **chave primária** da tabela.

- A **escolha da chave primária** de uma tabela é arbitrária, mas no entanto é usual escolher a **chave com o menor número de atributos**.

- No esquema de uma tabela, a **chave** primária é representada **sublinhada**.



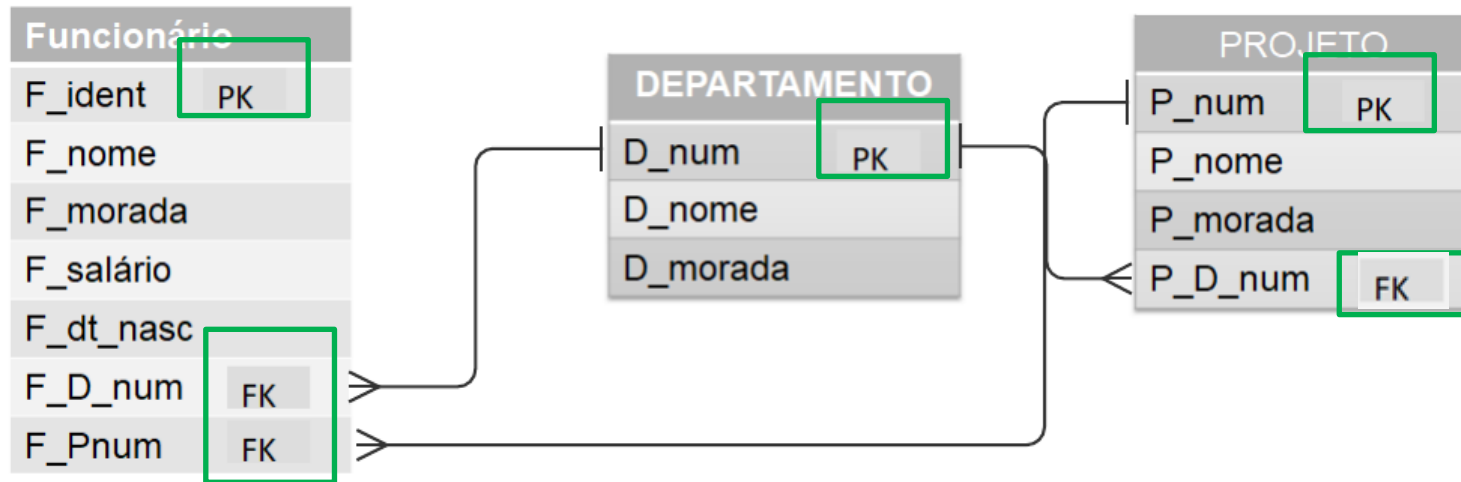
Modelo Relacional: Restrições de Integridade

- **Integridade de domínio:** Os valores de um atributo devem pertencer ao domínio do atributo.
- **Integridade da chave:** Não podem existir dois linhas de uma tabela com valores iguais na chave primária.
- **Integridade de entidade:** Os valores da chave primária não podem ser NULL.
- **Integridade referencial:** Um linha que referencia outra tabela tem de referenciar um linha existente nessa outra tabela (chave externa).

Modelo Relacional: Chave Externa

- **Chave externa ou chave estrangeira**
- Consideremos o exemplo:
 - Um funcionário trabalha num departamento. Os departamentos controlam vários projetos. Cada projeto apenas está associado a um departamento e ficou definido que um funcionário apenas participa num projeto de cada vez.

Modelo Relacional: Chave Externa

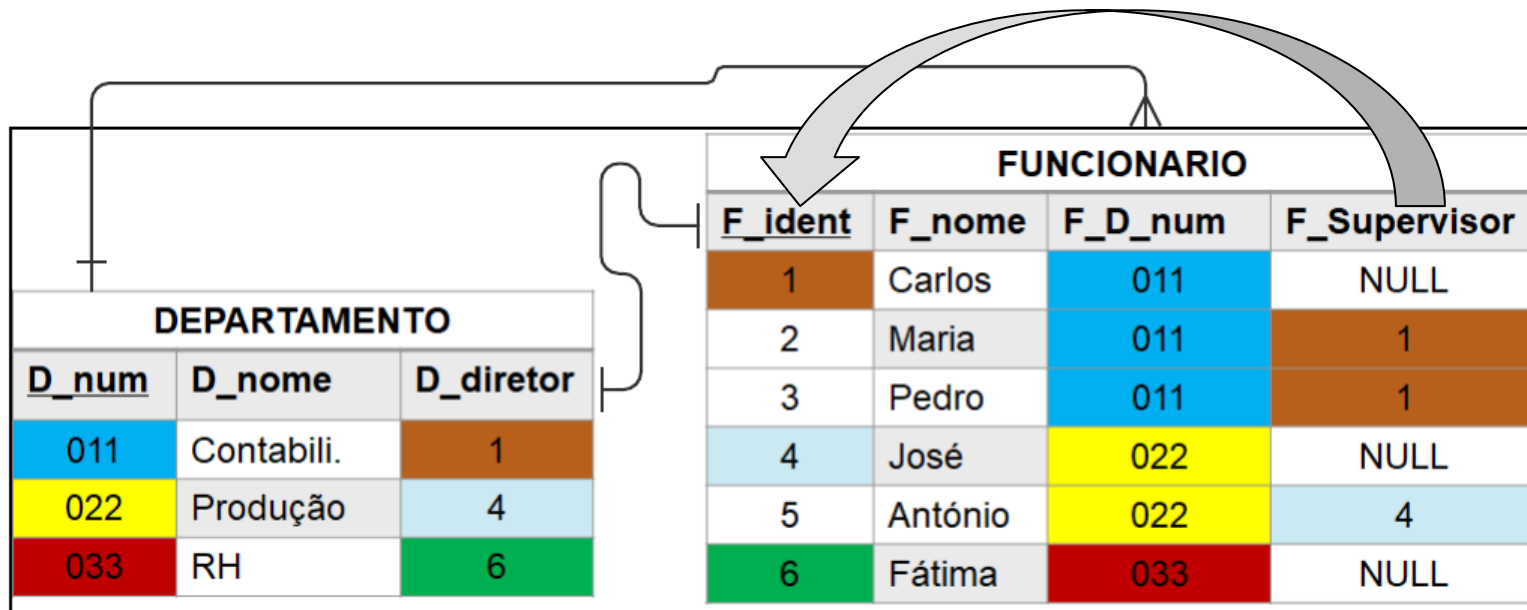


Quais chaves primárias?

Quais chaves estrangeiras?

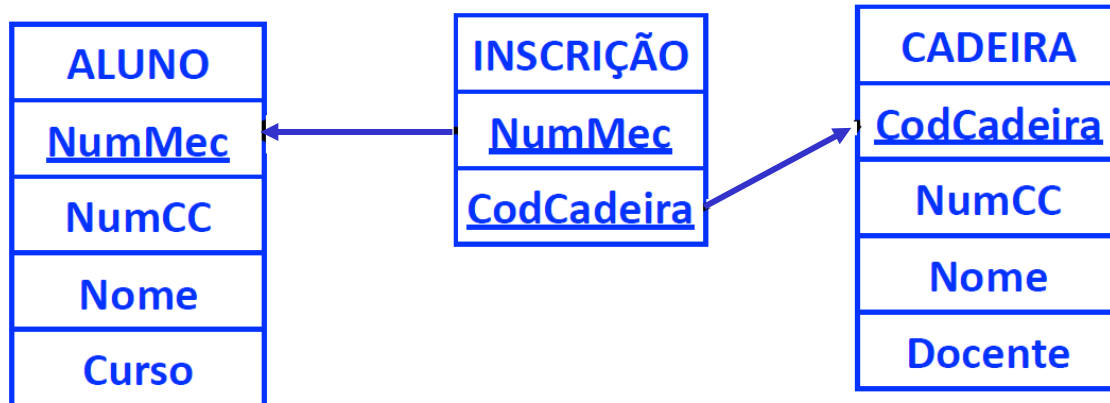
Modelo Relacional: Chave Externa

- Neste exemplo Empresa as **chaves externas não fazem parte da chave primária** de nenhuma das tabelas.
- Se nem todos os funcionários têm um supervisor, **podemos ter valores NULL** para o atributo FUNCIONÁRIO.Supervisor.

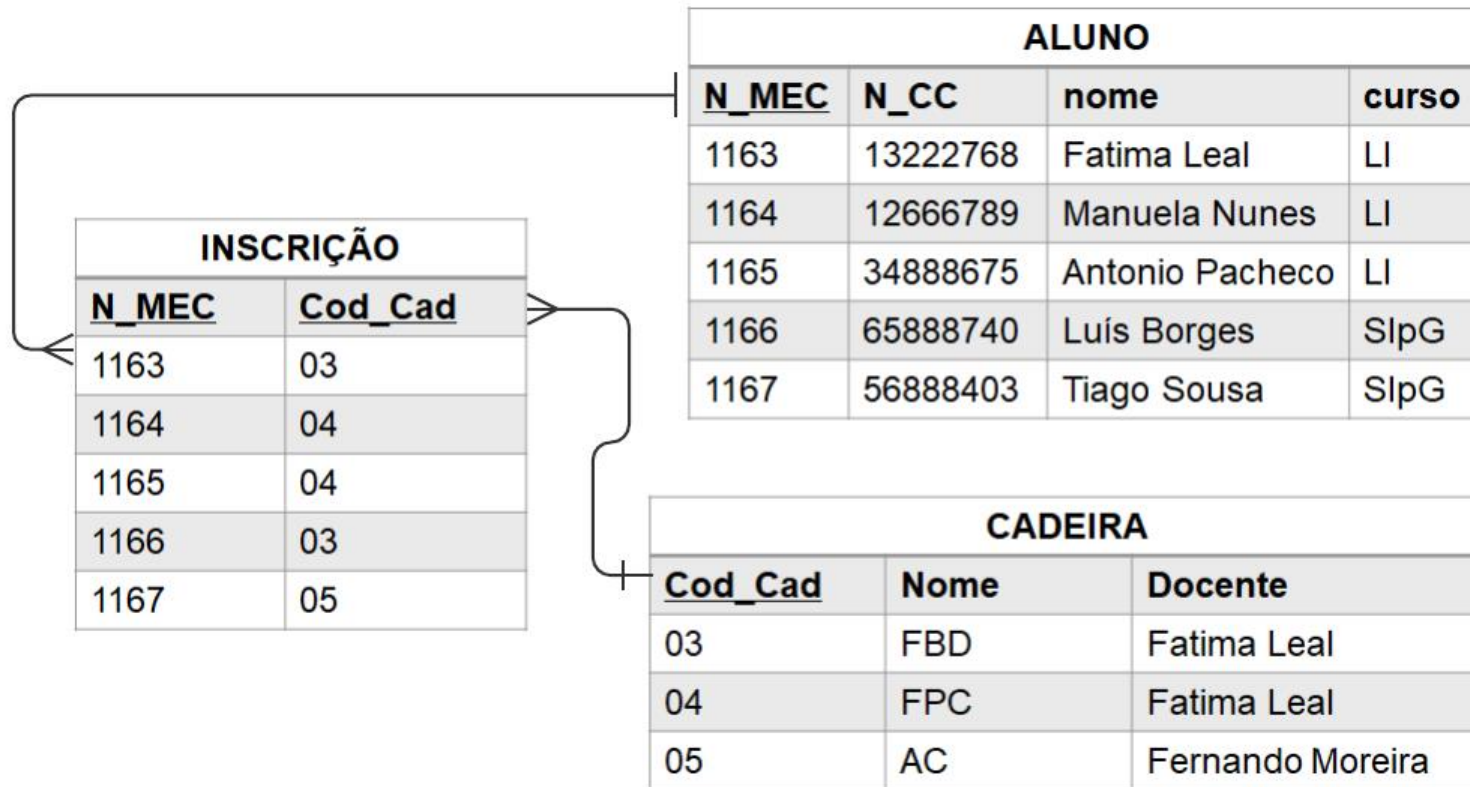


Modelo Relacional: Chave Externa

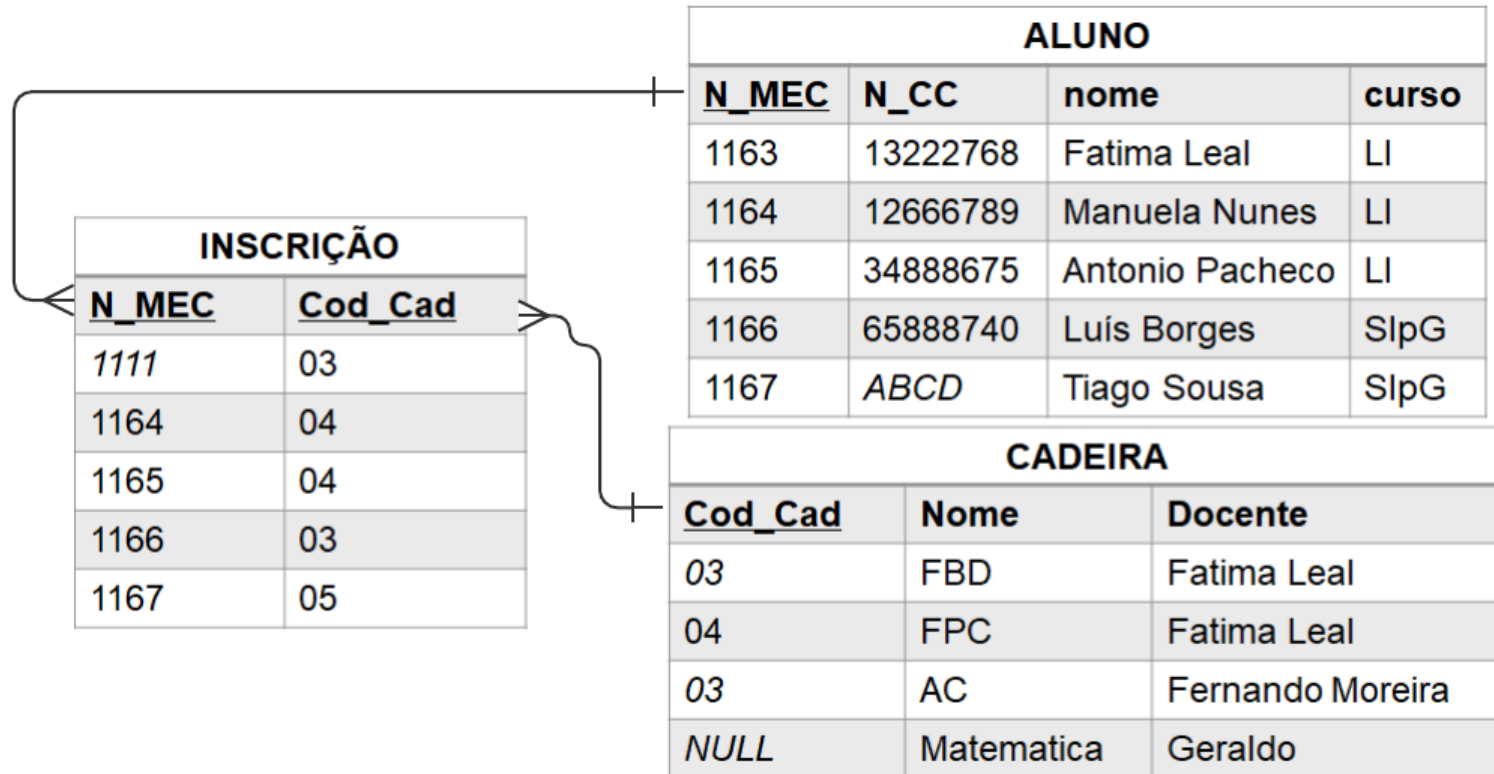
- INSCRIÇÃO tem 2 chaves externas: NumMec (chave primária de ALUNO) e CodCadeira (chave primária de CADEIRA) .
- Por sua vez, o par (NumMec, CodCadeira) é a chave primária de INSCRIÇÃO.



Modelo Relacional: Chave Externa



Modelo Relacional: Consistência de Dados

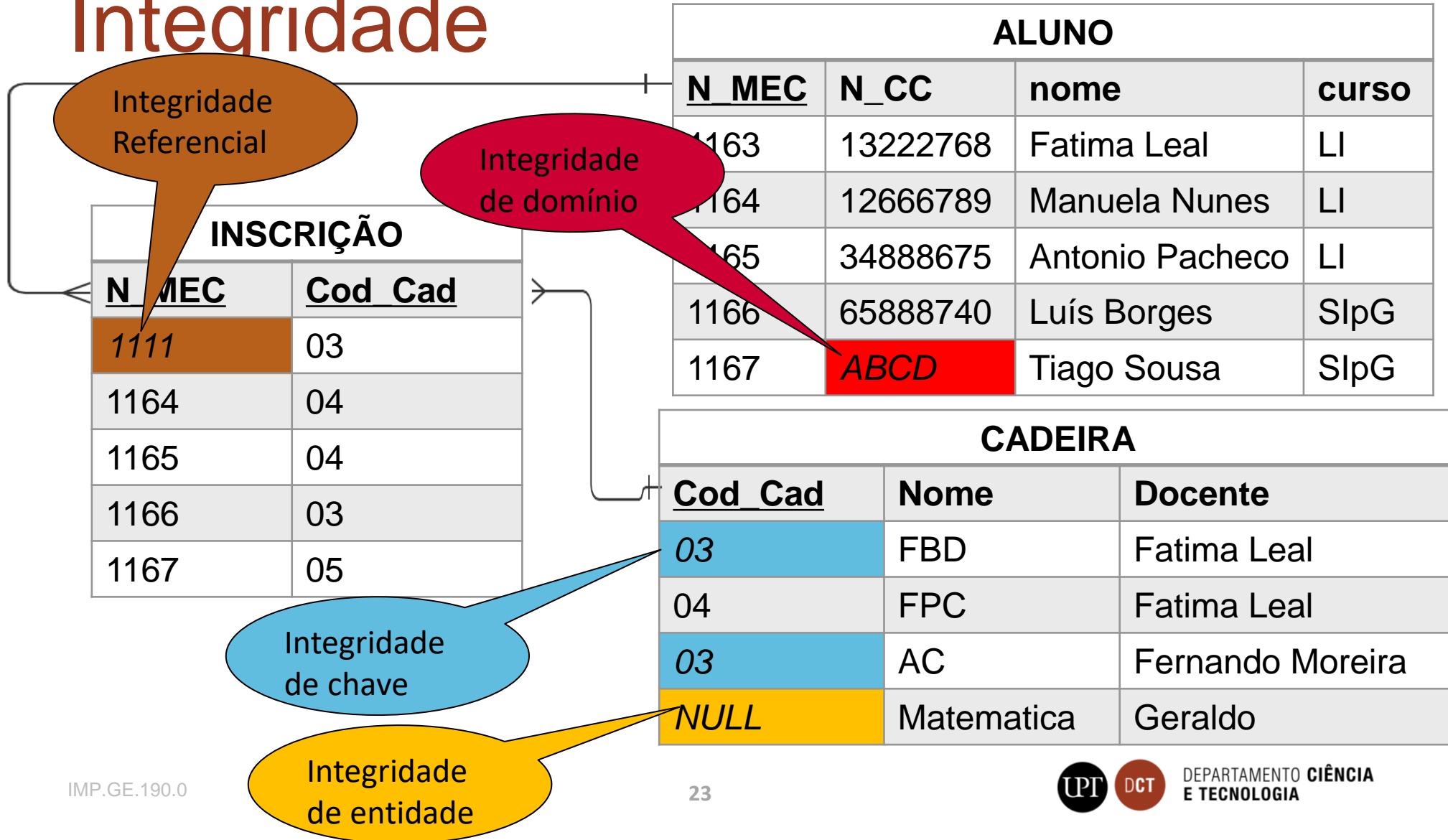


Quais são os problemas que identificas nesta base de dados? O que está em causa?

Modelo Relacional: Restrições de Integridade

- Por isso as restrições são muito importantes. Relembrando:
- **Integridade de domínio:** o valor de um atributo faz parte do domínio do atributo.
- **Integridade de entidade:** o valor da chave primária não pode ser NULL (sob pena de não conseguirmos identificar registos).
- **Integridade de chave:** dois registos da mesma tabela não podem ter valores iguais para uma chave primária.
- **Integridade referencial:** um valor definido para um atributo que seja chave externa deve referir-se a uma chave primária da tabela a que a chave externa se refere.

Modelo Relacional: Restrições de Integridade



Base de dados Relacional

- **Esquema da base de dados** = {esquema de tabelas}
- **Estado da base de dados** = {conteúdo das tabelas}
- **Esquema relacional de uma base de dados:**
 - Conjunto de esquemas de todas as suas tabelas
 - Conjunto de restrições de integridade
- **Restrições de integridade**
 - Uma base de dados deve satisfazer sempre as restrições de integridade. Quando isso não acontece, diz-se que a base de dados está num estado inválido.
- Existem **três operações básicas** que podem violar as restrições de integridade:
 - Insere (INSERT)
 - Remove (DELETE)
 - Atualiza (UPDATE)

Base de dados Relacional: Operações

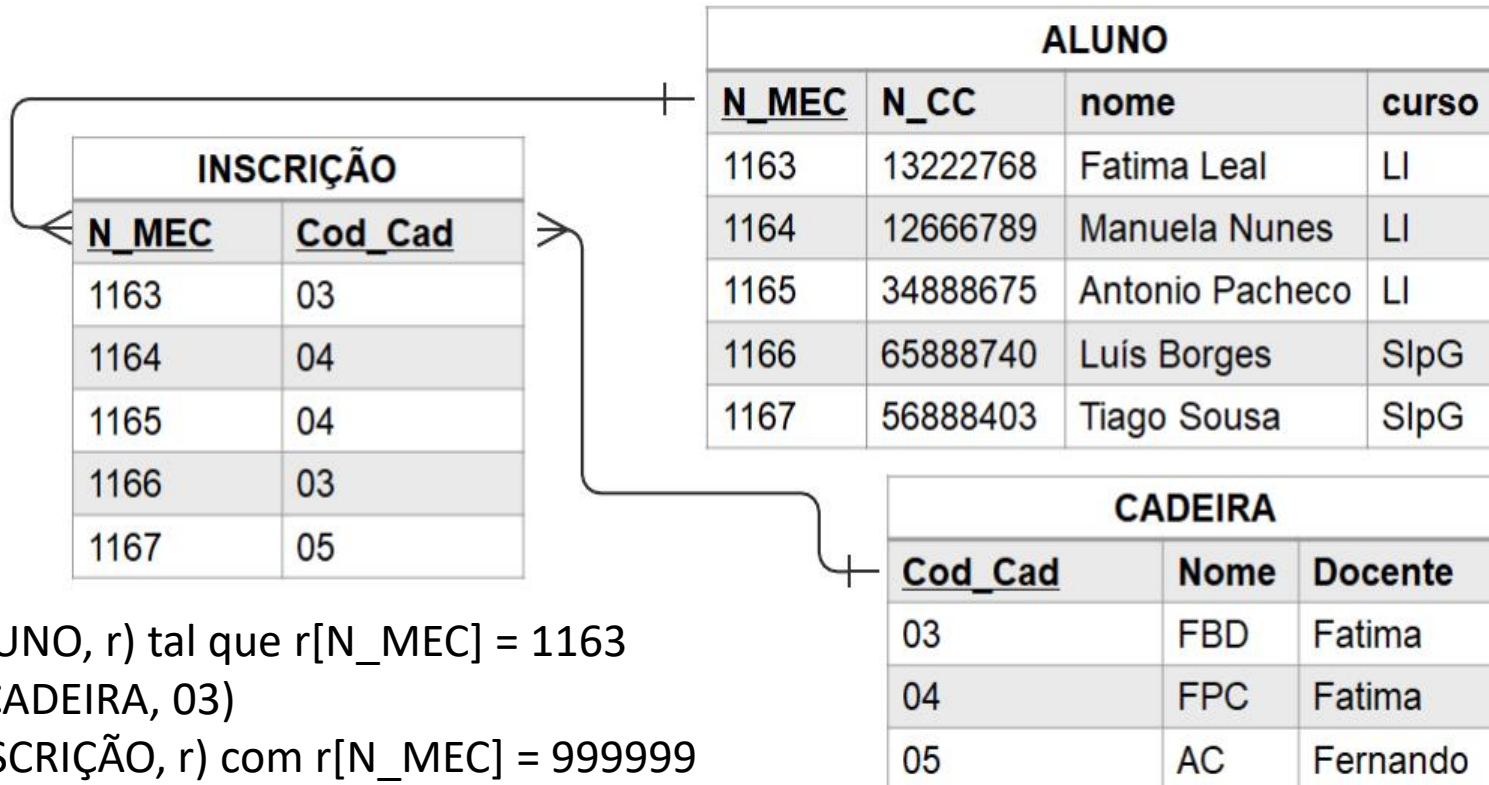
- **INSERE(T, r)** : insere novo registo r na tabela T
- **REMOVE(T, k)**: remove registo (que já exista) com chave primária k de T
- **ACTUALIZA(T, k, r)**: atualiza registo com chave primária k em T pelo registo r com a mesma chave primária (pode ser vista como uma remoção seguida de uma inserção)
- Estas operações irão corresponder às formas mais simples dos comandos SQL INSERT, DELETE, e UPDATE (a explorar nas próximas aulas).

Operações e restrições de integridade

- As operações consideradas podem ser inválidas se violarem restrição de integridade:
- $\text{INSERE}(T, r)$: insere novo registo r na tabela T — pode violar qualquer um dos tipos de restrições (domínio, entidade, chave, referencial).
- $\text{REMOVE}(T, k)$: remove registo (que já exista) com chave primária k de T — pode violar a integridade referencial se existir uma referência a k por via de uma chave externa.
- $\text{ACTUALIZA}(T, k, r)$: atualiza registo com chave primária k em T pelo registo r com a mesma chave primária — pode violar qualquer um dos tipos de restrição.
- Exemplos de como estas operações violam as restrições de integridade

Operações e restrições de integridade

Fatima Leal | February 21, 2021



Exemplos:

- INSERE(ALUNO, r) tal que $r[N_MEC] = 1163$
- REMOVE(CADEIRA, 03)
- INSERE(INSCRIÇÃO, r) com $r[N_MEC] = 999999$
- ACTUALIZA(ALUNO, 1163, r) com $r[N_CC] = 'ABCDE'$
- INSERE(ALUNO, r) com $r[N_MEC] = \text{NULL}$

Operações e restrições de integridade

- **INSERE(ALUNO, r)** tal que **r[N_MEC] = 1163** violaria integridade de chave p/ALUNO.
- **REMOVE(CADEIRA, 03)** violaria integridade referencial p/ INSCRIÇÃO.CodCad.
- **INSERE(INSCRIÇÃO, r)** com **r[N_MEC] = 999999** violaria integridade referencial p/ INSCRIÇÃO.N_MEC.
- **ACTUALIZA(ALUNO, 1163, r)** com **r[N_CC] = 'ABCDE'** violaria a integridade de domínio p/ ALUNO.NumCC.
- **INSERE(ALUNO, r)** com **r[N_MEC] = NULL** violaria a integridade de entidade p/ALUNO.

SGBDs e restrições de integridade

- Um SGBD deverá rejeitar uma operação que viole restrições de integridades, assinalando o erro.
- SGBDs maduros normalmente suportam todos os tipos de restrições de integridade que consideramos (domínio, entidade, chave, referencial).
- Há no entanto exceções que se prendem com escolhas feitas p/implementação de SGBDs, tipicamente por questões de complexidade de implementação/contexto de uso/desempenho. Por exemplo:
 - SQLite não valida restrições de domínio.
 - Versões antigas de MySQL não tinham suporte p/integridade referencial.



UNIVERSIDADE
PORTUCALENSE

Do conhecimento à prática.