

RESTRIÇÕES DE INTEGRIDADE

Um dos objetivos primordiais de um SGBD é a manutenção da integridade de dados sob seu controle. Dizer que os dados de um banco de dados estão íntegros significa dizer que eles refletem corretamente a realidade representada pelo banco de dados e que são consistentes entre si. Para tentar garantir a integridade de um banco de dados, os SGBDs oferecem o mecanismo de restrição de integridade. Uma restrição de integridade é uma regra de consistência de dados que é garantida pelo próprio SGBD. No caso da abordagem relacional, costuma-se classificar as restrições de integridade nas seguintes categorias:

Integridade de domínio: Restrições deste tipo especificam que o valor de um campo deve obedecer a definição de valores admitidos para a coluna (o domínio da coluna). Nos primeiros SGBDs relacionais, era possível usar apenas domínios pré-definidos (número inteiro, número real, alfanumérico de tamanho definido, data, ...). Em SGBDs mais recentes, o usuário pode definir domínios próprios de sua aplicação (por exemplo, o domínio dos dias da semana ou das unidades da federação).

Integridade de vazio: Através deste tipo de restrição de integridade é especificado se os campos de uma coluna podem ou não ser vazios (se a coluna é obrigatória ou opcional). Como já foi mencionado, campos que compõem a chave primária sempre devem ser diferentes de vazio.

Integridade de chave: Trata-se da restrição que define que os valores da chave primária e alternativa devem ser únicos.

Integridade referencial: É a restrição que define que os valores dos campos que aparecem em uma chave estrangeira devem aparecer na chave primária da tabela referenciada.

As restrições dos tipos acima especificados devem ser garantidas automaticamente por um SGBD relacional, isto é, não deve ser exigido que o programador escreva procedimentos para garanti-las explicitamente. Há outras restrições de integridade que não se encaixam nas categorias acima e que normalmente não são garantidas pelo SGBD. Essas restrições são chamadas de restrições semânticas. Alguns exemplos de restrições desse tipo poderiam ser:

Um empregado do departamento denominado "Finanças" não pode ter a categoria funcional "Engenheiro".

Um empregado não pode ter um salário maior que seu superior imediato.

RESTRIÇÕES DE INTEGRIDADE SOBRE RELAÇÕES

RESTRIÇÃO DE INTEGRIDADE

- Condição especificada no esquema do BD e restringe os dados que podem ser armazenados em uma instância do BD

Se uma instância satisfaz todas as restrições de integridade -> instância válida

UM SGB IMPÕE RESTRIÇÕES DE INTEGRIDADE!

RESTRIÇÕES DE CHAVE

- Declaração de que certo subconjunto mínimo dos campos de uma relação é um identificador único para uma tupla

Ex: 2 alunos não podem ter a mesma identificação

- Conjunto de campos que identifica uma tupla de acordo com uma restrição de chave -> **chave candidata**

Ex: id_aluno na relação Alunos

Na definição de chave

- Duas duplas distintas em instâncias podem ter valores idênticos em todos os campos de uma chave
- Nenhum subconjunto do conjunto de campo sem uma chave é um identificador único para uma tupla

Ex: {id_aluno, nome} não é uma chave para Alunos, mas é uma superchave

Superchave

- Subconjunto de atributos de R que identifique univocamente cada tupla
- Combinação de valores não se repete para superchave

Ex: Aluno = {Nome, Idade, Curso, Id_aluno}

• $SCH1(Aluno) = \{Nome, Id_aluno, Idade\}$

• $SCH2(Aluno) = \{Id_aluno, Nome\}$

Chave

- É uma superchave da qual não se pode retirar nenhum atributo e ainda preservar a propriedade de identificação unívoca

Chave Candidata

- Pode existir mais de uma chave para uma mesma relação
- Cada uma das chaves é chamada de Chave Candidata

$CH1(Aluno) = \{CPF\}$

$CH2(Aluno) = \{Id_aluno\}$

Chave Primária

- Escolhida entre as chaves candidatas (não nula)
- É freqüentemente utilizada para acesso à relação

$CH(Aluno) = \{Id_aluno\}$

Especificando Restrições de Chave em SQL

- Declaração de que um subconjunto das colunas de uma tabela constituem uma chave restrição -> **UNIQUE**
- Chave primária -> **PRIMARY KEY**

```
CREATE TABLE Alunos(
    id_aluno CHAR(20),
    nome CHAR(30),
    login CHAR(20),
    idade INTEGER,
    cpf CHAR(11),
    media REAL,
    UNIQUE (cpf),
    PRIMARY KEY (id_aluno)
);
```

RESTRIÇÕES DE CHAVE ESTRANGEIRA

- Informações armazenadas em uma relação estão ligadas a informações de outra relação

Manutenção de dados consistentes

- RI mais comum envolvendo duas relações -> chave estrangeira

```
Matriculado(  
    id_aluno: string,  
    id_disc: string,  
    nota: string  
);
```

- Garantir que apenas estudantes legítimos possam se matricular nas disciplinas

Valores do campo id_aluno da relação Matriculado deve aparecer no campo id_aluno na relação Alunos

O campo id_aluno de Matriculado é chamado Chave Estrangeira e se refere a Alunos

- A chave estrangeira na relação de referência deve corresponder à chave primária(ou a um campo unique) da relação referenciada

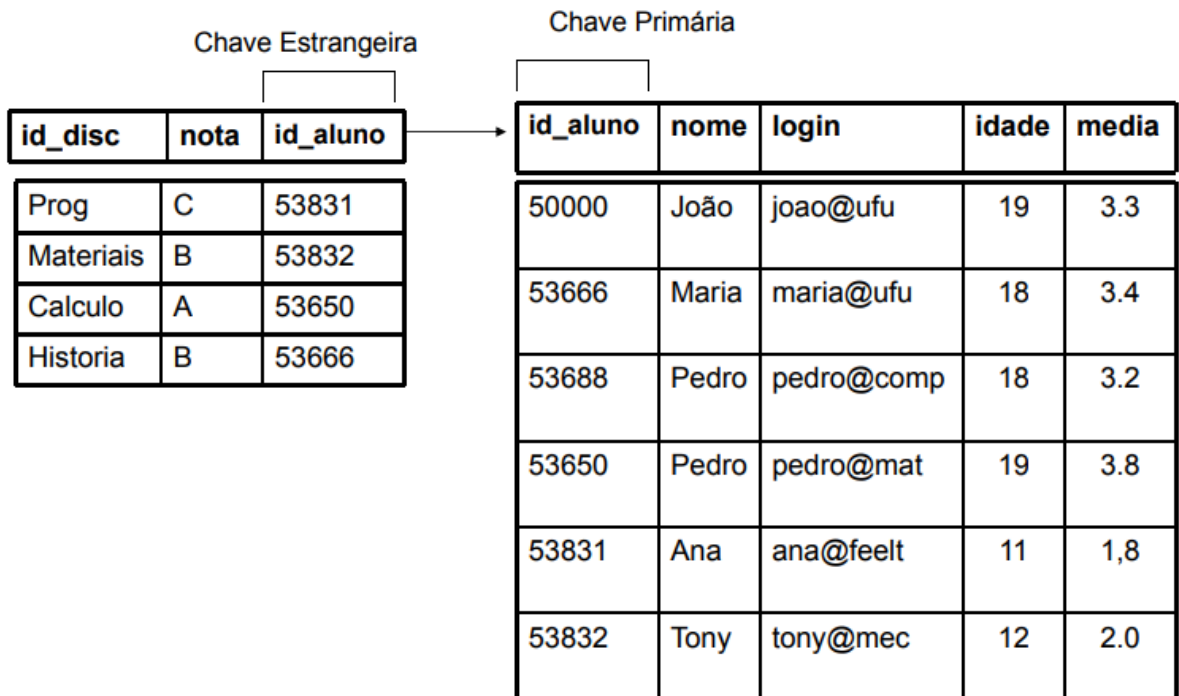
Mesmo número de colunas e tipos de dados compatíveis

Os nomes das colunas podem ser diferentes

- No exemplo matriculado

Todo valor de id_aluno que aparecem em Matriculado aparece na coluna de chave primária de Alunos

Podem existir tuplas de Alunos que não sejam referenciadas a partir de Matriculado



- Uma chave estrangeira pode referenciar a mesma relação

Ex: relação Alunos com uma coluna chamada parceiro

Se o aluno não tiver um parceiro -> usa-se null

A presença de null em um campo de chave estrangeira não viola a restrição de chave estrangeira

Restrições de Chave Estrangeira em SQL

FOREIGN KEY (atributos)

REFERENCES nome_relação (atributos)

[ON UPDATE [NO ACTION | CASCADE| SETNULL| SET DEFAULT]]
[ON DELETE [NO ACTION | CASCADE| SETNULL| SET DEFAULT]]

```
CREATE TABLE Matriculado(
    id_aluno CHAR(20),
    id_disc CHAR(20),
    nota CHAR(10),
    PRIMARY KEY(id_aluno, id_disc),
```

```
FOREIGN KEY(id_aluno)  
REFERENCES Alunos(id_aluno)  
)
```

id_disc também pode ser uma chave estrangeira referenciando uma tabela disciplinas

LISTA

1. Explique o que é uma chave primária.
2. Explique o que é uma chave estrangeira.
3. Descreva o que é integridade referencial da base de dados.
4. Considere o banco de dados relacional definido parcialmente abaixo (faltam as chaves da tabela Empregado):

Empregado (CodEmpregado, Nome, NoPIS-PASEP)

Dependente (CodEmpregado, NoDependente, Nome)

CodEmpregado referencia Empregado

Na tabela Empregado, tanto CodEmpregado quanto NoPIS-PASEP podem ser chave primária. Qual você escolheria como chave primária? Por quê?

5. A seguir aparece um esquema parcial para um banco de dados relacional. Identifique neste esquema as chaves primárias e as chaves estrangeiras:

Aluno (CodigoAluno, Nome, CodigoCurso)

Curso (CodigoCurso, Nome)

Disciplina (CodigoDisciplina, Nome, Creditos, CodigoDepartamento)

Curriculo (CodigoCurso, CodigoDisciplina, Obrigatoria-Opcional)

Conceito (CodigoAluno, CodigoDisciplina, Ano-semester, Conceito)

Departamento (CodigoDepartamento, Nome)

Explique quais verificações devem ser feitas pelo SGBD para garantir a integridade referencial nas seguintes situações:

- a. Uma linha é incluída na tabela Consulta.
- b. Uma linha é excluída da tabela Paciente.
- c. O código do CRM em uma linha de Consulta é alterado.
- d. O código do CRM em uma linha de Médico é alterado.

7. Usando alguma ferramenta CASE para modelagem de banco de dados, construa um esquema diagramático para o banco de dados cujo esquema textual aparece no exercício 4.

VERIFICANDO RESTRIÇÕES DE INTEGRIDADE

- RIs

São especificadas quando uma relação é criada e verificadas quando a mesma é modificada

Se um comando causa uma violação, ele é rejeitado

Toda violação é verificada no final da execução de cada instrução

ex.:

```
INSERT INTO Alunos (  
    id_aluno,  
    nome,  
    login,  
    idade,  
    media  
);
```

```
VALUES (null, 'Mike' , 'mike@ee', 17,3.4)
```

Viola a restrição de chave primária: Valor null no campo chave