

SQL

Restrições de integridade

Vinicius da S. Segalin
vinicius.segalin@posgrad.ufsc

Sumário

- Revisão
- Restrições de integridade
- Exercícios

Sumário

- ! " # \$ % & '
- Restrições de integridade
- Exercícios

Revisão

- Criação de chave primária simples

```
CREATE TABLE pessoa (  
  id_pessoa integer PRIMARY KEY  
);
```

```
CREATE TABLE pessoa (  
  id_pessoa integer,  
  PRIMARY KEY (id_pessoa)  
);
```

```
CREATE TABLE pessoa (  
  id_pessoa integer,  
  CONSTRAINT pk_pessoa PRIMARY KEY (id_pessoa)  
);
```

! () * + (, - . - .

Revisão

- Criação de chave primária composta

```
CREATE TABLE pessoa (  
  id_pessoa integer,  
  nome varchar (100),  
  PRIMARY KEY (id_pessoa, nome)  
);
```

```
CREATE TABLE pessoa (  
  id_pessoa integer,  
  nome varchar (100),  
  CONSTRAINT pk_pessoa PRIMARY KEY (id_pessoa, nome)  
);
```

! () * + (, - . - .

Revisão

- Criação de chave estrangeira simples

```
CREATE TABLE filho (  
  id_filho integer,  
  id_pessoa integer REFERENCES pessoa (id_pessoa)  
);
```

```
CREATE TABLE filho (  
  id_filho integer,  
  id_pessoa integer,  
  FOREIGN KEY (id_pessoa) REFERENCES pessoa (id_pessoa)  
);
```

```
CREATE TABLE filho (  
  id_filho integer,  
  id_pessoa integer,  
  CONSTRAINT fk_pessoa_filho FOREIGN KEY (id_pessoa) REFERENCES pessoa (id_pessoa)  
);
```

! () * + (, - . - .

Revisão

- Criação de chave estrangeira composta

```
CREATE TABLE filho (  
  id_filho integer,  
  id_pessoa integer,  
  nome_pessoa varchar (100),  
  FOREIGN KEY (id_pessoa, nome_pessoa) REFERENCES pessoa (id_pessoa, nome)  
);
```

```
CREATE TABLE filho (  
  id_filho integer,  
  id_pessoa integer,  
  nome_pessoa varchar (100),  
  CONSTRAINT fk_pessoa_filho FOREIGN KEY (id_pessoa, nome_pessoa)  
    REFERENCES pessoa (id_pessoa, nome)  
);
```

! () * + (, - . - .

Sumário

- Revisão
- ! "%/0\$12"%34"3\$5/"60\$474"
 - – "8\$5\$1&'
 - Domínio
 - Entidade
 - Referencial
- Exercícios

Restrições de integridade - Definição

- Restrições que garantem a exatidão e consistência do banco de dados
- Através da validação de restrições definidas sobre os dados na modelagem do BD
- Podem ser especificadas na criação da tabela (CREATE TABLE) ou na sua modificação (ALTER TABLE)
- Exemplos
 - A idade de uma pessoa deve ser maior ou igual a zero
 - A resposta para uma pergunta só pode ser SIM/NÃO
 - Uma cidade só pode ser cadastrada se pertencer a um país

Sumário

- Revisão
- ! "%/0\$12"%34"3\$5/"60\$474"
 - Definição
 - - ' 9:5\$'
 - Entidade
 - Referencial
- Exercícios

Restrições de integridade - Domínio

- Conjunto de valores válidos para um atributo

- Tipo do dado
- Comprimento (tamanho)
- Valor vazio
- Valor único
- Valores possíveis

```
CREATE TABLE pessoa (  
  id_pessoa integer NOT NULL,  
  nome varchar (100),  
  cpf numeric,  
  idade integer,  
  CONSTRAINT cpf_unico UNIQUE (cpf),  
  CONSTRAINT idade_maior CHECK (idade >= 18)  
);
```

Restrições de integridade - Domínio

- Conjunto de valores válidos para um atributo

```
CREATE TABLE pessoa (  
  id_pessoa integer NOT NULL,  
  nome varchar (100),  
  cpf numeric,  
  idade integer,  
  CONSTRAINT cpf_unico UNIQUE (cpf),  
  CONSTRAINT idade_maior CHECK (idade >= 18)  
);
```

```
INSERT INTO pessoa VALUES (1, 'João', 123456789, 20);  
INSERT INTO pessoa VALUES (2, 'Maria', 123456789, 30);  
INSERT INTO pessoa VALUES (3, 'José', 12345, 17);  
INSERT INTO pessoa VALUES ('Carlos', 'Carlos', 1234, 20);
```

ERROR: duplicate key value violates unique constraint "cpf_unico"
DETAIL: Key (cpf)=(123456789) already exists.

ERROR: new row for relation "pessoa" violates check constraint "idade_maior"
DETAIL: Failing row contains (3, José, 12345, 17).

ERROR: invalid input syntax for integer: "Carlos"
INSERT INTO pessoa VALUES ('Carlos', 'Carlos', 1234, 20);

Sumário

- Revisão
- ! "%/0\$12"%34"3\$5/"60\$474"
 - Definição
 - Domínio
 - (5/\$474"
 - Referencial
- Exercícios

Restrições de integridade - Entidade

- Restrição que garante que uma chave primária não pode ser nula e deve identificar um registro (ou seja, ser única)
 - NOT NULL + UNIQUE
 - Identifica um registro -> pode ser referenciado
 - Não é obrigatória, porém aconselhável
 - Pode ser formada por mais de um atributo

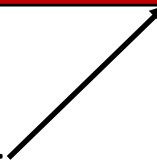
```
CREATE TABLE pessoa (  
    id_pessoa integer,  
    ...  
    CONSTRAINT pk_pessoa PRIMARY KEY (id_pessoa)  
);
```

Restrições de integridade - Entidade


```
CREATE TABLE pessoa (  
  id_pessoa integer,  
  nome varchar (100),  
  CONSTRAINT pk_pessoa PRIMARY KEY (id_pessoa)  
);
```

```
INSERT INTO pessoa (id_pessoa, nome) VALUES (1, 'João');  
INSERT INTO pessoa (id_pessoa, nome) VALUES (1, 'Maria');  
INSERT INTO pessoa (nome) VALUES ('José');
```

ERROR: duplicate key value violates unique constraint "pk_pessoa"
DETAIL: Key (id_pessoa)=(1) already exists.



ERROR: null value in column "id_pessoa" violates not-null constraint
DETAIL: Failing row contains (null, José).

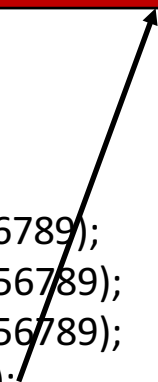


Restrições de integridade - Entidade

```
CREATE TABLE pessoa (  
  id_pessoa integer,  
  nome varchar (100),  
  cpf numeric,  
  CONSTRAINT pk_pessoa PRIMARY KEY (id_pessoa, nome)  
);
```

```
INSERT INTO pessoa (id_pessoa, nome, cpf) VALUES (1, 'João', 123456789);  
INSERT INTO pessoa (id_pessoa, nome, cpf) VALUES (1, 'Maria', 123456789);  
INSERT INTO pessoa (id_pessoa, nome, cpf) VALUES (2, 'Maria', 123456789);  
INSERT INTO pessoa (id_pessoa, nome, cpf) VALUES (2, 'Maria', 1234);
```

ERROR: duplicate key value violates unique constraint "pk_pessoa"
DETAIL: Key (id_pessoa, nome)=(2, Maria) already exists.



	id_pessoa integer	nome character varying(100)	cpf numeric
1	1	João	123456789
2	1	Maria	123456789
3	2	Maria	123456789

Sumário

- Revisão
- ! "%/0\$12"%34"3\$5/"60\$474"
 - Definição
 - Domínio
 - Entidade
 - ! "8"0"5;\$7<
- Exercícios

Restrições de integridade - Referencial

- Restrição que garante a referência entre tabela filha e tabela pai
 - Um registro não pode ser deletado se este for referenciado por outro
 - Um registro não pode ser modificado se este for referenciado por outro
 - Um registro só pode referenciar outro se este já existir
 - Um registro pode referenciar ninguém (null)

Restrições de integridade - Referencial

- Um registro não pode ser deletado se este for referenciado por outro

```
CREATE TABLE pai (  
  id_pai integer,  
  nome varchar (100),  
  CONSTRAINT pk_pai PRIMARY KEY (id_pai)  
);
```

	id_pai integer	nome character varying(100)
1	1	Pai do João

```
CREATE TABLE filho (  
  id_filho integer,  
  nome varchar (100),  
  id_pai integer,  
  CONSTRAINT fk_pai_filho FOREIGN KEY (id_pai) REFERENCES pai (id_pai)  
);
```

	id_filho integer	nome character varying(100)	id_pai integer
1	1	João	1

ERROR: update or delete on table "pai" violates foreign key constraint "fk_pai_filho" on table "filho"
DETAIL: Key (id_pai)=(1) is still referenced from table "filho".

```
INSERT INTO pai VALUES (1, 'Pai do João');  
INSERT INTO filho VALUES (1, 'João', 1);  
DELETE FROM pai WHERE id_pai = 1;
```

Restrições de integridade - Referencial

- Um registro não pode ser modificado se este for referenciado por outro

```
CREATE TABLE pai (  
  id_pai integer,  
  nome varchar (100),  
  CONSTRAINT pk_pai PRIMARY KEY (id_pai)  
);
```

	id_pai integer	nome character varying(100)
1	1	Pai do João

```
CREATE TABLE filho (  
  id_filho integer,  
  nome varchar (100),  
  id_pai integer,  
  CONSTRAINT fk_pai_filho FOREIGN KEY (id_pai) REFERENCES pai (id_pai)  
);
```

	id_filho integer	nome character varying(100)	id_pai integer
1	1	João	1

```
INSERT INTO pai VALUES (1, 'Pai do João');  
INSERT INTO filho VALUES (1, 'João', 1),  
UPDATE pai SET id_pai = 2;
```

ERROR: update or delete on table "pai" violates foreign key constraint "fk_pai_filho" on table "filho"
DETAIL: Key (id_pai)=(1) is still referenced from table "filho".

Restrições de integridade - Referencial

- Um registro só pode referenciar outro se este já existir

```
CREATE TABLE pai (  
  id_pai integer,  
  nome varchar (100),  
  CONSTRAINT pk_pai PRIMARY KEY (id_pai)  
);
```

	id_pai integer	nome character varying(100)
1	1	Pai do João

```
CREATE TABLE filho (  
  id_filho integer,  
  nome varchar (100),  
  id_pai integer,  
  CONSTRAINT fk_pai_filho FOREIGN KEY (id_pai) REFERENCES pai (id_pai)  
);
```

	id_filho integer	nome character varying(100)	id_pai integer
1	1	João	1

```
INSERT INTO pai VALUES (1, 'Pai do João');  
INSERT INTO filho VALUES (1, 'João', 1);  
INSERT INTO filho VALUES (2, 'Maria', 2);
```

ERROR: insert or update on table "filho" violates foreign key constraint "fk_pai_filho"
DETAIL: Key (id_pai)=(2) is not present in table "pai".

Restrições de integridade - Referencial

- Um registro pode referenciar ninguém (null)

```
CREATE TABLE pai (  
  id_pai integer,  
  nome varchar (100),  
  CONSTRAINT pk_pai PRIMARY KEY (id_pai)  
);
```

	id_pai integer	nome character varying(100)
1	1	Pai do João

```
CREATE TABLE filho (  
  id_filho integer,  
  nome varchar (100),  
  id_pai integer,  
  CONSTRAINT fk_pai_filho FOREIGN KEY (id_pai) REFERENCES pai (id_pai)  
);
```

```
INSERT INTO pai VALUES (1, 'Pai do João');  
INSERT INTO filho VALUES (1, 'João', 1);  
INSERT INTO filho VALUES (2, 'Maria');
```

	id_filho integer	nome character varying(100)	id_pai integer
1	1	João	1
2	2	Maria	

Restrições de integridade - Referencial

- E se eu quiser deletar uma PK referenciada?

- ON DELETE CASCADE

```
CREATE TABLE filho (
  id_filho integer,
  nome varchar (100),
  id_pai integer,
```

```
CONSTRAINT fk_pai_filho FOREIGN KEY (id_pai) REFERENCES pai (id_pai) * , 3- (=(>(3). ?). - (
);
```

```
ALTER TABLE filho
DROP CONSTRAINT fk_pai_filho,
ADD CONSTRAINT fk_pai_filho FOREIGN KEY (id_pai)
REFERENCES pai (id_pai) ON DELETE CASCADE;
```

	id_pai integer	nome character varying(100)
1	1	Pai do João
2	2	Pai da Maria

	id_filho integer	nome character varying(100)	id_pai integer
1	1	João	1
2	2	Irmão do João	1
3	3	Maria	2

```
DELETE FROM pai WHERE id_pai = 1;
```

	id_pai integer	nome character varying(100)
1	2	Pai da Maria

	id_filho integer	nome character varying(100)	id_pai integer
1	3	Maria	2

Restrições de integridade - Referencial

- E se eu quiser deletar uma PK referenciada?

- ON DELETE SET NULL

CREATE TABLE filho (

id_filho integer,

nome varchar (100),

id_pai integer,

CONSTRAINT fk_pai_filho FOREIGN KEY (id_pai) REFERENCES pai (id_pai) * , 3- (=(>{?(>3, @==

);

	id_pai integer	nome character varying(100)
1	1	Pai do João
2	2	Pai da Maria

	id_filho integer	nome character varying(100)	id_pai integer
1	1	João	1
2	2	Irmão do João	1
3	3	Maria	2

```
ALTER TABLE filho
DROP CONSTRAINT fk_pai_filho,
ADD CONSTRAINT fk_pai_filho FOREIGN KEY (id_pai)
REFERENCES pai (id_pai) ON DELETE SET NULL;
```

DELETE FROM pai WHERE id_pai = 1;

	id_pai integer	nome character varying(100)
1	2	Pai da Maria

	id_filho integer	nome character varying(100)	id_pai integer
1	3	Maria	2
2	1	João	
3	2	Irmão do João	

Restrições de integridade - Referencial

- E se eu quiser modificar uma PK referenciada?

- ON UPDATE CASCADE

```
CREATE TABLE filho (
  id_filho integer,
  nome varchar (100),
  id_pai integer,
  CONSTRAINT fk_pai_filho FOREIGN KEY (id_pai) REFERENCES pai (id_pai) ON UPDATE CASCADE;
);
```

```
ALTER TABLE filho
DROP CONSTRAINT fk_pai_filho,
ADD CONSTRAINT fk_pai_filho FOREIGN KEY (id_pai)
REFERENCES pai (id_pai) ON UPDATE CASCADE;
```

	id_pai integer	nome character varying(100)
1	1	Pai do João
2	2	Pai da Maria

	id_filho integer	nome character varying(100)	id_pai integer
1	1	João	1
2	2	Irmão do João	1
3	3	Maria	2

```
UPDATE pai
SET id_pai = 3
WHERE id_pai = 1;
```

	id_pai integer	nome character varying(100)
1	2	Pai da Maria
2	3	Pai do João

	id_filho integer	nome character varying(100)	id_pai integer
1	3	Maria	2
2	1	João	3
3	2	Irmão do João	3

Restrições de integridade - Referencial

- Combinações possíveis
 - ON DELETE CASCADE ON UPDATE CASCADE
 - ON DELETE CASCADE ON UPDATE SET NULL
 - ON DELETE SET NULL ON UPDATE CASCADE
 - ON DELETE SET NULL ON UPDATE SET NULL

Sumário

- Revisão
- Restrições de integridade
- $(B''0; :: \$' \%$