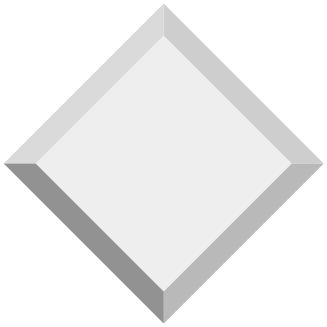
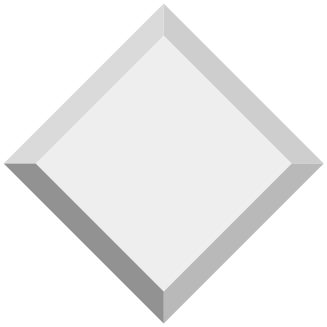


Capítulo 6
Regras de Integridade

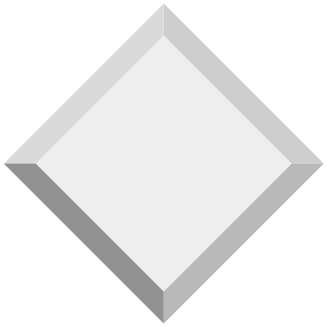
Murilo S. de Camargo

(Modificações M.A R. Dantas)



Restrições de Integridade

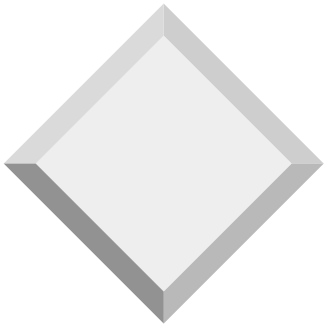
- Restrições de Domínio
- Integridade Referencial
- Asserções
- Gatilhos (Triggers)
- Dependências Funcionais



Observações Gerais

- Alguns autores usualmente denominam o conteúdo desse capítulo como ***constraints and triggers***.

Exemplo : Molina, H. G., Ullman, J. D. and Widom, J., *Database Systems - The Complete Book*, Prentice Hall, 2002.

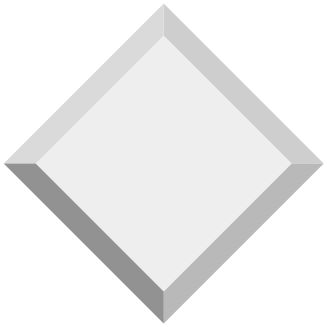


Restrições de Integridade

- Restrições de Domínio ✕
- Integridade Referencial
- Asserções
- Gatilhos (Triggers)
- Dependências Funcionais

Restrições de Domínio

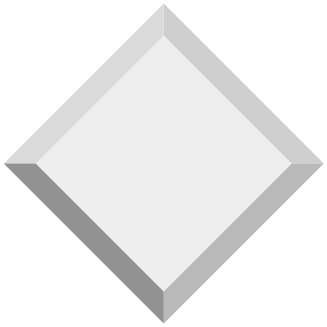
O que vem a ser uma restrição de domínio ?



Restrições de Domínio

- As restrições de integridade resguardam o Banco de Dados contra danos acidentais, assegurando que mudanças feitas ***por usuários autorizados*** não resultem na perda de consistência de dados.

Restrições de Domínio



Restrições de Domínio

- Restrições de domínio são a forma mais elementar de restrições de integridade.

- ***Estas*** testam valores inseridos no Banco de Dados, e testam (***efetuam***) consultas para assegurar que as comparações façam sentido.

Observações Gerais

- Considere os seguintes atributos :

  - Nome_cliente
  - Nome_empregado
  - Saldo
  - Nome_agência

  Caso 1 - É razoável imaginarmos que Nome_cliente e Nome_empregado estejam em um ***mesmo domínio***.

Observações Gerais

- Considerando ainda os atributos :

  - Nome_cliente
  - Nome_empregado
  - Saldo
  - Nome_agência

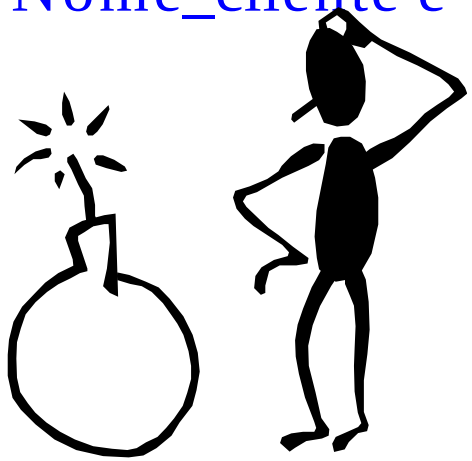
Caso 2 - É razoável imaginarmos que Saldo e Nome_agência estejam em ***domínios distintos***.

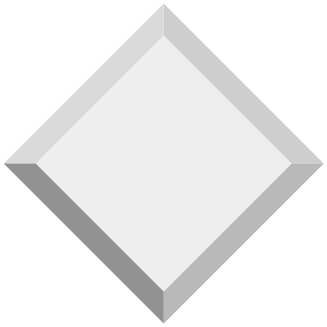
Observações Gerais

- Considerando ainda os atributos :

  - Nome_cliente
  - Nome_empregado
  - Saldo
  - Nome_agência

Caso 3 - É razoável imaginarmos que Nome_cliente e Nome_agência estejam em :
(a) um mesmo domínio,
(b) domínios distintos





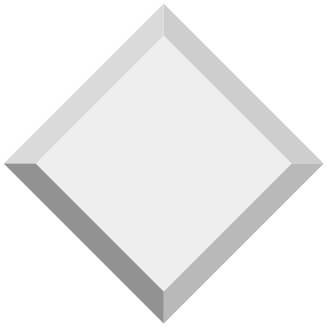
Restrições de Domínio

- A cláusula **check** em SQL-92 permite restringir domínios:

  – Exemplo1 :

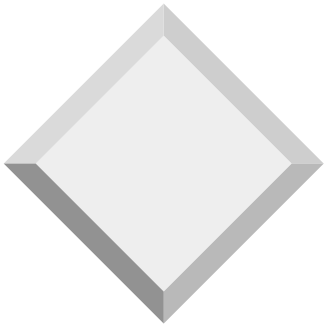
  Utilize a cláusula ***check*** para assegurar que um domínio de hora em Hora-Salario permita só valores maiores do que o especificado.

  create domain hora-salario **numeric(5,2)**
  constraint valor-teste **check**(valor >=4.00)



Restrições de Domínio

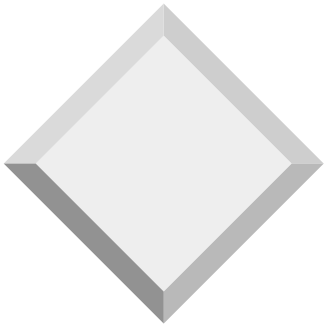
- O domínio de hora em **hora-salario** é declarado como um número decimal com 5 dígitos e 2 decimais.
- O domínio tem uma restrição que assegura que os valores do atributo "valor" deverá ser maior ou igual a 4.00.
- A cláusula **constraint** (que é opcional) serve para indicar qual a restrição que violou a atualização.



Restrições de Domínio

Suponha que a você foi solicitado que em um determinado Banco de Dados alguns valores não deveriam ser nulos.

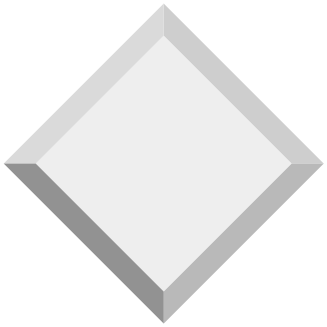
Como você faria ?



Restrições de Domínio

- A cláusula **check** usada para restringir os valores nulos em um domínio
  - Exemplo 2 :

```
create domain numero_conta char (10)
   constraint teste_nulo_nconta check(value not null)
```

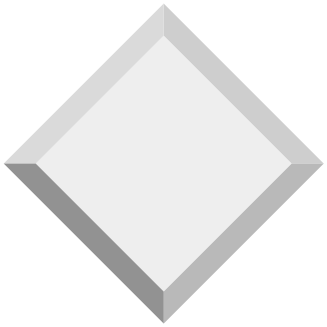


Restrições de Domínio

❖ A cláusula **check** usada para restringir um determinado conjunto de valores por meio do uso da cláusula **in**

– Exemplo 3 :

create domain tipo_conta **char (10)**
constraint teste_tipo_conta **check**(**value in (“corrente”, “Poupança”))**



Restrições de Integridade

- Restrições de Domínio
- Integridade Referencial ✕
- Asserções
- Gatilhos (Triggers)
- Dependências Funcionais

Integridade Referencial

- Assegura que um valor que aparece em uma relação (***tabela***) para um determinado conjunto de atributos apareça em outro conjunto de atributos em outra relação (***tabela***).
  - Exemplo:

Se "**Perryridge**" é um nome de filial que aparece na tupla (***linha***) da relação (***tabela***) **conta**, então deve existir uma tupla (***linha***) "**Perryridge**" na relação (***tabela***) **agencia**.

Observações Gerais

- Como as tabelas em essência são relações, utiliza-se os termos matemáticos ***relação*** e ***tupla,*** no lugar de ***tabela*** e ***linhas***. Assim :

tabela - relação
linha - tupla

- Como uma ***relação*** é um conjunto de ***tuplas***, podemos usar a notação matemática $t \in r$ para denotar que a ***tupla t*** está na ***relação r***.

Integridade Referencial

- Definição Formal:

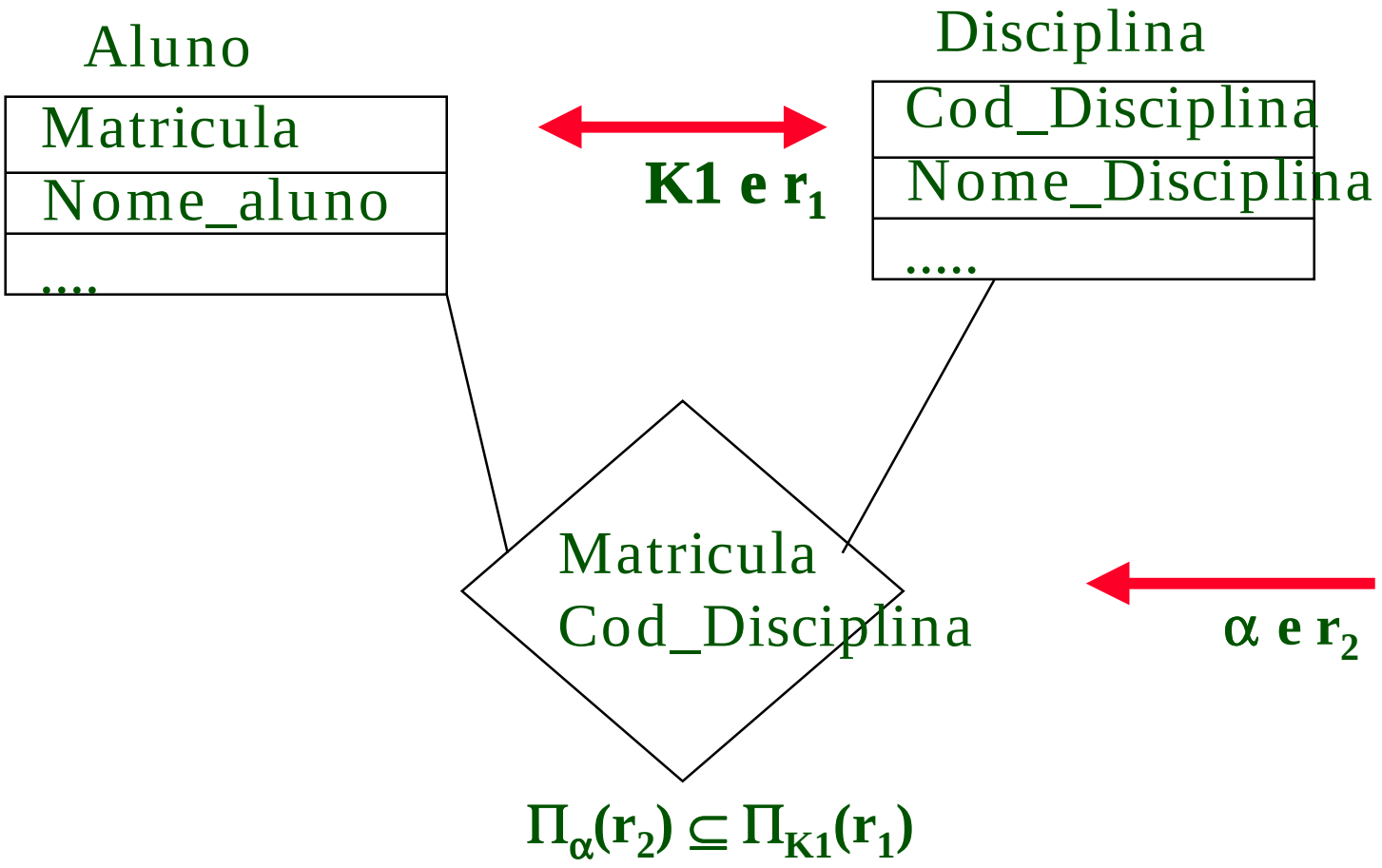
  - Sejam as relações $r_1(R_1)$ e $r_2(R_2)$ com chaves primárias K_1 e K_2, respectivamente.

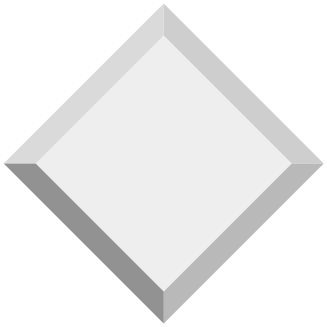
  - O subconjunto α de R_2 é uma **chave estrangeira** referenciando K_1 na relação r_1, se para toda relação t_2 em r_2 existir uma tupla t_1 em r_1 tal que $t_1[K_1]=t_2[\alpha]$.

  - Restrições de integridade podem ser descritas :

$$\Pi_{\alpha}(r_2) \subseteq \Pi_{K1}(r_1)$$

Restrições de Integridade



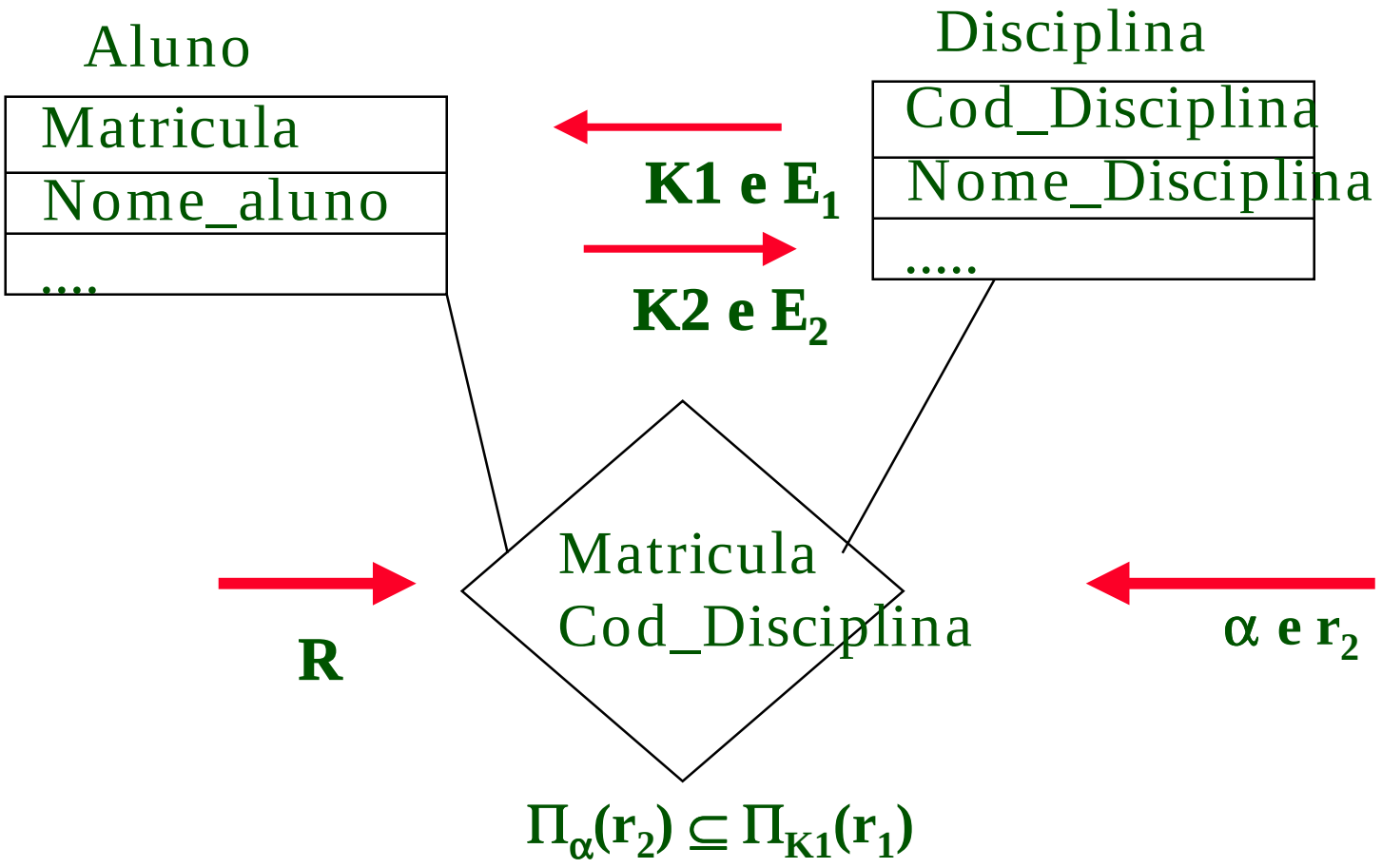


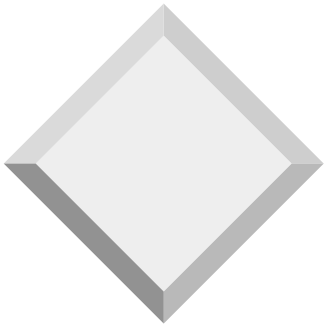
Integridade Referencial no Modelo E-R

❖ Considere o conjunto de relacionamentos R entre as entidades E_1 e E_2. O esquema relacional para R inclui as chaves primárias K_1 de E_1 e K_2 de E_2.

Então K_1 e K_2 formam chaves estrangeiras no esquema relacional de E_1 e E_2, respectivamente.

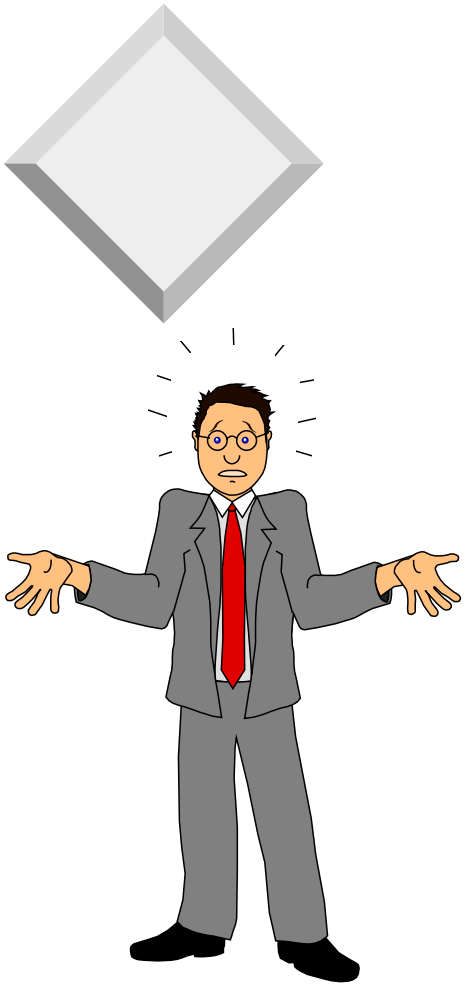
Restrições de Integridade





Integridade Referencial no Modelo E-R

❖ Entidades fracas também são uma fonte de restrições de integridade referencial. O esquema de relação para uma entidade fraca deve incluir a chave primária da entidade da qual ela depende.



Modificações no Banco de Dados

O que seriam modificações no Banco de Dados ?

- **Inserção**
- **Remoção**
- **Atualização**

Modificações no Banco de Dados

- Os testes apresentados a seguir devem ser efetuados para cada tipo de modificação no banco de dados, de maneira a preservar a seguinte restrição de integridade referencial:

$$\Pi_{\alpha}(r_2) \subseteq \Pi_{K}(r_1)$$

Modificações no Banco de Dados

- **Insert.** Se uma tupla (***linha***) t_2 é inserida em r_2 (***tabela***),o sistema precisa assegurar que existe uma tupla (***linha***) t_1 em r_1 (***na tabela***)tal que $t_1[K]=t_2[\alpha]$. Isto é:

$$t_2[\alpha] \in \Pi_K(r_1)$$

Vamos supor a inserção da aluna Andréia na *tabela* disciplina BD. A aluna deve existir na tabela r_1 para que a inserção ocorra sem erros (sem inconsistência no Banco de Dados).

Modificações no Banco de Dados

- **Delete.** Se uma tupla (***linha***) t_1 é removida de r_1 (***tabela***) o sistema precisa computar o conjunto de tuplas (***linhas***)em r_2 (***na tabela***) que referencia t_1 :

$$\sigma_{\alpha=t1[K]} (r_2)$$

Se este conjunto não estiver vazio, então ou o comando **delete** é rejeitado com um erro, ou as tuplas de t_1 devem ser removidas (cascateando a deleção se possível).

Modificações no Banco de Dados

- **Update -** Existem dois casos:
  - Se uma tupla t_2 é atualizada na relação r_2 e a atualização modifica valores para a chave estrangeira α, então é feito um teste similar ao caso de inserção. Seja t_2' denotando o novo valor da tupla t_2. O sistema deve assegurar que:

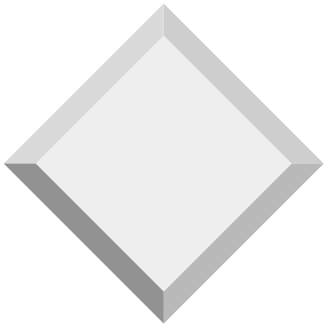
$$t_2'[\alpha] \in \Pi_K(r_1)$$

Modificações no Banco de Dados

- Se uma tupla t_1 é atualizada em r_1, e a atualização modifica valores da chave primária (K), então um teste similar ao caso de **delete** deve ser feito. O sistema precisa computar :

$$\sigma_{(\alpha = t1)}[K](r_2)$$

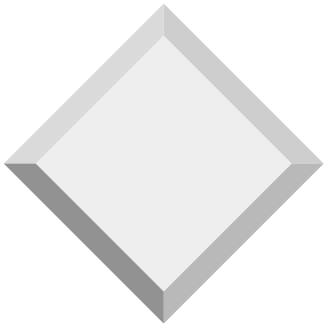
usando o antigo valor de t_1 (o valor anterior à aplicação da atualização). Se este conjunto não é vazio, a atualização é rejeitada com um erro, ou a atualização é cascateada nas tuplas do conjunto, ou ou as tuplas do conjunto podem ser removidas.



Integridade Referencial em SQL

- Chaves ***primárias***, ***candidatas*** e chaves ***estrangeiras*** podem ser especificadas como parte da declaração **create table** do SQL:

  - A cláusula **primary key** da declaração **create table** inclui uma lista de atributos que compreendem a chave primária.



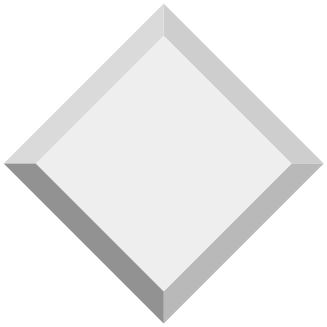
Integridade Referencial em SQL

- Chaves ***primárias***, ***candidatas*** e chaves ***estrangeiras*** podem ser especificadas como parte da declaração **create table** do SQL:
  - A cláusula **unique key** da declaração **create table** inclui uma lista de atributos que compreendem a chave candidata.

Integridade Referencial em SQL

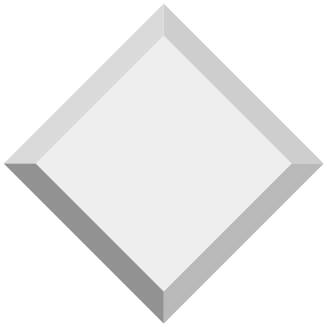
- Chaves ***primárias***, ***candidatas*** e chaves ***estrangeiras*** podem ser especificadas como parte da declaração **create table** do SQL:

  - A cláusula **foreing key** da declaração **create table** inclui uma lista de atributos que compreendem a chave estrangeira e o nome da relação referida pela chave estrangeira.



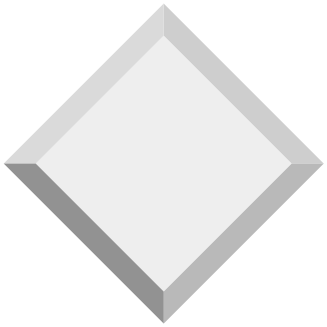
Integridade Referencial em SQL - Exemplo

```
create table cliente
        (nome-cliente      char(20) not null,
         rua                        char(30),
         cidade             char(30),
         primary key (nome-cliente))
```



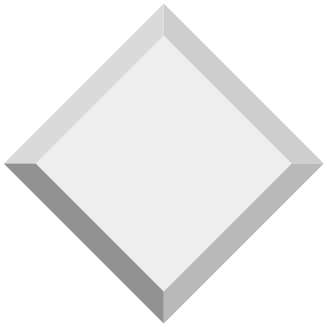
Integridade Referencial em SQL - Exemplo

```
create table agencia
        (nome-agencia      char(15) not null,
         cidade-agencia    char(30),
         ativos            integer,
         primary key (nome-agencia))
```



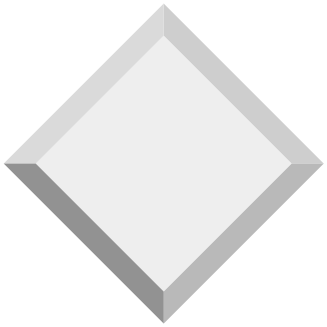
Integridade Referencial em SQL - Exemplo

```
create table conta
    (nome-agencia        char(15),
    numero-conta         char(10) not null,
    saldo                integer,
    primary key (numero-conta),
    foreign key (nome-agencia) references agencia)
```



Integridade Referencial em SQL - Exemplo

create table depositante
(nome-cliente char(20) **not nul**l,
numero-conta char(10) **not nul**l,
primary key (nome-cliente, numero-conta),
foreign key (numero-conta) **references c**onta,
foreign key (nome-cliente) **references c**liente)



Ações em Cascata em SQL

create table conta

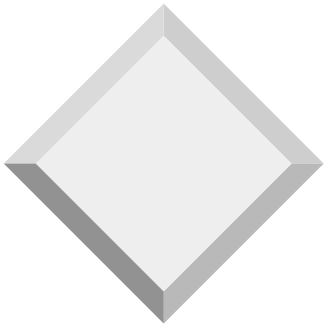
...

foreign key (nome-agencia) **references** agencia

on delete cascade

on update cascade,

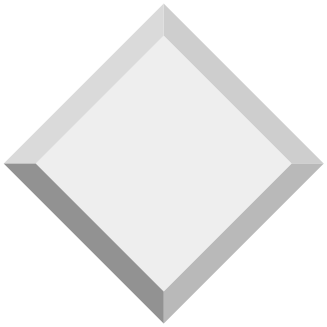
...)



Ações em Cascata em SQL

– Devido às cláusulas **on delete cascade,** se a remoção de uma tupla (***linha***) em a**gencia (*na tabela*)** resultar em violação da restrição de integridade referencial, a remoção é feita em "cascata" na relação (***tabela***) **conta,** removendo as tuplas (***linhas***) que se referem à agência que foi removida.

– Atualizações em "cascata" são semelhantes.

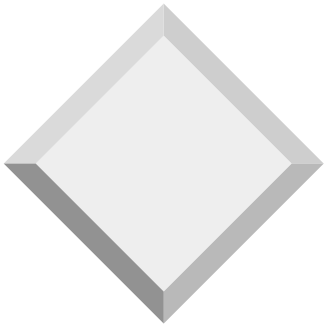


Ações em Cascata em SQL

- Se existe uma cadeia de dependências de chave estrangeira através de múltiplas relações, com

  on delete cascade

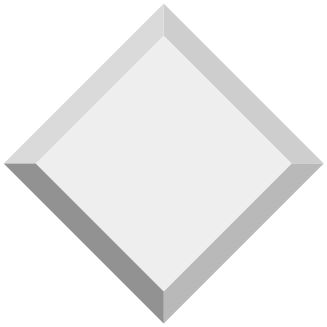
especificado para cada dependência, uma remoção ou atualização no final da cadeia pode se propagar através de toda a cadeia.



Ações em Cascata em SQL

- Se um "***cascateamento***" de atualização ou remoção causa uma violação de restrição que não pode ser tratada por uma operação em cascata subseqüente, o sistema aborta a transação.

Como resultado, todas as mudanças causadas por uma transação e suas ações de "***cascateamento***" serão desfeitas.



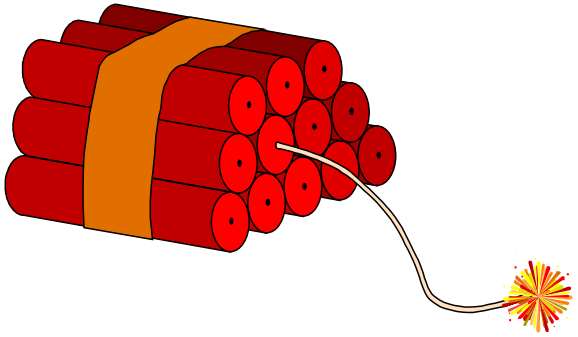
Restrições de Integridade

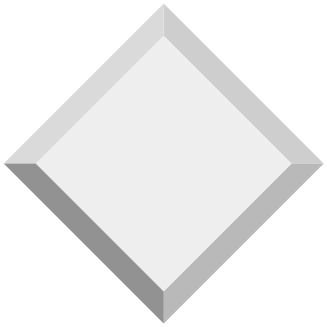
- Restrições de Domínio
- Integridade Referencial
- Asserções ✕
- Gatilhos (Triggers)
- Dependências Funcionais

Asserções

Asserções ?

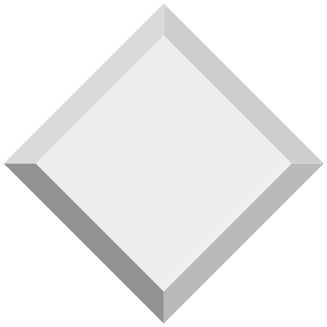
O que vem a ser ***isto ?***





Asserções

- Uma ***asserção*** é um predicado expressando uma condição que queremos que o Banco de Dados sempre satisfaça.



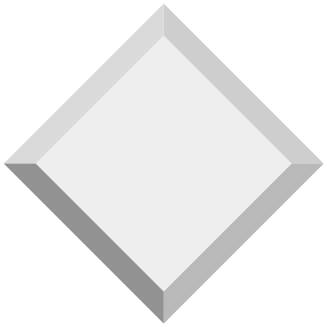
Asserções

- Uma asserção em SQL-92 tem a forma

create assertion <nome-asserção> **check** <predicado>

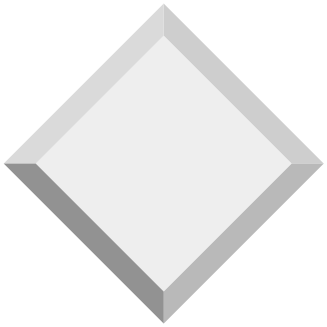
Quando uma asserção é feita, o sistema testa a sua validade. Este teste pode introduzir uma quantidade significativa de sobrecarga; assim as asserções devem ser usadas com grande cuidado.

E quanto a portabilidade ?



Asserções - Exemplo

- A soma de todos os totais dos empréstimos para cada agência deve ser menor do que a soma de todos os saldos das contas na agência.



Asserções - Exemplo

- **create assertion** restricao-soma **check**

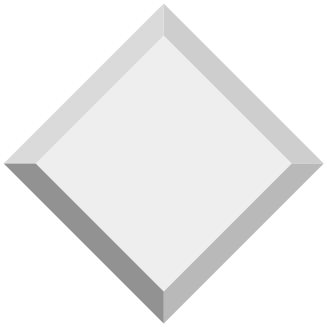
  (not exists (select * **from** agencia

  where (select sum(total) **from** emprestimo

  where emprestimo.nome-agencia = agencia. nome-agencia)

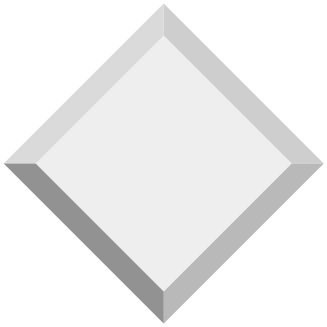
  >= (**select sum**(saldo) **from** conta

  where emprestimo. nome-agencia =agencia.nome-agencia)))



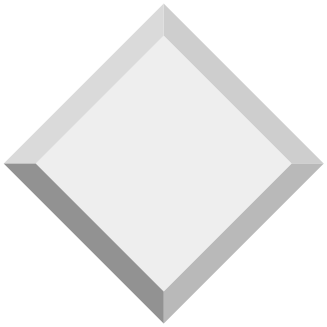
Asserções Exemplo

- Todo cliente de empréstimo precisa manter uma conta com o saldo mínimo de $1000.00.



Asserções Exemplo

```
create assertion restricao-saldo check
    (not exists (select * from emprestimo
      where not exists ( select *
        from devedor, depositante, conta
        where emprestimo.numero-emprestimo =
      devedor.numero-emprestimo
            and devedor.nome-cliente = depositante.nome-cliente
            and depositante.numero-conta = conta.numero-conta
            and conta.saldo >= 1000)))
```



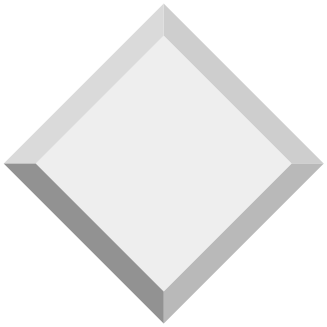
Restrições de Integridade

- Restrições de Domínio
- Integridade Referencial
- Asserções
- Gatilhos (Triggers) ✗
- Dependências Funcionais

Gatilhos (Triggers)

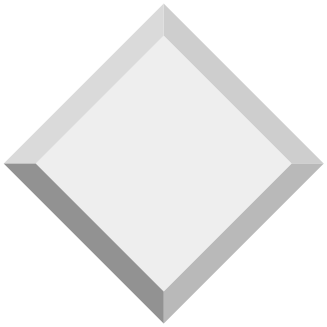
Gatilhos (Triggers)

O que vem a ser um gatilho em um Banco de Dados **?**



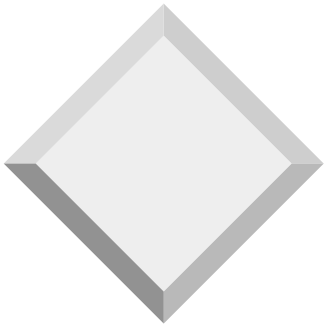
Gatilhos (Triggers)

- Um *gatilho* é um comando executado automaticamente pelo sistema como um efeito de uma modificação no Banco de Dados.



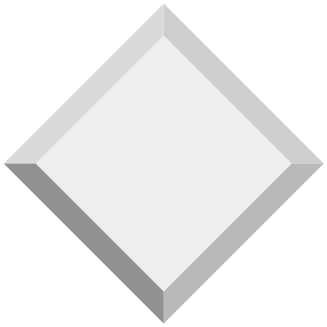
Gatilhos (Triggers)

- Para projetar um *gatilho*, precisamos:
  - Especificar as condições sob as quais o *gatilho* deve ser executado.
  - Especificar as ações a serem tomadas quando o *gatilho* é executado.
- O SQL-92 não inclui os *gatilhos*, mas muitas implementações suportam *gatilhos*.



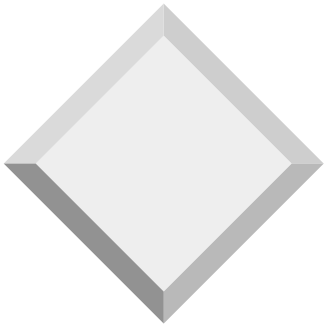
Exemplo Gatilhos

- Suponha que em vez de permitir saldos negativos, o banco trate saque a descoberto assim :
  - ajustando o saldo para zero
  - criando um empréstimo no valor da quantia saldo negativo
  - a este empréstimo é dado um número igual ao número da conta estourada



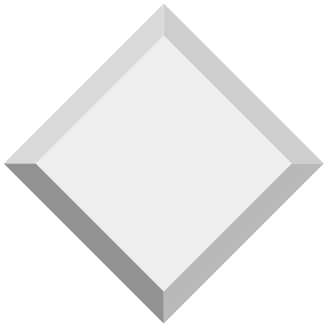
Exemplo Gatilhos

- A condição para executar o *trigger* é uma atualização na relação depósito que resulte em um valor de saldo negativo.



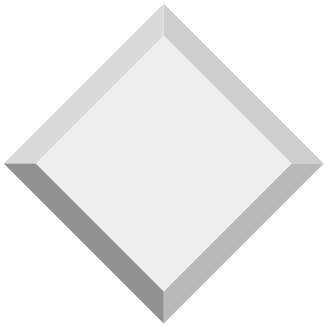
Exemplo Gatilhos

```
define trigger  saque-descoberto on update of conta T
   (if new T.saldo < 0
   then (insert into emprestimo values
                (T.nome-agencia,T.numero-conta, - new T.saldo)
          insert into devedor
                (select nome-cliente, numero-conta
                from depositante
                where T.numero-conta = depositante.numero-conta)
         update conta  S
         set  S.saldo =0
         where S.numero-conta =T.numero-conta))
```



Exemplo Gatilhos

A declaração **new** usado antes de T.saldo indica que o valor de T.saldo depois da atualização deve ser usado; se é omitido, o valor antes da atualização é usado.



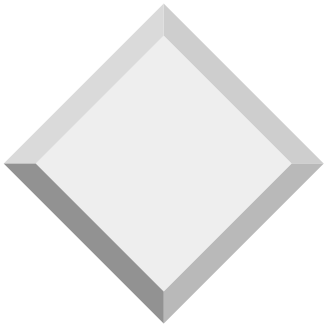
Restrições de Integridade

- Restrições de Domínio
- Integridade Referencial
- Asserções
- Gatilhos (Triggers)
- Dependências Funcionais ✗

Dependências Funcionais

O que vem a ser as dependências funcionais ?

Para que servem **?**



Dependências Funcionais

- Restrições ao conjunto de relações válidas.
- Requerem que o valor para um certo conjunto de atributos determine unicamente o valor para outro conjunto de atributos.
- A noção da dependência funcional generaliza a noção de superchave.

Dependências Funcionais

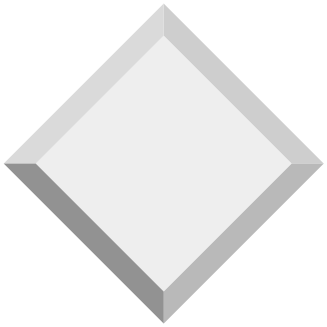
- Seja R o esquema de uma relação com $\alpha \subseteq \mathbf{R}, \beta \subseteq \mathbf{R}$

- A dependência funcional

$$\alpha \rightarrow \beta$$

realiza-se em R se, e somente se em qualquer relação válida r(R), sempre que duas tuplas t_1 e t_2 de r combinam nos atributos α, eles também combinam nos atributos β. Isto é,

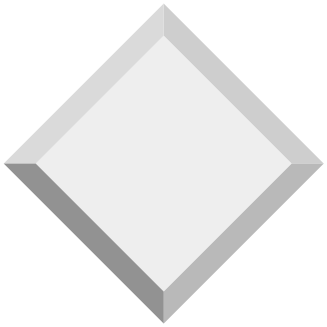
$$\mathbf{t_1[\alpha]=t_2[\alpha] \Rightarrow t_1[\beta]=t_2[\beta]}$$



Dependências Funcionais

- K é uma superchave para a relação R se e somente se $K \rightarrow R$

- K é uma chave candidata para R se, e somente se
  - $K \rightarrow R$, and
  - Para nenhum $\alpha \subset K$, $\alpha \rightarrow R$

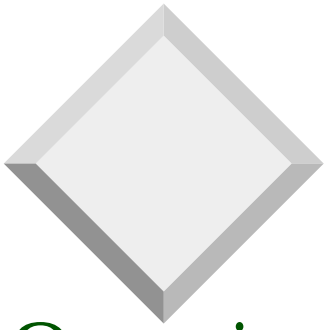


Dependências Funcionais

- A dependência funcional nos permite expressar restrições que as superchaves não expressam.

Considere o esquema:

esquema_info_emprestimo = (nome_agência, número_empréstimo, número_cliente, total)



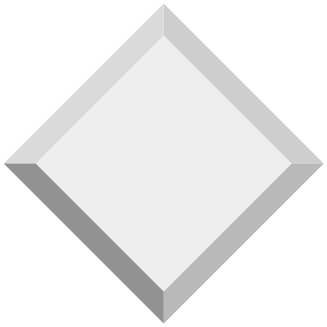
Dependências Funcionais

O conjunto de dependências funcionais que queremos garantir para esse esquema de relação é:

$$\text{numero\_emprestimo} \rightarrow \text{total}$$
$$\text{numero\_emprestimo} \rightarrow \text{nome\_agencia}$$

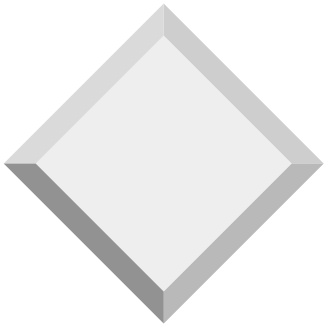
entretanto, não esperamos que a seguinte dependência funcional se verifique:

$$\text{numero\_emprestimo} \rightarrow \text{nome\_cliente}$$



Uso de Dependências Funcionais

- Usamos dependência funcional para:
  - testar relações para ver se elas são válidas sob um dado conjunto de dependências funcionais. Se uma relação R é válida sob um conjunto F de dependências funcionais, diz-se que r satisfaz F.



Uso de Dependências Funcionais

- Usamos dependência funcional para:
  - especificar restrições no conjunto de relações válidas; diz-se que F vale em R se todas as operações em R satisfazem o conjunto de dependências funcionais F.

Uso de Dependências Funcionais

- Nota:

Uma instância específica de um esquema de relação pode satisfazer uma dependência funcional mesmo se a dependência funcional não valha em todas as instâncias legais. Por exemplo, uma instância específica de esquema-emprestimo, por acaso satisfaz

numero-emprestimo → nome-cliente