

Linguagem SQL DDL Trigger e Sequence

Profa Dra Jeroniza Nunes Marchaukoski

Linguagem SQL DDL – Gatilho (trigger)

- Conceitos:
 - São programas que possuem declaração de variáveis, execução de comandos (if, while...) como toda linguagem estruturada.
 - Programas trigger também podem executar comandos SQL (Select...., insert... Update)

Linguagem SQL DDL – Gatilho (trigger)

- Conceitos:
 - O trigger se liga a um objeto do banco de dados e é dependente dele.
 - Um trigger é executado sempre que um evento de disparo ocorre no objeto ao qual ele está ligado.
 - Os eventos de Tabela que disparam um trigger são: **insert**, **update** ou **delete**.

Linguagem SQL DDL – Gatilho (trigger)

- Uso:
 - Para elaborar restrições de integridade complexas impossíveis de serem geradas na criação das tabelas.
 - Para registrar informações diversas sobre modificações de tabelas.
 - Para sinalizar outros programas das sobre modificações ocorridas.

Linguagem SQL DDL – Gatilho (trigger)

- Uso:
 - Aplicação de integridade referencial.
 - Registro de ações ocorridas e seu armazenamento em tabelas/arquivos de log da base de dados/S.O.
 - Auditoria.
 - Replicação sincronizada de tabelas.
 - Prevenção de transações inválidas.
 - Imposição de autorização de segurança.
 - Geração de serial para armazenar na tabela.

Linguagem SQL DDL – Gatilho (trigger)

- Sintaxe (Oracle - Tabela):
 - Create or replace trigger nome_trigger
 - Before/after insert or update or delete on nome_tabela
 - For each row
 - Declare
 - Variavel tipo
 - Begin
 - If inserting then
 - Comandos....
 - :new.campo := valor ...
 - Elself updating then
 - Comandos...
 - End if;
 - End;

Linguagem SQL DDL – Gatilho (trigger)

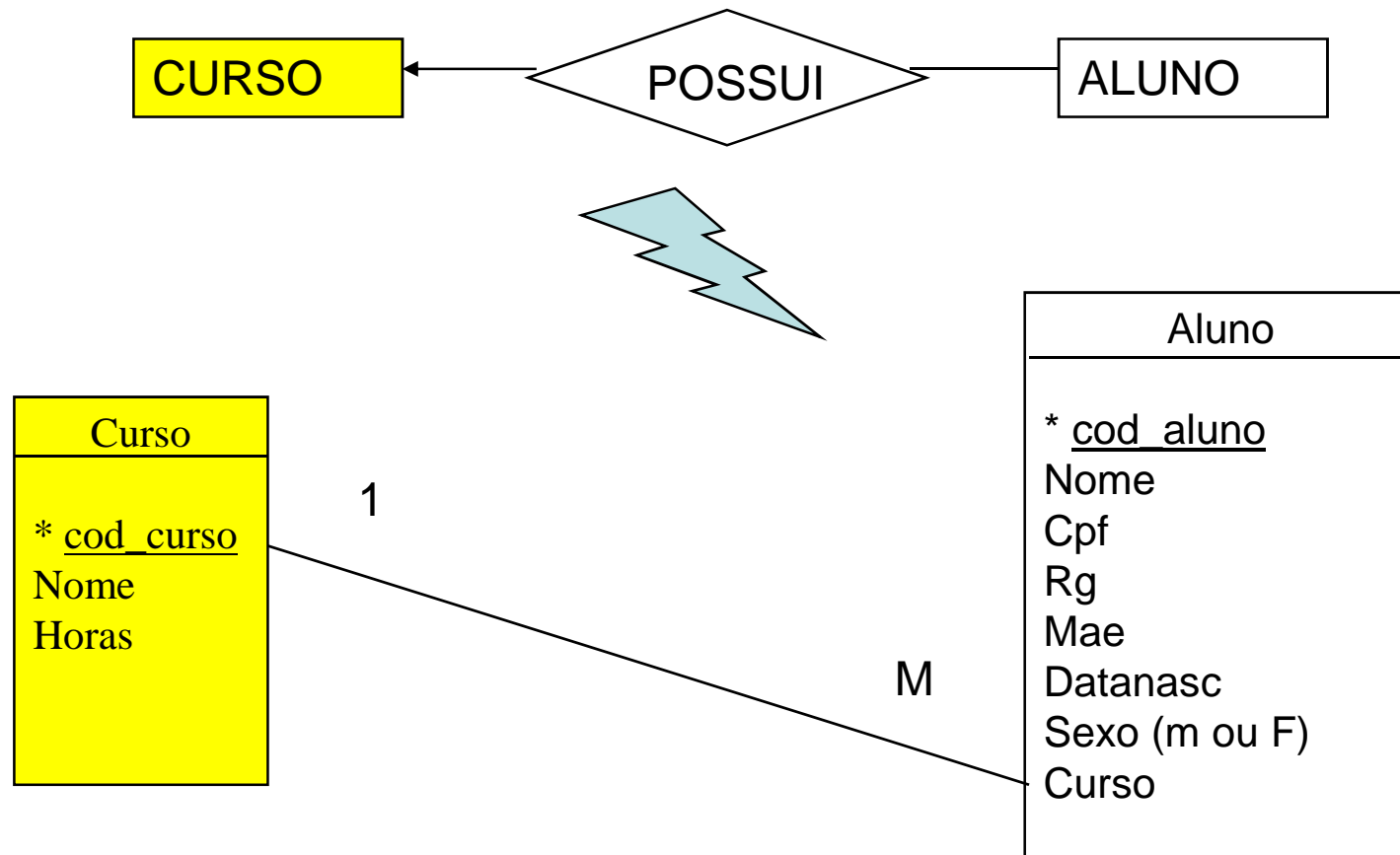
- Sintaxe:
 - **Create or replace trigger** **nome_trigger** **** cria ou atualiza o código de um objeto trigger
 - **Before/after** **** antes do evento ou depois do evento, só um deles pode ser utilizado
 - **insert or update or delete** **** todos os eventos podem ser usados ao mesmo tempo, em dupla ou individualmente
 - **on nome_tabela** **** tabela ligada ao trigger
 - **For each row** **** para cada linha alterada da tabela
 - **Declare** **** declaração de variável
 - Variavel tipo **** nome da var e seu tipo que pode ser varchar2, char, number...
 - **Begin** *** início do programa
 - **If inserting then** *** teste se for um insert. Somente se os demais eventos forem considerados
 - Comandos....
 - **:new.campo** := valor ... **** new é o valor que esta sendo colocado no campo
 - **If updating then** **** testa se for um update. Somente se os demais eventos forem considerados
 - Comandos...
 - **End;** **** final do programas

Linguagem SQL DDL – Gatilho (trigger)

Exemplo 1

Linguagem SQL DDL – Gatilho (trigger)

- Exemplo 1: Seja um curso que possui muitos alunos a ele vinculados.



Linguagem SQL DDL – Gatilho (trigger)

– Exemplo 1 (Oracle):

- create table curso (
 - cod_curso number(5),
 - nome varchar2(50),
 - horas number(5),
 - constraint pkcurso primary key (cod_curso))
- create table aluno (
 - cod_aluno number(5), nome varchar2(50), dt_nasc date,
 - mae varchar2(50), sexo char(1), curso number(5),
 - constraint pkaluno primary key (cod_aluno),
 - constraint ukaluno unique(nome, dt_nasc, mae),
 - constraint sexo check (sexo in('m','f')),
 - constraint fkcursoAluno foreign key (curso) references curso(cod_curso))

Linguagem SQL DDL – Gatilho (trigger)

- Exemplo 1:
 - Create or replace trigger tgCurso -- crie ou atualiza o trigger tgCurso
 - Before insert or update on Curso -- antes de inserir ou atualizar a tabela Curso
 - For each row --para cada linha inserida ou modificada
 - Declare – declare variável
 - Novo_cod number(5); -- variável novo_cód do tipo number
 - Begin -- comece o programa
 - If inserting then -- Se estiver ocorrendo um insert faça os comandos abaixo
 - Select max(cod_curso) + 1 into novo_cod from curso; -- gere o próximo e coloque na var (novo_cod).
 - :new.Cod_curso := novo_cod; -- coloca novo valor no campo cód_curso --coloca o novo valor da seqüência no campo cód_curso que é a chave primária.
 - :new.nome := (upper(:new.nome)); -- transforma o nome em maiúsculo
 -
 - Elsif updating then
 - :new.nome := (upper(:new.nome)); -- transforma o nome em maiúsculo
 - End if;
 - End; -- finaliza o programa

Linguagem SQL DDL – Gatilho (trigger)

- Exemplo 1... Exercício:
 - Crie este mesmo tipo de trigger para a tabela Pessoa

Linguagem SQL DDL – Sequencia (sequence)

- Conceitos: Cria um gerador de seqüência numérica.
- Sintaxe:
 - **CREATE SEQUENCE** nome-da sequencia
START WITH valor-inicio INCREMENT BY
valor-do-incremento;

Linguagem SQL DDL – Sequencia (sequence)

- Exemplo:
 - Create sequence sqCodCurso start with 1
increment by 1;
 - Retorna:
 - select sqCodCurso.nextval into vCod from
dual;

Linguagem SQL DDL – Gatilho (trigger)

Exemplo 2

Linguagem SQL DDL – Gatilho (trigger)

- Exemplo 2:
 - Create or replace trigger tgCurso -- crie ou atualiza o trigger tgCurso
 - Before insert or update on Curso -- antes de inserir ou atualizar a tabela Curso
 - For each row --para cada linha inserida ou modificada
 - Declare – declare variável
 - Novo_cod number(5); -- variável novo_cód do tipo number
 - Begin -- comece o programa
 - If inserting then -- Se estiver ocorrendo um insert faça os comandos abaixo
 - Select sqCodAluno.nextval into novo_cod from dual; -- chama sequence (sqCurso) selecione um novo valor (nextval) na seqüência e coloque na var (novo_cod).
 - :new.Cod_curso := novo_cod; -- coloca novo valor no campo cód_curso --coloca o novo valor da seqüência no campo cód_curso que é a chave primária.
 - :new.nome := (upper(:new.nome)); -- transforma o nome em maiúsculo
 -
 - Elsif updating then
 - :new.nome := (upper(:new.nome)); -- transforma o nome em maiúsculo
 - End if;
 - End; -- finaliza o programa

Linguagem SQL DDL – Gatilho (trigger)

- BEFORE (antes) Se o gatilho disparar antes do evento, o gatilho pode evitar a operação para a tupla atual, ou modificar a tupla sendo inserida.
- AFTER (depois) Se o gatilho disparar após o evento todas as modificações, incluindo a última inserção, atualização ou exclusão, são “visíveis” para o gatilho.
- Privilégios: Para criar um gatilho em uma tabela, o usuário deve possuir o privilégio TRIGGER na tabela.

Linguagem SQL DDL – Gatilho (trigger)

- NEW/ OLD
 - O **NEW** é válido nas regras
 - ON **INSERT** para se referir à nova linha sendo inserida
 - ON **UPDATE** para se referir à nova linha sendo atualizada.

Linguagem SQL DDL – Gatilho (trigger)

- NEW/ OLD
 - O NEW
 - Ele identifica a variável temporária que guarda o valor do campo da tabela no momento que o trigger está sendo executado.
 - Exemplo:
 - `:new.cod_curso` ... Identifica o valor `cod_curso` novo que vai entrar na tabela `curso`.
 - `:new.nome := upper(:new.nome)` ... Transforma para maiúsculo o valor `nome` novo que vai entrar na tabela `curso`.

Linguagem SQL DDL – Gatilho (trigger)

- NEW/ OLD
 - O OLD é válido nas regras
 - ON **UPDATE** para se referir à linha existente sendo atualizada.
 - ON **DELETE** para se referir à linha existente sendo excluída.

Linguagem SQL DDL – Gatilho (trigger)

- NEW/ OLD
 - O OLD
- Ele identifica a variável temporária que guarda o valor do campo da tabela, que está na tabela no momento que o trigger está sendo executado.
- Exemplo:
 - If (:new.nome = :old.nome) ... Compara o valor nome novo que vai entrar na tabela curso com o valor nome que já existe na tabela curso.