**Documento dei vincoli**

**“Statistiche Campionato di calcio”**

CREATE

DATABASE

TRIGGER

OR REPLACE

UPDATE

SELECT

IS NOT NULL

DELETE

COUNT (\*) INTO

FROM

RAISE\_APPLICATION\_ERROR

WHERE

FOR EACH ROW

DECLARE

BEGIN

IS NULL

AFTER INSERT

BEFORE UPDATE

**Realizzato da:**

* *Vincenzo Manfredi (N86001560);*
* *Andrea Francesco Scalella (N86001625).*

**Introduzione**

Essendo i trigger una procedura PL/SQL (Oracle), dunque associata ad una tabella e automaticamente richiamata dal DBMS (Database Management System) quando una certa modifica (o evento) avviene all’interno della tabella (operazioni di insert, update, delete), sono stati utilizzati per specificare e mantenere vincoli di integrità anche più complessi.

Per la realizzazione dei vincoli desiderati è stata aggiunta la clausola ‘for each row’ (trigger a livello di riga), differente dal trigger a livello di istruzione. Un trigger a livello di riga viene eseguito una volta per ogni riga dopo (o prima) l'evento che lo ha scatenato, al contrario, un trigger a livello di istruzione viene eseguito una volta dopo (o prima), indipendentemente da quante righe sono state interessate, dell'evento. Alcune caratteristiche importanti che riguardano i trigger di riga, che non sono fornite con i trigger di istruzione, sono le clausole ‘:new.<colonna>’ e ‘:old.<colonna>’, fondamentali per poter accedere ai valori degli attributi di una tupla prima e dopo la modifica . Per un update trigger sono entrambi disponibili, per un insert trigger solamente ‘:new.<colonna>’ mentre per un delete trigger solo ‘:old.<colonna>’.

**La gestione del database è affidata ai seguenti trigger:**

1. Il trigger **AGGIORNA\_CLASSIFICA**, viene eseguito in seguito ad un inserimento nella tabella *partita* effettuando un controllo sul risultato finale. Dopo aver analizzato i gol effettuati dalla squadra di casa e quelli effettuati dalla squadra ospite, aggiorna opportunamente i dati in *classifica* relativi alle due squadre ( punti, partite giocate, partite vinte, partite pareggiate, partite perse, gol fatti, gol subiti, differenza reti ).

**create or replace TRIGGER "AGGIORNA\_CLASSIFICA"**

after insert on partita

for each row

begin

if(:new.gol\_casa>:new.gol\_ospite) then

update classifica set punti=punti+3,partite\_giocate=partite\_giocate+1,partite\_vinte=partite\_vinte+1,gol\_fatti=gol\_fatti+:new.gol\_casa,gol\_subiti=gol\_subiti+:new.gol\_ospite,differenza\_reti=differenza\_reti+(:new.gol\_casa-:new.gol\_ospite) where (id\_competizione=:new.id\_competizione and nome\_squadra=:new.squadra\_casa);

update classifica set partite\_giocate=partite\_giocate+1,partite\_perse=partite\_perse+1,gol\_fatti=gol\_fatti+:new.gol\_ospite,gol\_subiti=gol\_subiti+:new.gol\_casa,differenza\_reti=differenza\_reti+(:new.gol\_ospite-:new.gol\_casa) where (id\_competizione=:new.id\_competizione and nome\_squadra=:new.squadra\_ospite);

end if;

if (:new.gol\_casa<:new.gol\_ospite) then

update classifica set partite\_giocate=partite\_giocate+1,partite\_perse=partite\_perse+1,gol\_fatti=gol\_fatti+:new.gol\_casa,gol\_subiti=gol\_subiti+:new.gol\_ospite,differenza\_reti=differenza\_reti+(:new.gol\_casa-:new.gol\_ospite) where (id\_competizione=:new.id\_competizione and nome\_squadra=:new.squadra\_casa);

update classifica set punti=punti+3,partite\_giocate=partite\_giocate+1,partite\_vinte=partite\_vinte+1,gol\_fatti=gol\_fatti+:new.gol\_ospite,gol\_subiti=gol\_subiti+:new.gol\_casa,differenza\_reti=differenza\_reti+(:new.gol\_ospite-:new.gol\_casa) where (id\_competizione=:new.id\_competizione and nome\_squadra=:new.squadra\_ospite);

end if;

if (:new.gol\_casa=:new.gol\_ospite) then

update classifica set punti=punti+1,partite\_giocate=partite\_giocate+1,partite\_pareggiate=partite\_pareggiate+1,gol\_fatti=gol\_fatti+:new.gol\_casa,gol\_subiti=gol\_subiti+:new.gol\_ospite where (id\_competizione=:new.id\_competizione and nome\_squadra=:new.squadra\_casa);

update classifica set punti=punti+1,partite\_giocate=partite\_giocate+1,partite\_pareggiate=partite\_pareggiate+1,gol\_fatti=gol\_fatti+:new.gol\_ospite,gol\_subiti=gol\_subiti+:new.gol\_casa where (id\_competizione=:new.id\_competizione and nome\_squadra=:new.squadra\_ospite);

end if;

end;

1. Il trigger **AGGIORNA\_GOL** offre la possibilità, aggiornando il tipo dell’evento relativo ad una determinata partita, di avere aggiornamenti istantanei anche sulla tabella *partita*, in particolare sul record interessato.

Se ad esempio aggiornassimo un evento del tipo=’ammonizione’ in tipo=’gol’, relativo ad un calciatore, questo comporterebbe l’aggiunta di un gol alla squadra di appartenenza del calciatore in riferimento alla partita in questione. Se l’aggiornamento cambia il tipo dell’evento da ‘gol’ a ( ‘ammoizione’ o ‘espulsione’ o ’sostituzione’ o ‘infortunio’) alla squadra di appartenenza del calciatore sarà sottratto un gol, sempre in riferimento alla partita in questione.

**create or replace TRIGGER "AGGIORNA\_GOL"**

after update on evento

for each row

declare

ck integer:=0;

sqc partita.squadra\_casa%type;

sqo partita.squadra\_ospite%type;

n\_sq contratto.nome\_squadra%type;

ndata partita.data%type;

begin

if(:new.tipo='gol' or :old.tipo='gol') then

select s.squadra\_casa into sqc from partita s where s.id\_partita=:new.id\_partita;

select s1.squadra\_ospite into sqo from partita s1 where s1.id\_partita=:new.id\_partita;

select p.data into ndata from partita p where p.id\_partita=:new.id\_partita;

select c.nome\_squadra into n\_sq from contratto c where (c.id\_persona=:new.id\_persona and (ndata between c.inizio\_contratto and c.fine\_contratto));

if(:new.tipo<>'gol' and :old.tipo='gol') then

if(n\_sq=sqc) then

update partita set gol\_casa=gol\_casa-1 where id\_partita=:new.id\_partita;

end if;

if (n\_sq = sqo) then

update partita set gol\_ospite=gol\_ospite-1 where id\_partita=:new.id\_partita;

end if;

end if;

if (:new.tipo='gol' and :old.tipo<>'gol') then

if(n\_sq = sqc)then

update partita set gol\_casa=gol\_casa+1 where id\_partita=:new.id\_partita;

end if;

if (n\_sq = sqo) then

update partita set gol\_ospite=gol\_ospite+1 where id\_partita=:new.id\_partita;

end if;

end if;

end if;

end;

1. Il trigger **AGGIORNA\_PARTITA**, simile al precedente (AGGIORNA\_GOL), effettua un aggiornamento sull’esito finale della *partita* in seguito all’inserimento di determinati valori sulla tabella *evento*.

**create or replace TRIGGER "AGGIORNA\_PARTITA"**

after insert on evento

for each row

declare

n\_sq contratto.nome\_squadra%type;

data partita.data%type;

sqc partita.squadra\_casa%type;

sqo partita.squadra\_ospite%type;

tipo persona.tipo%type;

begin

select p.tipo into tipo from persona p where p.id\_persona=:new.id\_persona;

if tipo<>'arbitro' then

select p.data into data from partita p where p.id\_partita=:new.id\_partita;

select c.nome\_squadra into n\_sq from contratto c where (c.id\_persona=:new.id\_persona and (data between c.inizio\_contratto and c.fine\_contratto));

select s.squadra\_casa into sqc from partita s where s.id\_partita=:new.id\_partita;

select s1.squadra\_ospite into sqo from partita s1 where s1.id\_partita=:new.id\_partita;

if(:new.tipo='gol') then

if(n\_sq = sqc)then

update partita set gol\_casa=gol\_casa+1 where id\_partita=:new.id\_partita;

end if;

if (n\_sq = sqo) then

update partita set gol\_ospite=gol\_ospite+1 where id\_partita=:new.id\_partita;

end if;

end if;

if (:new.tipo='entrata') then

update persona set presenze=presenze+1 where id\_persona=:new.id\_persona;

end if;

else

if (:new.tipo='entrata') then

update persona set presenze=presenze+1 where id\_persona=:new.id\_persona;

end if;

end if;

end;

1. Il trigger **AGGIORNA\_VINCOLO\_CLASSIFICA**, viene attivato in seguito ad un aggiornamento sulla tabella *partita*. I dati aggiornati saranno utilizzati per garantire la coerenza con le informazioni relative alle squadre in questione nella *classifica* di riferimento, dunque per permettere un aggiornamento di essa. Nel caso in cui l’aggiornamento relativo alla partita faccia riferimento ad una data diversa dalla precedente, verrà inserito un record anche nella tabella *rinvio.*

Per tenere traccia del vecchio e del nuovo risultato relativo ad una partita al fine di aggiornare correttamente la tabella *classifica,* è stata implementata la seguente funzione *RESULT* (gol1 partita.gol\_casa%type, gol2 partita.gol\_ospite%type)che ritorna un intero a seconda del risultato prima della modifica*:*

* **create or replace FUNCTION "RESULT"** (gol1 partita.gol\_casa%type, gol2 partita.gol\_ospite%type) return integer

is

ck integer:=0;

begin

if(gol1>gol2) then

ck:=1;

end if;

if(gol2>gol1) then

ck:=2;

end if;

return ck;

end;

**create or replace TRIGGER "AGGIORNA\_VINCOLO\_CLASSIFICA"**

after update on partita

for each row

declare

ck1 integer:=0;

ck2 integer:=0;

begin

if(:new.id\_competizione=:old.id\_competizione and :new.id\_partita=:old.id\_partita and :new.squadra\_casa=:old.squadra\_casa and :new.squadra\_ospite=:old.squadra\_ospite and (:new.gol\_casa<>:old.gol\_casa or :new.gol\_ospite<>:old.gol\_ospite)) then

ck1:=RESULT(:old.gol\_casa,:old.gol\_ospite);

ck2:=RESULT(:new.gol\_casa,:new.gol\_ospite);

if((ck1=0 and ck2=0) or (ck1=1 and ck2=1) or (ck1=2 and ck2=2)) then

update classifica set gol\_fatti=gol\_fatti+(:new.gol\_casa-:old.gol\_casa),gol\_subiti=gol\_subiti+(:new.gol\_ospite-:old.gol\_ospite),differenza\_reti=differenza\_reti+((:new.gol\_casa-:old.gol\_casa)-(:new.gol\_ospite-:old.gol\_ospite)) where (id\_competizione=:new.id\_competizione and nome\_squadra=:new.squadra\_casa);

update classifica set gol\_fatti=gol\_fatti+(:new.gol\_ospite-:old.gol\_ospite),gol\_subiti=gol\_subiti+(:new.gol\_casa-:old.gol\_casa),differenza\_reti=differenza\_reti+((:new.gol\_ospite-:old.gol\_ospite)-(:new.gol\_casa-:old.gol\_casa)) where (id\_competizione=:new.id\_competizione and nome\_squadra=:new.squadra\_ospite);

elsif (ck1=0 and ck2=1) then

update classifica set punti=punti+2,partite\_pareggiate=partite\_pareggiate-1, partite\_vinte=partite\_vinte+1,gol\_fatti=gol\_fatti+(:new.gol\_casa-:old.gol\_casa),gol\_subiti=gol\_subiti+(:new.gol\_ospite-:old.gol\_ospite),differenza\_reti=differenza\_reti+((:new.gol\_casa-:old.gol\_casa)-(:new.gol\_ospite-:old.gol\_ospite)) where (id\_competizione=:new.id\_competizione and nome\_squadra=:new.squadra\_casa);

update classifica set punti=punti-1,partite\_pareggiate=partite\_pareggiate-1,partite\_perse=partite\_perse+1,gol\_fatti=gol\_fatti+(:new.gol\_ospite-:old.gol\_ospite),gol\_subiti=gol\_subiti+(:new.gol\_casa-:old.gol\_casa),differenza\_reti=differenza\_reti+((:new.gol\_ospite-:old.gol\_ospite)-(:new.gol\_casa-:old.gol\_casa)) where (id\_competizione=:new.id\_competizione and nome\_squadra=:new.squadra\_ospite);

elsif (ck1=0 and ck2=2) then

update classifica set punti=punti-1,partite\_pareggiate=partite\_pareggiate-1,partite\_perse=partite\_perse+1,gol\_fatti=gol\_fatti+(:new.gol\_casa-:old.gol\_casa),gol\_subiti=gol\_subiti+(:new.gol\_ospite-:old.gol\_ospite),differenza\_reti=differenza\_reti+((:new.gol\_casa-:old.gol\_casa)-(:new.gol\_ospite-:old.gol\_ospite)) where (id\_competizione=:new.id\_competizione and nome\_squadra=:new.squadra\_casa);

update classifica set punti=punti+2,partite\_pareggiate=partite\_pareggiate-1, partite\_vinte=partite\_vinte+1,gol\_fatti=gol\_fatti+(:new.gol\_ospite-:old.gol\_ospite),gol\_subiti=gol\_subiti+(:new.gol\_casa-:old.gol\_casa),differenza\_reti=differenza\_reti+((:new.gol\_ospite-:old.gol\_ospite)-(:new.gol\_casa-:old.gol\_casa)) where (id\_competizione=:new.id\_competizione and nome\_squadra=:new.squadra\_ospite);

elsif (ck1=1 and ck2=0) then

update classifica set punti=punti-2,partite\_pareggiate=partite\_pareggiate+1,partite\_vinte=partite\_vinte-1,gol\_fatti=gol\_fatti+(:new.gol\_casa-:old.gol\_casa),gol\_subiti=gol\_subiti+(:new.gol\_ospite-:old.gol\_ospite),differenza\_reti=differenza\_reti+((:new.gol\_casa-:old.gol\_casa)-(:new.gol\_ospite-:old.gol\_ospite)) where (id\_competizione=:new.id\_competizione and nome\_squadra=:new.squadra\_casa);

update classifica set punti=punti+1,partite\_pareggiate=partite\_pareggiate+1,partite\_perse=partite\_perse-1,gol\_fatti=gol\_fatti+(:new.gol\_ospite-:old.gol\_ospite),gol\_subiti=gol\_subiti+(:new.gol\_casa-:old.gol\_casa),differenza\_reti=differenza\_reti+((:new.gol\_ospite-:old.gol\_ospite)-(:new.gol\_casa-:old.gol\_casa)) where (id\_competizione=:new.id\_competizione and nome\_squadra=:new.squadra\_ospite);

elsif (ck1=1 and ck2=2) then

update classifica set punti=punti-3,partite\_perse=partite\_perse+1,partite\_vinte=partite\_vinte-1,gol\_fatti=gol\_fatti+(:new.gol\_casa-:old.gol\_casa),gol\_subiti=gol\_subiti+(:new.gol\_ospite-:old.gol\_ospite),differenza\_reti=differenza\_reti+((:new.gol\_casa-:old.gol\_casa)-(:new.gol\_ospite-:old.gol\_ospite)) where (id\_competizione=:new.id\_competizione and nome\_squadra=:new.squadra\_casa);

update classifica set punti=punti+3,partite\_perse=partite\_perse-1,partite\_vinte=partite\_vinte+1,gol\_fatti=gol\_fatti+(:new.gol\_ospite-:old.gol\_ospite),gol\_subiti=gol\_subiti+(:new.gol\_casa-:old.gol\_casa),differenza\_reti=differenza\_reti+((:new.gol\_ospite-:old.gol\_ospite)-(:new.gol\_casa-:old.gol\_casa)) where (id\_competizione=:new.id\_competizione and nome\_squadra=:new.squadra\_ospite);

elsif (ck1=2 and ck2=0) then

update classifica set punti=punti+1,partite\_pareggiate=partite\_pareggiate+1,partite\_perse=partite\_perse-1,gol\_fatti=gol\_fatti+(:new.gol\_casa-:old.gol\_casa),gol\_subiti=gol\_subiti+(:new.gol\_ospite-:old.gol\_ospite),differenza\_reti=differenza\_reti+((:new.gol\_casa-:old.gol\_casa)-(:new.gol\_ospite-:old.gol\_ospite)) where (id\_competizione=:new.id\_competizione and nome\_squadra=:new.squadra\_casa);

update classifica set punti=punti-2,partite\_pareggiate=partite\_pareggiate+1,partite\_vinte=partite\_vinte-1,gol\_fatti=gol\_fatti+(:new.gol\_ospite-:old.gol\_ospite),gol\_subiti=gol\_subiti+(:new.gol\_casa-:old.gol\_casa),differenza\_reti=differenza\_reti+((:new.gol\_ospite-:old.gol\_ospite)-(:new.gol\_casa-:old.gol\_casa)) where (id\_competizione=:new.id\_competizione and nome\_squadra=:new.squadra\_ospite);

elsif (ck1=2 and ck2=1) then

update classifica set punti=punti+3,partite\_perse=partite\_perse-1,partite\_vinte=partite\_vinte+1,gol\_fatti=gol\_fatti+(:new.gol\_casa-:old.gol\_casa),gol\_subiti=gol\_subiti+(:new.gol\_ospite-:old.gol\_ospite),differenza\_reti=differenza\_reti+((:new.gol\_casa-:old.gol\_casa)-(:new.gol\_ospite-:old.gol\_ospite)) where (id\_competizione=:new.id\_competizione and nome\_squadra=:new.squadra\_casa);

update classifica set punti=punti-3,partite\_perse=partite\_perse+1,partite\_vinte=partite\_vinte-1,gol\_fatti=gol\_fatti+(:new.gol\_ospite-:old.gol\_ospite),gol\_subiti=gol\_subiti+(:new.gol\_casa-:old.gol\_casa),differenza\_reti=differenza\_reti+((:new.gol\_ospite-:old.gol\_ospite)-(:new.gol\_casa-:old.gol\_casa)) where (id\_competizione=:new.id\_competizione and nome\_squadra=:new.squadra\_ospite);

end if;

elsif (:new.data<>:old.data and :new.stadio=:old.stadio) then

insert into rinvio values (:old.id\_partita,'DA SPECIFICARE',:old.data,null);

elsif (:new.data<>:old.data and :new.stadio<>:old.stadio) then

insert into rinvio values (:old.id\_partita,'DA SPECIFICARE',:old.data,:old.stadio);

end if;

end;

1. Il trigger **VINCOLO\_AGGIORNAMENTO\_PARTITA**, eseguito prima dell’aggiornamento sulla tabella *partita,* fa si che alcune condizioni non vengano violate.

L’aggiornamento, possibile solo nel caso in cui si volessero modificare le colonne relative ai ‘GOL\_CASA’, ‘GOL\_OSPITE’, ‘DATA’, solleva un errore in linea con la reale impostazione dei campionati calcistici.

**create or replace TRIGGER "VINCOLO\_AGGIORNAMENTO\_PARTITA"**

before update on partita

for each row

declare

begin

if (:new.id\_competizione<>:old.id\_competizione or :new.giornata<>:old.giornata) then

raise\_application\_error(-20026,'Non è possibile trasferire partite tra competizioni o giornate di campionato');

elsif (:new.squadra\_casa<>:old.squadra\_casa or :new.squadra\_ospite<>:old.squadra\_ospite) then

raise\_application\_error(-20026,'Non è possibile cambiare le squadre partecipanti ad una partita');

elsif (:new.id\_partita<>:old.id\_partita) then

raise\_application\_error(-20037,'Non è possibile cambiare l''id\_della partita per motivi d''integrità');

end if;

end;

1. Il trigger **VINCOLO\_COMPETIZIONE** viene attivato prima di un inserimento nella tabella *competizione,* al fine di evitare l’esistenza di campionati/tornei con lo stesso nome, nello stesso anno.

**create or replace TRIGGER "VINCOLO\_COMPETIZIONE"**

before insert on competizione

for each row

declare

ck integer:=0;

begin

select count(\*) into ck from competizione c where c.nome=:new.nome and c.anno=:new.anno;

if (ck>0) then

raise\_application\_error(-20038,'Non è possibile che due competizioni nello stesso anno abbiano lo stesso nome');

end if;

end;

1. Il trigger **VINCOLO\_CONTRATTO**, eseguito prima dell’inserimento nella tabella *contratto,* garantisce il mantenimento della stabilità contrattuale tra professionisti (allenatore, giocatore) e società, oltre ad evitare di assegnare a più giocatori, nello stesso periodo, maglie con lo stesso numero.

**create or replace TRIGGER "VINCOLO\_CONTRATTO"**

before insert on contratto

for each row

declare

tipo persona.tipo%type;

nazione\_p persona.stato%type;

nazione\_s squadra.nazione%type;

city squadra.città%type;

ck integer:=0;

begin

select p.tipo into tipo from persona p where p.id\_persona=:new.id\_persona;

if(tipo='arbitro' or tipo='pensionato') then

raise\_application\_error(-20061,'Non è possibile assegnare un contratto ad un arbitro o ad un pensionato');

end if;

if (tipo='allenatore') then

if(:new.numero\_maglia is not null) then

raise\_application\_error(-20060,'Non è possibile assegnare un numero di maglia ad un allenatore');

end if;

select count(\*) into ck from (contratto c join persona p on c.id\_persona=p.id\_persona) where p.tipo='allenatore' and (c.nome\_squadra=:new.nome\_squadra) and ((:new.inizio\_contratto between c.inizio\_contratto and c.fine\_contratto) or (:new.fine\_contratto between c.inizio\_contratto and c.fine\_contratto) or (:new.inizio\_contratto<=c.inizio\_contratto and :new.fine\_contratto>=c.fine\_contratto));

if (ck>0) then

raise\_application\_error(-20014,'Una squadra non può avere due allenatori contemporaneamente');

end if;

end if;

if(:new.numero\_maglia is not null) then

select count(\*) into ck from contratto c where (c.nome\_squadra=:new.nome\_squadra) and ((:new.inizio\_contratto between c.inizio\_contratto and c.fine\_contratto) or (:new.fine\_contratto between c.inizio\_contratto and c.fine\_contratto) or (:new.inizio\_contratto<=c.inizio\_contratto and :new.fine\_contratto>=c.fine\_contratto)) and c.numero\_maglia=:new.numero\_maglia;

if (ck>0) then

raise\_application\_error(-20013,'Non è possibile assegnare lo stesso numero di maglia a due giocatori che giocano nella stessa squadra nello stesso periodo');

end if;

end if;

select s.città into city from squadra s where s.nome=:new.nome\_squadra;

if (city is not null) then

select count(\*) into ck from (contratto c join squadra s on c.nome\_squadra=s.nome) where c.id\_persona=:new.id\_persona and (s.città is not null) and ((:new.inizio\_contratto between c.inizio\_contratto and c.fine\_contratto) or (:new.fine\_contratto between c.inizio\_contratto and c.fine\_contratto)or (:new.inizio\_contratto<=c.inizio\_contratto and :new.fine\_contratto>=c.fine\_contratto));

if ck>0 then

raise\_application\_error(-20010,'Un giocatore non può stare in due squadre di club contemporaneamente');

end if;

else

select s.nazione into nazione\_s from squadra s where s.nome=:new.nome\_squadra;

select p.stato into nazione\_p from persona p where p.id\_persona=:new.id\_persona;

if (nazione\_p<>nazione\_s) then

raise\_application\_error(-20050,'Un giocatore non può giocare in una nazionale diversa dal suo stato di appartenenza!');

end if;

select count(\*) into ck from (contratto c join squadra s on c.nome\_squadra=s.nome) where c.id\_persona=:new.id\_persona and (s.città is null) and ((:new.inizio\_contratto between c.inizio\_contratto and c.fine\_contratto) or (:new.fine\_contratto between c.inizio\_contratto and c.fine\_contratto)or (:new.inizio\_contratto<=c.inizio\_contratto and :new.fine\_contratto>=c.fine\_contratto));

if ck>0 then

raise\_application\_error(-20010,'Un giocatore non può avere più contratti con la nazionale contemporaneamente');

end if;

end if;

if (:new.inizio\_contratto>:new.fine\_contratto) then

raise\_application\_error(-20050,'Fine contratto antecedente ad inizio contratto');

end if;

end;

1. Il trigger **VINCOLO\_EVENTO**, pensato per simulare e gestire in maniera efficiente tutto ciò che potrebbe accadere durante la disputa di una gara, viene attivato prima dell’inserimento nella tabella *evento* controllando perfino che una persona sia entrata in campo prima di poterle associare un determinato evento*.*

* La seguente procedura (*INFORTU)* è stata implementata per automatizzare la sostituzione di un calciatore con uno tra quelli presenti in rosa in seguito ad un infortunio.

Tramite l’utilizzo di un cursore vengono selezionati i calciatori in rosa presenti nella data di disputa della partita in questione e viene tenuta traccia della loro disponibilità in termini di quantità nella variabile *ck.* Solo successivamente si andrà ad iterare sul cursore fin quando il valore di ck sarà uguale a quello restituito dalle funzioni *round e dbms\_random.value* memorizzato in *rand,* che decreterà il calciatore che dovrà fare ingresso in campo.

**create or replace PROCEDURE "INFORTU" (pers persona.id\_persona%type,sq squadra.nome%type,ruolo evento.ruolo%type, tempo evento.minuto%type,par partita.id\_partita%type, data partita.data%type,capitano evento.capitano%type)**

is

rand integer:=0;

ck integer:=0;

cursor gio is ((select c.id\_persona from contratto c where (data between c.inizio\_contratto and c.fine\_contratto) and c.nome\_squadra=sq) minus (select e.id\_persona from evento e where e.id\_partita=par));

begin

select count(\*) into ck from ((select c.id\_persona from contratto c where (data between c.inizio\_contratto and c.fine\_contratto) and c.nome\_squadra=sq) minus (select e.id\_persona from evento e where e.id\_partita=par));

rand:=round(dbms\_random.value(1,ck));

ck:=0;

insert into evento values (tempo,ruolo,'sostituzione',capitano,pers,par);

for i in gio

loop

if (ck=rand) then

insert into evento values (tempo,ruolo,'entrata',NULL,i.id\_persona,par);

end if;

ck:=ck+1;

end loop;

end;

**create or replace TRIGGER "VINCOLO\_EVENTO"**

before insert on evento

for each row

declare

ck integer:=0;

data partita.data%type;

tipo persona.tipo%type;

sq squadra.nome%type;

sq1 squadra.nome%type;

sq2 squadra.nome%type;

cursor cpt is select e.id\_persona,e.capitano from evento e where (e.id\_partita=:new.id\_partita);

begin

select p.tipo into tipo from persona p where p.id\_persona=:new.id\_persona;

select p.data into data from partita p where p.id\_partita=:new.id\_partita;

if (tipo<>'arbitro') then

select c.nome\_squadra into sq from contratto c where c.id\_persona=:new.id\_persona and (data between c.inizio\_contratto and c.fine\_contratto);

end if;

select count(\*) into ck from evento e where (e.id\_persona=:new.id\_persona and e.id\_partita=:new.id\_partita and e.tipo='espulsione');

if (ck>0 and :new.tipo='ammonizione' and tipo='giocatore') then

RAISE\_APPLICATION\_ERROR (-20027, 'Una giocatore non può essere espulso più di una volta in una partita');

ck:=0;

end if;

if (:new.tipo='infortunio' and tipo='giocatore') then

INFORTU(:new.id\_persona,sq,:new.ruolo,:new.minuto,:new.id\_partita,data,:new.capitano);

end if;

if :new.tipo='ammonizione' and tipo='giocatore' then

select count(\*) into ck from evento e where (e.id\_persona=:new.id\_persona and e.id\_partita=:new.id\_partita and e.tipo='ammonizione');

if ck=1 then

insert into evento values (:new.minuto,:new.ruolo,'espulsione',:new.capitano,:new.id\_persona,:new.id\_partita);

elsif ck>1 then

RAISE\_APPLICATION\_ERROR (-20007, 'Una giocatore non può essere ammonito più di due volte in una partita');

end if;

end if;

if (tipo='giocatore' and :new.capitano is not null) then

for i in cpt

loop

select c.nome\_squadra into sq2 from contratto c where c.id\_persona=i.id\_persona and (data between c.inizio\_contratto and c.fine\_contratto);

if (sq=sq2 and i.id\_persona<>:new.id\_persona and i.capitano is not null) then

raise\_application\_error(-20012,'Non possono essere assegnati ad una squadra più capitani per una stessa partita');

end if;

end loop;

end if;

ck:=0;

select count(\*) into ck from evento e where (e.id\_persona=:new.id\_persona and e.id\_partita=:new.id\_partita and tipo='entrata');

if (ck>0 and :new.tipo='entrata') then

RAISE\_APPLICATION\_ERROR (-20002, 'Una persona non può entrare più volte in campo');

elsif (ck=0 and :new.tipo<>'entrata' and :new.tipo<>'ammonizione' and :new.tipo<>'espulsione') then

RAISE\_APPLICATION\_ERROR (-20035, 'Una persona deve entrare in campo prima di potergli assegnare un evento diverso da un''espulsione o ammonizione');

end if;

ck:=0;

select count(\*) into ck from evento e where (e.id\_persona=:new.id\_persona and e.id\_partita=:new.id\_partita and tipo='sostituzione');

if (ck>0 and :new.tipo='sostituzione') then

RAISE\_APPLICATION\_ERROR (-20004, 'Una persona non può essere sostituita più volte in una partita');

ck:=0;

end if;

select count(\*) into ck from evento e where (e.id\_persona=:new.id\_persona and e.id\_partita=:new.id\_partita and tipo='infortunio');

if (ck>0 and :new.tipo='infortunio') then

RAISE\_APPLICATION\_ERROR (-20006, 'Una persona non può infortunarsi più volte in una partita');

ck:=0;

end if;

select p.squadra\_casa into sq1 from partita p where p.id\_partita=:new.id\_partita;

select p.squadra\_ospite into sq2 from partita p where p.id\_partita=:new.id\_partita;

if (sq<>sq1 and sq<>sq2 and tipo<>'arbitro') then

RAISE\_APPLICATION\_ERROR (-20000, 'Non può essere assegnato un evento ad un giocatore/allenatore non appartente alle squadre in campo');

end if;

select count(\*) into ck from evento e where (e.id\_persona=:new.id\_persona and e.id\_partita=:new.id\_partita and (tipo='espulsione' or tipo='sostituzione') and e.minuto<:new.minuto);

if (ck>0) then

RAISE\_APPLICATION\_ERROR (-20001, 'Non può essere assegnato un evento ad una persona uscita dal campo');

end if;

if(:new.tipo='espulsione' or :new.tipo='sostituzione' and tipo<>'arbitro') then

select count(\*) into ck from evento e where (e.id\_persona=:new.id\_persona and e.id\_partita=:new.id\_partita) and (:new.minuto<e.minuto);

if(ck>0) then

RAISE\_APPLICATION\_ERROR (-20002, 'Non può essere sostituita/espulsa la data persona dato che ha eventi successivi a tale minuto');

end if;

end if;

if (tipo='allenatore' and :new.tipo<>'espulsione' and :new.tipo<>'entrata') or (tipo='arbitro' and :new.tipo<>'infortunio' and :new.tipo<>'sostituzione' and :new.tipo<>'entrata') then

raise\_application\_error(-20009,'Un allenatore può essere solo espulso ed un arbitro può solo infortunarsi ed essere sostituito');

end if;

if ((tipo='allenatore' or tipo='arbitro') and (:new.capitano is not null or :new.ruolo is not null)) then

raise\_application\_error(-20011,'Ad una persona diversa da un giocatore non si può assegnare un ruolo o la flag capitano');

end if;

if (:new.minuto=0 and :new.tipo<>'entrata') then

raise\_application\_error(-20011,'Al minuto zero è possibile inserire solo il tipo entrata');

end if;

end;

1. Il trigger **VINCOLO\_PARTITA**,innescato prima dell’inserimento dei dati all’interno della tabella *partita,* fa si che i nuovi record siano coerenti con il tipo di competizione scelta. Per coerenza si fa riferimento all’iscrizione delle squadre, partecipanti la gara, alla competizione scelta (presenza della squadra nella classifica di riferimento della competizione), alla data della disputa, ecc. Da sottolineare che questo trigger impedisce anche che due squadre disputino più partite nella stessa data/giornata.

**create or replace TRIGGER "VINCOLO\_PARTITA"**

before insert on partita

for each row

declare

ck integer:=0;

ck1 integer:=0;

tipo competizione.tipo%type;

girone competizione.numero\_gironi%type;

anno competizione.anno%type;

partite\_p classifica.partite\_perse%type;

partite\_p1 classifica.partite\_perse%type;

begin

select c.tipo into tipo from competizione c where c.id\_competizione=:new.id\_competizione;

select c.numero\_gironi into girone from competizione c where c.id\_competizione=:new.id\_competizione;

if (tipo='amichevole' and :new.giornata is not null) then

raise\_application\_error(-20015,'Non è possibile aggiungere le giornate ad una partita di una competizione amichevole');

end if;

if (tipo='campionato' and :new.giornata is null) then

raise\_application\_error(-20015,'E''necessario assegnare il numero di giornata dato che è una partita di campionato');

end if;

if (tipo='torneo' and :new.giornata is not null) then

raise\_application\_error(-20015,'Non è possibile assegnare la giornata a tale partita dato che appartiene ad un torneo');

end if;

select count(\*) into ck from partita p where (p.squadra\_casa=:new.squadra\_casa or p.squadra\_ospite=:new.squadra\_casa or p.squadra\_casa=:new.squadra\_ospite or p.squadra\_ospite=:new.squadra\_ospite) and p.data=:new.data;

if (ck>0) then

RAISE\_APPLICATION\_ERROR (-20005, 'Una squadra non può disputare più partite in una stessa data');

end if;

select count(\*) into ck1 from classifica c where c.id\_competizione=:new.id\_competizione and c.nome\_squadra=:new.squadra\_casa;

select count(\*) into ck from classifica c where c.id\_competizione=:new.id\_competizione and c.nome\_squadra=:new.squadra\_ospite;

if (ck1<>ck) then

raise\_application\_error(-20016,'Non è possibile inserire la partita dato che una delle due squadre non partecipa alla competizione');

end if;

select c.anno into anno from competizione c where c.id\_competizione=:new.id\_competizione;

if(anno<>extract(year from :new.data)) then

raise\_application\_error(-20046,'Non è possibile inserire la partita dato che la data non è in linea con l''anno della competizione');

end if;

if(tipo='torneo' and girone is null) then

select c.partite\_perse into partite\_p from classifica c where c.nome\_squadra=:new.squadra\_casa and c.id\_competizione=:new.id\_competizione;

if(partite\_p is not NULL and partite\_p<>0) then

raise\_application\_error(-20016,'La squadra di casa, avendo già perso una partita, non può disputare la gara');

end if;

select c1.partite\_perse into partite\_p1 from classifica c1 where c1.nome\_squadra=:new.squadra\_ospite and c1.id\_competizione=:new.id\_competizione ;

if(partite\_p1 is not NULL and partite\_p1<>0) then

raise\_application\_error(-20016,'La squadra ospite,avendo già perso una partita, non può disputare la gara');

end if;

end if;

select count(\*) into ck1 from partita p where p.squadra\_casa=:new.squadra\_casa and p.squadra\_ospite=:new.squadra\_ospite and p.id\_competizione=:new.id\_competizione;

if(ck1>0) then

raise\_application\_error(-20017,'La squadra di casa ha già affrontato tale club nella sua città in tale competizione');

end if;

select count(\*) into ck1 from partita p where (p.squadra\_casa=:new.squadra\_casa or p.squadra\_ospite=:new.squadra\_casa) and p.id\_competizione=:new.id\_competizione and p.giornata=:new.giornata;

if ck1>0 then

raise\_application\_error(-20040,'La squadra di casa ha già giocato per questa giornata');

end if;

select count(\*) into ck1 from partita p where (p.squadra\_casa=:new.squadra\_ospite or p.squadra\_ospite=:new.squadra\_ospite) and p.id\_competizione=:new.id\_competizione and p.giornata=:new.giornata;

if ck1>0 then

raise\_application\_error(-20040,'La squadra ospite ha già giocato per questa giornata');

end if;

end;

1. Il trigger **VINCOLO\_AGGIORNAMENTO\_EVENTO,** attivato prima dell’aggiornamento sulla tabella *evento,* fa si che alcune condizioni/regole vengano rispettate nel corso della gara.

E’ importante che una persona, che non sia più in campo, non possa compiere alcuna azione al fine di modificare l’esito della partita così come non deve essere possibile modificare il tipo degli eventi aventi conseguenze compromettenti sulla partita in corso.

**create or replace TRIGGER "VINCOLO\_AGGIORNAMENTO\_EVENTO"**

before update on evento

for each row

declare

begin

if (:new.id\_partita <> :old.id\_partita) then

raise\_application\_error(-20032,'Non è possibile cambiare la partita di appartenenza');

end if;

if(:new.tipo='gol' and (:old.tipo='sostituzione' or :old.tipo='espulsione')) or (:new.tipo='ammonizione' and (:old.tipo='sostituzione' or :old.tipo='espulsione')) or (:new.tipo='infortunio' and (:old.tipo='sostituzione' or :old.tipo='espulsione')) or (:new.tipo='espulsione' and :old.tipo='sostituzione' ) or (:new.tipo='sostituzione' and :old.tipo='espulsione') then

raise\_application\_error(-20033,'Non è possibile assegnare tale evento, la persona non è in campo');

end if;

if (:new.tipo='espulsione' or :new.tipo='sostituzione' or :new.tipo='infortunio') then

raise\_application\_error(-20034,'Non è data la possibilità di aggiornare il tipo di un evento con un''espulsione,infortunio o sostituzione per non compromettere la coerenza della partita. Usare inserimenti e cancellazioni');

end if;

end;

1. Il trigger **VINCOLO\_CLASSIFICA**, innescato prima dell’inserimento all’interno della tabella *classifica*, tiene traccia della corretta collocazione al fine di mantenere una coerenza tra le squadre inserite e la competizione di appartenenza

**create or replace TRIGGER "VINCOLO\_CLASSIFICA"**

before insert on classifica

for each row

declare

nazione competizione.nazione%type;

citta squadra.città%type;

nazione\_sq squadra.nazione%type;

begin

select c.nazione into nazione from competizione c where c.id\_competizione=:new.id\_competizione;

select s.città into citta from squadra s where s.nome=:new.nome\_squadra;

select s.nazione into nazione\_sq from squadra s where s.nome=:new.nome\_squadra;

if(nazione IS NOT NULL AND citta IS NOT NULL and nazione\_sq<>nazione and nazione<>'internazionale' ) then

raise\_application\_error(-20020,'E'' necessario inserire squadre di club appartenenti alla stessa nazione della competizione');

end if;

if(nazione IS NULL AND citta IS NOT NULL) then

raise\_application\_error(-20020,'E'' necessario inserire squadre nazionali');

end if;

if (nazione<>'internazionale' and nazione IS NOT NULL AND citta IS NULL) then

raise\_application\_error(-20020,'Tornei internazionali possono crearsi solo tra squadre di club');

end if;

end;

1. Il trigger **AGGIORNA\_PERSONA**, attivato prima dell’inserimento nella tabella *persona,* consente di trasformare, se non ha un contratto in corso, un giocatore in allenatore, arbitro o pensionato.

\*\*\*\*\*\*chek manuale\*\*\*\*

Con un controllo sul vecchio e nuovo *tipo*, non dà l’opportunità ad un allenatore o arbitro di diventare calciatori. E’ stata utilizzata la funzione *sysdate()* per ottenere la data corrente.

**create or replace TRIGGER "AGGIORNA\_PERSONA"**

before update on persona

for each row

declare

fin contratto.fine\_contratto%type;

ck integer:=0;

begin

if(:old.tipo='pensionato' and :new.tipo<>'pensionato') then

raise\_application\_error(-20067,'Un pensionato non può cambiare tipo');

end if;

if(:old.tipo='allenatore' and :new.tipo='giocatore') then

raise\_application\_error(-20062,'Un allenatore non può diventare calciatore!');

elsif(:old.tipo='arbitro' and :new.tipo='giocatore') then

raise\_application\_error(-20063,'Un arbitro non può diventare calciatore!');

elsif(:old.tipo='allenatore' and :new.tipo='arbitro') then

raise\_application\_error(-20064,'Un allenatore non può diventare arbitro!');

elsif(:old.tipo='arbitro' and :new.tipo='allenatore') then

raise\_application\_error(-20065,'Un arbitro non può diventare allenatore!');

end if;

if(:old.tipo='giocatore' and :new.tipo<>'giocatore') then

select count(\*) into ck from contratto c where c.id\_persona=:new.id\_persona;

if ck>0 then

select max(c.fine\_contratto) into fin from contratto c where c.id\_persona=:old.id\_persona;

if(sysdate() < fin) then

raise\_application\_error(-20066,'Non è possibile cambiare il tipo del giocatore dato che ha un contratto in corso!');

end if;

end if;

end if;

if(:old.tipo='allenatore' and :new.tipo<>'allenatore') then

select count(\*) into ck from contratto c where c.id\_persona=:new.id\_persona;

if ck>0 then

select max(c.fine\_contratto) into fin from contratto c where c.id\_persona=:old.id\_persona;

if(sysdate() < fin) then

raise\_application\_error(-20068,'Non è possibile cambiare il tipo dell''allenatore dato che ha un contratto in corso!');

end if;

end if;

end if;

end;

1. Il trigger **CANCELLA\_PARTITA,** innescato dopo la cancellazione di un record dalla tabella *partita,* modifica in maniera opportuna i dati relativi alle due squadre (*squadra\_casa e squadra\_ospite*) nella tabella *classifica.*

**create or replace TRIGGER "CANCELLA\_PARTITA"**

after delete on partita

for each row

declare

ck1 integer:=0;

begin

ck1:=RESULT(:old.gol\_casa,:old.gol\_ospite);

if (ck1=0) then

update classifica set punti=punti-1,partite\_giocate=partite\_giocate-1,partite\_pareggiate=partite\_pareggiate-1,gol\_fatti=gol\_fatti-:old.gol\_casa,gol\_subiti=gol\_subiti-:old.gol\_ospite,DIFFERENZA\_RETI=differenza\_reti-(:old.gol\_casa-:old.gol\_ospite) where (id\_competizione=:old.id\_competizione and nome\_squadra=:old.squadra\_casa);

update classifica set punti=punti-1,partite\_giocate=partite\_giocate-1,partite\_pareggiate=partite\_pareggiate-1,gol\_fatti=gol\_fatti-:old.gol\_ospite,gol\_subiti=gol\_subiti-:old.gol\_casa,DIFFERENZA\_RETI=differenza\_reti-(:old.gol\_ospite-:old.gol\_casa) where (id\_competizione=:old.id\_competizione and nome\_squadra=:old.squadra\_ospite);

elsif(ck1=1) then

update classifica set punti=punti-3,partite\_giocate=partite\_giocate-1,partite\_vinte=partite\_vinte-1,gol\_fatti=gol\_fatti-:old.gol\_casa,gol\_subiti=gol\_subiti-:old.gol\_ospite,DIFFERENZA\_RETI=differenza\_reti-(:old.gol\_casa-:old.gol\_ospite) where (id\_competizione=:old.id\_competizione and nome\_squadra=:old.squadra\_casa);

update classifica set partite\_giocate=partite\_giocate-1,partite\_perse=partite\_perse-1,gol\_fatti=gol\_fatti-:old.gol\_ospite,gol\_subiti=gol\_subiti-:old.gol\_casa,DIFFERENZA\_RETI=differenza\_reti-(:old.gol\_ospite-:old.gol\_casa) where (id\_competizione=:old.id\_competizione and nome\_squadra=:old.squadra\_ospite);

else

update classifica set partite\_giocate=partite\_giocate-1,partite\_perse=partite\_perse-1,gol\_fatti=gol\_fatti-:old.gol\_casa,gol\_subiti=gol\_subiti-:old.gol\_ospite,DIFFERENZA\_RETI=differenza\_reti-(:old.gol\_casa-:old.gol\_ospite) where (id\_competizione=:old.id\_competizione and nome\_squadra=:old.squadra\_casa);

update classifica set punti=punti-3,partite\_giocate=partite\_giocate-1,partite\_vinte=partite\_vinte-1,gol\_fatti=gol\_fatti-:old.gol\_ospite,gol\_subiti=gol\_subiti-:old.gol\_casa,DIFFERENZA\_RETI=differenza\_reti-(:old.gol\_ospite-:old.gol\_casa) where (id\_competizione=:old.id\_competizione and nome\_squadra=:old.squadra\_ospite);

end if;

end;

1. Il trigger **CANCELLA\_CLASSIFICA,** attivato prima della modifica nella tabella *classifica,* fa si che le squadre presenti all’interno di tale tabella siano in relazione con la nazione della competizione. La modifica della squadra non può avvenire se questa ha già disputato una qualche partita in tale competizione.

**create or replace TRIGGER "CANCELLA\_CLASSIFICA"**

before update on classifica

for each row

declare

stato\_c competizione.nazione%type;

stato\_s squadra.nazione%type;

città squadra.città%type;

begin

select c.nazione into stato\_c from competizione c where c.id\_competizione=:old.id\_competizione;

select s.nazione into stato\_s from squadra s where s.nome=:new.nome\_squadra;

select s.città into città from squadra s where s.nome=:new.nome\_squadra;

if(:old.nome\_squadra<>:new.nome\_squadra and :old.partite\_giocate>0) then

raise\_application\_error(-20044,'Non è possibile cambiare squadra dato che ha già disputato partite!');

elsif(:old.nome\_squadra<>:new.nome\_squadra and stato\_s<>stato\_c and stato\_c<>'internazionale') or (città is null and stato\_c is not null) or (città is not null and stato\_c is null) then

raise\_application\_error(-20043,'Non è possibile cambiare squadra con quella inserita poichè non fa parte della nazione della competizione o è di un tipo diverso!');

end if;

end;

1. Il trigger **VINCOLO\_PERSONA**, scatenato prima dell’inserimento nella tabella *persona*, controlla che i dati inseriti rispettino alcuni particolari di notevole importanza.

E’ necessario, ad esempio, che una persona di tipo ‘giocatore’ non sia contemporaneamente ‘ex-giocatore’. Questo vincolo è stato implementato con il seguente trigger e non con un semplice constraint in modo da poter visualizzare l’opportuno messaggio di errore.

**create or replace TRIGGER "VINCOLO\_PERSONA"**

before insert on persona

for each row

declare

begin

if (:new.tipo<>'giocatore' and :new.ex\_giocatore is null and :new.ruolo\_preferito is not null) then

raise\_application\_error(-20057,'Non essendo (o non essendo stato) un giocatore non è possibile assegnargli un ruolo preferito');

elsif (:new.tipo='giocatore' and (:new.ex\_giocatore is not null or :new.ruolo\_preferito is null)) then

raise\_application\_error(-20058,'Non è possibile che un giocatore sia già "EX" o/e invece necessario assegnargli un ruolo preferito');

end if;

:new.id\_persona:=seq\_persona.nextval;

end;

1. Il trigger **CANCELLA\_CONTRATTO,** innescato prima della cancellazione di un record dalla tabella *contratto,* verifica se rescindere il contratto tra una persona ed una società è un’operazione possibile.

**create or replace TRIGGER "CANCELLA\_CONTRATTO"**

before delete on contratto

for each row

declare

ck integer:=0;

begin

select count(\*) into ck from ( partita p join evento e on p.id\_partita=e.id\_partita) where (p.squadra\_casa=:old.nome\_squadra or p.squadra\_ospite=:old.nome\_squadra) and p.data>=:old.inizio\_contratto and e.id\_persona=:old.id\_persona ;

if ck<>0 then

raise\_application\_error(-20051,'Non è possibile eliminare tale contratto dato che ci sono eventi associati a questa persona per questa squadra');

end if ;

end;

1. Il trigger **CANCELLA\_EVENTO,** innescato dopo la cancellazione di un record dalla tabella *evento,* verifica se la cancellazione di un evento sia così importante da compromettere dati presenti in altre tabelle. E’ necessario, ad esempio, aggiornare il risultato finale di una partita se l’evento eliminato era di tipo ‘gol’ o aggiornare il numero di presenze nel caso in cui si elimini una persona partecipe alla gara.

**create or replace TRIGGER "CANCELLA\_EVENTO"**

after delete on evento

for each row

declare

sq squadra.nome%type;

casa squadra.nome%type;

quando partita.data%type;

begin

if (:old.tipo='gol') then

select p.data into quando from partita p where p.id\_partita=:old.id\_partita;

select c.nome\_squadra into sq from contratto c where c.ID\_PERSONA=:old.id\_persona and quando BETWEEN c.inizio\_contratto and c.FINE\_CONTRATTO;

select p.squadra\_casa into casa from partita p where p.id\_partita=:old.id\_partita;

if casa=sq then

update partita set gol\_casa=gol\_casa-1 where id\_partita=:old.id\_partita;

else

update partita set gol\_ospite=gol\_ospite-1 where id\_partita=:old.id\_partita;

end if;

end if;

if (:old.tipo='entrata') then

update persona set presenze=presenze-1 where id\_persona=:old.id\_persona;

end if;

end;

1. Il trigger **CANCELLA\_PERSONA,** innescato prima della cancellazione di un record dalla tabella *persona,* verifica se la cancellazione di una determinata persona sia considerevole a tal punto da compromettere dati presenti in altre tabelle. Ad esempio, se alla persona che si vuole eliminare è associata almeno una tupla nella tabella evento, non deve poter essere eliminata. Gli ‘eventi’ ad essa associati risulterebbero non validi, privi di un interprete.

**create or replace TRIGGER "CANCELLA\_PERSONA"**

before delete on persona

for each row

declare

ck integer:=0;

begin

select count(\*) into ck from evento e where e.id\_persona=:old.id\_persona;

if ck<>0 then

raise\_application\_error(-20055,'Non è possibile eliminare la persona dato che altri dati dipendono da essa. Eliminare tutte le dipendenze e poi riprovare');

else

delete from contratto where id\_persona=:old.id\_persona;

end if;

end;

1. Il trigger **CATENA\_EVENTO,** innescato prima della cancellazione di un record dalla tabella *partita* fa si che la cancellazione, di una partita, si ripercuota sulla tabella *evento*, in particolare sui record relativi alla partita in esame.

**create or replace TRIGGER "CATENA\_EVENTO"**

before delete on partita

for each row

declare

begin

delete from evento where id\_partita=:old.id\_partita;

end;

1. Il trigger **CATENA\_PARTITA,** innescato prima della cancellazione di un record dalla tabella *competizione* fa si che la rimozione di una determinata competizione si ripercuota su tutti i record della tabella classifica e della tabella partita aventi associati la competizione prossima ad essere depennata.

**create or replace TRIGGER "CATENA\_PARTITA"**

before delete on competizione

for each row

declare

begin

delete from partita where id\_competizione=:old.id\_competizione;

delete from classifica where id\_competizione=:old.id\_competizione;

end;

1. Il trigger **NUMERO\_GIOCATORI,** innescato prima dell’inserimento di un nuovo record nella tabella *evento* controlla che la persona alla quale si sta associando un evento, di tipo diverso da ‘entrata’, faccia parte di una squadra già al completo, con undici calciatori in campo.

Il controllo avviene anche nel caso si tenti di far entrare una nuova persona ma la sua squadra è già composta da undici persone.

**create or replace TRIGGER "NUMERO\_GIOCATORI"**

before insert on evento

for each row

declare

ck integer :=0;

tipo persona.tipo%type;

quando partita.data%type;

sq contratto.nome\_squadra%type;

sqc partita.squadra\_casa%type;

sqo partita.squadra\_ospite%type;

allen persona.id\_persona%type;

begin

select p.tipo into tipo from persona p where p.id\_persona=:new.id\_persona;

if tipo<>'arbitro' then

select p.data into quando from partita p where p.id\_partita=:new.id\_partita;

select c.nome\_squadra into sq from contratto c where c.id\_persona=:new.id\_persona and quando BETWEEN c.inizio\_contratto and c.fine\_contratto;

select p.squadra\_casa into sqc from partita p where p.id\_partita=:new.id\_partita;

select p.squadra\_ospite into sqo from partita p where p.id\_partita=:new.id\_partita;

if(tipo='giocatore' and sq=sqc) then

select count(\*) into ck from formazione f where f.partita=:new.id\_partita and f.squadra=sq;

if (ck<11 and :new.tipo<>'entrata') then

raise\_application\_error(-20014,'Non si può assegnare tale evento, la squadra di casa non ha ancora undici giocatori in campo');

elsif(ck>10 and :new.tipo='entrata' and :new.minuto=0) then

raise\_application\_error(-20014,'La squadra di casa ha già undici giocatori in campo');

elsif(ck=1) then

select r.id\_persona into allen from contratto r where r.nome\_squadra=sqc and r.numero\_maglia is null and quando between r.inizio\_contratto and r.fine\_contratto;

insert into evento values (0,null,'entrata',null,allen,:new.id\_partita);

end if;

elsif(tipo='giocatore' and sq=sqo) then

select count(\*) into ck from formazione f where f.partita=:new.id\_partita and f.squadra=sq;

if (ck<11 and :new.tipo<>'entrata') then

raise\_application\_error(-20014,'Non si può assegnare tale evento, la squadra ospite non ha ancora undici giocatori in campo');

elsif(ck>10 and :new.tipo='entrata' and :new.minuto=0) then

raise\_application\_error(-20014,'La squadra ospite ha già undici giocatori in campo');

elsif(ck=1) then

select r.id\_persona into allen from contratto r where r.nome\_squadra=sqo and r.numero\_maglia is null and quando between r.inizio\_contratto and r.fine\_contratto;

insert into evento values (0,null,'entrata',null,allen,:new.id\_partita);

end if;

end if;

end if;

end;