

FUNZIONI

Questa funzione è progettata per essere eseguita come trigger prima di un'operazione DELETE sulla tabella progetto.studente

Viene eseguita un'operazione di INSERT nella tabella progetto.memoria_studente utilizzando i valori presenti nella vecchia riga (OLD) della tabella progetto.studente (campi matricola, nome, cognome, email).

Viene eseguito un controllo per determinare se la funzione è stata chiamata a seguito di un'operazione di DELETE. In caso affermativo, la funzione restituirà la vecchia riga (OLD).

Se la funzione non è stata chiamata da un'operazione di DELETE, procede eliminando la riga corrispondente dalla tabella progetto.studente utilizzando il valore della matricola dalla vecchia riga.

Infine, la funzione restituisce NULL, indicando che l'operazione è stata gestita correttamente.

Per la gestione delle eccezioni, la funzione include un blocco EXCEPTION che solleva una notifica con un messaggio di errore se si verifica un problema durante l'esecuzione.

Infine, viene creato un trigger chiamato delete_studente che esegue la funzione progetto.salvataggio_studente_in_memoria() prima di ogni operazione di DELETE sulla tabella progetto.studente

```
CREATE OR REPLACE FUNCTION progetto.salvataggio_studente_in_memoria()
RETURNS TRIGGER AS $$
BEGIN

    INSERT INTO progetto.memoria_studente(matricola, nome, cognome, email)
    VALUES (OLD.matricola, OLD.nome, OLD.cognome, OLD.email);

    -- Controlla se siamo chiamati da DELETE
    IF TG_OP = 'DELETE' THEN
        RETURN OLD;
    END IF;

    DELETE FROM progetto.studente WHERE matricola = OLD.matricola;

    RETURN NULL;
EXCEPTION
    WHEN OTHERS THEN
        RAISE NOTICE 'Errore durante l'inserimento in memoria_studente: %', SQLERRM;
        RETURN NULL;
END;
$$ LANGUAGE plpgsql;

delete from memoria_studente where nome = 'Marco';

-- Creazione del trigger
CREATE TRIGGER delete_studente
BEFORE DELETE ON progetto.studente
FOR EACH ROW
EXECUTE FUNCTION progetto.salvataggio_studente_in_memoria();
```

La funzione `progetto.check_date_exam()` opera come trigger prima di un'operazione di inserimento (INSERT) sulla tabella `progetto.appelli`. La sua funzione principale è controllare se esiste già un appello con la stessa data di svolgimento e lo stesso anno accademico associato a un insegnamento. In caso affermativo, viene sollevata un'eccezione con un messaggio informativo. Altrimenti, la funzione consente l'inserimento della nuova riga nella tabella `progetto.appelli`.

Il trigger associato, denominato `check_date_exam_trigger`, viene attivato prima di ogni inserimento sulla tabella `progetto.appelli` e esegue la funzione `progetto.check_date_exam()`. Questo meccanismo garantisce che non ci siano sovrapposizioni indesiderate nelle date degli appelli per uno stesso insegnamento.

```
-- funzione per controllare la correttezza del calendario d'esame
CREATE OR REPLACE FUNCTION progetto.check_date_exam() RETURNS TRIGGER AS $$
DECLARE
    anno_insegnamento INTEGER;
BEGIN
    -- Ottenere l'anno accademico associato all'insegnamento
    SELECT i.anno INTO anno_insegnamento
    FROM progetto.insegnamenti i
    WHERE i.codice = NEW.id;

    -- Controllare se esiste già un appello per la stessa data e anno accademico
    IF EXISTS (
        SELECT 1
        FROM progetto.appelli a
        INNER JOIN progetto.insegnamenti i ON i.codice = a.id
        WHERE NEW.data_svolgimento = a.data_svolgimento
        AND anno_insegnamento = i.anno
    ) THEN
        RAISE EXCEPTION 'Esiste già un appello per la stessa data e anno accademico';
    END IF;

    RETURN NEW;
END;
$$ LANGUAGE plpgsql;

-- Creare il trigger
CREATE TRIGGER check_date_exam_trigger
BEFORE INSERT ON progetto.appelli
FOR EACH ROW
EXECUTE FUNCTION progetto.check_date_exam();
```

La funzione `carriera_completa_studente` è una procedura che permette di ottenere una visione completa della carriera accademica di uno studente. Prende in input la matricola dello studente e restituisce una tabella con le seguenti colonne: `codice_ins` (codice dell'insegnamento), `nomeIns` (nome dell'insegnamento), `voto` (voto ottenuto) e `data_svolgimento` (data in cui l'esame è stato svolto).

La funzione utilizza un costrutto `RETURN QUERY` per restituire i risultati di una query SQL. In particolare, la query seleziona le informazioni necessarie dalla tabella `memoria_esame`, che contiene i dettagli degli esami sostenuti dagli studenti, e dalla tabella `insegnamenti` utilizzando una clausola `JOIN` per collegare i codici degli insegnamenti.

La condizione `WHERE` limita i risultati solo agli esami dello studente specificato dalla matricola di input.

In sintesi, questa funzione fornisce un modo semplice per visualizzare tutti gli esami sostenuti da uno studente, inclusi i dettagli come il nome dell'insegnamento, il voto ottenuto e la data in cui l'esame è stato svolto.

```
● -- Funzione che mi permette di visualizzare la carriera completa di uno studente
CREATE OR REPLACE FUNCTION carriera_completa_studente(matricola_input int)
RETURNS TABLE (
    codice_ins int,
    nomeIns varchar(50),
    voto int,
    data_svolgimento date
) AS $$
BEGIN
    RETURN QUERY
    SELECT
        me.codice_ins,
        i.nome AS nomeIns,
        me.voto,
        me.data_svolgimento
    FROM memoria_esame me
    JOIN insegnamenti i ON me.codice_ins = i.codice
    WHERE me.matricola = matricola_input;
END;
$$ LANGUAGE plpgsql;
```

La funzione `carriera_valida_studente` prende in input la matricola di uno studente e restituisce una tabella contenente le seguenti colonne: `codice_ins` (codice dell'insegnamento), `nomeIns` (nome dell'insegnamento), `voto` (voto ottenuto) e `data_svolgimento` (data in cui l'esame è stato svolto).

Questa funzione fornisce una panoramica della carriera accademica di uno studente, limitando i risultati agli esami in cui lo studente ha ottenuto un voto uguale o superiore a 18. La funzione utilizza una query SQL per estrarre le informazioni necessarie dalle tabelle `esame` e `insegnamenti`, collegando i codici degli insegnamenti.

In breve, la funzione identifica e restituisce gli esami validi (con un voto di 18 o superiore) di uno studente specifico, mostrando il codice dell'insegnamento, il nome dell'insegnamento, il voto ottenuto e la data in cui l'esame è stato svolto.

```
● -- Funzione che mi permette di visualizzare la carriera valida di uno studente
create or replace function carriera_valida_studente(matricola_input int)
returns table (
    codice_ins int,
    nomeIns varchar(50),
    voto int,
    data_svolgimento date
) as $$
begin
    return QUERY
    select
        e.codiceIns,
        i.nome AS nomeIns,
        e.voto,
        e.data_svolgimento
    from esame e
    join insegnamenti i on e.codice_ins = i.codice
    where e.matricola = matricola_input and e.voto >= 18;
end;
$$ language plpgsql;
```

La funzione `cdl_info` riceve in input il codice di un corso di laurea e restituisce una tabella contenente le seguenti colonne: `codiceIns` (codice dell'insegnamento), `nomeIns` (nome dell'insegnamento), `descrizione` (descrizione dell'insegnamento), `annoIns` (anno dell'insegnamento), `nomeDocente` (nome del docente responsabile), `cognomeDocente` (cognome del docente responsabile).

Questa funzione fornisce informazioni riguardanti un corso di laurea specifico, identificato dal suo codice. Essa estrae dati dalla tabella `insegnamenti`, limitando i risultati agli insegnamenti associati al corso di laurea indicato.

In breve, la funzione restituisce una tabella con dettagli sugli insegnamenti del corso di laurea specificato, mostrando il codice dell'insegnamento, il nome dell'insegnamento, la descrizione, l'anno, il nome e il cognome del docente responsabile.

```
-- Funzione che mi permette di riprodurre le informazioni riguardante un corso di laurea
create or replace function cdl_info(codice_corso int)
returns table (
  codiceIns int,
  nomeIns varchar(50),
  descrizione varchar(100),
  annoIns int,
  nomeDocente varchar(20),
  cognomeDocente varchar(20)
) as $$
begin
  return QUERY
  select
    i.codice as codiceIns,
    i.nome as nomeIns,
    i.descrizione as descrizione,
    i.anno as annoIns,
    d.nome as nomeDocente,
    d.cognome as cognomeDocente
  from insegnamenti i
  where i.cdl_codice = codice_corso;
end;
$$ language plpgsql;
```