Funzioni Realizzate

Nozioni necessarie alla comprensione del documento

La tabella delle propedeuticità contiene record composti da due identificatori di corsi:
- identificatore del corso propedeutico
- identificatore del corso a cui è riferita la propedeuticità

Dati due insegnamenti X ed Y, dove l'insegnamento X è propedeutico all'insegnamento Y, tale propedeuticità può essere salvata all'interno della tabella in due modalità differenti da cui scaturiscono le nomenclature propedeuticità diretta e transitiva utilizzati all'interno del documento.

La propedeuticità tra due insegnamenti X ed Y, dove l'insegnamento X è propedeutico all'insegnamento Y, è definita diretta se la propedeuticità è definita dalla presenza di un record all'interno della tabella delle propedeuticità che ha come identificatore del corso propedeutico l'identificatore del corso X e come identificatore del corso a cui è riferita la propedeuticità l'identificatore del corso Y.

La propedeuticità tra due insegnamenti X ed Y, dove l'insegnamento X è propedeutico all'insegnamento Y, è definita transitiva se la propedeuticità è definita come segue:
- vi è un insieme di insegnamenti $I = \{ X_1, X_2, ..., X_n \}$ dove $\forall X_i \in I$ l'insegnamento X_i è propedeutico all'insegnamento X_{i+1} per propedeuticità diretta
- l'insegnamento X è propedeutico all'insegnamento X_1 per propedeuticità diretta
- l'insegnamento X_n è propedeutico all'insegnamento Y per propedeuticità diretta

Definizioni in aiuto alla comprensione del documento

```
create domain anno_insegnamento as integer check (value > 0 and value < 4);
create domain valutazione_esame as integer check (value >= 0 and value <= 30);
create domain tipo_corso_laurea as char(1) check (value in ('T', 'M'));
```

Funzioni

```
function controllo_anno_insegnamento() returns trigger
```

La funzione `controllo_anno_insegnamento` restituisce un trigger che si occupa di verificare che nell'inserimento o nell'aggiornamento di un nuovo insegnamento, l'anno previsto da quest'ultimo sia coerente con quelli relativi al tipo di corso di laurea, ovvero, per le lauree di tipo triennale da 1 a 3 e per le lauree di tipo magistrale da 1 a 2.

Nel caso in cui l'anno non è coerente con il tipo di corso di laurea, viene sollevata un'eccezione che non permette di portare a termine l'operazione.

La funzione viene chiamata da un trigger che è impostato **`before update or insert`** sulla tabella `insegnamenti`.

```
function controllo_numero_insegnamenti_docente() returns trigger
```

La funzione `controllo_numero_insegnamenti_docente` restituisce un trigger che si occupa di verificare che nell'inserimento o nell'aggiornamento di un nuovo insegnamento, il docente segnato come responsabile di tale insegnamento, non superi il numero massimo di corsi di cui può occuparsi.

Nel caso in cui il docente supererebbe il numero massimo di corsi di cui può occuparsi, viene sollevata un'eccezione che non permette di portare a termine l'operazione.

La funzione viene chiamata da un trigger che è impostato **`before update or insert`** sulla tabella `insegnamenti`.

```
function controllo_propedeuticità_iscrizione_appello() returns trigger
```

La funzione `controllo_propedeuticità_iscrizione_appello` restituisce un trigger che si occupa di verificare che nell'inserimento o nell'aggiornamento di una iscrizione ad un appello d'esame lo studente abbia superato le propedeuticità (propedeuticità dirette) necessarie per sostenere l'esame dell'insegnamento tenuto in tale appello.

Si noti che, se l'ultimo appello sostenuto da uno studente di un dato insegnamento ha avuto esito positivo ma lo studente si è iscritto ad un nuovo appello di cui non ha ancora ricevuto la valutazione, l'insegnamento non è considerato come superato ai fini del rispetto delle propedeuticità.

Nel caso in cui lo studente non avesse ancora superato tutti gli esami di insegnamenti propedeutici all'insegnamento dell'appello al quale sta tentando l'iscrizione con successo (ovvero, l'ultimo voto registrato per tali esami ha esito positivo), viene sollevata un'eccezione che non permette di portare a termine l'operazione.

La funzione viene chiamata da un trigger che è impostato **`before update or insert`** sulla tabella `esami`.

```
function get_appelli_studente_non_iscritto_futuri(s integer) returns table(
     appello integer,
     data_appello date,
     cdl varchar(15),
     codice_insegnamento varchar(15),
     nome_insegnamento varchar(128),
     descrizione_insegnamento text,
     anno_insegnamento anno_insegnamento
)
```

Parametri:
- `s integer`: deve essere la matricola identificativa di uno studente

La funzione `controllo_propedeuticità_iscrizione_appello` restituisce una tabella che contiene tutte le informazioni relative agli appelli che hanno data maggiore di quella corrente ad eccezione degli appelli a cui lo studente di matricola `s` è già iscritto.

```
function get_insegnamenti_a_cui_propedeutico(cl varchar(15), ci varchar(15)) returns setof insegnamenti
```

Parametri:
- `cl varchar(15)`: deve essere il codice identificativo di un corso di laurea
- `ci varchar(15)`: deve essere il codice identificativo di un insegnamento

La funzione `get_insegnamenti_a_cui_propedeutico` restituisce una tabella contenente tutti gli insegnamenti che hanno l'insegnamento identificato da `cl` e `ci` segnato come propedeutico (propedeuticità diretta).

```
function get_insegnamenti_aggiungibili_come_propedeutici(cl varchar(15), ci varchar(15)) returns setof insegnamenti
```

Parametri:
- `cl varchar(15)`: deve essere il codice identificativo di un corso di laurea
- `ci varchar(15)`: deve essere il codice identificativo di un insegnamento

La funzione `get_insegnamenti_aggiungibili_come_propedeutici` restituisce una tabella contenente tutti gli insegnamenti che possono essere aggiunti come propedeutici all'insegnamento identificato da `cl` e `ci` (senza tenere conto di eventuali loop di transitività, prevenuti dalla funzione `previeni_inserimento_propedeuticità_transitive_loop` descritta successivamente).

Più nel dettaglio, vengono mostrati tutti gli insegnamenti che:
- fanno parte del medesimo corso di laurea dell'insegnamento identificato da `cl` e `ci`
- non sono già segnati come propedeutici (propedeuticità diretta) per l'insegnamento identificato da `cl` e `ci`
- non hanno il corso identificato da `cl` e `ci` segnato come propedeutico (propedeuticità diretta) per essi

```
function get_insegnamenti_propedeutici(cl varchar(15), ci varchar(15)) returns setof insegnamenti
```

Parametri:
- `cl varchar(15)`: deve essere il codice identificativo di un corso di laurea
- `ci varchar(15)`: deve essere il codice identificativo di un insegnamento

La funzione `get_insegnamenti_propedeutici` restituisce una tabella contenente tutti gli insegnamenti segnati come propedeutici (propedeuticità diretta) per l'insegnamento identificato da `cl` e `ci`.

```
function get_iscrizioni_appello(id_appello integer) returns table(
     matricola_studente integer,
     email_studente varchar(254),
     nome_studente varchar(128),
     cognome_studente varchar(128),
     valutazione_esame valutazione_esame
)
```

Parametri:
- `id_appello integer`: deve essere il codice identificativo di un appello

La funzione `get_iscrizioni_appello` restituisce una tabella contenente tutte le informazioni relative agli studenti e alla relativa valutazione che sono iscritti all'appello identificato da `id_appello`.

```
function get_iscrizioni_attive_appelli_studente(s integer) returns table(
     appello integer,
     data_appello date,
     valutazione valutazione_esame,
     cdl varchar(15),
     codice_insegnamento varchar(15),
     nome_insegnamento varchar(128),
     descrizione_insegnamento text,
     anno_insegnamento anno_insegnamento,
     email_docente varchar(254),
     nome_docente varchar(128),
     cognome_docente varchar(128)
)
```

Parametri:
- `s integer`: deve essere la matricola identificativa di uno studente

La funzione `get_iscrizioni_attive_appelli_studente` restituisce una tabella contenente tutti gli appelli, l'insegnamento a cui l'appello fa riferimento e le informazioni sul docente di tale insegnamento, ai quali lo studente identificato dalla matricola `s` ha un'iscrizione attiva.

Più nel dettaglio, un'iscrizione ad un appello è considerata attiva quando:
- la data dell'appello è una data maggiore rispetto a quella corrente
- la valutazione non è ancora stata fornita

```
function previeni_appelli_per_insegnamenti_stesso_anno() returns trigger
```

La funzione `previeni_appelli_per_insegnamenti_stesso_anno` restituisce un trigger che si occupa di prevenire che vengano creati più appelli per insegnamenti del medesimo corso di laurea dello stesso anno lo stesso giorno.

Nel caso in cui viene identificata la situazione sopra descritta, viene sollevata un'eccezione che non permette di portare a termine l'operazione.

La funzione viene chiamata da un trigger che è impostato `before update or insert` sulla tabella `appelli`.

```
function previeni_eliminazione_appello_passato() returns trigger
```

La funzione `previeni_eliminazione_appello_passato` restituisce un trigger che si occupa di prevenire l'eliminazione di un appello che è già avvenuto.

Nel caso in cui viene identificata la situazione sopra descritta, viene sollevata un'eccezione che non permette di portare a termine l'operazione.

La funzione viene chiamata da un trigger che è impostato `before delete` sulla tabella `appelli`.

```
function previeni_inserimento_propedeuticità_cdl_differenti() returns trigger
```

La funzione `previeni_inserimento_propedeuticità_cdl_differenti` restituisce un trigger che si occupa di prevenire la creazione di propedeuticità tra insegnamenti di corsi di laurea differenti.

Nel caso in cui viene identificata la situazione sopra descritta, viene sollevata un'eccezione che non permette di portare a termine l'operazione.

La funzione viene chiamata da un trigger che è impostato `before insert or update` sulla tabella `propedeuticità`.

```
function previeni_inserimento_propedeuticità_medesimo_insegnamento() returns trigger
```

La funzione `previeni_inserimento_propedeuticità_medesimo_insegnamento` restituisce un trigger che si occupa di prevenire la creazione di propedeuticità (propedeuticità diretta) sul medesimo corso, ovvero di creare propedeuticità che affermino che un insegnamento X è propedeutico a se stesso.

Nel caso in cui viene identificata la situazione sopra descritta, viene sollevata un'eccezione che non permette di portare a termine l'operazione.

La funzione viene chiamata da un trigger che è impostato **`before insert or update`** sulla tabella `propedeuticità`.

```
function previeni_inserimento_propedeuticità_transitive_loop() returns trigger
```

La funzione `previeni_inserimento_propedeuticità_transitive_loop` restituisce un trigger che si occupa di prevenire la creazione di situazioni in cui si viene a creare un circolo di propedeuticità (propedeuticità transitive) che porta a non potersi iscrivere a nessun insegnamento coinvolto in tale circolo.

Si supponga di avere 3 insegnamenti A, B e C e di avere la seguente situazione:
- l'insegnamento A è segnato come propedeutico a B
- l'insegnamento B è segnato come propedeutico a C
- l'insegnamento C è segnato come propedeutico a A

La situazione che ne consegue è che, supponendo la presenza di uno studente S che non ha ancora sostenuto con successo nessuno degli insegnamenti A, B, C:
- se lo studente U prova ad iscriversi ad un appello dell'insegnamento A questo gli viene impedito in quanto non ha sostenuto con successo l'insegnamento C
- se lo studente U prova ad iscriversi ad un appello dell'insegnamento B questo gli viene impedito in quanto non ha sostenuto con successo l'insegnamento A
- se lo studente U prova ad iscriversi ad un appello dell'insegnamento C questo gli viene impedito in quanto non ha sostenuto con successo l'insegnamento B

Si osserva uno stallo che porta l'utente a non potersi iscrivere a nessuno degli appelli degli insegnamenti coinvolti in questa situazione.

Ovviamente, il problema delle propedeuticità transitive può coinvolgere un numero arbitrario di insegnamenti.

Grazie alla funzione `previeni_inserimento_propedeuticità_transitive_loop` viene prevenuto l'inserimento di propedeuticità che portano allo scaturire del problema sopra descritto.

Nel caso in cui viene identificata tale situazione, viene sollevata un'eccezione che non permette di portare a termine l'operazione.

La funzione viene chiamata da un trigger che è impostato **`before insert or update`** sulla tabella `propedeuticità`.

```
function previeni_iscrizione_appello_data_passata() returns trigger
```

La funzione `previeni_iscrizione_appello_data_passata` restituisce un trigger che si occupa di prevenire l'iscrizione di uno studente ad un appello con una data che è minore della data corrente, ovvero, ad un appello passato.

Nel caso in cui viene identificata la situazione sopra descritta, viene sollevata un'eccezione che non permette di portare a termine l'operazione.

La funzione viene chiamata da un trigger che è impostato **`before insert or update`** sulla tabella `esami`.

```
function previeni_iscrizione_appello_insegnamento_non_in_cdl() returns trigger
```

La funzione `previeni_iscrizione_appello_insegnamento_non_in_cdl` restituisce un trigger che si occupa di prevenire l'iscrizione di uno studente ad un appello di un insegnamento di un corso di laurea differente da quello a cui è iscritto lo studente.

Nel caso in cui viene identificata la situazione sopra descritta, viene sollevata un'eccezione che non permette di portare a termine l'operazione.

La funzione viene chiamata da un trigger che è impostato **`before insert or update`** sulla tabella `esami`.

```
function produci_carriera_completa_studente(s integer) returns table(
    appello integer,
    data_appello date,
    valutazione valutazione_esame,
    cdl varchar(15),
    codice_insegnamento varchar(15),
    nome_insegnamento varchar(128),
    descrizione_insegnamento text,
    anno_insegnamento anno_insegnamento,
    email_docente varchar(254),
    nome_docente varchar(128),
    cognome_docente varchar(128)
)
```

Parametri:
- `s integer`: deve essere la matricola identificativa di uno studente

La funzione `produci_carriera_completa_studente` restituisce una tabella contenente tutti gli esami, con le relative valutazioni, sostenuti dallo studente identificato dalla matricola `s`.

Ad ogni esame è inoltre associato:
- l'appello nel quale è stato sostenuto
- l'insegnamento oggetto dell'esame
- il docente responsabile di tale insegnamento

Un'esame è considerato sostenuto (sia esso con esito positivo o negativo) se la data dell'appello è minore della data corrente (ovvero l'appello è di fatto avvenuto) o se è stata assegnata una valutazione, ad esempio, un appello posto in data futura per la verbalizzazione di un voto a cui però il docente ha già inserito la valutazione viene considerato sostenuto senza dover aspettare l'effettiva data dell'appello.

```
function produci_carriera_completa_studente_storico(s integer) returns table(
     appello integer,
     data_appello date,
     valutazione valutazione_esame,
     cdl varchar(15),
     codice_insegnamento varchar(15),
     nome_insegnamento varchar(128),
     descrizione_insegnamento text,
     anno_insegnamento anno_insegnamento,
     email_docente varchar(254),
     nome_docente varchar(128),
     cognome_docente varchar(128)
)
```

Funzione speculare alla precedente ma che agisce sullo storico studenti e non sugli studenti attivi.

```
function produci_carriera_valida_studente(s integer) returns table(
     appello integer,
     data_appello date,
     valutazione valutazione_esame,
     cdl varchar(15),
     codice_insegnamento varchar(15),
     nome_insegnamento varchar(128),
     descrizione_insegnamento text,
     anno_insegnamento anno_insegnamento,
     email_docente varchar(254),
     nome_docente varchar(128),
     cognome_docente varchar(128)
)
```

Parametri:
- `s integer`: deve essere la matricola identificativa di uno studente

La funzione `produci_carriera_valida_studente` restituisce una tabella contenente tutte le informazioni di tutti gli insegnamenti che lo studente ha superato insieme alle informazioni dei relativi docenti e dell'appello grazie al quale l'insegnamento viene considerato superato.

Più nel dettaglio, un insegnamento viene considerato superato quando:
- lo studente identificato dalla matricola `s` ha sostenuto almeno un esame di tale insegnamento
- l'ultimo esame sostenuto da tale studente ha avuto esito positivo, ovvero una valutazione corrispondente o superiore a 18

```
function produci_carriera_valida_studente_storico(s integer) returns table(
     appello integer,
     data_appello date,
     valutazione valutazione_esame,
     cdl varchar(15),
     codice_insegnamento varchar(15),
     nome_insegnamento varchar(128),
     descrizione_insegnamento text,
     anno_insegnamento anno_insegnamento,
     email_docente varchar(254),
     nome_docente varchar(128),
     cognome_docente varchar(128)
)
```

Funzione speculare alla precedente ma che agisce sullo storico studenti e non sugli studenti attivi.

```
function produci_informazioni_corso_laurea(cdl varchar(15)) returns table(
     codice varchar(15),
     nome varchar(15),
     descrizione text,
     anno anno_insegnamento,
     email_docente varchar(254),
     nome_docente varchar(128),
     cognome_docente varchar(128)
)
```

Parametri:
- `cdl varchar(15)`: deve essere il codice identificativo di un corso di laurea

La funzione `produci_informazioni_corso_laurea` restituisce una tabella contenente tutti le informazioni degli insegnamenti e dei relativi docenti facenti parte del corso di laurea identificato dal codice `cdl`.

```
function salvataggio_studente_in_storico() returns trigger
```

La funzione `salvataggio_studente_in_storico` restituisce un trigger che si occupa, in fase di eliminazione di uno studente, di spostare le informazioni relative a quest'ultimo e alla sua carriera nelle relative tabelle di storico prima di procedere.

La funzione viene chiamata da un trigger che è impostato **`before delete`** sulla tabella `studenti`.

```
create or replace procedure delete_docente(
     doc varchar(254),
     cl_arr varchar(15)[],
     ins_arr varchar(15)[],
     doc_arr varchar(254)[]
)
```

Parametri:
- `doc varchar(254)`: deve essere l'indirizzo email identificativa di un docente
- `cl_arr varchar(15)[]`: deve essere un array di codici identificativi di corsi di laurea
- `ins_arr varchar(15)[]`: deve essere un array di codici identificativi di insegnamenti
- `doc_arr varchar(254)[]`: deve essere un array di email identificative di docenti

La funzione `delete_docente` permette di eliminare il docente identificato dall'indirizzo email `doc`.

Essendo che un insegnamento non può rimanere senza un docente, vengono richiesti anche dei docenti da assegnare ai corsi che prima gestiva il docente che si sta tentando di eliminare.

In particolare, `cl_arr`, `ins_arr` e `doc_arr` devono essere array di lunghezza pari al numero di insegnamenti assegnati al docente identificato dall'indirizzo email `doc`.

Tali array vengono così interpretati dalla funzione: per ogni indice i compreso tra 0 e il numero di corsi assegnati al docente identificato dall'indirizzo email `doc`, aggiorna l'insegnamento identificato dal codice `ins_arr[i]`, facente parte del corso di laurea identificato dal codice `cl_arr[i]` e precedentemente tenuto dal docente identificato dall'indirizzo email `doc`, sostituendo il docente con quello identificato dall'indirizzo email `doc_arr[i]`.

Nel caso in cui gli array non rappresentino dei dati integri (es. dimensione differente tra loro, dimensione differente dal numero di insegnamenti del docente in eliminazione) viene sollevata un'eccezione.

Nel caso in cui uno dei docenti che si sta tentando di assegnare ad un corso abbia già raggiunto il numero massimo di corsi viene sollevata un eccezione.

Nel caso in cui gli array contengano codici che identificano un corso non di proprietà del docente che si sta tentando di eliminare viene sollevata un'eccezione.