RELAZIONE PROGETTO DATABASE

Giacomo Lucca 02678A

Contenuto del file

Descrizione delle tabelle, delle funzioni e prove di funzionamento

Tabelle

Utenti

Tabella contenente i vari utenti della piattaforma universitaria.

Ogni Utente dovrà possedere un **codice fiscale (cf)** obbligatorio. Serve nel caso in cui si cancella un utente. Per specifica, dato che ogni utente cancellato finirà nella tabella “Utenti_old”, risultando di fatti non più accessibile, per recuperalo verrà effettuata una ricerca per **codice fiscale** e verrà riportato in questa tabella.

Il **codice fiscale** è vincolato da questa Espressione Regolare (ER) e se non rispettata, riporterà errore:

```
constraint codice_fiscale_ER check ( cf ~ '^[A-Z]{6}[0-9]{2}[A-Z][0-9]{2}[A-Z][0-9]{3}[A-Z]$' )
```

Stessa cosa viene eseguita sulla email che dovrà essere di questo tipo:

```
constraint email_check check ( email ~ '^([a-z]+)[.]([a-z]+)([0-9]*)[@]((studenti)|(docente)|(segreteria))[.](unimi)[.](it)$')
```

Invece, per la tipologia verrà eseguito un controllo se il valore appartiene a uno di questi tre valori:

- Studente
- Segretario
- Docente

Ognuno di queste tipologie possederà dei privilegi nel database:

- Lo studente potrà iscriversi ad esami ed assistere alla propria carriera universitaria.
- Il docente potrà gestire i vari esami degli insegnamenti per cui è responsabile creandoli, eliminarli e inserendo i voti ai vari esami.
- Il segretario potrà creare nuovi studenti e gestirli. Può anche creare e gestire i vari corsi di laurea con i propri insegnamenti al loro interno.

L’email verrà calcolata sulla base della ridondanza della prima parte del nome e della prima parte del cognome (ovvero il primo nome e il primo cognome).

La password sarà salvata in formato md5.

In base alla tipologia dello user, è stata creata una policy specifica.

- Lo studente può vedere solo se stesso e i docenti e come attributo può vedere tutto tranne la password e il codice fiscale.
- Il docente vede se stesso, i suoi colleghi e gli studenti del corso di laurea in cui ha una materia e come attributo può vedere tutto tranne la password e il codice fiscale.
- Il segretario può vedere tutti e non può vedere come attributo la password.

Studenti

Se l'utente appartiene alla tipologia Studente, verrà inserito un record all'interno di questa tabella in cui verrà specificata la matricola dello studente (uno smallint), stabilire se lo studente è in corso con le tempistiche per la laurea che frequenta e il corso che frequenta lo studente.

Per specifica, lo studente può seguire un corso per volta. Se necessario, un segretario può cambiare il corso dello studente. I vari esami non validi per il nuovo corso, non verranno considerati per la carriera dello studente ma saranno presenti all'interno del Database.

(unicamente gli utenti di tipologia "studente" può possedere un record al suo interno).

In base alla tipologia dello user, è stata creata una policy specifica.

- Lo studente può visualizzare solo il proprio record.
- Il segretario qualunque cosa può vedere.
- Il docente solo gli studenti del corso di laurea in cui ha una materia.

Laurea

Verranno conservate le varie classi di lauree offerte dalla università.

Corso

Contiene le informazioni di un corso di laurea. Potrebbero esistere all'interno dell'università una classe di Laurea (es: L-31), per ottenere questa laurea posso eseguire vari corsi differenti (es: posso ottenerla se completo interamente il corso: Informatica oppure Informatica per la comunicazione digitale, Informatica Musicale, …).

Ogni corso verrà affiancata alla sua tipologia di laurea, ovvero:

- Triennale: 3 anni
- Magistrale: 2 anni
- Magistrale a ciclo unico: 5 anni
- Magistrale a ciclo unico: 6 anni (es: Medicina)

Insegnamento

Comprende le informazioni di ogni materia di uno specifico Corso di laurea contente le varie informazioni su di esso.

Le informazioni sono:

- Quanti CFU offre l'insegnamento.
- Il mese in cui viene erogato.

- L’anno in cui è previsto in cui uno studente frequenta il medesimo corso.
- Una sua descrizione.
- Il nome.
- Il codice identificativo del corso. Potrebbe darsi che ci siano corsi con lo stesso nome, quindi si è scelto questo attributo come chiave primaria.

Propedeutico

Ognuno di questi insegnamenti potrebbe darsi che siano vincolati da altri esami. Le informazioni inerenti a questi vincoli è contenuto dentro a questa tabella.

L’attributo **insegnamento** indica l’insegnamento in cui è vincolato l’insegnamento **propedeutico**, per cui uno studente dovrà prima aver registrato con esito positivo (e accettato l’esito) dell’esame nell’attributo **insegnamento** per poter sostenere l’esame di quel determinato corso.

Responsabile

Contiene le informazioni inerenti a chi sono i docenti di un determinato corso.

Esistono due tipologie differenti di docenti:

- Docente responsabile: mostrato con l’attributo 1.
- Docente normale: mostrato con l’attributo 2.

La differenza consiste nel fatto che unicamente il docente responsabile può gestire gli aspetti burocratici del corso, quindi può:

- Creare un esame per quell’insegnamento.
- Eliminare un determinato esame del corso.
- Inserire i vari voti di quell’esame.

Per specifica, un docente può essere responsabile al massimo di tre diversi insegnamenti, ma può essere potenzialmente un docente normale per infiniti corsi diversi.

Esame

Verranno memorizzati i dati dei vari esami di un insegnamento.

Verranno confermate le informazioni su:

- Data e ora dell’esame.
- Insegnamento in cui fa riferimento.
- Il docente che effettuerà l’esame (ovviamente solo docente responsabile).
- Quali lettere del cognome ammette (solo nella forma A-Z e controllato la sua correttezza attraverso una ER: `constraint lettere_check check (lettere ~ '^[A-Z]-[A-Z]$')`).

Se il docente dell'esame è stato cancellato, l'esame non verrà eliminato dalla tabella (altrimenti alcuni studenti che hanno sostenuto l'esame con quello studente dovranno farsi rivalutare) il docente verrà fissato con il valore 'Docente cancellato'.

Gli esami dei corsi verranno visualizzati entro una settimana dalla data in cui si eseguiranno, oltre tale data non sarà possibile disiscriversi da quell'esame.

In base alla tipologia dello user, è stata creata una policy specifica.

- Gli studenti possono vedere unicamente gli esami del proprio corso di laurea.
- Il docente può vedere gli esami degli insegnamenti in cui insegna (anche se non è responsabile).
- Il segretario vede tutto.

Esami_studenti

Verranno conservati i dati dell'esame relativi ad un singolo studente per record e le varie iscrizione degli esami che verranno sostenuti.

La differenza tra valutazione e iscrizione dell'esame e che l'iscrizione ha una data maggiore di quella corrente.

Il risultato dell'appello può essere anche null e verrà considerato come 'assente' non venendo nemmeno contato nella carriera dello studente.

Il parametro lode può essere dato esclusivamente se il voto è uguale a 30.

In base alla tipologia dello user, è stata creata una policy specifica.

- Gli studenti possono vedere solamente i propri esami del corso di laurea che frequentano.
- I docenti vedono solamente gli esami che gli competono (ovvero gli esami degli studenti in cui si sono iscritti o compiuti della propria materia).
- I segretari vedono tutto.

Utenti_old

Contiene tutti gli utenti che sono stati cancellati dalla piattaforma.

Quando necessario, possono essere rispostati nella tabella Utenti con tutti i suoi dati attraverso il controllo dell'esistenza del codice fiscale.

Il codice fiscale, a contrario della tabella Utenti che è unique, non è unique perché si è previsto che una determinata persona può essere (ovviamente in tempi diversi) anche tutte e tre le tipologie permesse nel Database. L'unica cosa che cambierà sarà il dominio dell'email.

Studenti_old

Contiene le informazioni degli utenti “studente” cancellati dalla base di dati.

Valgono le stesse proprietà della tabella Studenti.

Tabella Esami_studenti_old

Contiene i dati relativi agli esami di studenti cancellati, ovvero di utenti che sono contenuti nella tabella Utenti_old e Studenti_old.

Valgono le stesse proprietà della tabella Esami_studenti.

Quando lo studente verrà ripristinato dalla tabella Utenti_old a Utenti, tutti gli esami verrà spostati sulla tabella Esami_studenti (anche quelli che non c’entrano niente con il corso di laurea dello studente per evitare violazione dell’integrità referenziale).

Funzioni

crea_email( )

Crea l'email dell'utente sulla base di:

- La prima parte dell'email è determinata dal primo nome dello user (fino al primo spazio).
- La seconda parte dell'email è determinata dalla prima parte del cognome (fino al primo spazio).
- Quanti omonimi esistono (stesso nome e stesso cognome identico). In base a questo parametro verrà aggiunto un numero che identifica unicamente l'email. Per esempio se esistono due "Giacomo Lucca" nel database, verrà create il primo utente (inserito in ordine cronologico) con nome *giacomo.lucca@studenti.unimi.it* e il secondo con email *giacomo.lucca1@studenti.unimi.it* e via dicendo. Ovviamente va in base unicamente del tipo, se ci sono due "Giacomo Lucca" ma uno è uno studente e uno è un docente, il numero non verrà inserito.
- Inserimento del dominio dell'email in base al tipo dell'utente inserito: *@studenti.unimi.it* se è uno studente, *@docente*.unimi.it se è un docente oppure *@segretaria.unimi.it* se è un segretario.

Verrà restituita l'email generata alla fine della funzione. È possibile richiamare la funzione unicamente in fase di creazione dello user.

Se la tipologia non è docente, studente o segretario, la funzione solleverà un eccezione.

crea_user_before( )

Funzione eseguita a livello di amministratore del database (security definer) a causa della istruzione:
`execute 'create user '||quote_ident(new.email)||' with login encrypted password '||quote_literal(new.password);`

che permetta la creazione di un nuovo user con la password scelta.

Viene eseguita prima di una "insert into Utenti " e permette di controllare se esiste già un utente con quel codice fiscale e tipologia all'interno della tabella Utenti_old. Se presente, cambia i dati attuali dell'utente "passato" alla funzione e viene cambiato con le informazione della tabella Utente_old tranne che per la password, altrimenti esegue la funzione *crea_email()*.

Se la password non è stata inserita, viene generata con la funzione genera_passwrod_random e la mostrerà con una raise notice.

Se la tipologia dell'utente non è studente, docente o segretario solleverà un eccezione.

crea_user_before( )

Funzione e seguita a livello di amministratore del database a casa della istruzione:
`execute 'set session_replication_role = replica';` e `execute 'grant [...] to '|| quote_ident(new.email);`

che permette di disattivare i trigger di quella sessione per evitare di rendere impossibile il reinserimento degli esami dentro la tabella Esami_studenti dalla tabella Esami_studenti_old. Successivamente verranno riattivati i trigger con: `execute 'set session_replication_role = default';`

Viene anche reinserito nella tabella Studente il record della tabella Studente_old.

Queste istruzioni sono eseguite unicamente se esiste un utente dentro Utenti_old con quel codice fiscale e se è un utente. Anche se non è uno studente viene l'utente dentro Utenti_old viene eseguita una "delete" dell'utente_old.

Se non c'è l'utente in Utente_old ed è uno studente, viene creato un record dentro Studenti con dei valori di default.

Il calcolo della matricola viene seguito controllando il massimo tra le due tabelle Studenti e Studenti_old e, il risultato, viene incrementato di 1.

Alla fine viene dato il ruolo allo user del database creato.

Funzione trigger attivata dopo la "insert into Utenti".

check_lode( )

Funzione che controlla unicamente se posso impostare il valore lode al voto dell'esame.

Posso dare la lode unicamente se il voto è 30, altrimenti sollevo un eccezione.

Funzione eseguita prima di una "update esame_studenti".

inserisci_corso_before( )

Controlla se l'anno dell'insegnamento che voglio inserire è lecito o meno controllando il valore massimo che il corso accetta. Esempio: se è una triennale accetta solo anni fino a 3 o Facoltativo (salvato nel database con il valore 0).

Funzione eseguita prima di una "insert into Insegnamento".

update_password( )

Funzione e seguita a livello di amministratore del database a casa della istruzione:

```
execute 'alter user '||quote_ident(new.email)||' with encrypted password '||quote_literal(new.password);
```

Aggiorna la password dello user.

Se lo user selezionato che sta tentando di cambiare password è uno studente o un docente e sta provando a cambiare una password che non è la propria, solleva una eccezione. L'unico autorizzato a fare questo tipo di operazione è il segretario.

Successivamente imposta la password dello user in md5.

Funzione eseguita prima di una “update utente” con password diverso da null.

enera_password_random( )

Funzione che genera una password random con caratteri sia maiuscoli sia minuscoli e la restiuisce.

posta_utente_before( )

Funzione e seguita a livello di amministratore del database a casa della istruzione: `execute 'drop user '||quote_ident(old.email);`

Viene eseguita prima di una “delete from utenti” e sposta gli utenti che sta per cancellare nella rispettive tabelle old con tutti i dati (se presenti) contenute in Studente e Esami_studenti.

Alla fine cancella lo user del database.

sposta_utente_before( )

Funziona che controlla se un determinato studente (stud) può ottenere la laurea.

Per ottenere una laurea bisogna soddisfare questi parametri:

- Avere tutti gli esami obbligatori (quelli in cui l’anno non è 0, ovvero che non siano facoltativi) con esame almeno sufficiente ed accettato.
- Avere almeno tanti CFU quanti anni sono previsti per quel corso di laurea. Sono 60 CFU per anno richiesti. Può darsi che non sia possibile laurearsi con solo gli esami obbligatori, quindi all’utente sarà richiesto di iscriversi anche a corsi facoltavi.

Restituisce true se è possibile ottenere la laurea, altrimenti restituisce false.

visualizza_carriera( )

È una sorta di vista, perché restituisce i dati relativi alla carriera completa dello studente.

Per restituire i dati della carriera, verrà utilizzato il tipo (realizzato per l’occasione) **carriera_type** in cui è contenuto:

- Nome dell’insegnamento.
- Data dell’esame.
- Il voto e con l’eventuale lode.
- Se è stato o meno accettato.
- I CFU della materia.

La funzione può essere eseguita solamente dall'utente postgres oppure se l'utente è un segretario o un docente, in caso negativo solleva un eccezione.

visualizza_carriera_valida( )

Estensione della funzione *visualizza_carrira()* perché restituisce solamente i record in cui il voto è maggiore di 18 ed è stato accettato.

La funzione può essere eseguita solamente dall'utente postgres oppure se l'utente è un segretario o un docente, in caso negativo solleva un eccezione.

prof_responsabile( )

Funzione che verifica se un docente è responsabile al massimo per 3 insegnamenti, se si tenta di inserirne un altro, solleva una eccezione.

Funzione eseguita prima di una "insert into responsabile".

responsabile_before( )

Funziona che controlla se l'utente che sto per inserire è o meno un docente. Se non lo è, solleva un eccezione.

Verifica anche se non sono ancora stati registrati docenti responsabili per la materia. Il primo docente che bisogna inserire è obbligatoriamente un docente responsabile, altrimenti solleva un eccezione.

Funzione eseguita prima di una "insert into responsabile".

controllo_propedeuticità_esame( )

Controlla se uno studente ha o meno il diritto di iscriversi ad un esame. Viene controllato se l'esame possiede delle propedeuticità e cerca se lo studente ha già superato gli esami con esito positivo ed accettato che sono propedeutici a quel determinato esame. Se non è verificato, solleva un eccezione.

Funzione eseguita prima di una "insert into esami_studenti".

Funzione *controllo_propedeuticità* ( )

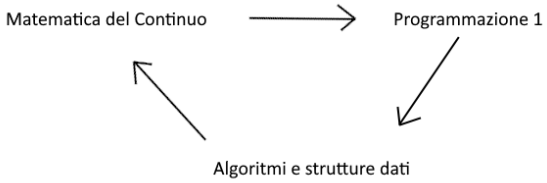
Funzione che controlla se creo o meno di cicli con le propedeuticità se viene aggiunta quella propedeuticità con qui determinati dati. Verrà eseguita una BFS con le varie propedeuticità e quelle nuove che sto per aggiungere e salvo i vari risultati in "array" appositi (non sono propriamente degli array essendo che SQL non

gli possiede come tipo). Se vede che una propedeuticità è già stata esaminata, solleva un eccezione perché altrimenti creerei un ciclo, e risulterebbe impossibile eseguire quei determinati esami.

Esempio:

sopponiamo che Programmazione 1 è dipendente da Matematica del continuo, che Basi di dati sia limitato da Programmazione 1 e che Matematica del continuo sia fattibile sono dopo aver fatto Base di Dati, ovvero così:

In questo caso la funzione restituirà una eccezione perché risulterà infattibile il superamento delle tre materie.



Funzione eseguita prima di una “insert into propedeutico”.

iscrizione_esame ( )

Funzione che controlla se è o meno possibile iscriversi ad un determinato esame. Controlla se la data dell’esame (meno sette giorni) è minore della data corrente. In caso in cui risulta superiore la data odierna, solleva una eccezione.

Il controllo dei giorni meno sette, è dovuto al fatto che le iscrizione per un esame chiudono sette giorni prima dello svolgimento dell’esame, risultando come esame già avvenuto.

Funzione eseguita prima di una “insert into esami_studenti”

inserire_esame ( )

Funzione e seguita a livello di amministratore del database a casa della istruzione: `execute 'set session_replication_role = replica'`; e `execute 'set session_replication_role = default'`

Inserisce un nuovo esame in quella data. Essendo che è possibile che ci siano più responsabili per insegnamento e che ogni docente responsabile si prende a carico solamente alcuni studenti suddivisi per lettera del cognome, verranno aggiunti tanti esami con lettere distinte quanti sono i docenti responsabili nella stessa giornata. Il range di lettere viene calcolato qui e tiene in considerazione solamente il numero di docenti e, di conseguenza, gli studenti verrà divisi in maniera non equa per ogni esame.

Verrà controllato anche se è possibile inserire l’esame in quella specifica data. Non è possibile inserirlo se il giorno è già occupato da un altro esame dello stesso anno della materia in cui si sta aggiungendo l’esame, oppure se non si da un preavviso di almeno 14 giorni.

Funzione eseguita prima di una “insert into esame”

update_esame( )

Funzione e seguita a livello di amministratore del database a casa della istruzione: execute 'set session_replication_role = replica'; e execute 'set session_replication_role = default'

Aggiorna i dati relativi alla data dell'esame. Se sono presenti più esami dello stesso insegnamento, è possibile modificare o la singola ora dell'esame, oppure cambiare la data dell'intero blocco di esami della materia.

Controlla anche se chi sta aggiornando un esame è un docente responsabile di quel corso, se non lo è solleva una eccezione.

Funzione eseguita prima di una "update esame".

cancella_esame( )

Funzione e seguita a livello di amministratore del database a casa della istruzione: execute 'set session_replication_role = replica'; e execute 'set session_replication_role = default'

Cancella in blocco gli esami di quella materia se sono presenti più docenti responsabili.

Controlla anche se chi sta aggiornando un esame è un docente responsabile di quel corso, se non lo è solleva una eccezione.

Funzione eseguita prima di una "delete from esame".

controllo_lettera_esame( )

Funzione che controlla se un utente può iscriversi in quel determinato esame con quel docente. Viene controllato se la prima lettere del suo cognome è compreso tra le lettere che quell'esame con quel docente accetta, se non lo accetta solleva un eccezione.

Funzione eseguita prima di una "insert into esami_studenti".

controllo_lettera_esame( )

Funzione e seguita a livello di amministratore del database a casa del non permesso di accedere alla tabella da parte dell'utilizzatore principale, ovvero lo user *accesso.*

Controlla se esiste un utente con quella password (criptata in md5), se esiste solleva la tipologia dello studente.

È l'unica istruzione che può eseguire *accesso*.

elimina_esame_before( )

Controlla se è possibile o meno cancellare un esame da parte di uno studente.

Se è già stato fatto l'esame, non sarà possibile e solleverà un eccezione.

Funzione eseguita prima di una "insert into esami_studenti".

elimina_esame_before( )

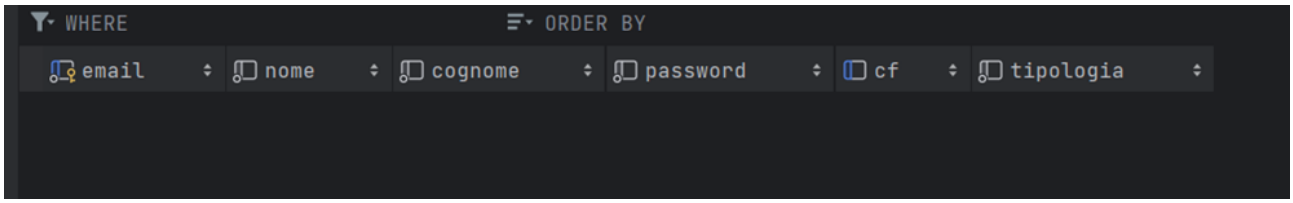
Controlla se il voto che si sta aggiornando all'esito dell'appello di esame sia sufficiente o meno. Se non lo è, imposta automaticamente accettato a false.

Funzione eseguita prima di una "update esami_studenti".

Prova di funzionamento

1) Inserimento di un utente

Tabella Utenti:




```
insert into Utenti(nome, cognome, password, cf, tipologia) values
('Giacomo','Lucca','1234','AJFRFA62M82C675S','studente')
```

Tabella Utenti:

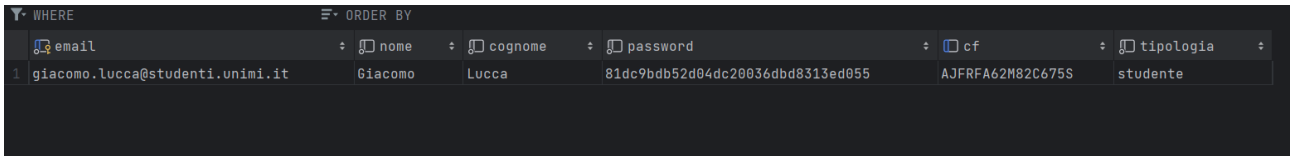
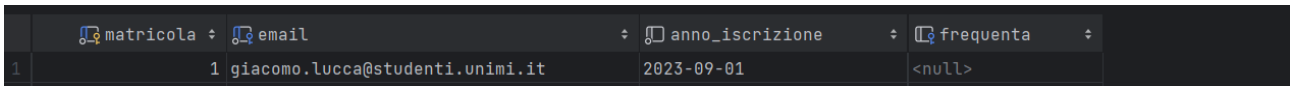


Tabella Studenti:



2) Inserimento di altri due utenti

```
insert into Utenti(nome, cognome, password, cf, tipologia) values
('Giacomo','Lucca','1234','AJFRFT62M82C675S','studente'),
('Giacomo','Lucca','1234','AJQRFT62M82C675S','studente')
```

Tabella Utenti:

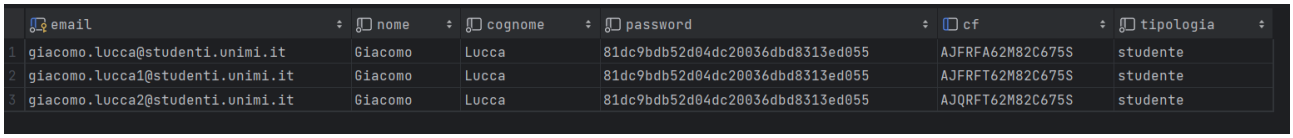
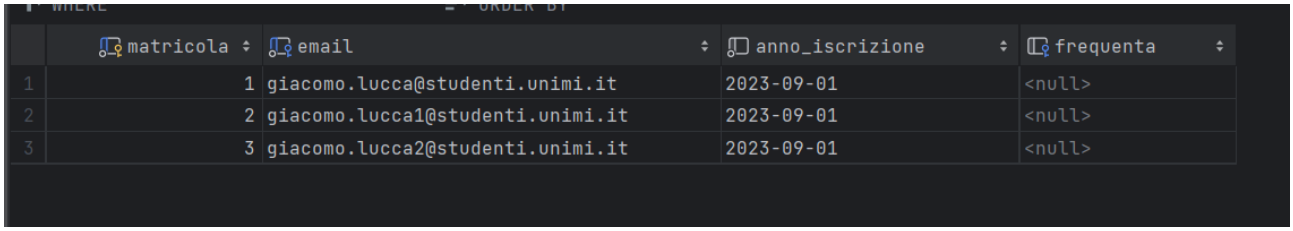


Tabella Studenti:



3) Inserimento di un utente con un utente cancellato (ovvero che risiede in Utente_old e Studente_old).

Tabella Utenti:

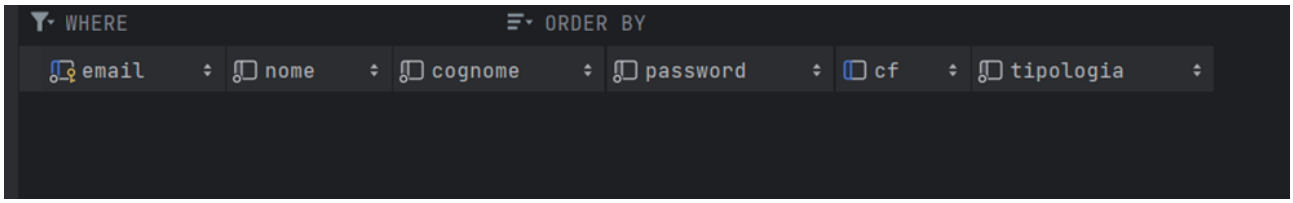
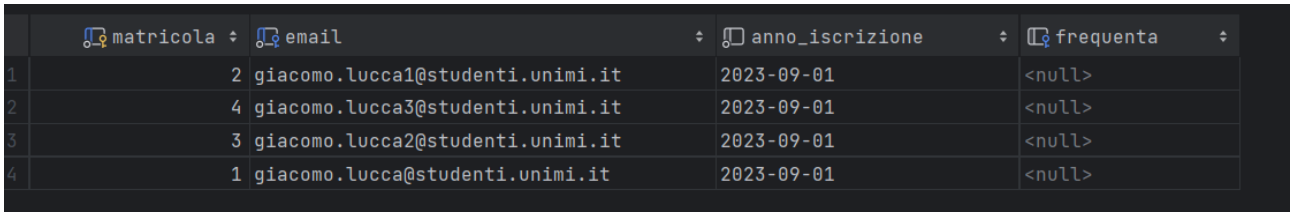


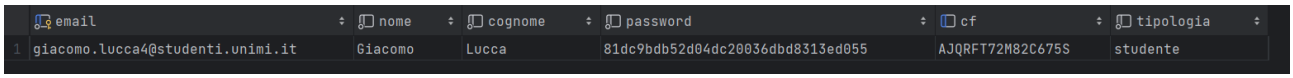
Tabella Studenti_old:



	matricola	email	anno_iscrizione	frequenta
1	2	giacomo.lucca1@studenti.unimi.it	2023-09-01	<null>
2	4	giacomo.lucca3@studenti.unimi.it	2023-09-01	<null>
3	3	giacomo.lucca2@studenti.unimi.it	2023-09-01	<null>
4	1	giacomo.lucca@studenti.unimi.it	2023-09-01	<null>

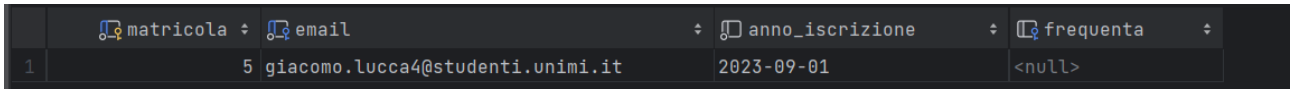
```
insert into Utenti(nome, cognome, password, cf, tipologia) values
('Giacomo','Lucca','1234','AJQRFT72M82C675S','studente')
```

Tabella Utenti:



	email	nome	cognome	password	cf	tipologia
1	giacomo.lucca4@studenti.unimi.it	Giacomo	Lucca	81dc9bdb52d04dc20036dbd8313ed055	AJQRFT72M82C675S	studente

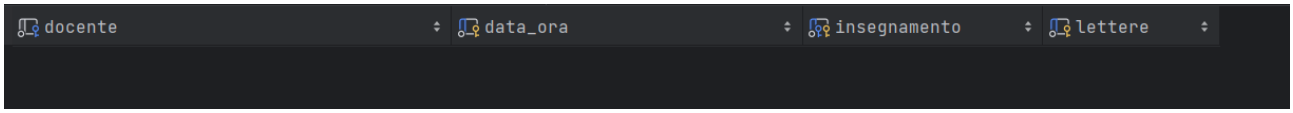
Tabella Studenti:



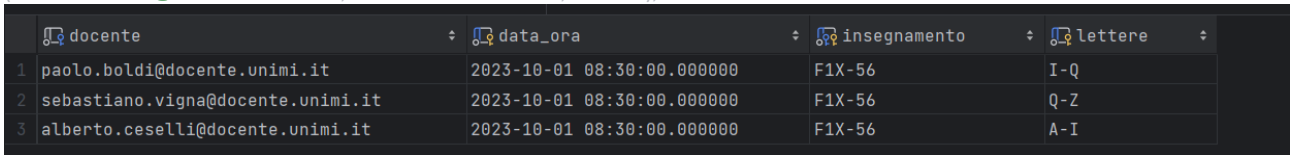
	matricola	email	anno_iscrizione	frequenta
1	5	giacomo.lucca4@studenti.unimi.it	2023-09-01	<null>

4) Inserimento di un esame con più docenti responsabili (3 docenti responsabili) :

Tabella Esame:




```
insert into esame(docente, data_ora, insegnamento) values
('alberto.ceselli@docente.unimi.it','2023-10-01 08:30:00','F1X-56');
```



	docente	data_ora	insegnamento	lettere
1	paolo.boldi@docente.unimi.it	2023-10-01 08:30:00.000000	F1X-56	I-Q
2	sebastiano.vigna@docente.unimi.it	2023-10-01 08:30:00.000000	F1X-56	Q-Z
3	alberto.ceselli@docente.unimi.it	2023-10-01 08:30:00.000000	F1X-56	A-I

5) Cancellazione dell’esame con più docenti responsabili:

```
DELETE FROM public.esame
WHERE data_ora = '2023-10-01 08:30:00.000000' AND insegnamento = 'F1X-56'
```

Tabella Esame:

docente	data_ora	insegnamento	lettere

6) Propedeuticità con ciclo:

Terminale per le query:

```
177 insert into propedeutico(propedeutico, insegnamento) values
178 ( propedeutico: 'F1X-56', insegnamento: 'F1X-59'),
179 ( propedeutico: 'F1X-52', insegnamento: 'F1X-56'),
180 ( propedeutico: 'F1X-59', insegnamento: 'F1X-52');
181
```

```
[P0001] ERRORE: Impossibile inserire la propedeuticità. Risulterà impossibile eseguirlo se si aggiungesse
Dove: funzione PL/pgSQL "controllo_propedeuticità"() riga 34 a RAISE
```

7) Rimozione di uno studente con esami fatti:

Tabella Esami_Studenti:

	studente	esame_insegnamento	esame_lettere	esame_data_ora	voto	lode	accettato
1	2	F1X-56	I-Q	2021-09-01 08:30:00.000000	9	false	false
2	2	F1X-56	I-Q	2022-09-01 08:30:00.000000	29	false	true
3	2	F1X-52	A-Z	2022-09-02 14:30:00.000000	26	false	true
4	2	F1X-77	A-N	2022-09-03 09:30:00.000000	28	false	true
5	2	F1X-95	A-Z	2022-09-04 00:00:00.000000	25	false	true
6	2	F1X-51	A-Z	2022-09-05 14:30:00.000000	25	false	true
7	2	F1X-97	A-Z	2022-09-06 08:30:00.000000	25	false	true
8	2	F1X-94	A-Z	2022-09-07 00:00:00.000000	25	false	true
9	2	F1X-54	A-Z	2022-09-08 00:00:00.000000	25	false	true
10	2	F1X-71	A-Z	2022-09-09 09:30:00.000000	25	false	true
11	2	F1X-114	A-Z	2022-09-10 14:30:00.000000	25	false	true
12	2	F1X-43	A-Z	2022-09-11 00:00:00.000000	25	false	true
13	2	F1X-81	A-Z	2022-09-12 09:30:00.000000	25	false	true
14	2	F9X-35	A-Z	2022-09-13 08:30:00.000000	25	false	true
15	2	F1X-99	A-Z	2022-09-14 00:00:00.000000	25	false	true
16	2	F1X-90	A-Z	2022-09-15 00:00:00.000000	25	false	true
17	2	F1X-67	A-Z	2022-09-16 15:30:00.000000	25	false	true
18	2	F1X-116	A-Z	2022-09-17 13:30:00.000000	25	false	true
19	2	F1X-115	A-Z	2022-09-18 00:00:00.000000	25	false	true
20	2	F1X-59	A-Z	2022-09-19 12:00:00.000000	25	false	true
21	2	F1X-88	A-Z	2022-09-20 00:00:00.000000	25	false	true
22	2	F1X-128	A-Z	2022-09-21 15:00:00.000000	25	false	true
23	2	F1X-125	A-Z	2022-09-22 00:00:00.000000	25	false	true
24	2	F1X-50	A-Z	2022-09-23 00:00:00.000000	25	false	true
25	2	F1X-78	A-Z	2022-09-24 00:00:00.000000	25	false	true
26	2	F1X-122	A-Z	2022-09-25 00:00:00.000000	25	false	true

(in totale ce ne sono 33)

Tabella Studenti:

	matricola	email	anno_iscrizione	frequenta
1	2	giacomo.lucca1@studenti.unimi.it	2023-09-01	Informatica

```
DELETE FROM public.utenti
WHERE email LIKE 'giacomo.lucca1@studenti.unimi.it'
```

Tabella Esami_studenti_old:

	studente	esame_insegnamento	esame_lettere	esame_data_ora	voto	lode	accettato
1	2	F1X-56	I-Q	2021-09-01 08:30:00.000000	9	false	false
2	2	F1X-56	I-Q	2022-09-01 08:30:00.000000	29	false	true
3	2	F1X-52	A-Z	2022-09-02 14:30:00.000000	26	false	true
4	2	F1X-77	A-N	2022-09-03 09:30:00.000000	28	false	true
5	2	F1X-95	A-Z	2022-09-04 00:00:00.000000	25	false	true
6	2	F1X-51	A-Z	2022-09-05 14:30:00.000000	25	false	true
7	2	F1X-97	A-Z	2022-09-06 08:30:00.000000	25	false	true
8	2	F1X-94	A-Z	2022-09-07 00:00:00.000000	25	false	true
9	2	F1X-54	A-Z	2022-09-08 00:00:00.000000	25	false	true
10	2	F1X-71	A-Z	2022-09-09 09:30:00.000000	25	false	true
11	2	F1X-114	A-Z	2022-09-10 14:30:00.000000	25	false	true
12	2	F1X-43	A-Z	2022-09-11 00:00:00.000000	25	false	true
13	2	F1X-81	A-Z	2022-09-12 09:30:00.000000	25	false	true
14	2	F9X-35	A-Z	2022-09-13 08:30:00.000000	25	false	true
15	2	F1X-99	A-Z	2022-09-14 00:00:00.000000	25	false	true
16	2	F1X-90	A-Z	2022-09-15 00:00:00.000000	25	false	true
17	2	F1X-67	A-Z	2022-09-16 15:30:00.000000	25	false	true
18	2	F1X-116	A-Z	2022-09-17 13:30:00.000000	25	false	true
19	2	F1X-115	A-Z	2022-09-18 00:00:00.000000	25	false	true
20	2	F1X-59	A-Z	2022-09-19 12:00:00.000000	25	false	true
21	2	F1X-88	A-Z	2022-09-20 00:00:00.000000	25	false	true
22	2	F1X-128	A-Z	2022-09-21 15:00:00.000000	25	false	true
23	2	F1X-125	A-Z	2022-09-22 00:00:00.000000	25	false	true
24	2	F1X-50	A-Z	2022-09-23 00:00:00.000000	25	false	true
25	2	F1X-78	A-Z	2022-09-24 00:00:00.000000	25	false	true
26	2	F1X-122	A-Z	2022-09-25 00:00:00.000000	25	false	true

(in totale ce ne sono 33)

Tabella Esami_studenti:

studente	esame_insegnamento	esame_lettere	esame_data_ora	voto	lode	accettato

8) Riportare uno studente da Utenti_old a Utenti

Tabella Esami_studenti:

studente	esame_insegnamento	esame_lettere	esame_data_ora	voto	lode	accettato

Tabella Studenti_old:

	matricola	email	anno_iscrizione	frequenta
1	2	giacomo.lucca1@studenti.unimi.it	2023-09-01	Informatica

Tabella Utenti_old:

	email	nome	cognome	cf	tipologia
1	giacomo.lucca1@studenti.unimi.it	Giacomo	Lucca	AJFRFT62M82C675S	studente

Tabella Esami_studenti_old:

	studente	esame_insegnamento	esame_lettere	esame_data_ora	voto	lode	accettato
1	2	F1X-56	I-Q	2021-09-01 08:30:00.000000	9	false	false
2	2	F1X-56	I-Q	2022-09-01 08:30:00.000000	29	false	true
3	2	F1X-52	A-Z	2022-09-02 14:30:00.000000	26	false	true
4	2	F1X-77	A-N	2022-09-03 09:30:00.000000	28	false	true
5	2	F1X-95	A-Z	2022-09-04 00:00:00.000000	25	false	true
6	2	F1X-51	A-Z	2022-09-05 14:30:00.000000	25	false	true
7	2	F1X-97	A-Z	2022-09-06 08:30:00.000000	25	false	true
8	2	F1X-94	A-Z	2022-09-07 00:00:00.000000	25	false	true
9	2	F1X-54	A-Z	2022-09-08 00:00:00.000000	25	false	true
10	2	F1X-71	A-Z	2022-09-09 09:30:00.000000	25	false	true
11	2	F1X-114	A-Z	2022-09-10 14:30:00.000000	25	false	true
12	2	F1X-43	A-Z	2022-09-11 00:00:00.000000	25	false	true
13	2	F1X-81	A-Z	2022-09-12 09:30:00.000000	25	false	true
14	2	F9X-35	A-Z	2022-09-13 08:30:00.000000	25	false	true
15	2	F1X-99	A-Z	2022-09-14 00:00:00.000000	25	false	true
16	2	F1X-90	A-Z	2022-09-15 00:00:00.000000	25	false	true
17	2	F1X-67	A-Z	2022-09-16 15:30:00.000000	25	false	true
18	2	F1X-116	A-Z	2022-09-17 13:30:00.000000	25	false	true
19	2	F1X-115	A-Z	2022-09-18 00:00:00.000000	25	false	true
20	2	F1X-59	A-Z	2022-09-19 12:00:00.000000	25	false	true
21	2	F1X-88	A-Z	2022-09-20 00:00:00.000000	25	false	true
22	2	F1X-128	A-Z	2022-09-21 15:00:00.000000	25	false	true
23	2	F1X-125	A-Z	2022-09-22 00:00:00.000000	25	false	true
24	2	F1X-50	A-Z	2022-09-23 00:00:00.000000	25	false	true
25	2	F1X-78	A-Z	2022-09-24 00:00:00.000000	25	false	true
26	2	F1X-122	A-Z	2022-09-25 00:00:00.000000	25	false	true

```
insert into Utenti(nome, cognome, password, cf, tipologia) values
('Giacomo','Lucca','1234','AJFRFT62M82C675S','studente');
```

Tabella Esami_studenti:

	studente	esame_insegnamento	esame_lettere	esame_data_ora	voto	lode	accettato
1	2	F1X-56	I-Q	2021-09-01 08:30:00.000000	9	false	false
2	2	F1X-56	I-Q	2022-09-01 08:30:00.000000	29	false	true
3	2	F1X-52	A-Z	2022-09-02 14:30:00.000000	26	false	true
4	2	F1X-77	A-N	2022-09-03 09:30:00.000000	28	false	true
5	2	F1X-95	A-Z	2022-09-04 00:00:00.000000	25	false	true
6	2	F1X-51	A-Z	2022-09-05 14:30:00.000000	25	false	true
7	2	F1X-97	A-Z	2022-09-06 08:30:00.000000	25	false	true
8	2	F1X-94	A-Z	2022-09-07 00:00:00.000000	25	false	true
9	2	F1X-54	A-Z	2022-09-08 00:00:00.000000	25	false	true
10	2	F1X-71	A-Z	2022-09-09 09:30:00.000000	25	false	true
11	2	F1X-114	A-Z	2022-09-10 14:30:00.000000	25	false	true
12	2	F1X-43	A-Z	2022-09-11 00:00:00.000000	25	false	true
13	2	F1X-81	A-Z	2022-09-12 09:30:00.000000	25	false	true
14	2	F9X-35	A-Z	2022-09-13 08:30:00.000000	25	false	true
15	2	F1X-99	A-Z	2022-09-14 00:00:00.000000	25	false	true
16	2	F1X-90	A-Z	2022-09-15 00:00:00.000000	25	false	true
17	2	F1X-67	A-Z	2022-09-16 15:30:00.000000	25	false	true
18	2	F1X-116	A-Z	2022-09-17 13:30:00.000000	25	false	true
19	2	F1X-115	A-Z	2022-09-18 00:00:00.000000	25	false	true
20	2	F1X-59	A-Z	2022-09-19 12:00:00.000000	25	false	true
21	2	F1X-88	A-Z	2022-09-20 00:00:00.000000	25	false	true
22	2	F1X-128	A-Z	2022-09-21 15:00:00.000000	25	false	true
23	2	F1X-125	A-Z	2022-09-22 00:00:00.000000	25	false	true
24	2	F1X-50	A-Z	2022-09-23 00:00:00.000000	25	false	true
25	2	F1X-78	A-Z	2022-09-24 00:00:00.000000	25	false	true
26	2	F1X-122	A-Z	2022-09-25 00:00:00.000000	25	false	true

Tabella Utenti:

	email	nome	cognome	password	cf	tipologia
1	giacomo.lucca1@studenti.unimi.it	Giacomo	Lucca	81dc9bdb52d04dc20036dbd8313ed055	AJFRFT62M82C675S	studente

9) Iscrizione in un appello che non è il proprio