Progetto DB 1314

Argomento proposto

Si vuole realizzare un'applicazione database per gestire le informazioni relative all'utilizzo dei locali e delle risorse di un'Università da parte degli studenti dei Corsi di Laurea e del personale, docente e non docente, dell'Università stessa. Gli accessi ad alcuni locali dell'Università sono riservati a persone autorizzate in base alle seguenti modalità. Un utente può avere per un locale un solo permesso. Ogni permesso appartiene ad una certa tipologia che specifica tra l'altro, per ciascuno dei sei giorni lavorativi, gli intervalli temporali di accesso. Gli accessi sono controllati e rilevati tramite la lettura di una tessera magnetica assegnata a ciascun utente. Vi sono alcuni locali per cui bisogna tenere traccia di tutti gli accessi, rilevando per ogni utente: data, ora e identificazione dell'utente. Per altri locali bisogna tenere traccia solo del numero di accessi giornalieri. Tutte le volte che il sistema rifiuta l'accesso ad un locale, si deve tenere traccia dell'evento, memorizzando tutte le informazioni relative. In particolare, interessa evidenziare i casi in cui per uno stesso utente e per lo stesso locale ci siano stati più di tre rifiuti giornalieri. Tra i locali dell'Università vi sono dei laboratori didattici che contengono un insieme di posti di lavoro ed un insieme di risorse. Ad ogni posto di lavoro sono assegnate alcune risorse (quali ad esempio, unità di calcolo, stampanti, software applicativo). Alcuni posti di lavoro sono resi disponibili a tutti gli studenti senza controlli, altri vengono mensilmente assegnati a particolari studenti, quali ad esempio laureandi, previa autorizzazione di un docente. Infine, in un laboratorio vi possono essere un certo numero di posti di lavoro prenotabili giornalmente da parte di un singolo studente oppure da parte di un docente titolare di un insegnamento; per questi posti di lavoro si deve tenere traccia di tutte le prenotazioni e di tutti le utilizzazioni da parte degli studenti. Per le prenotazioni devono essere garantite le seguenti regole di non sovrapposizione:

1. in una certa ora, un posto può essere prenotato da un solo studente;

2. in una certa ora, uno studente non può avere la prenotazione per due posti differenti.

Ogni laboratorio ha come responsabile organizzativo un docente e come responsabile operativo un tecnico dell'Università. Alcune unità di calcolo richiedono un account per accedervi; gli account vengono rilasciati automaticamente, su un insieme di server, agli studenti sulla base della loro

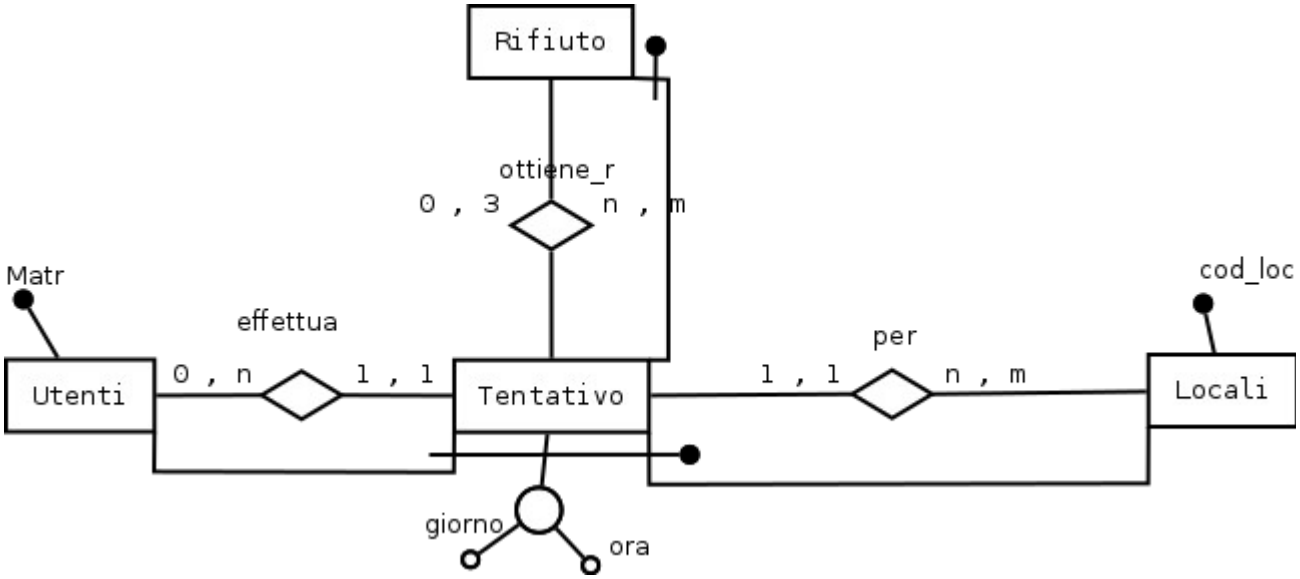
iscrizione ad un corso di laurea e all’anno di iscrizione. A ciascuno dei gruppi così individuato il System Administrator assegna e gestisce determinati privilegi e risorse. Per ogni account viene gestita la configurazione del profilo utente, memorizzando informazioni quali lo “shell” e gli applicativi utilizzati.

Schema ER

Diviso in sezioni

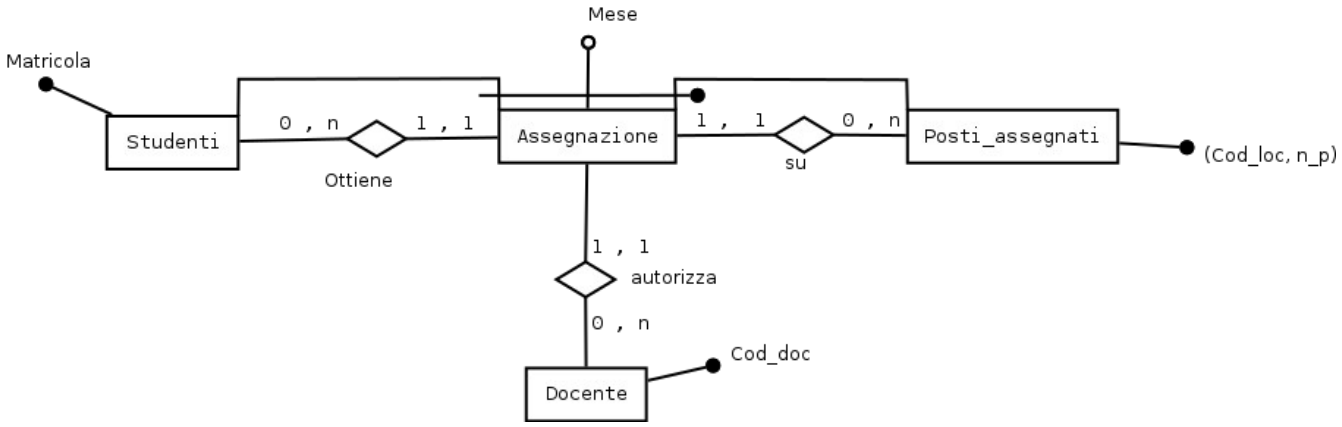
- Accessi-rifiuti

Ho creato un Entità supporto “Rifiuto” per mantenere ogni tentativo non andato a buon fine dell'utente di accedere a un determinato locale, qualora questa tabella avrà più di 3 record con gli stessi valori “Matr, cod_loc, giorno” verrà segnalato grazie a un trigger in una colonna dell'Utente



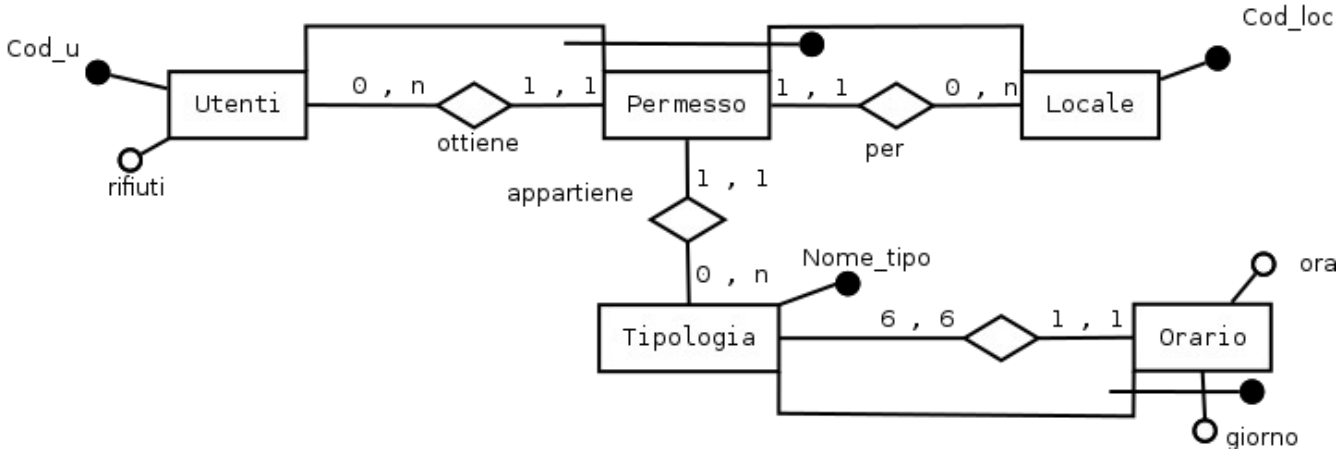
- Assegnazione

L'assegnazione di un posto avviene mensilmente e autorizzata da un docente



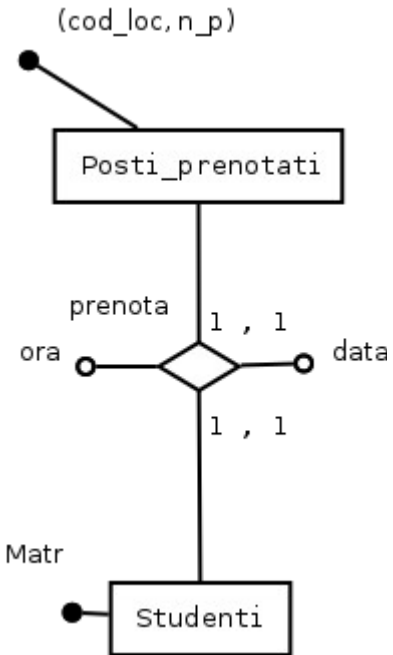
- Permesso

Un utente può avere un solo permesso per un dato Locale, il permesso appartiene a una certa tipologia che ne specifica, tra l'altro, l'orario



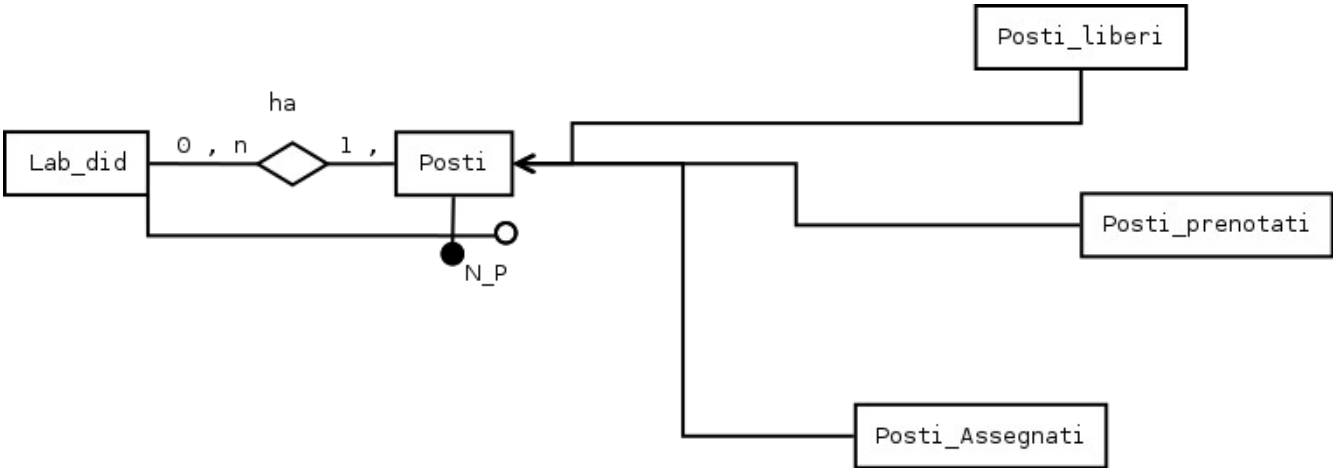
- Prenotazioni

In questo modo uno stupente può prenotare un posto in una data a una certa ora



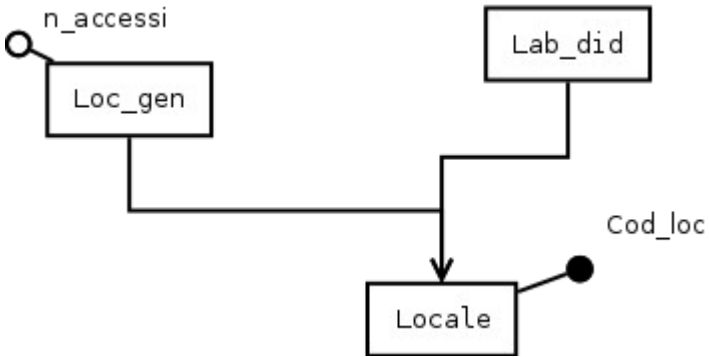
- Posti

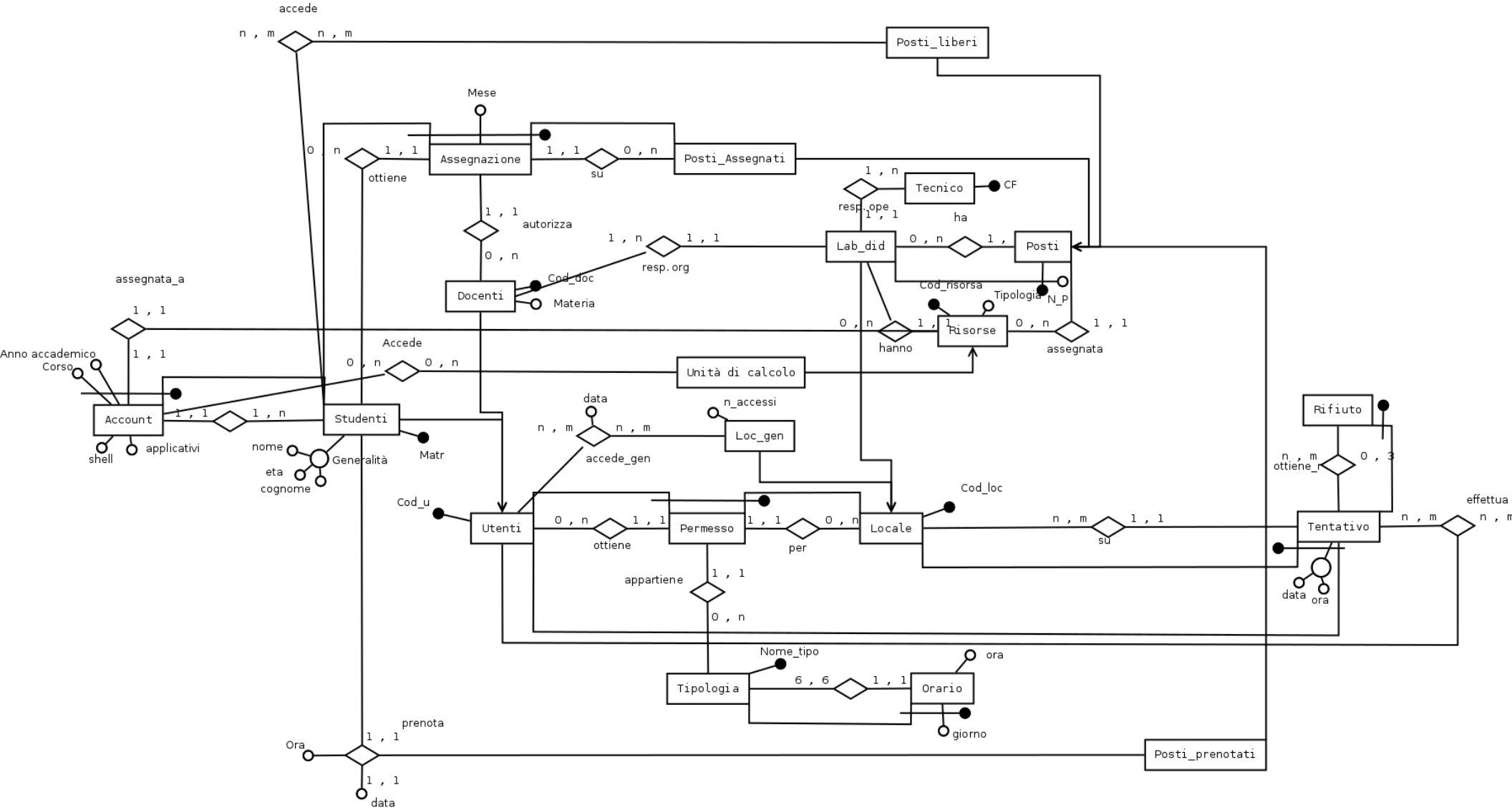
ho deciso di creare tre entità specifiche: posti liberi, prenotati e assegnati, i primi non avranno alcun controllo per il loro uso, ai secondi vi potranno accedere solo coloro che hanno prenotato e ai secondi solo coloro che hanno l'assegnazione



- Locali

Ho creato due entita Locale generico e laboratorio didattico per consentire l'accesso senza permesso ai primi e associare i posti ai secondi



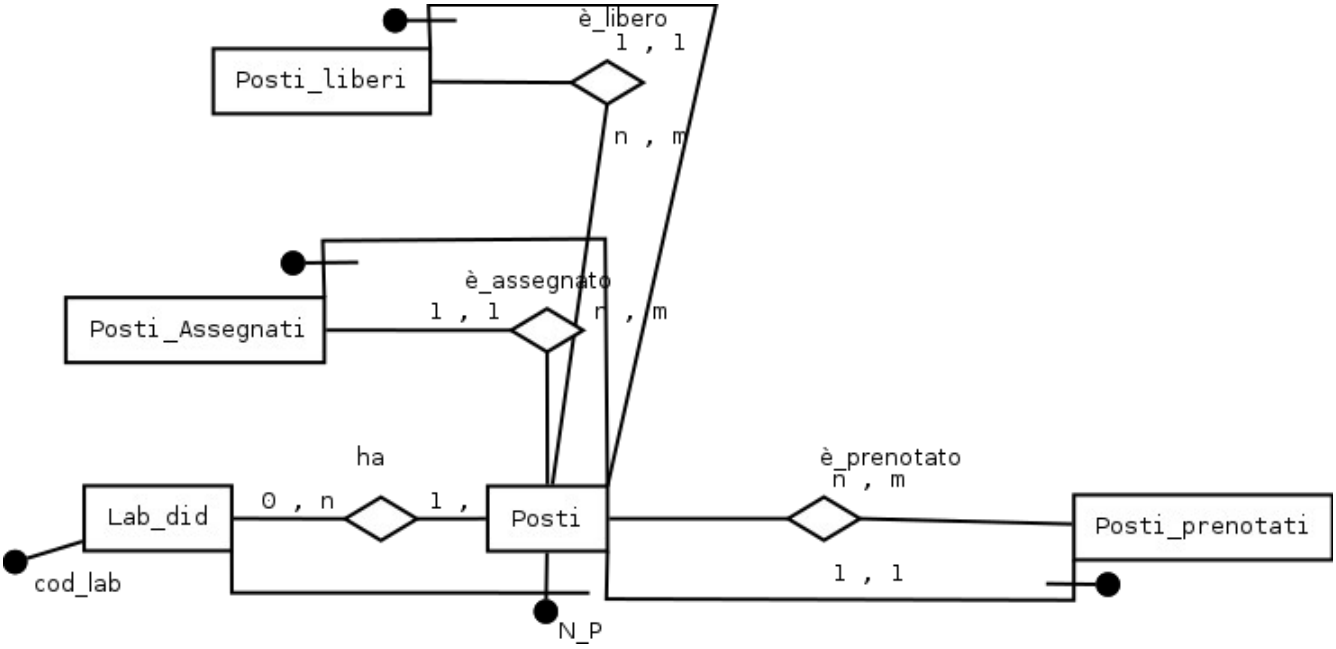

accede
n , m
n , m
Posti_liberi
Mese
0 , n
1 , 1
Assegnazione
1 , 1
su
0 , n
Posti_Assegnati
ottiene
1 , n
Tecnico
CF
resp. ope
1 , 1
ha
1 , 1
autorizza
0 , n
1 , n
1 , 1
resp. org
Lab_did
0 , n
1 ,
Posti
assegnata_a
Cod_doc
Docenti
Materia
Cod_risorsa
Tipologia
N_P
1 , 1
0 , n
1 , 1
Risorse
0 , n
1 , 1
hanno
assegnata
Anno accademico
Corso
Accede
0 , n
0 , n
Unità di calcolo
Account
1 , 1
1 , n
Studenti
shell
applicativi
nome
eta
cognome
Generalità
Matr
data
n_accessi
n , m
n , m
Loc_gen
accede_gen
Rifiuto
n , m
ottiene_r
0 , 3
Cod_u
Utenti
0 , n
1 , 1
ottiene
Permesso
1 , 1
0 , n
per
Locale
Cod_loc
n , m
1 , 1
su
Tentativo
n , m
n , m
effettua
data
ora
appartiene
1 , 1
0 , n
Nome_tipo
ora
Tipologia
6 , 6
1 , 1
Orario
giorno
prenota
Ora
1 , 1
1 , 1
data
Posti_prenotati

Progetto logico

Normalizzazzioni

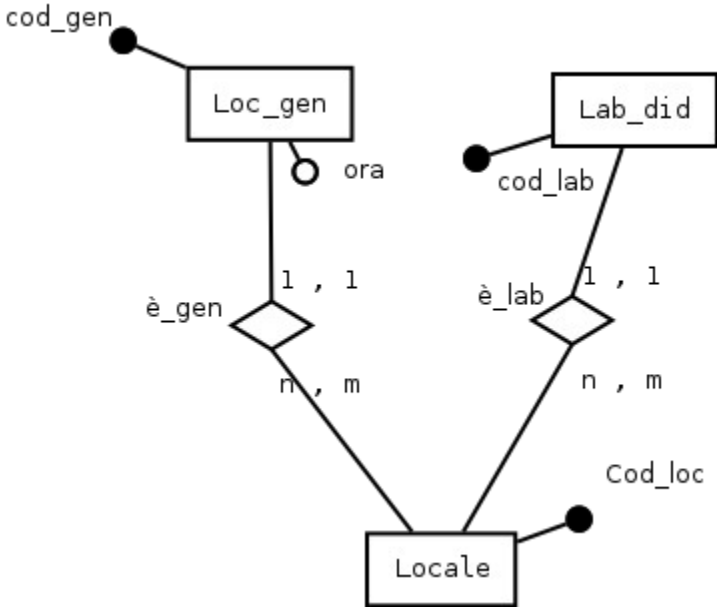
- Posti

Ho collassato verso il basso Posti liberi assegnati e prenotati di modo da poterli usare singolarmente, ho però mantenuto la chiave esterna di posti



- Locali

Anche in locali ho collassato verso il basso creando loc_gen e lab_did, entrambe con chiave proprietaria.



Schema logico

Studenti (Matr, Nome, Cognome, Eta)

Account (shell, applicativi, Matr, Corso, Anno Accademico)

    FK: Matr **reference** Studenti

Utenti (cod_u, rifiuti)

in_studenti (Matr, cod_u)

    FK: Matr **reference** Studenti
    FK: cod_u **reference** Utenti

Docenti (cod_doc, Materia)

in_docenti (cod_u, cod_doc)

    FK: cod_u **reference** Utenti
    FK: cod_doc **reference** Docenti

Assegnazione (Mese, N_P, cod_lab, Matr)

    FK: Matr **reference** Studenti
    FK: N_P, cod_lab **reference** Posti_Assegnati

autorizza (cod_doc, Matr,Mese,N_P,cod_lab)

    FK: cod_doc **reference** Docenti
    FK: Matr, Mese, N_P, cod_lab **reference** Assegnazione

Posti_Assegnati (N_P, cod_lab)

    FK: N_P, cod_lab **reference** Posti

Posti (N_P, cod_lab)

Posti_prenotati (N_P, cod_lab)

    FK: N_P, cod_lab **reference** Posti

prenota (ora, Data , N_P, cod_lab, Matr)

    FK: Nome_giorno **reference** Giorno
    FK: Matr **reference** Studenti
    FK: N_P, cod_lab **reference** Posti_prenotati

Posti_liberi (N_P, cod_lab)

    FK: N_P, cod_lab **reference** Posti

accede_liberamente(N_P, cod_lab, Matr)

    FK: N_P, cod_lab **reference** Posti_liberi
    FK: Matr **reference** Studenti

Lab_did (cod_lab)

resp.org (cod_lab, cod_doc)

    FK: cod_lab **reference** Lab_did
    FK: cod_doc **reference** Docenti

resp.ope (cod_lab, CF)

    FK: cod_lab **reference** Lab_did
    FK: CF **reference** Tecnico

Tecnico(CF)

assegnata (N_P,cod_lab, Cod_risorsa)

    FK: N_P, cod_lab **reference** Posti
    FK: Cod_risorsa **reference** Risorse

Risorse (Cod_risorsa, Tipologia)
Unita di calcolo (Cod_risorsa)

    FK: Cod_risorsa **reference** Risorse

Accede (Corso, anno accademico, Matr, Cod_risorsa)

    FK: Corso, anno accademico, Matr **reference** Account
    FK: Cod_risorsa **reference** Unita di calcolo

System Admin (id)

SA_assegna(id, Cod_risorsa, Matr, anno accademico, corso)

    FK: id **reference** System Admin
    FK: Cod_risorsa **reference** Risorse
    FK: anno accademico, corso, Matr **reference** Account

è_lab (cod_lab, cod_loc)

    FK: cod_lab **reference** Lab_did
    FK: cod_loc **reference** Locale

Permesso (Cod_u, cod_loc)

    FK: Utenti **reference** cod_u
    FK: cod_loc **reference** Locale

Tipologia (Nome_tipo)

appartiene(Nome_tipo,cod_loc,cod_u)

    FK: Nome_tipo **reference** Tipologia
    FK: cod_loc, cod_u **reference** Permesso

Orario (Nome_tipo, giorno, ora)

    FK: Nome_tipo **reference** Tipologia

Locale(cod_loc)

è_gen (cod_gen, cod_loc)

    FK: cod_loc **reference** Locale
    FK: cod_gen **reference** Loc_gen

Log_gen (cod_gen, n_accessi)

accede_gen (cod_gen, cod_u, data)

    FK: cod_gen **reference** Loc_gen
    FK: cod_u **reference** Utenti

Tentativo (cod_loc, cod_u , data, ora)

    FK: cod_loc **reference** Locale
    FK: cod_u **reference** Utenti

Rifiuto (cod_loc, cod_u, data)

    FK: cod_loc **reference** Locale
    FK: cod_u **reference** Utenti

Codice per la creazione delle tabelle

```
CREATE TABLE studenti
  (
     nome    CHAR(20),
     cognome CHAR(20),
     eta     INTEGER,
     matr    CHAR(30) UNIQUE NOT NULL,
     PRIMARY KEY (matr),
     CHECK ( eta >= '18' )
  );

CREATE TABLE sa_account
  (
     shell           TEXT,
     applicativi     TEXT,
     corso           CHAR(20) NOT NULL,
     anno_accademico INTEGER NOT NULL,
     matr            CHAR(30) NOT NULL,
     PRIMARY KEY (corso, anno_accademico, matr),
     FOREIGN KEY (matr) referenceENCES studenti(matr)
  );

CREATE TABLE utenti
  (
     rifiuti INTEGER,
     cod_u   CHAR(20) NOT NULL,
     PRIMARY KEY (cod_u)
  );

CREATE TABLE in_studenti
  (
     matr  CHAR(30) NOT NULL,
     cod_u CHAR(20) NOT NULL,
     FOREIGN KEY (matr) referenceENCES studenti(matr),
     FOREIGN KEY (cod_u) referenceENCES utenti(cod_u)
  );

CREATE TABLE docenti
  (
     nome    CHAR(20),
     cognome CHAR(20),
     materia CHAR(20),
     eta     INTEGER,
     cod_doc CHAR(30) NOT NULL,
     PRIMARY KEY (cod_doc)
  );

CREATE TABLE in_docenti
  (
     cod_u   CHAR(20) NOT NULL,
     cod_doc CHAR(30) NOT NULL,
     FOREIGN KEY (cod_u) referenceENCES utenti(cod_u),
     FOREIGN KEY (cod_doc) referenceENCES docenti(cod_doc)
```

```
  );

CREATE TABLE lab_did
  (
     cod_lab CHAR(20) NOT NULL,
     PRIMARY KEY (cod_lab)
  );

CREATE TABLE posti
  (
     n_p     INTEGER NOT NULL,
     cod_lab CHAR(20) NOT NULL,
     PRIMARY KEY (n_p, cod_lab),
     FOREIGN KEY (cod_lab) referenceENCES lab_did(cod_lab)
  );

CREATE TABLE posti_assegnati
  (
     n_p     INTEGER NOT NULL,
     cod_lab CHAR(20) NOT NULL,
     PRIMARY KEY (n_p, cod_lab),
     FOREIGN KEY (n_p, cod_lab) referenceENCES posti(n_p, cod_lab)
  );

CREATE TABLE assegnazione
  (
     mese    CHAR(20) NOT NULL,
     matr    CHAR(30) NOT NULL,
     cod_lab CHAR(20) NOT NULL,
     n_p     INTEGER NOT NULL,
     PRIMARY KEY (mese, matr, n_p, cod_lab),
     FOREIGN KEY (matr) referenceENCES studenti(matr),
     FOREIGN KEY (n_p, cod_lab) referenceENCES posti_assegnati(n_p, cod_lab),
     CHECK ( mese = 'gennaio' OR mese = 'febbraio' OR mese = 'marzo' OR mese =
     'aprile' OR mese = 'maggio' OR mese = 'giugno' OR mese = 'luglio' OR mese =
     'agosto' OR mese = 'settembre' OR mese = 'ottobre' OR mese = 'novembre' OR
     mese = 'dicembre')
  );

CREATE TABLE autorizza
  (
     cod_doc CHAR(30) NOT NULL,
     cod_lab CHAR(20) NOT NULL,
     matr    CHAR(30) NOT NULL,
     mese    CHAR(20) NOT NULL,
     n_p     INTEGER NOT NULL,
     PRIMARY KEY (matr, mese, n_p, cod_lab, cod_doc),
     FOREIGN KEY (cod_doc) referenceENCES docenti(cod_doc),
     FOREIGN KEY (matr, mese, n_p, cod_lab) referenceENCES assegnazione(matr,
mese,
     n_p, cod_lab)
  );

CREATE TABLE posti_prenotati
  (
     n_p     INTEGER NOT NULL,
     cod_lab CHAR(20) NOT NULL,
     PRIMARY KEY (n_p, cod_lab),
```

```
    FOREIGN KEY (n_p, cod_lab) referenceENCES posti(n_p, cod_lab)
  );

CREATE TABLE prenota
  (
    ora     INTEGER NOT NULL,
    n_p     INTEGER NOT NULL,
    cod_lab CHAR(20) NOT NULL,
    matr    CHAR(30) NOT NULL,
    data    DATE NOT NULL,
    PRIMARY KEY (data, matr, n_p, cod_lab),
    FOREIGN KEY (matr) referenceENCES studenti(matr),
    FOREIGN KEY (n_p, cod_lab) referenceENCES posti_prenotati(n_p, cod_lab),
    CHECK (ora >= 6 AND ora<=20)
  );

CREATE TABLE posti_liberi
  (
    n_p     INTEGER NOT NULL,
    cod_lab CHAR(20) NOT NULL,
    PRIMARY KEY (n_p, cod_lab),
    FOREIGN KEY (n_p, cod_lab) referenceENCES posti(n_p, cod_lab)
  );

CREATE TABLE accede_liberamente
  (
    n_p     INTEGER NOT NULL,
    matr    CHAR(30) NOT NULL,
    cod_lab CHAR(20) NOT NULL,
    PRIMARY KEY (n_p, cod_lab, matr),
    FOREIGN KEY (n_p, cod_lab) referenceENCES posti_liberi(n_p, cod_lab),
    FOREIGN KEY (matr) referenceENCES studenti(matr)
  );

CREATE TABLE resp_org
  (
    cod_lab CHAR(20) NOT NULL,
    cod_doc CHAR(30) NOT NULL,
    PRIMARY KEY (cod_doc, cod_lab),
    FOREIGN KEY (cod_lab) referenceENCES lab_did(cod_lab),
    FOREIGN KEY (cod_doc) referenceENCES docenti(cod_doc)
  );

CREATE TABLE tecnico
  (
    cognome CHAR(20),
    nome    CHAR(20),
    eta     INTEGER,
    cf      CHAR(20) NOT NULL,
    PRIMARY KEY (cf)
  );

CREATE TABLE resp_ope
  (
    cod_lab CHAR(20) NOT NULL,
    cf      CHAR(20) NOT NULL,
    PRIMARY KEY (cf, cod_lab),
    FOREIGN KEY (cod_lab) referenceENCES lab_did(cod_lab),
```

```
    FOREIGN KEY (cf) referenceENCES tecnico(cf)
  );

CREATE TABLE risorsa
  (
    tipologia   CHAR(20),
    cod_risorsa CHAR(20) NOT NULL,
    PRIMARY KEY (cod_risorsa)
  );

CREATE TABLE assegnata
  (
    n_p         INTEGER NOT NULL,
    cod_lab     CHAR(20) NOT NULL,
    cod_risorsa CHAR(20) NOT NULL,
    PRIMARY KEY (cod_risorsa, n_p, cod_lab),
    FOREIGN KEY (n_p, cod_lab) referenceENCES posti(n_p, cod_lab),
    FOREIGN KEY (cod_risorsa) referenceENCES risorsa(cod_risorsa)
  );

CREATE TABLE unita_di_calcolo
  (
    cod_risorsa CHAR(20) NOT NULL,
    PRIMARY KEY (cod_risorsa),
    FOREIGN KEY (cod_risorsa) referenceENCES risorsa(cod_risorsa)
  );

CREATE TABLE accede
  (
    corso           CHAR(20) NOT NULL,
    cod_risorsa     CHAR(20) NOT NULL,
    matr            CHAR(30) NOT NULL,
    anno_accademico INTEGER NOT NULL,
    PRIMARY KEY (corso, anno_accademico, matr, cod_risorsa),
    FOREIGN KEY (corso, anno_accademico, matr) referenceENCES sa_account(corso,
    anno_accademico, matr),
    FOREIGN KEY (cod_risorsa) referenceENCES unita_di_calcolo(cod_risorsa)
  );

CREATE TABLE assegnata_a
  (
    matr            CHAR(30) NOT NULL,
    cod_risorsa     CHAR(20) NOT NULL,
    anno_accademico INTEGER NOT NULL,
    corso           CHAR(20) NOT NULL,
    PRIMARY KEY (corso, anno_accademico, matr, cod_risorsa),
    FOREIGN KEY (cod_risorsa) referenceENCES risorsa(cod_risorsa),
    FOREIGN KEY (anno_accademico, corso, matr) referenceENCES sa_account(
    anno_accademico, corso, matr)
  );

CREATE TABLE locale
  (
    cod_loc CHAR(20) NOT NULL,
    PRIMARY KEY (cod_loc)
  );

CREATE TABLE loc_gen
```

```
  (
     cod_gen   CHAR(20) NOT NULL,
     n_accessi INTEGER,
     PRIMARY KEY (cod_gen)
  );

CREATE TABLE accende_gen
  (
     cod_gen CHAR(20) NOT NULL,
     cod_u   CHAR(20) NOT NULL,
     data    DATE NOT NULL,
     PRIMARY KEY (cod_gen, cod_u),
     FOREIGN KEY (cod_gen) referenceENCES loc_gen(cod_gen),
     FOREIGN KEY (cod_u) referenceENCES utenti(cod_u)
  );

CREATE TABLE is_gen
  (
     cod_gen CHAR(20) NOT NULL,
     cod_loc CHAR(20) NOT NULL,
     PRIMARY KEY (cod_gen, cod_loc),
     FOREIGN KEY (cod_gen) referenceENCES loc_gen(cod_gen),
     FOREIGN KEY (cod_loc) referenceENCES locale(cod_loc)
  );

CREATE TABLE is_lab
  (
     cod_lab CHAR(20) NOT NULL,
     cod_loc CHAR(20) NOT NULL,
     PRIMARY KEY (cod_lab, cod_loc),
     FOREIGN KEY (cod_lab) referenceENCES lab_did(cod_lab),
     FOREIGN KEY (cod_loc) referenceENCES locale(cod_loc)
  );

CREATE TABLE permesso
  (
     cod_u   CHAR(20) NOT NULL,
     cod_loc CHAR(20) NOT NULL,
     PRIMARY KEY (cod_u, cod_loc),
     FOREIGN KEY (cod_u) referenceENCES utenti(cod_u),
     FOREIGN KEY (cod_loc) referenceENCES locale(cod_loc)
  );

CREATE TABLE tipologia
  (
     nome_tipo CHAR(20) NOT NULL,
     PRIMARY KEY (nome_tipo)
  );

CREATE TABLE appartiene
  (
     nome_tipo CHAR(20) NOT NULL,
     cod_u     CHAR(20) NOT NULL,
     cod_loc   CHAR(20) NOT NULL,
     PRIMARY KEY (nome_tipo, cod_loc, cod_u),
     FOREIGN KEY (nome_tipo) referenceENCES tipologia(nome_tipo),
     FOREIGN KEY (cod_loc, cod_u) referenceENCES permesso(cod_loc, cod_u)
  );
```

```
CREATE TABLE orario
  (
     giorno    CHAR(3) NOT NULL,
     nome_tipo CHAR(3) NOT NULL,
     ora       INTEGER NOT NULL,
     PRIMARY KEY (nome_tipo),
     FOREIGN KEY (nome_tipo) referenceENCES tipologia(nome_tipo)
  );

CREATE TABLE tentativo
  (
     data    DATE NOT NULL,
     ora     TIME NOT NULL,
     cod_loc CHAR(20) NOT NULL,
     cod_u   CHAR(30) NOT NULL,
     PRIMARY KEY (cod_u, cod_loc, data, ora),
     FOREIGN KEY (cod_loc) referenceENCES locale(cod_loc),
     FOREIGN KEY (cod_u) referenceENCES utenti(cod_u)
  );

CREATE TABLE rifiuti
  (
     data    DATE NOT NULL,
     cod_loc CHAR(20) NOT NULL,
     cod_u   CHAR(30) NOT NULL,
     PRIMARY KEY (cod_u, cod_loc, data),
     FOREIGN KEY (cod_loc) referenceENCES locale(cod_loc),
     FOREIGN KEY (cod_u) referenceENCES utenti(cod_u)
  );
```

Popolamento delle tabelle

```
--Studenti
INSERT INTO studenti
VALUES      ('Matteo',
             'Triggiani',
             '25',
             'MatrMatteoTriggiani25'),
            ('Marcello',
             'Fregni',
             '20',
             'MatrMarcelloFregni20'),
            ('Francesco',
             'Bianchi',
             '23',
             'MatreferencerancescoBianchi23'),
            ('Michela',
             'Rossi',
             '20',
```

```
             'MatrMichelaRossi20'),
            ('Vincenzo',
             'Neri',
             '21',
             'MatrVincenzoNeri21'),
            ('Giulia',
             'Furia',
             '25',
             'MatrGiuliaFuria25'),
            ('Ilaria',
             'Condizionata',
             '24',
             'MatrIlariaCondizionata24'),
            ('Rezza',
             'Capa',
             '20',
             'MatrRezzaCapa20'),
            ('Fabrizio',
             'Cherubini',
             '23',
             'MatreferenceabrizioCherubini23'),
            ('Marco',
             'Giudici',
             '21',
             'MatrMarcoGiudici21'),
            ('Massimo',
             'Della Pena',
             '22',
             'MatrMassimoDellaPena22'),
            ('Giuditta',
             'Penna',
             '23',
             'MatrGiudittaPenna23'),
            ('Filippo',
             'Verucchi',
             '26',
             'MatreferenceilippoVerucchi26'),
            ('Franco',
             'Neri',
             '20',
             'MatreferencerancoNeri20'),
            ('Lucia',
             'Melella',
             '25',
             'MatrLuciaMelella25');

--Docenti
INSERT INTO docenti
```

```
VALUES      ('Mario',
             'Rossi',
             'Storia Romana',
             '49',
             'cod-docMarioRossi49'),
            ('Paolo',
             'Notai',
             'Filosofia',
             '44',
             'cod-docPaoloNotai44'),
            ('Giovanna',
             'Lapico',
             'Finanza',
             '59',
             'cod-docGiovannaLapico59'),
            ('Vincenzo',
             'Di Matteo',
             'Management Aziendale',
             '52',
             'cod-docVincenzoDiMatteo52'),
            ('Venezia',
             'Ragazzi',
             'Arte Moderna',
             '46',
             'cod-docVeneziaRagazzi46'),
            ('Stefano',
             'Ferraresi',
             'Diritto Romano',
             '36',
             'cod-docStefanoFerraresi36'),
            ('Barbara',
             'Vincenzi',
             'Diritto Civile',
             '50',
             'cod-docBarbaraVincenzi50'),
            ('Luca',
             'Rossi',
             'Linguaggi Dinamici',
             '44',
             'cod-docLucaRossi44');

--Account
INSERT INTO sa_account
VALUES      ('bash',
             'gimp',
             'Informatica',
             '2010',
             'MatrMatteoTriggiani25'),
```

```
('bash',
 'gimp',
 'Giurisprudenza',
 '2014',
 'MatrMarcelloFregni20'),
('bash',
 'gimp',
 'Farmacologia',
 '2012',
 'MatreferencerancescoBianchi23'),
('bash',
 'gimp',
 'Filosofia',
 '2011',
 'MatrMichelaRossi20'),
('bash',
 'gimp',
 'Lettere',
 '2010',
 'MatrVincenzoNeri21'),
('bash',
 'gimp',
 'Biologia',
 '2012',
 'MatrGiuliaFuria25'),
('bash',
 'gimp',
 'Ingegneria Meccanica',
 '2012',
 'MatrIlariaCondizionata24'),
('bash',
 'gimp',
 'Lingue Orientali',
 '2011',
 'MatrRezzaCapa20'),
('bash',
 'gimp',
 'Informatica',
 '2013',
 'MatreferenceabrizioCherubini23'),
('bash',
 'gimp',
 'Arte Romana',
 '2015',
 'MatrMarcoGiudici21'),
('bash',
 'gimp',
 'Scienze Politiche',
```

```
             '2012',
             'MatrMassimoDellaPena22'),
            ('bash',
             'gimp',
             'Matematica',
             '2011',
             'MatrGiuditta Penna23'),
            ('bash',
             'gimp',
             'Chimica Farmaceutica',
             '2010',
             'MatreferenceilippoVerucchi26'),
            ('bash',
             'gimp',
             'Economia Aziendale',
             '2011',
             'MatreferencerancoNeri20'),
            ('bash',
             'gimp',
             'Lingue Orientali',
             '2013',
             'MatrLuciaMelella25');

--Utenti
INSERT INTO utenti
VALUES      ('cod u'),
            ('cod u1'),
            ('cod u2'),
            ('cod u3'),
            ('cod u4'),
            ('cod u5'),
            ('cod u6'),
            ('cod u7'),
            ('cod u8'),
            ('cod u9'),
            ('cod u10'),
            ('cod u11'),
            ('cod u12'),
            ('cod u13'),
            ('cod u14'),
            ('cod u15'),
            ('cod u16'),
            ('cod u17'),
            ('cod u18'),
            ('cod u19'),
            ('cod u20'),
            ('cod u21'),
            ('cod u22'),
```

```
            ('cod u23');

--utenti in studenti
INSERT INTO in_studenti
VALUES      ('MatrMatteoTriggiani25',
             'cod u'),
            ('MatrMarcelloFregni20',
             'cod u1'),
            ('MatreferencerancescoBianchi23',
             'cod u2'),
            ('MatrMichelaRossi20',
             'cod u3'),
            ('MatrVincenzoNeri21',
             'cod u4'),
            ('MatrGiuliaFuria25',
             'cod u5'),
            ('MatrIlariaCondizionata24',
             'cod u6'),
            ('MatrRezzaCapa20',
             'cod u7'),
            ('MatreferenceabrizioCherubini23',
             'cod u8'),
            ('MatrMarcoGiudici21',
             'cod u9'),
            ('MatrMassimoDellaPena22',
             'cod u10'),
            ('MatrGiudittaPenna23',
             'cod u11'),
            ('MatreferenceilippoVerucchi26',
             'cod u12'),
            ('MatreferencerancoNeri20',
             'cod u13'),
            ('MatrLuciaMelella25',
             'cod u14');

--utenti in docenti
INSERT INTO in_docenti
VALUES      ('cod u15',
             'cod-docMarioRossi49'),
            ('cod u16',
             'cod-docPaoloNotai44'),
            ('cod u17',
             'cod-docGiovannaLapico59'),
            ('cod u18',
             'cod-docVincenzoDiMatteo52'),
            ('cod u19',
             'cod-docVeneziaRagazzi46'),
            ('cod u20',
```

```
             'cod-docStefanoFerraresi36'),
            ('cod u21',
             'cod-docBarbaraVincenzi50'),
            ('cod u22',
             'cod-docLucaRossi44');

--laboratorio didattico
INSERT INTO lab_did
VALUES      ('cod lab'),
            ('cod lab1'),
            ('cod lab2'),
            ('cod lab3');

--posti
INSERT INTO posti
VALUES      ('1',
             'cod lab'),
            ('2',
             'cod lab'),
            ('3',
             'cod lab'),
            ('4',
             'cod lab'),
            ('5',
             'cod lab'),
            ('6',
             'cod lab'),
            ('7',
             'cod lab'),
            ('8',
             'cod lab'),
            ('9',
             'cod lab'),
            ('10',
             'cod lab'),
            ('1',
             'cod lab1'),
            ('2',
             'cod lab1'),
            ('3',
             'cod lab1'),
            ('4',
             'cod lab1'),
            ('1',
             'cod lab2'),
            ('2',
             'cod lab2'),
            ('3',
```

```
             'cod lab2'),
            ('4',
             'cod lab2'),
            ('5',
             'cod lab2'),
            ('6',
             'cod lab2'),
            ('7',
             'cod lab2'),
            ('8',
             'cod lab2'),
            ('9',
             'cod lab2'),
            ('10',
             'cod lab2'),
            ('11',
             'cod lab2'),
            ('1',
             'cod lab3'),
            ('2',
             'cod lab3'),
            ('3',
             'cod lab3'),
            ('4',
             'cod lab3'),
            ('5',
             'cod lab3'),
            ('6',
             'cod lab3');

--posti assegnati
INSERT INTO posti_assegnati
VALUES      ('1',
             'cod lab'),
            ('2',
             'cod lab'),
            ('3',
             'cod lab'),
            ('4',
             'cod lab'),
            ('5',
             'cod lab'),
            ('6',
             'cod lab'),
            ('7',
             'cod lab'),
            ('8',
             'cod lab'),
```

```
            ('9',
             'cod lab');

--assegnazione
INSERT INTO assegnazione
VALUES      ('gennaio',
             'MatrMarcelloFregni20',
             'cod lab',
             '1'),
            ('gennaio',
             'MatrIlariaCondizionata24',
             'cod lab',
             '2'),
            ('febbraio',
             'MatrGiuliaFuria25',
             'cod lab',
             '3'),
            ('luglio',
             'MatrMarcelloFregni20',
             'cod lab',
             '1'),
            ('marzo',
             'MatrGiudittaPenna23',
             'cod lab',
             '2'),
            ('marzo',
             'MatrLuciaMelella25',
             'cod lab',
             '1'),
            ('aprile',
             'MatrMarcelloFregni20',
             'cod lab',
             '2'),
            ('luglio',
             'MatrLuciaMelella25',
             'cod lab',
             '4'),
            ('gennaio',
             'MatrGiudittaPenna23',
             'cod lab',
             '5');

--autorizzazione
INSERT INTO autorizza
            (cod_doc,
             mese,
             matr,
             cod_lab,
```

```
             np)
VALUES       ('cod-docGiovannaLapico59',
              'gennaio',
              'MatrMarcelloFregni20',
              'cod lab',
              '1'),
             ('cod-docStefanoFerraresi36',
              'gennaio',
              'MatrIlariaCondizionata24',
              'cod lab',
              '2'),
             ('cod-docBarbaraVincenzi50',
              'febbraio',
              'MatrGiuliaFuria25',
              'cod lab',
              '3'),
             ('cod-docBarbaraVincenzi50',
              'luglio',
              'MatrMarcelloFregni20',
              'cod lab',
              '1'),
             ('cod-docPaoloNotai44',
              'marzo',
              'MatrGiudittaPenna23',
              'cod lab',
              '2'),
             ('cod-docPaoloNotai44',
              'marzo',
              'MatrLuciaMelella25',
              'cod lab',
              '1'),
             ('cod-docVincenzoDiMatteo52',
              'aprile',
              'MatrMarcelloFregni20',
              'cod lab',
              '2'),
             ('cod-docVincenzoDiMatteo52',
              'luglio',
              'MatrLuciaMelella25',
              'cod lab',
              '4'),
             ('cod-docGiovannaLapico59',
              'gennaio',
              'MatrGiudittaPenna23',
              'cod lab',
              '5');

--Posti prenotati
```

```
INSERT INTO posti_prenotati
VALUES      ('1',
             'cod lab2'),
            ('2',
             'cod lab2'),
            ('3',
             'cod lab2'),
            ('4',
             'cod lab2'),
            ('5',
             'cod lab2'),
            ('6',
             'cod lab2'),
            ('7',
             'cod lab2'),
            ('8',
             'cod lab2'),
            ('9',
             'cod lab2'),
            ('10',
             'cod lab2'),
            ('11',
             'cod lab2');

--Prenotazioni
INSERT INTO prenota
VALUES      ('6',
             '1',
             'cod lab2',
             'MatreferenceabrizioCherubini23',
             '14/04/2015'),
            ('8',
             '1',
             'cod lab2',
             'MatrLuciaMelella25',
             '12/04/2015'),
            ('10',
             '3',
             'cod lab2',
             'MatreferenceabrizioCherubini23',
             '15/04/2015'),
            ('12',
             '6',
             'cod lab2',
             'MatrLuciaMelella25',
             '16/03/2015'),
            ('7',
             '11',
```

```
             'cod lab2',
             'MatrGiudittaPenna23',
             '11/04/2015'),
            ('14',
             '4',
             'cod lab2',
             'MatrMichelaRossi20',
             '22/04/2015'),
            ('17',
             '2',
             'cod lab2',
             'MatreferenceilippoVerucchi26',
             '01/02/2015');

--Posti Liberi
INSERT INTO posti_liberi
VALUES      ('1',
             'cod lab1'),
            ('2',
             'cod lab1'),
            ('3',
             'cod lab1'),
            ('4',
             'cod lab1');

--accessi liberi
INSERT INTO accede_liberamente
VALUES      ('1',
             'MatrRezzaCapa20',
             'cod lab1'),
            ('2',
             'MatrVincenzoNeri21',
             'cod lab1'),
            ('3',
             'MatreferencerancoNeri20',
             'cod lab1');

--resposabile organizzativo
INSERT INTO resp_org
            (cod_lab,
             cod_doc)
VALUES      ('cod lab',
             'cod-docGiovannaLapico59'),
            ('cod lab',
             'cod-docPaoloNotai44'),
            ('cod lab',
             'cod-docVincenzoDiMatteo52'),
            ('cod lab',
```

```
             'cod-docBarbaraVincenzi50');

--tecnici
INSERT INTO tecnico
VALUES      ('Donati',
             'Mario',
             '30',
             'DNTMRA00XX00X000X'),
            ('Verucchi',
             'Maria',
             '40',
             'VRCMRA00XX00X000X'),
            ('Benvatto',
             'Lorenzo',
             '35',
             'BNVLRN00XX00X000X'),
            ('Tavoni',
             'Franco',
             '38',
             'TVNFRN00XX00X000X'),
            ('Ronchi',
             'Laura',
             '29',
             'RNCLRA00XX00X000X'),
            ('Signorini',
             'Luca',
             '42',
             'SGNLCU00XX00X000X');

--responsabile operativo
INSERT INTO resp_ope
            (cod_lab,
             cf)
VALUES      ('cod lab',
             'VRCMRA00XX00X000X'),
            ('cod lab',
             'BNVLRN00XX00X000X'),
            ('cod lab',
             'SGNLCU00XX00X000X'),
            ('cod lab',
             'DNTMRA00XX00X000X');

--risorse
INSERT INTO risorsa
VALUES      ('Stampante',
             'cod risorsa1'),
            ('Fax',
             'cod risorsa2'),
```

```
            ('Drone',
             'cod risorsa3'),
            ('Large Hadron Collider',
             'cod risorsa4'),
            ('Velocipede',
             'cod risorsa5'),
            ('Unità di calcolo',
             'cod risorsa6');

--UC
INSERT INTO unita_di_calcolo
VALUES      ('cod risorsa6');

INSERT INTO accede
VALUES      ('Arte Romana',
             'cod risorsa6',
             '2015',
             'MatrMarcoGiudici21');

INSERT INTO assegnata_a
VALUES      ('MatrMarcoGiudici21',
             'cod risorsa6',
             '2015',
             'Arte Romana');

INSERT INTO locale
VALUES      ('cod loc'),
            ('cod loc1'),
            ('cod loc2'),
            ('cod loc3'),
            ('cod loc4'),
            ('cod loc5'),
            ('cod loc6'),
            ('cod loc7');

INSERT INTO is_lab
VALUES      ('cod lab',
             'cod loc'),
            ('cod lab1',
             'cod loc1'),
            ('cod lab2',
             'cod loc2'),
            ('cod lab3',
             'cod loc3');

INSERT INTO permesso
VALUES      ('cod u',
             'cod loc');
```

```
INSERT INTO tipologia
VALUES      ('nome tipo');

INSERT INTO appartiene
VALUES      ('nome tipo',
             'cod u',
             'cod loc' );

INSERT INTO orario
VALUES      ('Lunedi',
             'nome tipo',
             '13');

INSERT INTO tentativo
            (cod_loc,
             cod_u,
             data,
             ora)
VALUES      ('cod loc',
             'cod u',
             '14/04/2014',
             '15:00');
```

Codice per l'automazione del DBM

```
--gestione Rifiuti
CREATE FUNCTION controllo
  () returns TRIGGER
AS
  $$
BEGIN
  IF (NEW.cod_u, NEW.cod_loc) NOT IN
    (
           SELECT cod_u,
                  cod_loc
           FROM   permesso p
           WHERE  NEW.cod_u = p.cod_u
           AND    NEW.cod_loc = p.cod_loc) THEN
    INSERT INTO rifiuti VALUES
                (
                           NEW.data,
                           NEW.cod_loc,
                           NEW.cod_u
                );

    endif;
    RETURN NEW;
  END;
  $$language 'plpgsql';
```

```
CREATE TRIGGER gestione_rifiuti BEFORE
INSERT
ON tentativo FOR EACH ROW EXECUTE PROCEDURE controllo ();

--gestione accesso genericoCREATE FUNCTION generico ()
returns TRIGGER AS $$
BEGIN
  UPDATE loc_gen
  SET    n_accessi = n_accessi +1
  WHERE  NEW.cod_gen = cod_gen;

  RETURN NEW;
END;
$$language 'plpgsql';
CREATE TRIGGER gestione_generici BEFORE
INSERT
ON accede_gen FOR EACH ROW EXECUTE PROCEDURE generico ();

--gestione SessioneCREATE FUNCTION sessione ()
returns TRIGGER AS $$
DECLARE
  n INTEGER;
BEGIN
  SELECT count(*)
  INTO   n
  FROM   rifiuti
  WHERE  matr = NEW.matr
  AND    data = NEW.data
  AND    cod_loc = NEW.cod_loc;

  IF n = 3 THEN
    RAISE
  EXCEPTION
    'Hai gia` ricevuto 3 rifiuti, non hai il permesso per entrare qui';
  END IF;
  RETURN NEW;
END;
$$language 'plpgsql';
CREATE TRIGGER gestione_accessi BEFORE
INSERT
ON rifiuti FOR EACH ROW EXECUTE PROCEDURE sessione();

--Assegnazioni univocheCREATE FUNCTION assegnazioni ()
returns TRIGGER AS $$
BEGIN
  IF NEW.cod_lab = a.cod_lab
    AND
    NEW.matr = a.matr
    AND
    NEW.mese = a.mese THEN
    RAISE
  EXCEPTION
    'una sola assegnazionezione';
    endif;
    RETURN NEW;
  END;
  $$language 'plpgsql';
  CREATE TRIGGER gestione_assegnazioni BEFORE
```

```
INSERT
ON assegnazione FOR EACH ROW EXECUTE PROCEDURE assegnazioni();

--Permessi univociCREATE FUNCTION permessi ()
returns TRIGGER AS $$
BEGIN
  IF NEW.cod_u IN
    (
           SELECT cod_u
           FROM   permesso p
           WHERE  NEW.cod_loc = p.cod_loc
           AND    NEW.cod_u = p.cod_u
    )
    THEN
    RAISE
  EXCEPTION
    'una solo permesso';
    endif;
    RETURN NEW;
  END;
  $$language 'plpgsql';
  CREATE TRIGGER gestione_permessi BEFORE
  INSERT
  ON permesso FOR EACH ROW EXECUTE PROCEDURE permessi();
```