

PROGETTO BASI DI DATI

Servizio web di rete sociale dedicato allo sviluppo di contatti professionali

Progetto per l'anno accademico

2016/2017

Sessione estiva

Realizzato da:

Marrancone Davide [273567]

Indice

Analisi dei requisiti	3
1.1 Specifiche	3
Glossario	4
1.3 Concetti chiave riformulati	4
2 Operazioni	5
2.1 Interrogazioni:	5
2.2 Aggiornamenti	5
3 Progettazione concettuale	5
3.1 Scelte progettuali	5
3.2 Sviluppo dello schema E-R	6
3.3 Utenti.....	7
3.4 Standard	7
3.5 Premium	8
3.6 Offerte	8
3.7 Feedback	9
3.8 Gruppi.....	9
3.9 Candidature	9
3.10 Relazioni	10
3.11 Schema E-R finale.....	10
4 Progettazione logica	12
4.1 Ottimizzazione	12
4.2 Semplificazione	13
4.3 Traduzione	14
4.4 Normalizzazione	14
5. MySql query	15

1 Analisi dei requisiti

1.1 Specifiche

Il fine del progetto è quello di realizzare un servizio che permette di mettere in contatto singoli utenti con altri in ambito socio-lavorativo.

Lo scopo principale della piattaforma è di connettere utenti fra loro all'interno di una rete sociale, che può essere trasformata a piacere dall'utente.

Per rendere possibile la comunicazione tra gli utenti si utilizzano messaggi in forma di candidature e/o richieste.

Ogni singolo utente ha una rete di contatti a sua disposizione, comprende delle connessioni dirette, quelle di secondo e di terzo grado: partendo dal singolo nodo, si definisce diretta la connessione in assenza di nodi intermedi, mentre per secondo e terzo grado si intendono connessioni con, rispettivamente, uno e due nodi intermedi.

Nel profilo personale vi è inoltre una sezione riservata a riconoscimenti e premi ottenuti, feedback di competenza delle capacità attestate da altri utenti.

Gli utenti si registrano alla piattaforma con i propri dati anagrafici presentando il proprio curriculum e le proprie competenze nel profilo personale.

Una volta registrato l'utente sarà inserito in uno o più gruppi secondo dettagli in comune, competenze, ambito di lavoro, candidati o stessa provenienza geografica di ricerca.

Nella piattaforma è possibile l'inserimento di annunci da parte di datori di lavoro (utenti premium) e la ricerca delle singole offerte da parte dei candidati (utenti standard); sarà possibile essere presentati ad altri utenti e quindi creare una nuova connessione tramite contatti creando così un proprio capitale sociale.

1.2 Glossario

Termine	Descrizione	Sinonimi	Collegamenti
Utente	L'utente può essere standard e quindi ad accesso libero o in caso di datori di lavoro, premium quindi a pagamento.	Candidati	Gruppo Feedback Candidature
Profilo utente	Area in cui sono dettagliate tutte le specifiche del singolo utente, sia anagrafiche che curriculari, nonché feedback, premi e riconoscimenti ottenuti.		Utente Feedback
Gruppo	Insieme numerabile di utenti aventi una caratteristica comune.		Utente
Feedback	Feedback rilasciato da un utente.		Profilo personale Utente
Connessioni	Legame logico fra uno o più utenti.		Utente
Offerte	Proposte di lavoro.	Annunci	Utente
Candidature	Comunicazione fra datori di lavoro e candidati	Messaggi	Utente

1.3 Concetti chiave riformulati

- **UTENTI:** gli utenti possono registrarsi come standard o come premium per poter fruire dei servizi della piattaforma.
- **GRUPPI:** ogni utente può essere inserito in uno o più gruppi in base alla chiave di ricerca e quindi alla tipologia di gruppo; dividiamo i gruppi in 2 tipologie, stesso settore lavorativo o medesima Provenienza geografica.

- OFFERTE: sono le offerte di lavoro erogate dagli utenti premium contenenti il settore, la retribuzione, la posizione geografica e le competenze necessarie.
- CANDIDATURE: le candidature sono testi scambiati fra candidati e datori di lavoro, contrassegnati da una data e un testo.
- FEEDBACK: i feedback sono messaggi composti da una data, un testo e un utente destinatario.

2 Operazioni

Riportiamo le interrogazioni e gli aggiornamenti che la base di dati deve soddisfare.

2.1 Interrogazioni:

La base deve:

- Permettere agli utenti standard di cercare offerte nel settore di lavoro scelto
- Permettere agli utenti standard di cercare offerte per competenze necessarie richieste
- Permettere agli utenti premium di ricevere le candidature
- Permettere agli utenti standard di presentare le candidature

2.2 Aggiornamenti

La base deve:

- Poter registrare un utente premium
- Poter registrare un utente standard
- Poter eliminare un utente premium
- Poter eliminare un utente premium
- permettere agli utenti premium di aggiungere offerte
- permettere agli utenti premium di eliminare offerte
- permettere all'utente standard di rilasciare feedback
- Permettere all'utente premium di rilasciare feedback

3 Progettazione concettuale

3.1 Scelte progettuali

L'idea è quella di partire da uno schema scheletro che evidenzi i concetti principali, dividere poi i requisiti e le entità in singoli componenti distinte, infine ricostruire uno schema complesso e

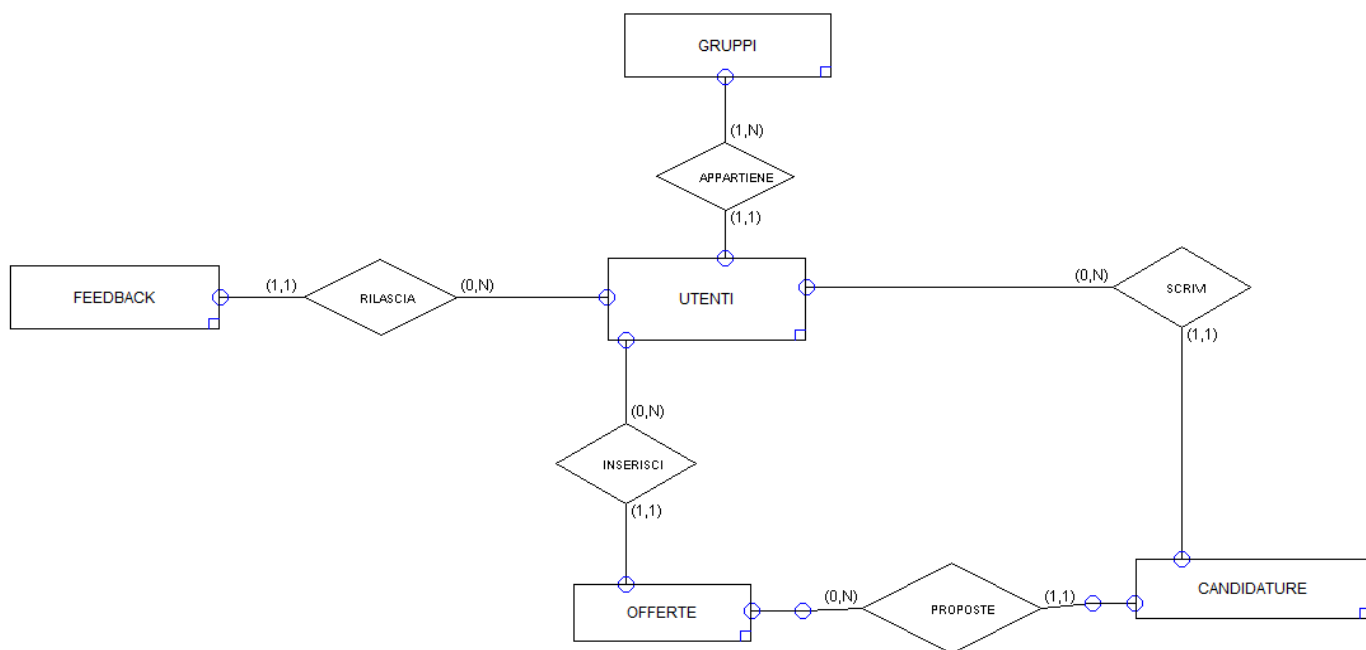
dettagliato: per il progetto viene quindi utilizzata l'unione fra strategia bottom-up e top-down, scomponendo e ricomponendo l'intero schema.

3.2 Sviluppo dello schema E-R

Inizialmente concretizziamo in uno schema scheletro, in seguito ci soffermiamo sui singoli sottoschemi rappresentanti le entità definite nello scheletro: lavorando iterativamente su ogni sottoschema miglioriamo e affiniamo gli attributi delle singole entità.

Infine tutti i sottoschemi prodotti vengono integrati in un unico grande schema seguendo sempre la linea guida dello schema scheletro.

Viene di seguito riportato lo schema scheletro.



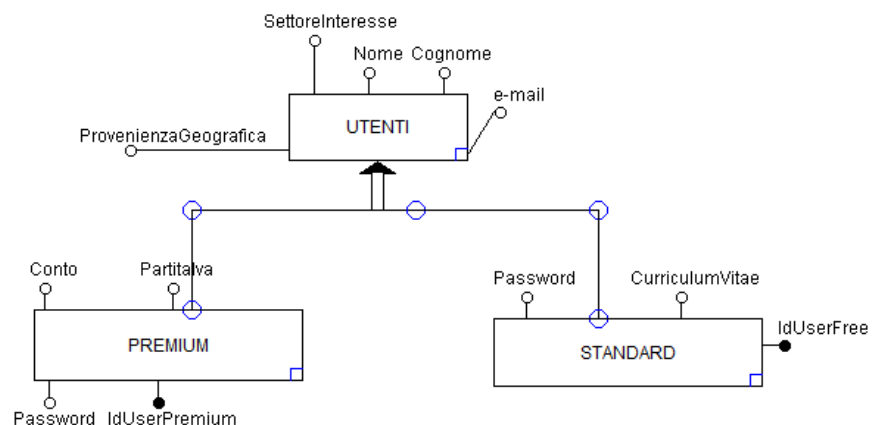
Andiamo ora a vedere le singole entità.

3.3 Utenti

Utenti è l'entità padre di Premium e Standard ed è colui che può e desidera utilizzare l'applicazione; quest'ultima può essere utilizzata da utenti Premium o da utenti Standard distinguendo i servizi richiesti.

Gli attributi generali associati sono:

- Nome
- Cognome
- Email
- Provenienza geografica
- Settore d'interesse

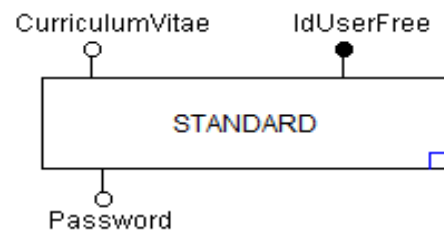


3.4 Standard

L'entità Standard è la classe figlia della classe Utenti, in particolare un utente Standard può ricercare e proporre candidature per offerte di lavoro e rilasciare feedback a qualsiasi altro utente registrato alla piattaforma.

Gli attributi associati sono:

- UserStandard (username utente standard)
- Password
- Curriculum vitae

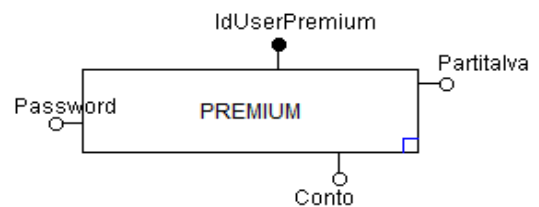


3.5 Premium

L'entità Premium è la classe figlia della classe Utenti, in particolare un utente Premium può inserire o rimuovere offerte di lavoro e rilasciare feedback a qualsiasi altro utente registrato alla piattaforma. Ogni utente Premium può appartenere ad uno o più gruppi in base alle proprie caratteristiche.

Gli attributi associati sono:

- UserPremium
- Password
- Partita IVA
- Conto

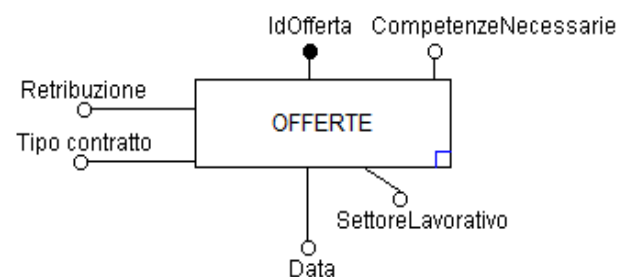


3.6 Offerte

L'entità Offerte è accessibile da entrambi i tipi di utenti in modalità differenti: gli utenti Premium possono inserire e cancellare Offerte mentre gli utenti Standard possono ricercarle ed eventualmente presentare la propria candidatura.

Ogni offerta indica:

- IdOfferta (identificativo dell'offerta)
- Tipo contratto (a tempo determinato, indeterminato, a progetto, ecc...)
- Retribuzione (stipendio fornito)
- Competenze necessarie
- Data
- Settore lavorativo

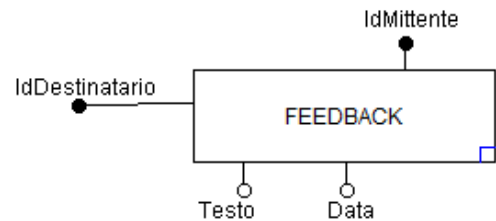


3.7 Feedback

L'entità feedback è accessibile da ogni tipo di utente e consente di rilasciare un feedback in forma testuale riferito ad un altro utente registrato.

È necessario indicare:

- Data
- Testo
- IdMittente (identificatore di colui che scrive la segnalazione)
- IdDestinatario (identificatore di colui che riceve la segnalazione)

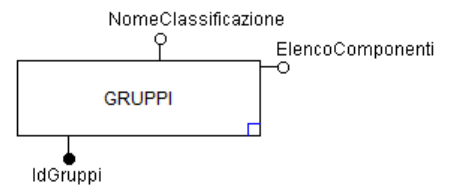


3.8 Gruppi

L'entità gruppi è accessibile da ogni tipo di utente e li classifica secondo due caratteristiche: Provenienza geografica e settore lavorativo.

Attributi:

- IdGruppo (identificativo del gruppo)
- Nome Classificazione (Provenienza geografica o settore)
- Elenco Componenti (lista nominativa degli utenti che appartengono al determinato gruppo)

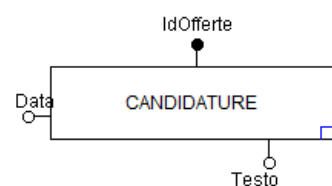


3.9 Candidature

L'entità Candidature è accessibile solo da utenti Standard, rappresenta la volontà di proporsi per un lavoro e si riferisce ad una specifica offerta.

È composta da:

- Data
- Testo
- IdOfferta



3.10 Relazioni

Analizziamo le relazioni presenti fra le entità al fine di costruire lo schema finale:

- Relazione Utenti-Feedback, ogni utente può rilasciare (0,N) feedback e ogni feedback può essere scritto da uno e un solo utente (1,1);
- Relazione Utenti-Gruppi, ogni utente può appartenere ad (1,N) mentre ogni gruppo è composto da (1,N) utenti;
- Relazione Premium-Offerte, ogni utente Premium può inserire (0,N) offerte di lavoro ed ogni offerta è inserita da uno ed un solo (1,1) utente Premium;
- Relazione Standard-Candidature, ogni utente Standard può scrivere da (0,N) candidature differenti e ogni candidatura può essere presentata da uno ed un solo (1,1) utente Standard;
- Relazione Candidature-Offerte, ogni candidatura viene proposta ad una sola offerta (1,1) mentre ogni offerta può ricevere da (0,N) candidature differenti.

3.11 Schema E-R finale

Schema E-R nella sua forma finale, prima della progettazione logica, si presenta così:

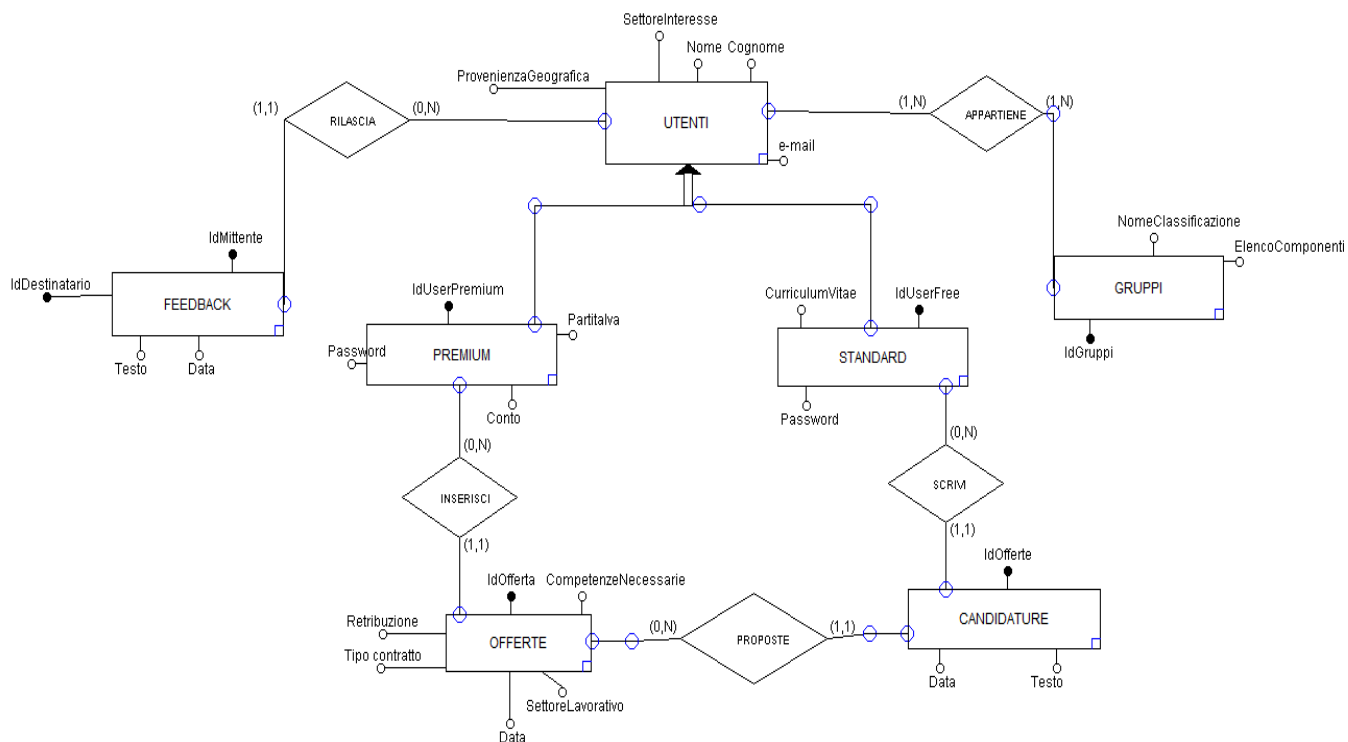


Tabella delle entità

Entità	Descrizione	Attributi	Identificatore
Utenti	Classe padre, qualsiasi utente appartiene ad un gruppo può rilasciare feedback.	Nome Cognome Email Provenienza geografica Settore lavorativo	
Premium	Classe figlio, eredita le caratteristiche del padre, inoltre può inserire e rimuovere offerte.	UserPremium Password Partita IVA Conto	IdPremium
Standard	Classe figlio, eredita le caratteristiche del padre, inoltre può presentare candidature.	UserStandard Password Curriculum vitae	IdStandard
Offerte	Opportunità di lavoro pubblicata da un utente premium e a cui può candidarsi un utente standard.	IdOfferta Tipo contratto Retribuzione Competenze necessarie Data Settore lavorativo	IdOfferta
Feedback	Recensioni che un utente può rilasciare su un altro utente.	Data Testo IdMittente IdDestinatario	
Gruppi	Classificazione degli utenti in base a caratteristiche comuni.	IdGruppo Nome Classificazione Elenco Componenti	IdGruppo
Candidature	Messaggi testuali che un utente standard può scrivere rispetto ad una determinata offerta.	Data Testo IdOfferta	

Tabella delle associazioni

Associazione	Descrizione	Componenti
Rilascia	Aggiunge una segnalazione	Feedback Utenti
Appartiene	Inserisce l'utente in un gruppo	Gruppi Utenti
Inserisci	Aggiunge un'offerta lavorativa	Premium Offerte
Scrivi	Permette candidarsi per un lavoro	Standard Candidature
Proposta	Collega una candidatura alla specifica offerta lavorativa	Candidature Offerte

4 Progettazione logica

La progettazione logica della base di dati in esame è suddivisa in 4 parti:

1. ottimizzazione;
2. semplificazione;
3. traduzione;
4. normalizzazione.

4.1 Ottimizzazione

Nell'ottimizzazione andremo a inserire la tabella delle operazioni e dei volumi riguardanti il database allo scopo di approssimare l'indice di ottimizzazione.

Concetto	Tipo	Volume
Utenti	E	1500
Premium	E	500
Standard	E	1000
Offerte	E	2000
Feedback	E	3000
Gruppi	E	750
Candidature	E	3000
Rilascia	R	2000
Appartiene	R	500
Inserisci	R	2000
Scrivi	R	200
Proposta	R	100
Candida	R	150

Tabella delle operazioni

Operazione	Tipo	Frequenza
Registrazione Premium	I	15 al giorno
Registrazione Standard	I	30 al giorno
Aggiunta offerta	I	30 ogni settimana
Rimozione offerta	I	10 alla settimana
Invio Candidature	I	70 alla settimana
Rilascio feedback	I	50 ogni due settimane
Mostrare lista candidati	B	20 al giorno
Mostrare gruppo utenti	B	25 ogni tre giorni
Eliminazione candidati	I	20 al mese

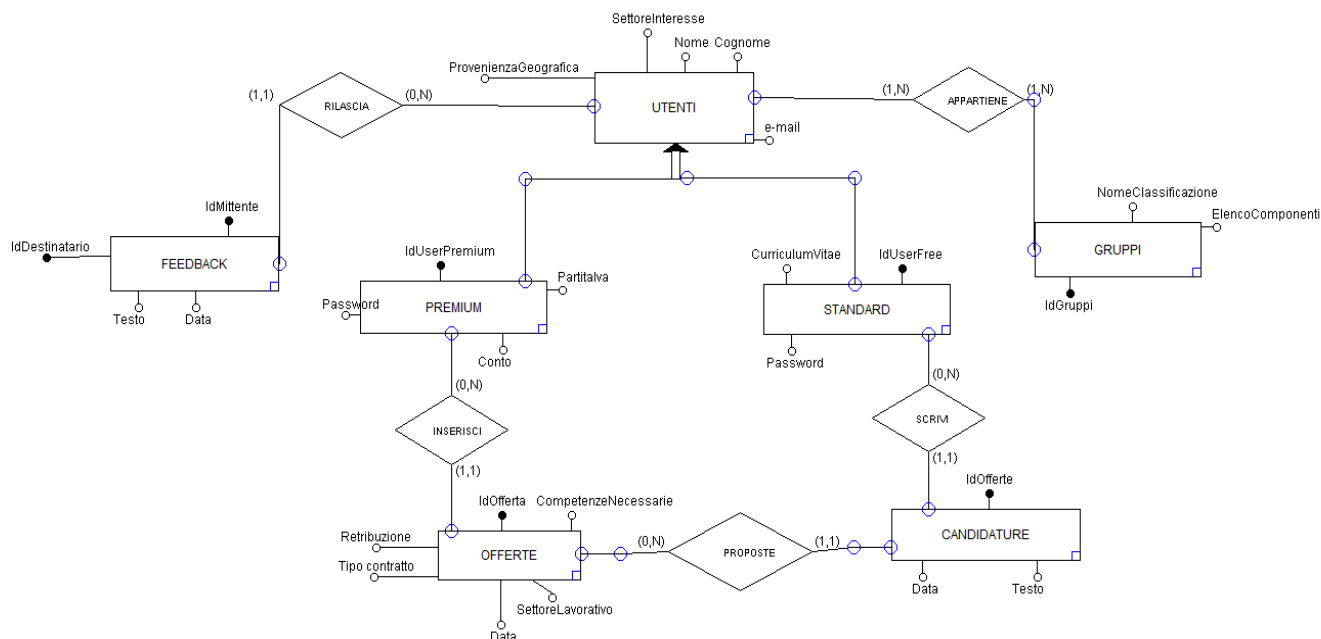
4.2 Semplificazione

La stesura dello schema E-R finale è avvenuta attraverso una serie di miglioramenti e osservazioni, dall'analisi iniziale fino alla progettazione, con l'intento di migliorare l'efficienza del progetto, snellendolo, per ottenere una base di dati valida.

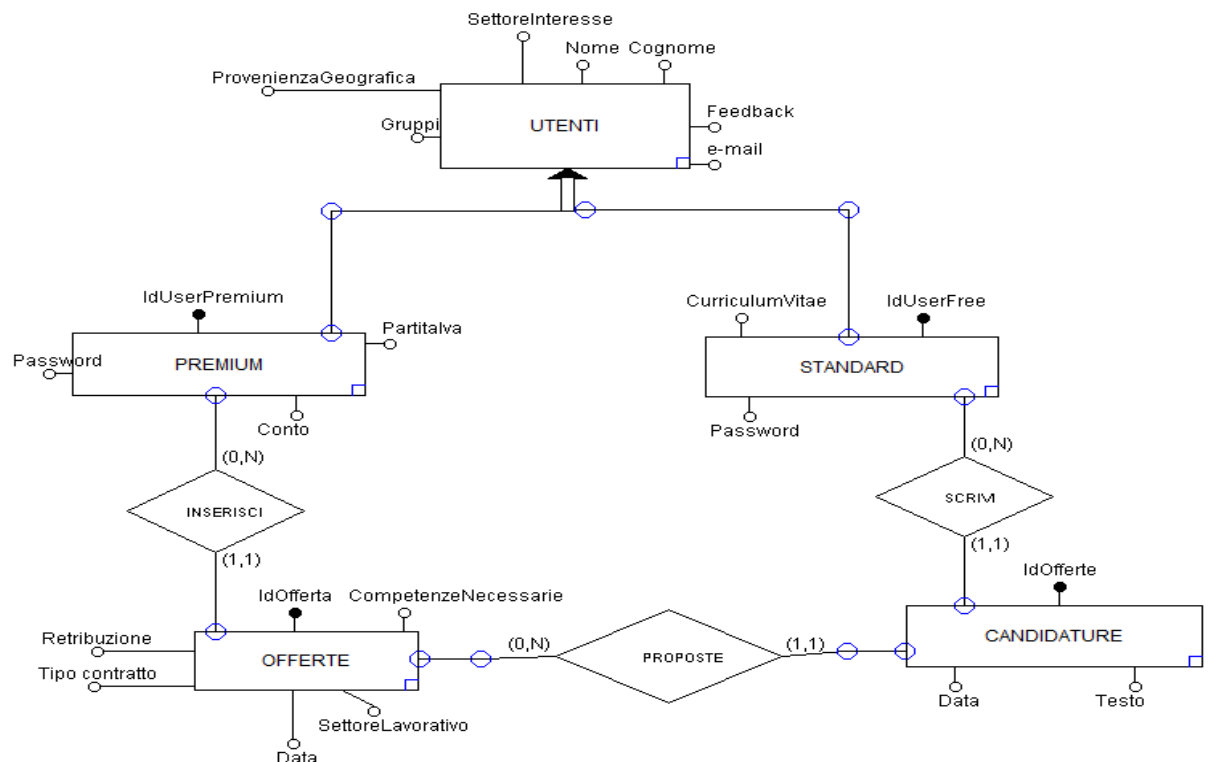
I cambiamenti sostanziali sono:

- Una scelta è stata quella di pensare a FEEDBACK non come una tabella separata e quindi come singola entità, bensì come un concetto associato all'entità UTENTI, trasformandolo così in un attributo di quest'ultima: otteniamo uno schema con una tabella in meno.
- Una scelta è stata quella di pensare a GRUPPI non come una tabella separata e quindi come singola entità, bensì come un concetto associato all'entità UTENTI, trasformandolo così in un attributo di quest'ultima: otteniamo uno schema con una tabella in meno.

Di seguito vedremo la variazione grafica dello schema E-R finale. Precedentemente lo schema finale si presentava così:



Dopo aver applicato la semplificazione lo schema E-R avrà questo aspetto:



4.3 Traduzione

Il nostro schema E-R viene adesso tradotto nel relativo schema relazionale:

GRUPPI (IdGruppo, Nome classificazione, Elenco componenti)

UTENTI (Nome, Cognome, Email, Provenienza geografica, Settore lavorativo, Elenco recensioni)

PREMIUM (UserPremium, Password, Partita IVA, Conto)

STANDARD (UserStandard, Password, Curriculum vitae)

OFFERTE (IdOfferta, Tipo contratto, Retribuzione, Competenze necessarie, Data, Settore Lavorativo, Elenco candidati)

4.4 Normalizzazione

La normalizzazione è un processo che ha lo scopo finale di certificare la qualità dello schema del database.

Per far tutto ciò si esaminano le relazioni delle tabelle precedenti prendendo in esame una tabella per volta e si verifica

il soddisfacimento delle seguenti forme normali, che garantiscono l'eliminazione di qualsiasi anomalia o incoerenze all'interno delle relazioni.

1. Gli attributi devono essere definiti su un dominio con valori atomici.
2. quando è in 1NF e per ogni relazione tutti gli attributi non chiave dipendono funzionalmente dall'intera chiave composta cioè dall'insieme di chiavi minime che identifica in modo univoco i record della tabella.
3. quando è in 2NF e ogni attributo non chiave dipende solamente dalla chiave e non da altri attributi non chiave.
4. Forma normale di Boyce e Codd (BCNF) se per ogni dipendenza funzionale $X \rightarrow A$ (relazioni tra attributi di una tabella), X deve contenere una chiave della relazione.

Si può vedere che tutti gli attributi delle tabelle hanno un dominio di valori atomici ed è quindi soddisfatta la 1FN. In maniera alquanto banale è soddisfatta la 2FN visto che in ogni tabella restante la chiave minima è unica. La terza forma normale è soddisfatta perché ogni attributo non chiave non dipende da altri attributi non chiave. Infine per lo stesso motivo della 3NF è garantita anche la forma normale di Boyce e Codd.

5. MySQL query

In questa sezione verrà illustrato il codice **SQL** utilizzato per la creazione delle tabelle

Tabella gruppi

```
CREATE TABLE gruppi(  
  id_gruppi int(3) NOT NULL AUTO_INCREMENT,  
  nome_classificazione varchar(50) NOT NULL,  
  elenco_componenti varchar(500),  
  PRIMARY KEY (id_gruppi)  
);
```

Tabella utenti premium

```
CREATE TABLE premium(  
  id_user_premium int(3) NOT NULL AUTO_INCREMENT,  
  partita_iva varchar(15),  
  conto_corrente varchar(50) NOT NULL,  
  password varchar(20) NOT NULL,
```

```
nome varchar(50) NOT NULL,  
cognome varchar(50) NOT NULL,  
settore_interesse varchar(50),  
feedback varchar(500),  
area_geografica varchar(50),  
email varchar(50),  
PRIMARY KEY (id_user_premium)  
);
```

Tabella utenti standard

```
CREATE TABLE standard(  
    id_user_standard int(3) NOT NULL AUTO_INCREMENT,  
    curriculum_vitae varchar(5000),  
    password varchar(20) NOT NULL,  
    nome varchar(50) NOT NULL,  
    cognome varchar(50) NOT NULL,  
    settore_interesse varchar(50),  
    feedback varchar(500),  
    area_geografica varchar(50),  
    email varchar(50),  
    PRIMARY KEY (id_user_standard)  
);
```

Tabella offerte

```
CREATE TABLE offerte(  
    id_offerta int(3) NOT NULL AUTO_INCREMENT,  
    competenze_necessarie varchar(2000),  
    settore_lavorativo varchar(50) NOT NULL,  
    retribuzione decimal(10,2),  
    tipo_contratto varchar(50),  
    elenco_candidati varchar(500),  
    data date NOT NULL,  
    PRIMARY KEY (id_offerta),  
    FOREIGN KEY (elenco_candidati) REFERENCES standard(cognome)  
);
```

Tabella candidature

```
CREATE TABLE candidature(  
    id_offerte int(3) NOT NULL AUTO_INCREMENT,  
    data date NOT NULL,  
    testo varchar(500),  
    PRIMARY KEY (id_offerte)  
);
```


Verranno ora inserite le interrogazioni in codice SQL.

1) Permettere agli utenti standard di cercare offerte settore

```
SELECT * FROM offerte WHERE settore_lavorativo = ""
```

2) Permettere agli utenti standard di cercare offerte per competenze necessarie

```
SELECT * FROM offerte WHERE competenze_necessarie = ""
```

3) Permettere agli utenti premium di ricevere le candidature

```
SELECT elenco_candidati FROM offerte
```

4) Permettere agli utenti standard di presentare le candidature

```
INSERT INTO candidature(testo) values("")
```

5) Poter registrare un utente premium

```
INSERT INTO  
premium(partita_iva,password,nome,cognome,settori_interesse,feedback,area_geografica,email)  
values("", "", "", "", "", "", "", "")
```

6) Poter registrare un utente standard

```
INSERT INTO  
standard(curriculum_vitae,password,nome,cognome,settori_interesse,feedback,area_geografica,  
email)values("", "", "", "", "", "", "", "")
```

7) Poter eliminare un utente premium

```
DELETE FROM premium WHERE id_user_premium = ""
```

8) Poter eliminare un utente standard

```
DELETE FROM standard WHERE id_user_standard = ""
```

9) Permettere agli utenti premium di inserire offerte

```
INSERT INTO  
offerte(competenze_necessarie,settori_lavorativo,retribuzione,tipo_contratto,data)values  
("", "", "", "", "")
```

10) Permettere agli utenti premium di eliminare offerte

```
DELETE FROM offerte WHERE id_offerta = ""
```

11) Permettere all'utente premium di rilasciare i feedback

```
UPDATE premium SET feedback="" WHERE id_user_premium=""
```

12) Permettere all'utente standard di rilasciare i feedback

```
UPDATE standard SET feedback="" WHERE id_user_standard = ""
```

Strumenti usati

Per la realizzazione del progetto sono stati usati prodotti open source, per la realizzazione dei diagrammi E-R è stato utilizzato Java Diagram ER mentre per verificare il funzionamento delle query è stato creato un database su altervista.org tramite l'uso di phpmyadmin.