



Progetto di Basi di Dati

Piattaforma d'incontri

Progetto di Basi di Dati

- Piattaforma d'incontri –

1. Raccolta delle specifiche della realtà d'interesse

Descrizione

Il nostro intento è quello di creare una base di dati che gestisca una **piattaforma d'incontri**, permettendo a persone con interessi affini di conoscersi e interagire.

Per accedere alla piattaforma si richiede la creazione di un profilo, univocamente identificato dal numero di telefono, si vuole conoscere nome e cognome dell'utente, il suo genere ed orientamento; si ha bisogno di una mail e una password per poter accedere alla piattaforma. Inoltre, il profilo conserva la data d'iscrizione e può contenere più foto.

Un profilo in più può lasciar mi piace ad un altro profilo, il quale viene conservata la data di quando si è lasciato.

L'utente ha la possibilità di scrivere una bio, evidenziando la propria professione, un profilo social e vi è la possibilità di scrivere un breve testo. In più può essere presente il motivo (amicizia, relazione sentimentale o un semplice incontro) per il quale ci si è iscritti alla piattaforma.

Un profilo può essere di tipo normale, che ha un numero massimo di match, e premium, il quale specifica un costo e la data di sottoscrizione al servizio. Per essere tale un utente premium effettua un pagamento che è identificato dal numero della carta e conserva i dati del titolare, del tipo di circuito e la data di scadenza. Gli utenti premium possono entrare all'interno di una classifica, in cui per ogni utente possiamo consultare il numero della posizione del profilo e il rate d'interesse determinato dal numero di mi piace. L'ulteriore vantaggio degli utenti premium è quello di poter interagire con utenti con i quali non si è riscontrato un match. In più quest'ultimi possono vedere il numero di mi piace che hanno ricevuto.

L'affinità tra due profili permette di favorire il match, individuato da un ID, si conserva anche la data in cui è avvenuto e il numero totale verificatosi fino ad ora. Una volta in cui si è verificato un match l'utente (normale) sblocca la possibilità di interagire con l'utente in questione.

A loro volta le interazioni sono identificate da un codice univoco e ne viene conservata una data e un testo.

Specifiche della realtà d'interesse

La realtà che andremo a sviluppare è quella di una piattaforma d'incontri; i quali grazie all'ingente sviluppo tecnologico avuto in questi ultimi anni, queste piattaforme hanno giocato un ruolo importante nel modo in cui le persone sviluppano le relazioni e il modo in cui le vivono. Ad oggi le piattaforme d'incontri sono diventate un vero e proprio luogo virtuale dove conoscere e interagire con persone di tutto il mondo.

Nella realtà che andiamo a rappresentare, è necessaria la creazione di un profilo affinché si possa interagire con altri utenti e che si venga visualizzati a sua volta. L'utente può essere di due tipi: normale, che ha dei limiti al proprio profilo, e premium, che oltre a non avere i limiti di un utente normale ha accesso a particolari funzioni. In più la piattaforma a quest'ultimi riserva una classifica, nella quale può far aumentare il rate d'interesse ed entrare in una graduatoria a seconda di esso con un massimo di 10 posti. Un qualsiasi tipo di profilo, può scrivere una bio di massimo 500 caratteri visualizzabile a schermo all'interno del profilo, nella quale può aggiungere la propria professione un profilo sociale. In più può, a detta dell'utente, aggiungerne il motivo per cui è in piattaforma specificandone il tipo di relazione per cui si è alla ricerca.

Di un profilo si tiene anche traccia dell'orientamento, che può essere di 8 tipi, scelto dall'utente al momento della creazione del profilo. La compatibilità tra due profilo fa avvenire il match, la cui piattaforma tiene traccia la data di avvenuta compatibilità. Una volta verificatosi permette di sbloccare l'interazione, che per gli utenti premium può essere compiuta anche senza l'avvenuto match, cioè un utente può scrivere un messaggio di testo, di massimo 1000 caratteri, per interagire con un altro profilo.

La base di dati inoltre tiene traccia anche del numero di match e del numero di mi piace che ha avuto un determinato profilo, visualizzabile unicamente dal proprietario di quest'ultimo, tranne per il numero di mi piace visualizzabile unicamente da profili premium

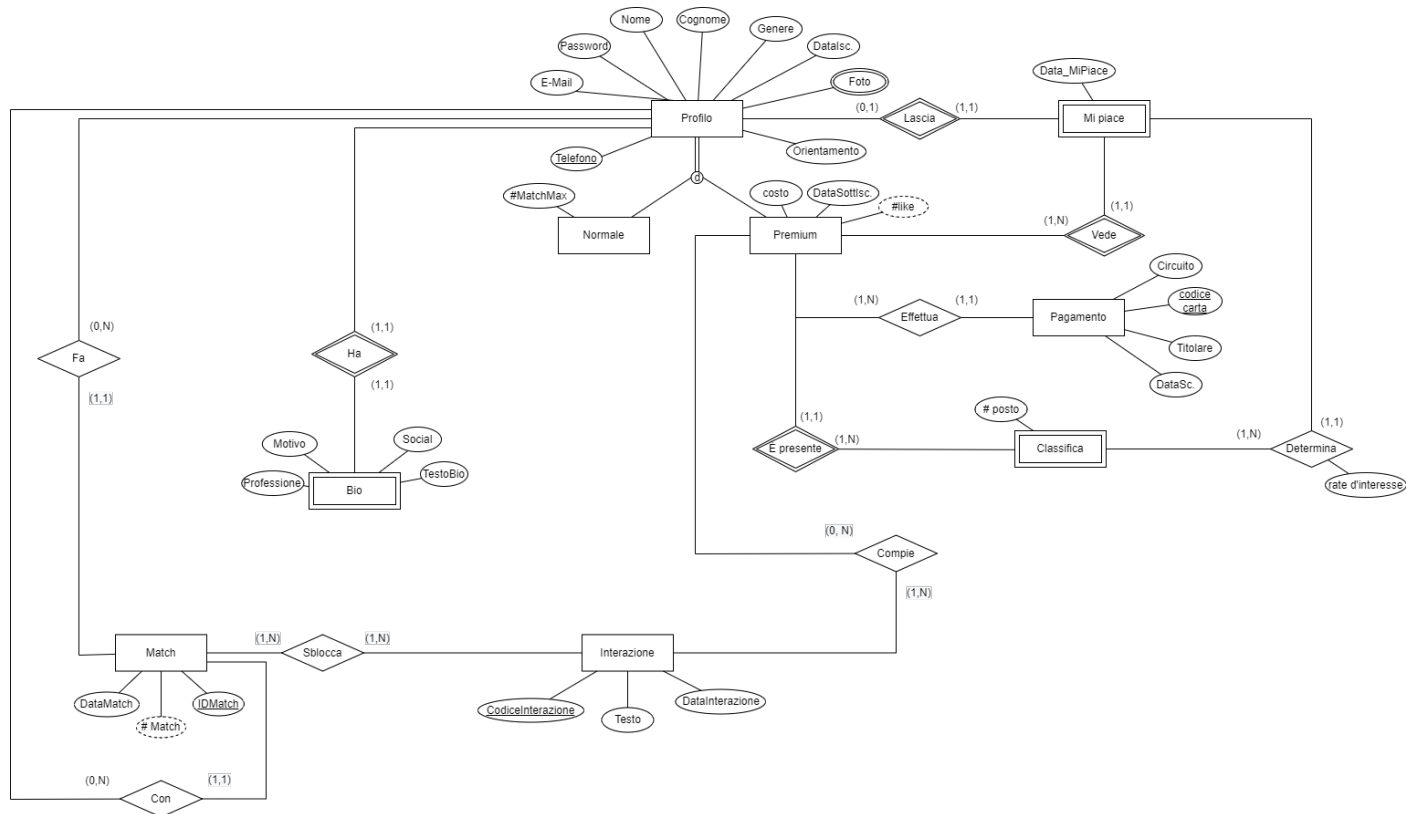
Glossario dei termini

Termine	Significato
Profilo	Identità che contiene l'insieme di dati di un utente che permette di essere visualizzato ad altri utenti e di accedere all'interno della piattaforma
Normale	Utente base della piattaforma d'incontri, che ha dei limiti al proprio profilo
Premium	Utente che ha sottoscritto un abbonamento (di solito mensile/annuale) che permette di avere accesso a particolari funzioni non disponibili per un normale utente
Pagamento	Mezzo per cui si acquista un determinato bene
Classifica	Graduatoria di tutti gli utenti premium
Mi piace	Apprezzamento di un profilo da un punto di vista soggettivo
Bio	Si tratta di poche righe in cui è scritta una sorta di presentazione della persona
Match	La compatibilità tra due utenti, che hanno gli stessi interessi e orientamento

2. Progettazione concettuale della base di dati

Schema EER

In seguito, dopo l'astrazione dei concetti, si passa alla progettazione concettuale, ottenendo il seguente schema EER:



Vincoli non esprimibili nello schema

Oltre a ciò che è evidente all'interno dello schema EER, si tiene conto anche di vincoli non esprimibili:

- L'attributo "NumeroPosti" dell'entità Classifica ha un valore massimo di 10.
- L'attributo "Testo" dell'entità Bio deve avere un valore massimo di caratteri minore o uguale a 500.
- L'attributo "Social" dell'entità Bio può contenere un unico valore.
- L'attributo "Testo" dell'entità Interazione deve avere un valore massimo di caratteri minore o uguale a 1000.
- L'attributo "NumeroMatchMax" dell'entità Normale ha un valore massimo di 100.

Dizionario delle entità

Legenda: sotto-entità, attributo multivalore, attributo ridondante, entità debole, chiave candidata

Entità	Descrizione	Attributi	Identificatore
Profilo	Identità di un utente sulla piattaforma	<ul style="list-style-type: none"> - Telefono - E-mail - Password - Nome - Cognome - Genere - DataIscrizione - Foto - Orientamento 	Telefono
Normale	Profilo di base	<ul style="list-style-type: none"> - NumeroMatchMax 	-
Premium	Profilo con particolari caratteristiche	<ul style="list-style-type: none"> - Costo - DataSottoscrizione - NumeroLike 	-
Mi piace	Apprezzamento di un profilo da un punto di vista soggettivo	<ul style="list-style-type: none"> - Data_MiPiace 	-
Pagamento	Entità che conserva i dati di una carta per acquistare un determinato bene	<ul style="list-style-type: none"> - CodiceCarta - Circuito - Titolare - DataScadenza 	CodiceCarta
Classifica	Graduatoria degli utenti premium	<ul style="list-style-type: none"> - NumeroPosto 	-
Bio	Possibilità di scrivere una breve presentazione del profilo	<ul style="list-style-type: none"> - Professione - Motivo - Social - TestoBio 	-
Match	Compatibilità fra due profili	<ul style="list-style-type: none"> - IDMatch - Data - NumeroMatch 	-
Interazione	Messaggio di testo che permette di comunicare tra due utenti	<ul style="list-style-type: none"> - CodiceInterazione - TestoInterazione - DataInterazione 	CodiceInterazione

Dizionario delle relazioni

Relazione	Descrizione	Entità coinvolte	Attributi
Lascia	Un profilo emette un like ad un ulteriore profilo	Profilo(0,1) Mi Piace(1,1)	-
Vede	Un profilo premium vede i mi piace ricevuti al proprio profilo	Premium(1,N) Mi Piace(1,1)	
Ha	All'interno del profilo l'utente scrive una bio	Profilo(1,1) Bio(1,1)	-
È presente	Un profilo premium è presente in una classifica	Premium(1,1) Classifica(1,1)	-
Determina	Il numero di mi piace determina il posto in classifica calcolandone il rate d'interesse	Mi Piace(1,1) Classifica(1,N)	Rate d'interesse
Fa	Un profilo compie un match	Profilo(0,N) Match(1,1)	
Con	Un profilo fa match con un ulteriore profilo	Match(1,1) Profilo(0,N)	
Compie	Un profilo premium interagisce con un profilo	Premium(0,N) Interazione(1,N)	-
Sblocca	Un profilo una volta avvenuto il match può interagire con l'utente	Match(1,N) Interazione(1,N)	-

3. Definizione delle procedure per la gestione della base di dati

Tavola dei volumi

Definiamo di seguito la tavola dei volumi della base di dati

Concetto	Tipo	Carico Applicativo
Profilo	E	200
Normale	E	120
Premium	E	80
Mi piace	E	350
Classifica	E	5
Pagamento	E	320
Bio	E	200
Match	E	150
Interazione	E	600
<i>Lascia</i>	R	400
<i>Vede</i>	R	200
<i>È presente</i>	R	20
<i>Effettua</i>	R	400
<i>Determina</i>	R	400
<i>Compie</i>	R	160
<i>Ha</i>	R	400
<i>Fa</i>	R	300
<i>Con</i>	R	300
<i>Sblocca</i>	R	300

Tavola delle operazioni

	Operazione	Tipo	Frequenza
1	Selezionare gli utenti di un dato match e data	I	3/sett.
2	Leggere l'id di un match legato ad una interazione	I	2/mm
3	Creazione di un nuovo profilo	I	20/sett.
4	Sbloccamento di un match da un profilo che si è iscritto alla piattaforma il 16/07/2020	B	3/mm
5	Selezionare il motivo di iscrizione di un utente alla piattaforma	I	1/mm
6	Visualizzare data sottoscrizione di un utente premium	I	2/sett.
7	Selezionare il rate d'interesse e il numero di posto in classifica di un dato profilo utente	I	2/mm
8	Profili raggruppati per nome in senso decrescente che hanno orientamento 'Eterosessuale'	I	4/mm
9	Lettura dei numeri di match di tutti gli utenti della piattaforma	B	5/mm
10	Modifica password di un profilo	I	1/mm
11	Inserimento di un like lasciato da un profilo con una e-mail che inizia con 'L'	I	2/mm
12	Selezionare utenti con stessa professione e che hanno impostato il loro genere a 'Donna'	I	1/sett.
13	Selezionare tutti gli ID match con data oltre il 15/03/2020	I	3/sett.
14	Password ed e-mail di tutti gli utenti normali	B	1/mm
15	Trovare codice interazione attraverso la data di un match	I	3/sett.
16	Selezionare il titolare di tutti i pagamento che non hanno circuito VISA	I	4/mm
17	Selezionare il testo di un'interazione sbloccata da un match non in data 25/12/2019	I	2/sett.
18	Profili premium ordinati per cognome	I	3/mm
19	Numero di mi piace di un utente premium	I	5/mm

20	Utenti di genere maschile raggruppati per orientamento che hanno almeno una foto	I	3/mm
21	Numero di utenti con stesse motivazioni	B	4/mm
22	Profili che non hanno effettuato un match con un dato profilo	I	4/mm

4. Progettazione logica

Analisi delle ridondanze

La seguente base di dati presenta ben due dati ridondanti cioè il “NumeroMatch” e “NumeroLike” dell’entità “Match” e “Premium”. Dunque, potremmo ottenere il numero di Match effettuati di un profilo attraverso il conto delle partecipazioni “Profilo Fa Match” e “Match Con Profilo” ed il numero di Like attraverso il conto delle partecipazioni “Premium vede Mi piace”. Supponendo che l’attributo abbia un peso di 4 byte, e considerando il volume dell’entità “Profilo” che è uguale a 200, il dato andrà ad occupare uno spazio totale che è di circa **800 byte**. Nel caso di “Premium” il suo volume è pari 80, quindi il dato andrà ad occupare uno spazio totale di circa **320 byte**.

Per determinare se conservare o meno i due dati ridondanti, si va a calcolare, per le operazioni che lo coinvolgono, la differenza nel numero di accessi con o senza ridondanza.

Tavola degli Accessi

Le operazioni che coinvolgono il dato ridondante (NumeroMatch) sono:

Op.4, Op.9

Operazione		Tipo	Frequenza
4	Avvenimento di un match grazie alla stesura di una bio in un dato profilo	B	2/mm
9	Lettura dei numeri di match di tutti gli utenti della piattaforma	B	5/mm

Le operazioni che coinvolgono il dato ridondante (NumeroLike) sono:

Op.11, Op.19

Operazione		Tipo	Frequenza
11	Inserimento di un like lasciato da un profilo con una e-mail che inizia con ‘L’	I	2/mm
19	Lista dei like di un utente premium	I	5/mm

Analisi delle ridondanze

Per NumeroMatch

Operazione 4

Calcolo con ridondanza				Calcolo senza ridondanza			
Concetto	Costrutto	Accesso	Tipo di accesso	Concetto	Costrutto	Accesso	Tipo di accesso
Profilo	E	1	S	Profilo	E	1	S
Con	R	2	S	Con	R	2	S
Fa	R	2	S	Fa	E	2	S
Match	R	2	L				
Match	E	2	S				
Totale	$[2 + (1+2+2+2)*2]*3 = 48/\text{mese}$			$[(1+2+2)*2]*3 = 30/\text{mese}$			

Operazione 9

Calcolo con ridondanza				Calcolo senza ridondanza			
Concetto	Costrutto	Accesso	Tipo di accesso	Concetto	Costrutto	Accesso	Tipo di accesso
Match	E	200	L	Match	L	200	L
				Con	L	$200*2$	L
				Fa	L	$200*2$	L
Totale	$[200]*5 = 1000/\text{mese}$			$[200+400+400]*5 = 5000/\text{mese}$			

Totale accessi con ridondanza

$$(48 + 1000) \backslash \text{mm} =$$

$$1048/\text{mm} + 800 \text{ byte}$$

Totale accessi senza ridondanza

$$(30 + 5000) \backslash \text{mm} =$$

$$5030/\text{mm}$$

Conviene mantenere il dato ridondante NumeroDiMatch

Poiché il numero di accessi mantenendo il dato ridondante è molto minore rispetto al numero di accessi senza ridondanza, conviene mantenere il dato ridondante.

Analisi delle ridondanze

Per NumeroLike

Operazione 11

Calcolo con ridondanza				Calcolo senza ridondanza			
Concetto	Costrutto	Accesso	Tipo di accesso	Concetto	Costrutto	Accesso	Tipo di accesso
Profilo	E	1	S	Profilo	E	1	S
Lascia	R	2	S	Lascia	R	2	S
Mi piace	E	2	S	Mi piace	E	2	S
Vede	R	4	S	Vede	R	4	S
Premium	E	4	L				
Premium	E	4	S				
Totale	$[4 + (1+2+2+4+4)*2] * 2 = 60/\text{mese}$			$[(1+2+2+4)*2] * 2 = 36/\text{mese}$			

Operazione 19

Calcolo con ridondanza				Calcolo senza ridondanza			
Concetto	Costrutto	Accesso	Tipo di accesso	Concetto	Costrutto	Accesso	Tipo di accesso
Premium	E	1	L	Premium	E	1	L
				Vede	R	2	L
				Mi Piace	E	2	L
				Lascia	R	$\cong 4$	L
Totale	$[(1 * 5)] = 5/\text{mese}$			$[(1+2+2+4)] * 5 = 45/\text{mese}$			

Totale accessi con ridondanza

$$(60 + 5) \backslash \text{mm} =$$

$$65/\text{mm} + 320 \text{ byte}$$

Totale accessi senza ridondanza

$$(36+45) \backslash \text{mm} =$$

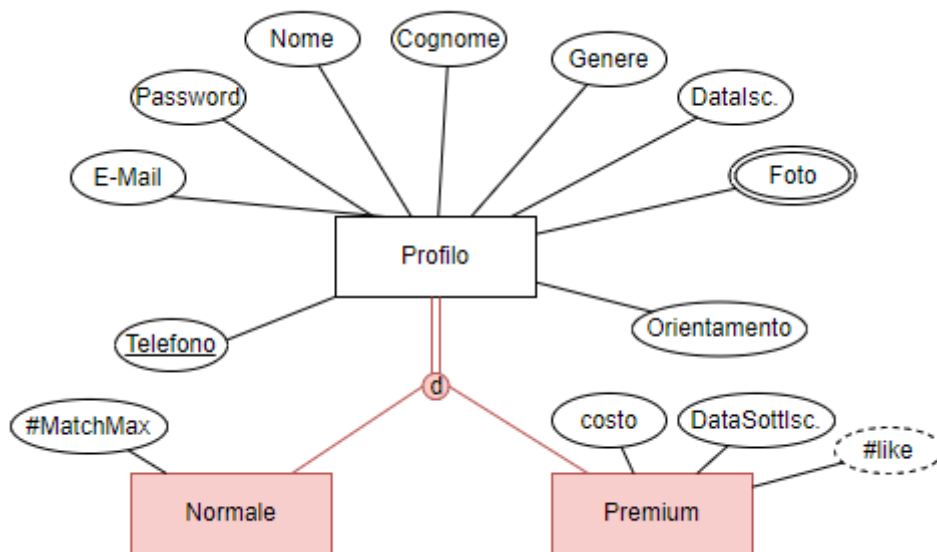
$$81/\text{mm}$$

Conviene mantenere il dato ridondante NumeroDiLike

Poiché il numero di accessi mantenendo il dato ridondante è minore rispetto al numero di accessi senza ridondanza, conviene mantenere il dato ridondante.

Eliminazione delle gerarchie

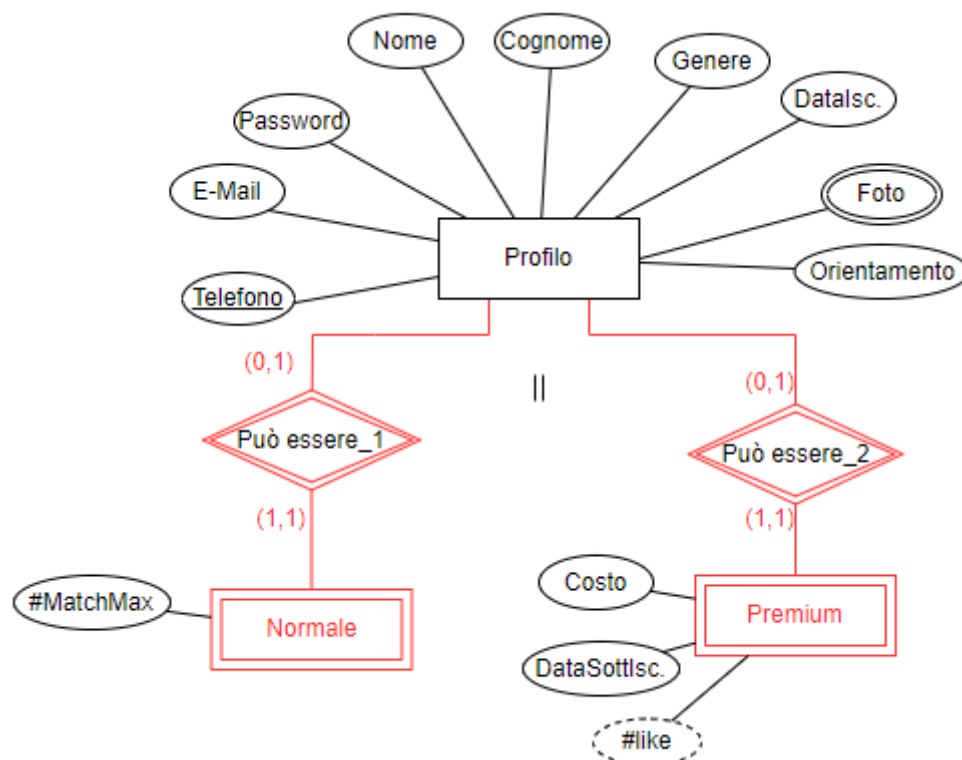
Dallo schema EER di partenza, risulta all'entità "Profilo" una specializzazione totale e disgiunta



Nella seguente fase, oltre all'analisi del dato ridondante, bisogna trovare un metodo risolutivo che vada ad eliminare tale gerarchia.

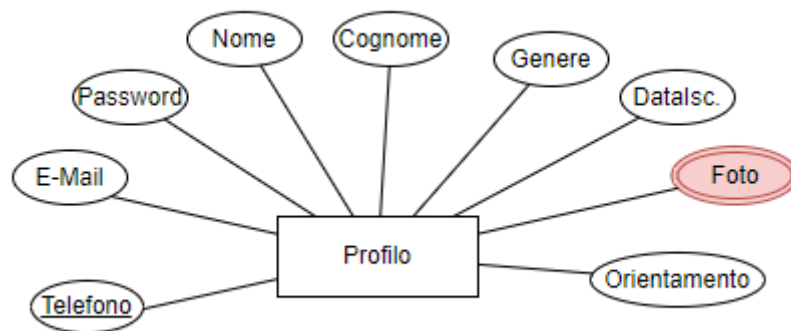
Il metodo scelto sarà la creazione di due nuove relazioni, collegate all'entità "Profilo" e deboli rispetto quest'ultimo.

Questa scelta è dovuta dal fatto che le entità figlie possiedono a loro volta degli attributi che non possono essere inglobate dal padre, in più l'entità Premium ha una vera e propria relazione che è legata solo a quel tipo di entità, ed eviteremo inoltre anche valori NULL

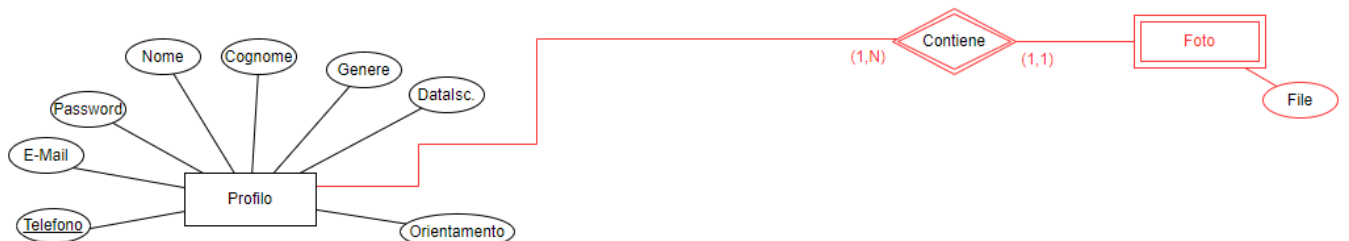


Eliminazione dell'attributo multivalued

L'entità "Profilo", oltre a contenere una gerarchia, contiene anche l'attributo multivalued "Foto"

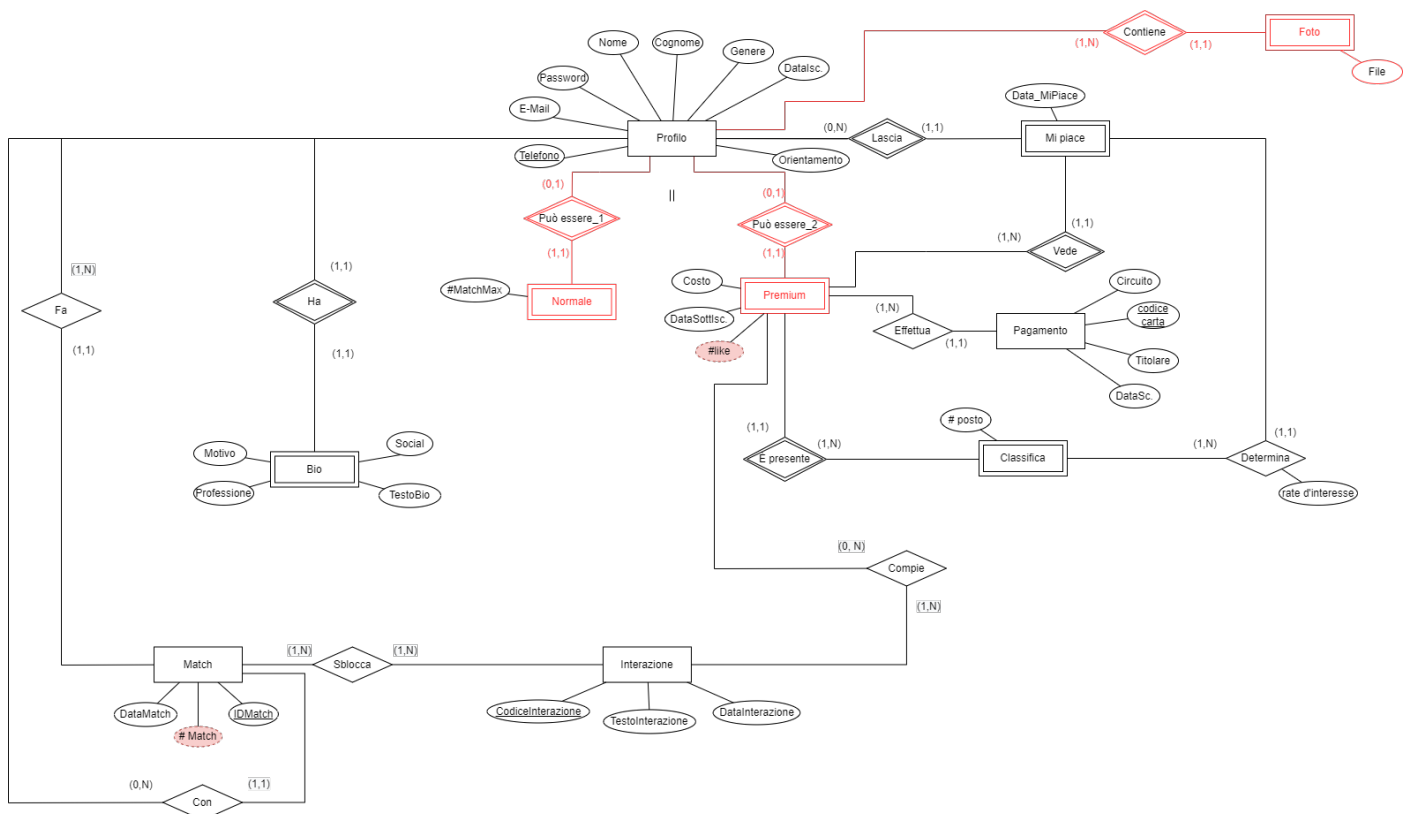


Nella progettazione logica l'attributo multivalued, al contrario della progettazione concettuale, si sceglie di definirlo con un'entità, debole rispetto profilo, tramite una relazione



Schema EER Ristrutturato

In seguito alla progettazione logica, lo schema EER sarà raffigurato in modo seguente:



Schema Relazionale

In seguito alla ristrutturazione dello schema EER, si procede al mapping della base di dati

Profilo (Telefono, E-mail, Password, Nome, Cognome, Genere, DataIscrizione, Orientamento)

Foto (Profilo ↑, File)

Normale (Profilo ↑, MatchMax)

Premium (Profilo ↑, Costo, DataSottoscrizione, Numero_di_like)

Mi Piace(Profilo ↑, Premium ↑, Classifica ↑, Data_mi_piace, rate_d'interesse)

Pagamento(Codice carta, Circuito, Titolare, DataScadenza)

Classifica(Premium ↑, Numero_Posto)

Bio (Profilo ↑, Testo_Bio, Professione, Motivo, Social)

Match(ID, Data_match, Numero_di_match)

Interazione (Codice, Testo_interazione, Data_interazione)

Effettua (Premium ↑, Pagamento ↑)

Fa(Profilo ↑, Match ↑)

Con(Match ↑, Profilo ↑)

Sblocca (Match ↑, Interazione ↑)

Compie (Premium ↑, Interazione ↑)

Con (Profilo ↑, IDMATCH ↑)

Sblocca (Match ↑, Interazione ↑)

Compie (Premium ↑, Interazione ↑)

Normalizzazione

La base di dati si presenta già normalizzata.

È infatti in **prima forma normale** in quanto tutti gli attributi sono atomici dopo la ristrutturazione. È in **seconda forma normale** perché, oltre ad essere già in 1NF, quando è presente una chiave primaria composta da più attributi tutte le dipendenze funzionali che la riguardano sono piene e non parziali. È in **terza forma normale** perché, oltre ad essere già in 2NF, in tutte le tabelle non sono presenti dipendenze transitive fra attributi non chiave e la chiave primaria.

5. Realizzazione della base di dati con MySQL

Di seguito, la realizzazione della base di dati con istruzioni MySQL

```
1 • drop database if exists incontri;
2
3 • CREATE database incontri;
4
5 • USE incontri;
6
7 • create table profilo
8   ( telefono bigint not null,
9     email varchar(45) not null,
10    nome varchar(15) not null,
11    cognome varchar(15) not null,
12    password varchar(25) not null,
13    genere varchar(20) default 'non specificato',
14    dataiscrizione date not null,
15    orientamento varchar(20) not null,
16    primary key(telefono)
17  );
18
19 • create table foto
20   ( profilo bigint not null,
21     file varchar(20),
22     primary key(profilo),
23     foreign key(profilo) references profilo(telefono)
24   );
25
26 • create table bio (
27   profilo bigint not null,
28   testo_bio varchar(50) default 'non specificato',
29   professione varchar(20) default 'non specificato',
30   motivo varchar(30) default 'non specificato',
31   social varchar(20) default 'non specificato',
32   foreign key(profilo) references profilo(telefono)
33 );
34
35
```

```
36 • ⊖ create table normale (  
37     profilo bigint primary key,  
38     maxmatch int not null,  
39     foreign key(profilo) references profilo(telefono)  
40 );  
41  
42 • ⊖ create table premium (  
43     profilo bigint primary key,  
44     costo double not null,  
45     datasottoscrizione date not null,  
46     numero_di_like int default 0,  
47     foreign key(profilo) references profilo(telefono)  
48 );  
49  
50 • ⊖ create table mi_piace (  
51     premium bigint primary key,  
52     profilo bigint not null,  
53     classifica bigint not null,  
54     data_mi_piace date not null,  
55     ratedinteresse double not null,  
56     foreign key(premium) references premium(profilo)  
57 );  
58  
59 • ⊖ create table pagamento (  
60     numero_carta int primary key,  
61     circuito varchar(20) not null,  
62     titolare varchar(45) not null,  
63     datascadenza date not null  
64 );  
65  
66 • ⊖ create table classifica (  
67     premium bigint not null,  
68     numero_posto int not null,  
69     foreign key(premium) references premium(profilo)  
70 );  
71
```



```
72 • ⊖ create table match_ (  
73     id int primary key,  
74     datamatch date not null,  
75     numero_di_match int not null  
76 );  
77  
78 • ⊖ create table interazione (  
79     codiceinterazione int primary key,  
80     testo_interazione varchar(50) not null,  
81     data_interazione date not null  
82 );  
83  
84  
85 • ⊖ create table effettua(  
86     premium bigint not null,  
87     pagamento int not null,  
88     foreign key(premium) references premium(profilo),  
89     foreign key(pagamento) references pagamento(numero_carta)  
90 );  
91  
92 • ⊖ create table fa(  
93     profilo bigint not null,  
94     match_ int not null,  
95     foreign key(profilo) references profilo(telefono),  
96     foreign key(match_) references match_(id)  
97 );  
98  
99 • ⊖ create table con(  
100     match_ int not null,  
101     profilo bigint not null,  
102     foreign key(profilo) references profilo(telefono),  
103     foreign key(match_) references match_(id)  
104 );  
105
```

```
106 • ⊖ create table sblocca(  
107     match_ int not null,  
108     interazione int not null,  
109     foreign key(match_) references match_(id),  
110     foreign key(interazione) references interazione(codiceinterazione)  
111 );  
112  
113 • ⊖ create table compie(  
114     premium bigint not null,  
115     interazione int not null,  
116     foreign key(premium) references premium(profilo),  
117     foreign key(interazione) references interazione(codiceinterazione)  
118 );  
---
```

6. Implementazione query SQL

Operazione 1:

//SELEZIONARE GLI UTENTI DI UNA DATA MATCH E DATA

```
SELECT nome, cognome, datamatch
FROM match_ AS m, profilo AS p, fa AS f
WHERE p.telefono=f.profilo AND
      m.id= ? AND
      m.id=f.match_
```

Operazione 2:

//Leggere l'id di un match legato ad un'interazione

```
SELECT s.match_
FROM interazione as i, sblocca AS s
WHERE i.codiceinterazione= ? AND
      s.interazione=i.codiceinterazione
```

Operazione 3:

//Creazione di un nuovo profilo

```
insert into profilo(telefono, e-mail, nome, cognome, password, genere, dataiscrizione,
orientamento)
```

```
VALUES (
    ?, ?, ?, ?, ?, ?, ?
);
```

Operazione 4:

//Sblocco di un match da un profilo che si è iscritto alla piattaforma il 16.07.2020

```
SELECT m.id
FROM match_ AS m, profilo AS p, fa AS f
WHERE dataiscrizione=20200716 AND
      f.profilo=p.telefono AND
      m.id=f.match_
```

Operazione 5:

//Selezionare il motivo d'iscrizione di un utente alla piattaforma

```
SELECT b.motivo
FROM profilo AS p, bio AS b
WHERE b.profilo=p.telefono AND
      p.telefono= ?
```

Operazione 6:

//Visualizzare data sottoscrizione di un utente premium

```
SELECT datasottoscrizione
FROM premium AS pm, profilo AS p
WHERE pm.profilo=p.telefono AND
      p.telefono= ?
```

Operazione 7:

//Selezionare il rate d'interesse e il numero di posto in classifica di un dato profilo premium

```
SELECT ratedinteresse, numero_posto
FROM premium AS pm, profilo AS p, classifica as c, mi_piace as mp
WHERE c.premium=mp.classifica AND
      pm.profilo=mp.premium AND
      pm.profilo=p.telefono AND
      p.telefono= ?
```

Operazione 8:

//Profili raggruppati per nome in senso decrescente che hanno orientamento 'Eterosessuale'

```
SELECT nome, cognome
FROM profilo
WHERE orientamento='etero'
ORDER BY nome DESC
```

Operazione 9:

//Lettura di numero di match di tutti gli utenti

```
SELECT nome, cognome, count(*) as c
FROM profilo AS p
WHERE EXISTS(SELECT *
             FROM fa, con
             WHERE fa.profilo=p.telefono OR
                   con.profilo=p.telefono)
GROUP BY p.telefono
```

Operazione 10:

//Modifica password di un profilo

```
UPDATE profilo SET password= ?
WHERE telefono= ?
```

Operazione 11:

//Seleziona un like lasciato da un profilo con l'e-mail che comincia con 'L'

```
Insert into mi_piace(profilo, premium, classifica,data_mi_piace,ratedinteresse)
values(3465473334, ?, ?, 20220301, 50)
```

Operazione 12:

//Selezionare utenti con stessa professione e di genere 'Donna'

```
SELECT DISTINCT  nome, cognome
FROM profilo AS p, bio AS b
WHERE EXISTS( SELECT *
              FROM bio AS b1
              WHERE b.professione=b1.professione) AND
              p.genere= 'Donna'
GROUP BY b.professione
```

Operazione 13:

//Selezionare tutti gli id match con data oltre il 15-5-2020

```
SELECT id
FROM match_
WHERE data>2020515
```

Operazione 14:

//Password ed email di tutti gli utenti normali

```
SELECT MD5(password) AS password, email
FROM profilo AS p, normale AS n
WHERE n.profilo=p.telefono
```

Operazione 15:

//Codice interazione attraverso una data di un match

```
SELECT codiceinterazione
FROM match_ as m, interazione as i, sblocca as s
WHERE datamatch= ? AND
      i.codiceinterazione=s.interazione AND
      m.id=s.match_
```

Operazione 16:

//Selezionare il titolare di tutti i pagamenti che non hanno circuito VISA

```
SELECT p1.titolare
FROM pagamento as p1
WHERE 'VISA' NOT IN(SELECT p2.circuito
                    FROM pagamento as p2
                    WHERE p1.numero_carta = p2.numero_carta)
```

Operazione 17:

//Selezionare il testo di un'interazione sbloccata da un match non in data 25-12-2019

```
SELECT testo_interazione
FROM interazione as i, sblocca as s
WHERE NOT EXISTS(SELECT *
                  FROM match_ as m
                  WHERE m.datamatch=2019-25-12) AND
                  s.interazione=i.codiceinterazione
```

Operazione 18:

//I profili premium ordinati per cognome che hanno avuto un posto nella classifica

```
SELECT nome, cognome
FROM profilo AS p, premium AS pm, classifica AS c
WHERE c.premium=pm.profilo AND
      pm.profilo=p.telefono
ORDER BY p.cognome
```

Operazione 19:

//Numero di mi piace di un utente premium

```
SELECT count(*) AS co
FROM mi_piace as mp, premium AS pm
WHERE pm.profilo= ? AND
      pm.profilo=mp.premium
GROUP BY pm.profilo
```

Operazione 20:

//Utenti maschili raggruppati per tipo di orientamento che hanno postato almeno una foto

```
SELECT nome, cognome
FROM profilo AS p
WHERE p.genere='Uomo' AND EXISTS(SELECT *
                                FROM foto as f
                                WHERE f.profilo=p.telefono)
GROUP BY p.orientamento
```

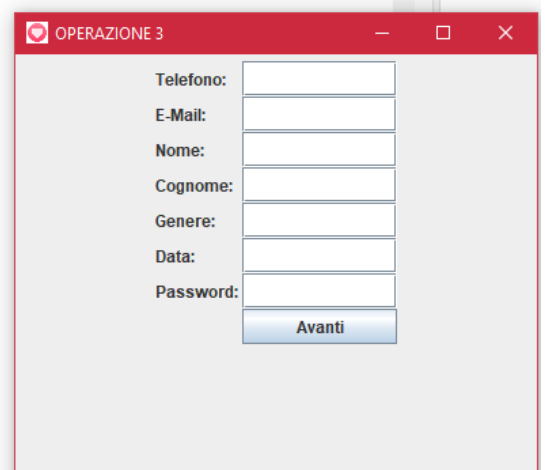
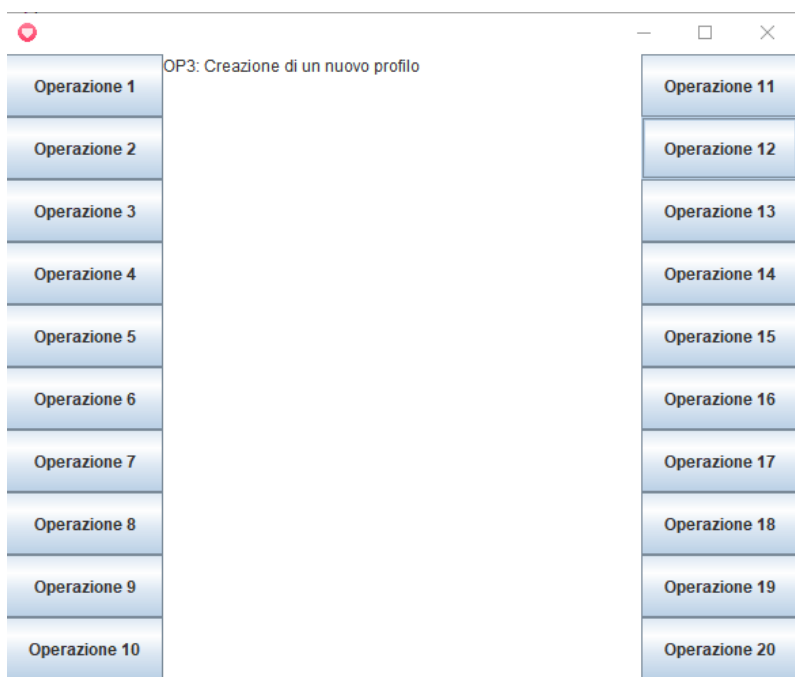
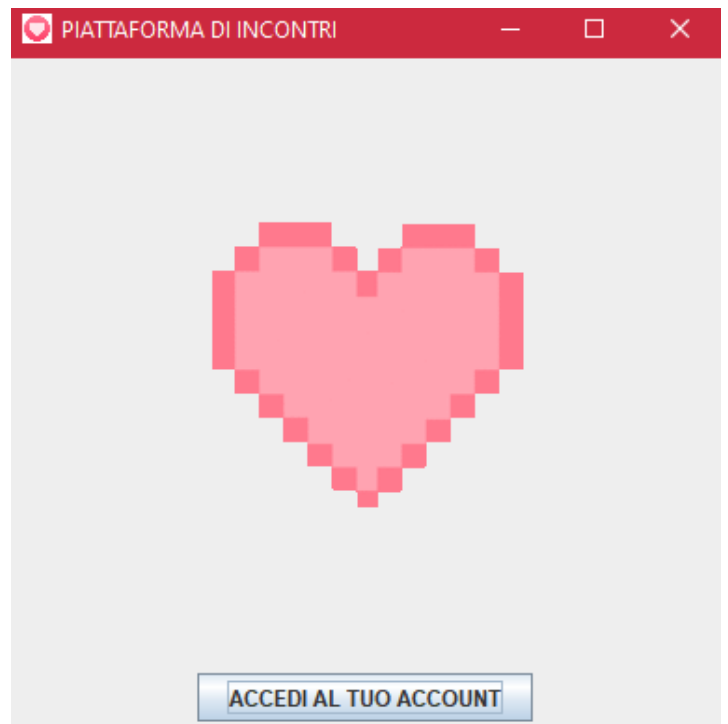
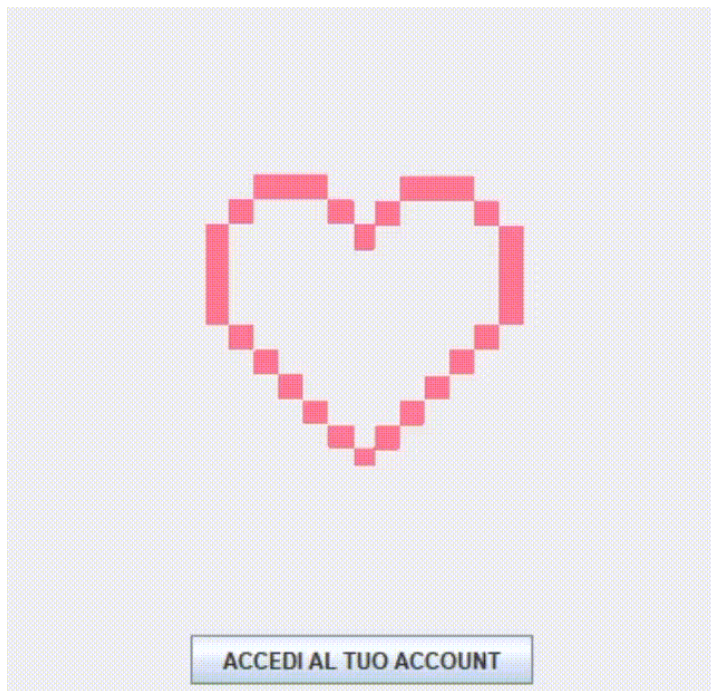
Operazione 21:

//Numero di utenti con stesse motivazioni

```
SELECT motivo, count(*) as c
FROM bio as b
GROUP BY motivo
```


7. Test dell'applicazione Java

All'avvio dell'applicazione, si simula l'accesso alla piattaforma nella quale poi è possibile effettuare le operazioni



OP6: Visualizzare data sottoscrizione di un utente premium

Operazione 1	Operazione 11
Operazione 2	Operazione 12
Operazione 3	Operazione 13
Operazione 4	Operazione 14
Operazione 5	Operazione 15
Operazione 6	Operazione 16
Operazione 7	Operazione 17
Operazione 8	Operazione 18
Operazione 9	Operazione 19
Operazione 10	Operazione 20

OPERAZIONE 6

Telefono:

Avanti

Data di sottoscrizione al servizio premium: 2020-08-20