

Progettazione e sviluppo di una Base di dati relazionale per la gestione di una Galleria Fotografica Geolocalizzata

Michela Pollio

Elisa Tiberio

N86003697

N86003682

Indice

1	Descrizione del progetto	4
1.1	Traccia	4
2	Progettazione Concettuale	5
2.1	Introduzione	5
2.2	Class Diagram UML	6
2.3	Ristrutturazione del class Diagram	7
2.3.1	Analisi delle ridondanze	7
2.3.2	Analisi degli identificatori	7
2.3.3	Rimozione degli attributi multipli	7
2.3.4	Partizione/Accorpamento delle Generalizzazioni/Specializzazioni	7
2.3.5	Attributi Composti	7
2.3.6	Attributi Derivabili	7
2.4	Class Diagram Ristrutturato	8
2.5	Diagramma ER	9
2.6	Dizionario delle classi	10
2.7	Dizionario delle Associazioni	11
3	Schema Logico	12
4	Progettazione Fisica	14
4.1	Definizione delle tabelle	14
4.1.1	Creazione Schema	14
4.1.2	Utente	14
4.1.3	Dispositivo	14
4.1.4	Foto	15
4.1.5	Luogo	15
4.1.6	LuogoFoto	15
4.1.7	Soggetto	16
4.1.8	SoggFoto	16
4.1.9	Tag	16
4.1.10	Album	16
4.1.11	FotoAlbum	17
4.1.12	AlbumUtente	17
4.2	Definizione dei Trigger	17
4.2.1	Trigger - Creazione album personale	17
4.2.2	Trigger - Aggiungi alla galleria	18
4.2.3	Trigger - Aggiungi data corrente album	18
4.2.4	Trigger - Aggiungi data corrente foto	18
4.2.5	Trigger - Aggiungi partecipazione ad album	19
4.2.6	Trigger - Cancellazione foto	19
4.3	Definizione delle funzioni	20

4.3.1	Function - getsameplace	20
4.3.2	Function - getsamesubject	20
4.4	View	21
5	Popolamento della Base di Dati	22

Capitolo 1

Descrizione del progetto

1.1 Traccia

Questo progetto consiste nello sviluppo di una base di dati relazionale, per la gestione di collezioni fotografiche geolocalizzate condivise. Ogni **utente** che accede al sistema ha la possibilità di poter accedere alla propria **galleria personale**, composta esclusivamente dalle foto scattate da lui. L'utente potrà inoltre accedere a **gallerie condivise** con altri utenti del sistema. Le **foto** presenti nella base di dati sono identificate univocamente e descritte tenendo conto della **data** dello scatto, del **dispositivo** con cui è stata scattata, dei **soggetti** o utenti presenti all'interno della foto e, se presente, del **luogo** in cui è stata scattata.

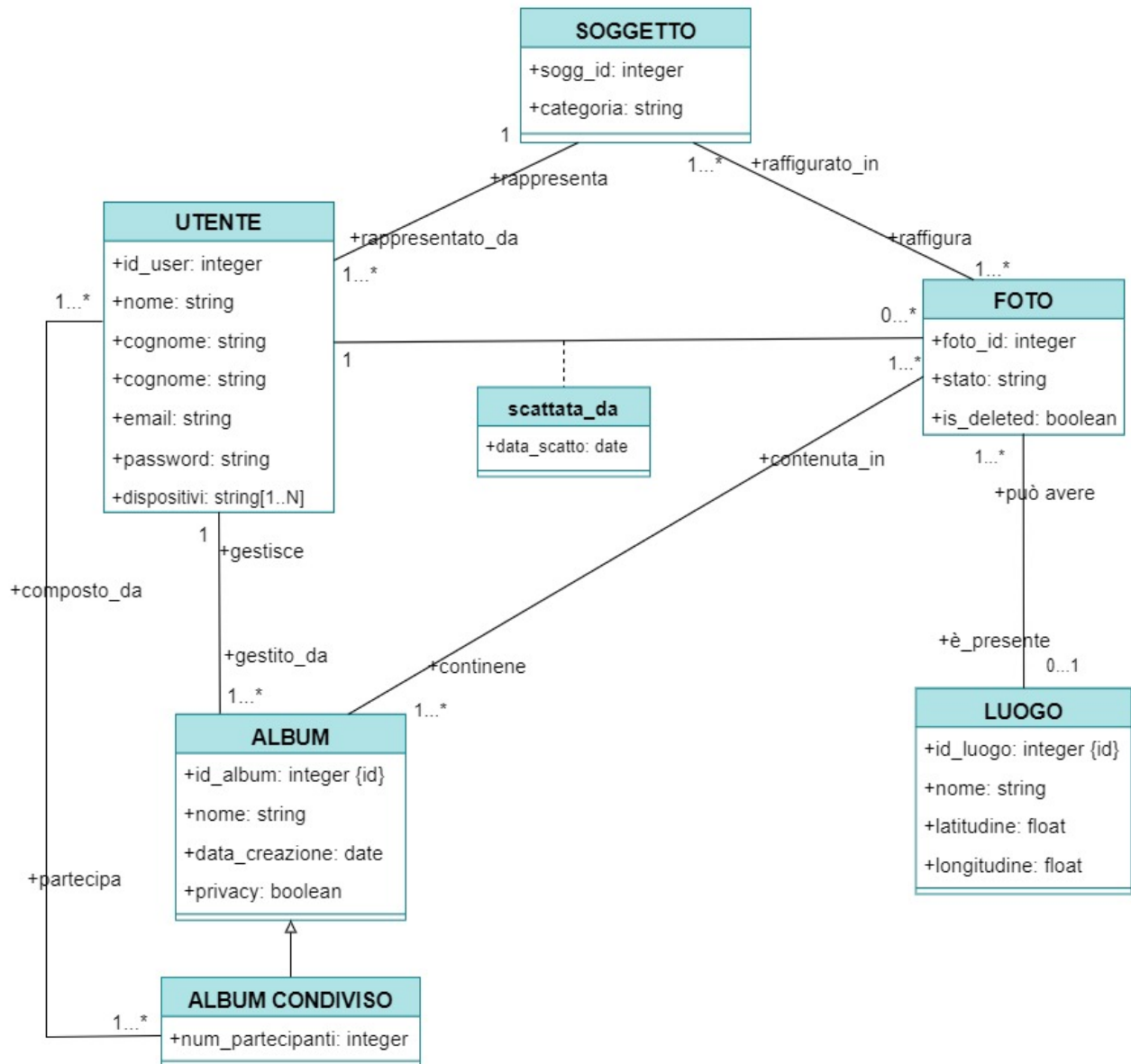
Capitolo 2

Progettazione Concettuale

2.1 Introduzione

Il diagramma è composto inizialmente da 5 classi: La classe **utente** contenente tutti gli attributi relativi alla persona fisica che accederà al sistema (id_utente, nome, cognome, email, password, numero di dispositivi collegati all'account), collegata alla classe **foto** (id_foto, stato, is_deleted) tramite la relazione che identifica lo scatto della fotografia da parte dell'utente. La classe foto a sua volta è associata alle classi **Soggetto** (sogg_id, categoria) e la classe **Luogo** (id_luogo, nome, latitudine, longitudine) tramite due relazioni distinte che, rispettivamente, rappresentano la presenza di soggetti all'interno della foto e la possibile presenza di un luogo dove quest'ultima è stata scattata. La foto, inoltre, è in relazione con la classe **Album** (id_album, nome, data_creazione, privacy), in quanto contenuta all'interno di esso. La classe **Album Condiviso** (album a cui possono partecipare più utenti) è stata raffigurata come specializzazione della classe Album.

2.2 Class Diagram UML



2.3 Ristrutturazione del class Diagram

2.3.1 Analisi delle ridondanze

Il class Diagram non presenta ridondanze.

2.3.2 Analisi degli identificatori

Ogni classe presenta un attributo che la identifica univocamente, in particolare: Per la classe foto è l'attributo *foto_id*, per Utente è *id_user*, per la classe Soggetto è *sogg_id*, per la classe Luogo è *id_luogo*, per la classe Album è *id_album*.

2.3.3 Rimozione degli attributi multipli

L'unico attributo multiplo è l'attributo **dispositivo** contenuto all'interno dell'entità **Utente**. Infatti, ogni utente può avere associati diversi dispositivi attraverso i quali scattare le foto. Per rimuovere tale attributo abbiamo optato per l'aggiunta di un'ulteriore classe **DISPOSITIVO** posta in relazione con la classe **Utente** e con la classe **Foto** per quanto spiegato prima.

2.3.4 Partizione/Accorpamento delle Generalizzazioni/Specializzazioni

L'unico caso di specializzazione presente riguarda la classe **ALBUM CONDIVISO** in quanto specializzazione della classe **ALBUM**. Tale specializzazione è stata risolta accorpando la classe figlia nella classe padre.

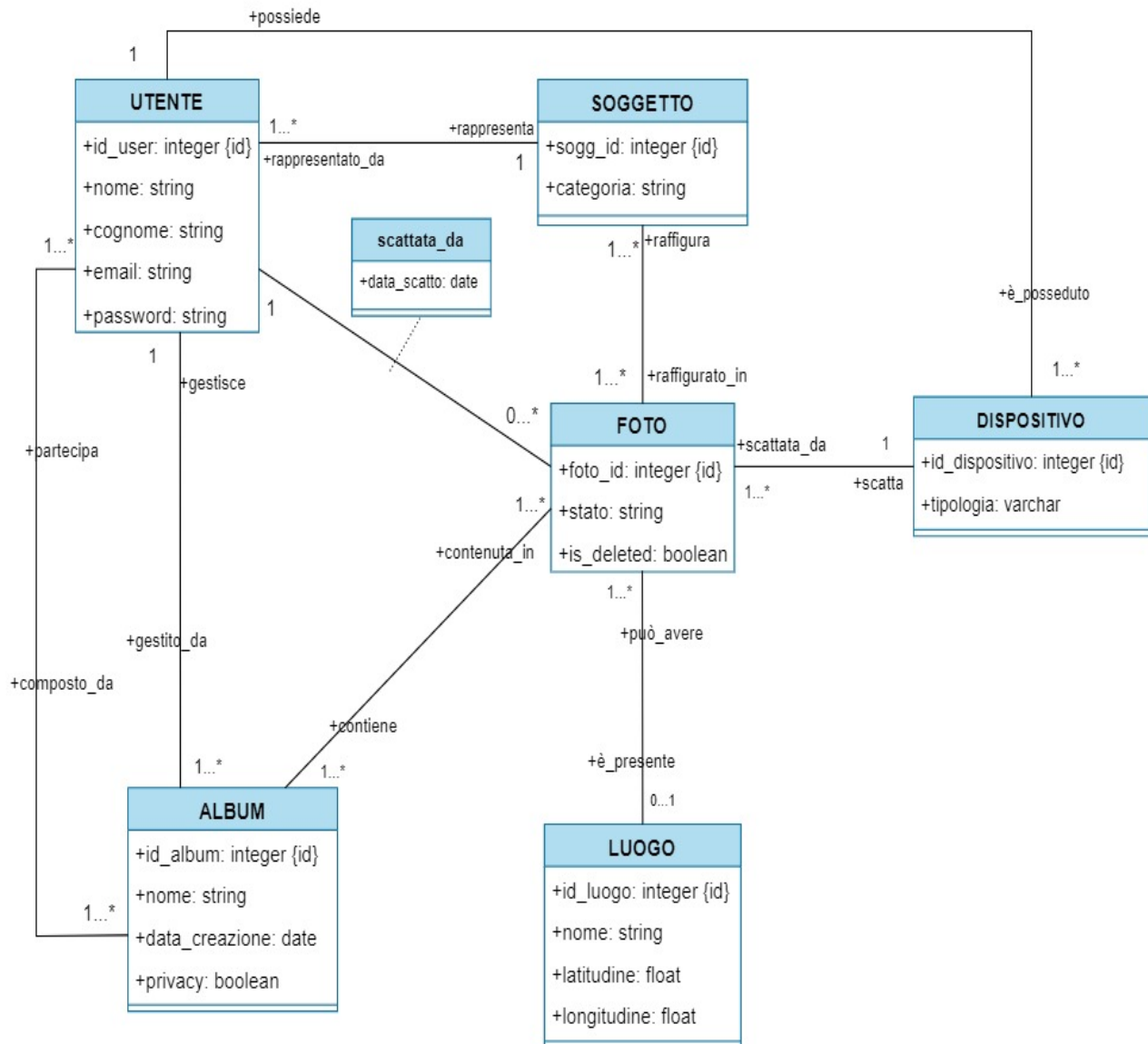
2.3.5 Attributi Composti

La classe non presenta attributi composti.

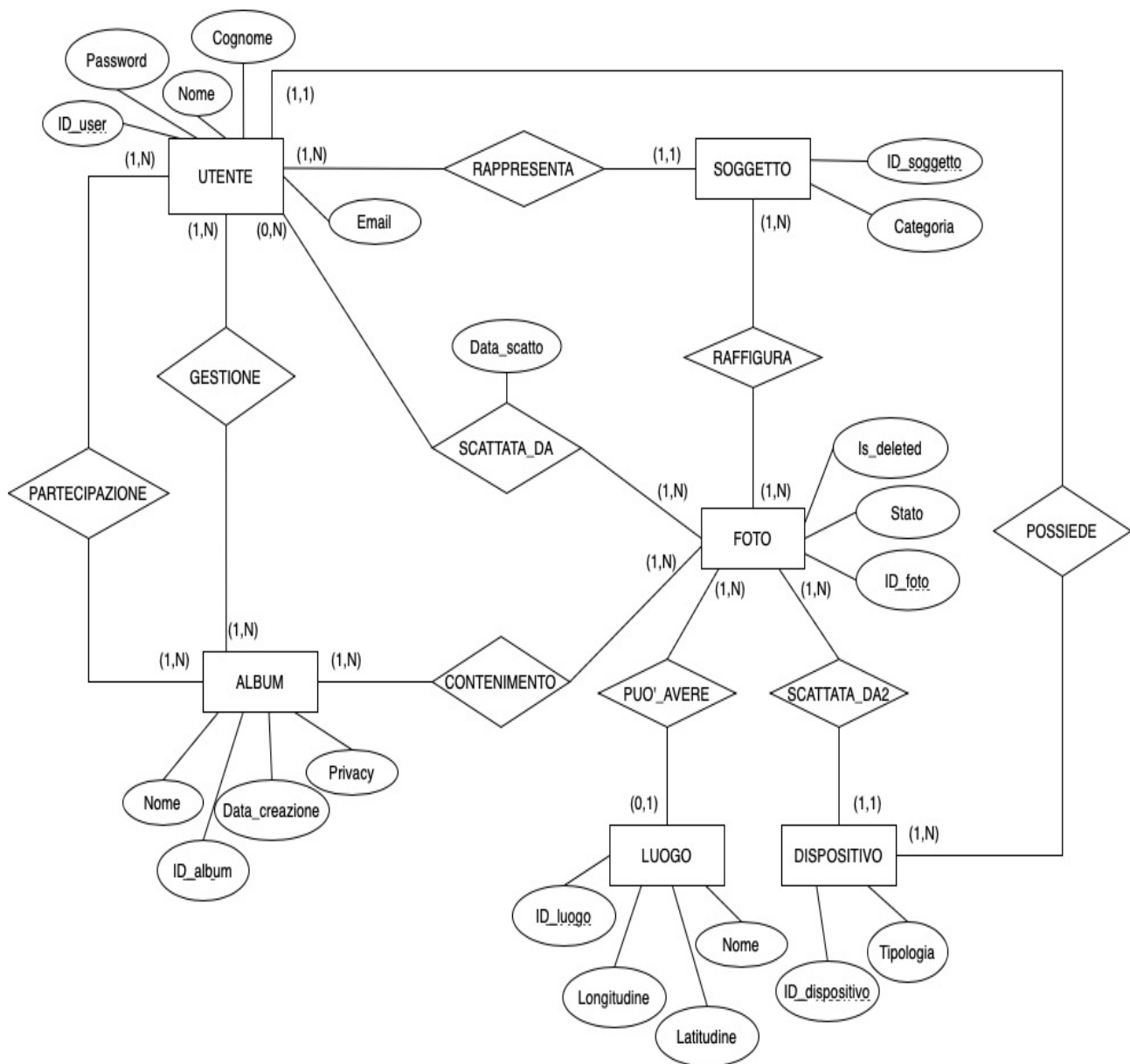
2.3.6 Attributi Derivabili

Non sono presenti attributi derivabili.

2.4 Class Diagram Ristrutturato



2.5 Diagramma ER



2.6 Dizionario delle classi

Classe	Attributi
Utente	id_user (<i>int</i>): identificativo per l'utente nome (<i>varchar(50)</i>): nome dell'utente cognome (<i>varchar(50)</i>): cognome dell'utente email (<i>varchar(100)</i>): email personale attraverso cui l'utente potrà accedere alla galleria password (<i>varchar(100)</i>): password scelta dall'utente utilizzata per accedere alla galleria
Dispositivo	id_dispositivo (<i>int</i>): identificativo per i dispositivi tipologia (<i>varchar(50)</i>): descrizione della tipologia di dispositivo registrato e utilizzato per scattare le foto
Foto	foto_id (<i>int</i>): identificativo per la fotografia stato (<i>varchar(20)</i>): descrizione della visibilità della foto, cioè "pubblica" o "privata"
Luogo	id_luogo (<i>int</i>): identificativo per il luogo nome (<i>varchar(100)</i>): nome unico per identificare la posizione geografica dove la foto è stata scattata latitudine (<i>float</i>): coordinata geografica della foto longitudine (<i>float</i>): coordinata geografica della foto
Album	id_album (<i>int</i>): identificativo per l'album nome (<i>varchar(50)</i>): nome personalizzato per l'album data_creazione (<i>date</i>): data in cui è stato creato l'album privacy (<i>boolean</i>): descrizione della visibilità dell'album, ovvero "0" se l'album è privato, "1" se l'album è condivisibile

2.7 Dizionario delle Associazioni

Descrizione dettagliata delle associazioni che intercorrono tra le classi

Nome	Descrizione	Classi Coinvolte
scattata_da	caricamento della foto da parte dell'utente	Foto[1] una foto può essere scattata da un solo utente Utente[0..*] un utente può scattare da zero a molte foto
può avere	presenza o meno del luogo nella foto	Foto[1..*] più foto possono essere scattate nello stesso luogo Luogo[0..1] un luogo può essere al massimo presente una volta in una foto
raffigura	presenza o meno di soggetti nella foto	Foto[1..*] una foto può raffigurare da uno a molti soggetti Soggetto[1..*] un soggetto può essere presente all'interno di più fotografie
contenimento	composizione dell'album	Foto[1..*] una foto può essere contenuta in più album Album[1..*] un album contiene da uno a molte foto
rappresenta	rappresentazione dell'utente all'interno della foto	Utente [1] un utente può essere rappresentato da un unico soggetto Soggetto[1] un soggetto raffigura un utente
gestione	gestione della galleria personale da parte dell'utente	Utente[1..*] un utente può gestire da uno a molti album Album[1] l'album è gestito da un solo utente
partecipazione	partecipazione di un utente ad un album condiviso	Utente[1..*] un utente può partecipare da uno a molti album condivisi Album[1..*] all'album possono partecipare da uno a molti utenti
scattata2	scatto della foto	Foto[1] la foto è scattata da un solo dispositivo Dispositivo[1..*] un dispositivo può scattare da uno a molte foto
possiede	relazione tra l'utente e i dispositivi associati	Utente[1..*] un utente può associare da uno a molti dispositivi Dispositivo[1] un dispositivo è associato ad un solo utente

Capitolo 3

Schema Logico

Traduzione dello schema concettuale ristrutturato, visto nel capitolo 2, nello schema logico. Indicheremo con la sottolineatura le chiavi primarie delle relazioni, mentre le chiavi esterne verranno raffigurate con affianco (*)

UTENTE (id_user, nome, cognome, email, password)

SOGGETTO (sogg_id, id_utente*, categoria)

id_utente → UTENTE.id_user

DISPOSITIVO (is_dispositivo, id_utente*, categoria)

id_utente → UTENTE.id_user

LUOGO (id_luogo, nome, latitudine, longitudine)

FOTO (id_foto, stato, is_deleted, id_owner*, id_dispositivo*, id_luogo*)

id_owner → UTENTE.id_user

id_dispositivo → DISPOSITIVO.id_dispositivo

id_luogo → LUOGO.id_luogo

ALBUM (id_album, nome, data_creazione, privacy, id_owner*)

id_owner → UTENTE.id_user

USERCOLLECTIONS (id_owner*, id_album*)

id_owner → UTENTE.id_user

id_album → ALBUM.id_album

TAG (id_utente*, id_foto*)

id_utente → UTENTE.id_user

id_foto → FOTO.id_foto

SOGGETTOFOTO (id_sogg*, id_foto*)

id_sogg → SOGGETTO.sogg_id

id_foto → FOTO.id_foto

ALBUMFOTO(id_album*, id_foto*)

id_foto → FOTO.id_foto

id_album → ALBUM.id_album

Capitolo 4

Progettazione Fisica

4.1 Definizione delle tabelle

4.1.1 Creazione Schema

```
1 DROP SCHEMA IF EXISTS s1 CASCADE;  
2 CREATE SCHEMA s1;
```

4.1.2 Utente

```
1 CREATE TABLE s1.Utente  
2 (  
3     IdUtente SERIAL,  
4     Nome      VARCHAR(50) NOT NULL,  
5     Cognome   VARCHAR(50) NOT NULL,  
6     Email     VARCHAR(50) NOT NULL,  
7     Password  VARCHAR(50) NOT NULL,  
8     CONSTRAINT PK_Utente PRIMARY KEY (IdUtente)  
9 );
```

4.1.3 Dispositivo

```
1 CREATE TABLE s1.Dispositivo  
2 (  
3     IdDispositivo SERIAL,  
4     Tipologia     VARCHAR(50) NOT NULL,  
5     IdUtente      INTEGER     NOT NULL,  
6     CONSTRAINT PK_Dispositivo PRIMARY KEY (IdDispositivo),  
7     CONSTRAINT FK_Dispositivo_Utente FOREIGN KEY (IdUtente)  
8     REFERENCES s1.Utente (IdUtente)  
9 );
```

4.1.4 Foto

```
1 CREATE TABLE s1.Foto
2 (
3     IdFoto          SERIAL,
4     datascatto      date,
5     Stato           stato,
6     IdUtente        INTEGER NOT NULL,
7     IdDispositivo   INTEGER NOT NULL,
8     CONSTRAINT PK_Foto PRIMARY KEY (IdFoto),
9     CONSTRAINT FK_Foto_Utente FOREIGN KEY (IdUtente) REFERENCES s1.
    Utente (IdUtente),
10    CONSTRAINT FK_Foto_Dispositivo FOREIGN KEY (IdDispositivo)
    REFERENCES s1.Dispostivo (IdDispositivo)
11 );
```

4.1.5 Luogo

```
1 CREATE TABLE s1.Luogo
2 (
3     IdLuogo         SERIAL,
4     Nome            VARCHAR(50),
5     Latitudine      DECIMAL(10, 8) NOT NULL,
6     Longitudine     DECIMAL(11, 8) NOT NULL,
7     CONSTRAINT PK_Luogo PRIMARY KEY (IdLuogo),
8     CONSTRAINT UQ_Coordinate UNIQUE (Latitudine, Longitudine),
9     CONSTRAINT UQ_Nome UNIQUE (Nome)
10 );
```

4.1.6 LuogoFoto

```
1 CREATE TABLE s1.LuogoFoto
2 (
3     IdLuogo INTEGER NOT NULL,
4     IdFoto  INTEGER NOT NULL,
5     CONSTRAINT PK_LuogoFoto PRIMARY KEY (IdLuogo, IdFoto),
6     CONSTRAINT FK_LuogoFoto_Luogo FOREIGN KEY (IdLuogo) REFERENCES
    s1.Luogo (IdLuogo),
7     CONSTRAINT FK_LuogoFoto_Foto FOREIGN KEY (IdFoto) REFERENCES s1.
    Foto (IdFoto)
8 );
```

4.1.7 Soggetto

```
1 CREATE TABLE s1.Soggetto
2 (
3     IdSoggetto SERIAL,
4     Categoria VARCHAR(50) NOT NULL,
5     CONSTRAINT PK_Soggetto PRIMARY KEY (IdSoggetto)
6 );
```

4.1.8 SoggFoto

```
1 CREATE TABLE s1.SoggettoFoto
2 (
3     IdSoggetto INTEGER NOT NULL,
4     IdFoto      INTEGER NOT NULL,
5     CONSTRAINT PK_SoggettoFoto PRIMARY KEY (IdSoggetto, IdFoto),
6     CONSTRAINT FK_SoggettoFoto_Soggetto FOREIGN KEY (IdSoggetto)
7     REFERENCES s1.Soggetto (IdSoggetto),
8     CONSTRAINT FK_SoggettoFoto_Foto FOREIGN KEY (IdFoto) REFERENCES
9     s1.Foto (IdFoto)
10 );
```

4.1.9 Tag

```
1 CREATE TABLE s1.Tag
2 (
3     IdUtente INTEGER NOT NULL,
4     IdFoto    INTEGER NOT NULL,
5     CONSTRAINT PK_Tag PRIMARY KEY (IdUtente, IdFoto),
6     CONSTRAINT FK_Tag_Utente FOREIGN KEY (IdUtente) REFERENCES s1.
7     Utente (IdUtente),
8     CONSTRAINT FK_Tag_Foto FOREIGN KEY (IdFoto) REFERENCES s1.Foto (
9     IdFoto)
10 );
```

4.1.10 Album

```
1 CREATE TABLE s1.Album
2 (
3     IdAlbum      Serial,
4     Nome          VARCHAR(50) NOT NULL,
5     DataCreazione DATE,
6     IdOwner       INTEGER     NOT NULL,
7     Privacy       BOOLEAN     NOT NULL,
8     CONSTRAINT PK_Album PRIMARY KEY (IdAlbum),
9     CONSTRAINT FK_Album_Utente FOREIGN KEY (IdOwner) REFERENCES s1.
10    Utente (IdUtente)
11 );
```


4.1.11 FotoAlbum

```
1 CREATE TABLE s1.FotoAlbum
2 (
3     IdFoto    INTEGER NOT NULL,
4     IdAlbum   INTEGER NOT NULL,
5     CONSTRAINT PK_FotoAlbum PRIMARY KEY (IdFoto, IdAlbum),
6     CONSTRAINT FK_FotoAlbum_Foto FOREIGN KEY (IdFoto) REFERENCES s1.
7     Foto (IdFoto),
8     CONSTRAINT FK_FotoAlbum_Album FOREIGN KEY (IdAlbum) REFERENCES
9     s1.Album (IdAlbum)
10 );
```

4.1.12 AlbumUtente

```
1 CREATE TABLE s1.AlbumUtente
2 (
3     IdAlbum   INTEGER NOT NULL,
4     IdUtente  INTEGER NOT NULL,
5     CONSTRAINT PK_AlbumUtente PRIMARY KEY (IdAlbum, IdUtente),
6     CONSTRAINT FK_AlbumUtente_Album FOREIGN KEY (IdAlbum) REFERENCES
7     s1.Album (IdAlbum),
8     CONSTRAINT FK_AlbumUtente_Utente FOREIGN KEY (IdUtente)
9     REFERENCES s1.Utente (IdUtente)
10 );
```

4.2 Definizione dei Trigger

4.2.1 Trigger - Creazione album personale

```
1 create or replace function f1() returns trigger as
2 $$
3 declare
4     nome1 varchar(50);
5 begin
6     nome1 = concat(NEW.nome, ' ', 'album personale');
7     insert into s1.album (nome, datacreazione, idowner, privacy)
8     values (nome1, CURRENT_TIMESTAMP, NEW.idutente, false);
9
10    return NEW;
11 END;
12 $$ language plpgsql;
13
14 create trigger t1
15     after insert
16     on s1.utente
17     for each row
18 execute function f1();
```

4.2.2 Trigger - Aggiungi alla galleria

questo trigger viene azionato dopo l'inserimento di una nuova foto. quest'ultima verrà infatti caricata direttamente nell'album personale dell'utente che l'ha scattata

```
1 CREATE OR REPLACE FUNCTION s1.add_to_gallery() RETURNS trigger
2 AS
3 $$
4 BEGIN
5     INSERT INTO s1.fotoalbum (idalbum, idfoto)
6     SELECT idalbum, NEW.idfoto
7     FROM s1.album
8     WHERE idowner = NEW.idutente;
9     RETURN NEW;
10 END;
11 $$ language plpgsql;
12
13
14 CREATE OR REPLACE TRIGGER add_to_gallery_trigger
15 AFTER INSERT
16 ON s1.foto
17 for each row
18 execute function s1.add_to_gallery();
```

4.2.3 Trigger - Aggiungi data corrente album

```
1 create or replace function s1.add_data() returns trigger as
2 $$
3 BEGIN
4     new.datacreazione = CURRENT_TIMESTAMP;
5     RETURN NEW;
6 end;
7 $$ language plpgsql;
8
9 create trigger t4
10 before insert on s1.album
11 for each row
12 execute function s1.add_data();
```

4.2.4 Trigger - Aggiungi data corrente foto

```
1 create or replace function s1.add_datafoto() returns trigger as
2 $$
3 BEGIN
4     new.datascatto = CURRENT_TIMESTAMP;
5     return new;
6 end;
7 $$ language plpgsql;
8
9 create trigger t5
10 before insert on s1.foto
11 for each row
12 execute function s1.add_datafoto();
```

4.2.5 Trigger - Aggiungi partecipazione ad album

```
1 create or replace function s1.add_utentealbum() returns trigger as
2 $$
3 BEGIN
4     insert into s1.albumutente(idalbum, idutente)
5     VALUES (NEW.idalbum, NEW.idowner);
6     return new;
7 end;
8 $$ language plpgsql;
9
10 create trigger t5
11     after insert
12     on s1.album
13     for each row
14 execute function s1.add_utentealbum();
```

4.2.6 Trigger - Cancellazione foto

```
1 create or replace function s1.f2() returns trigger as
2 $$
3     declare
4         c1 cursor for select idalbum from s1.album a where a.idowner
5         = OLD.idutente and a.privacy = true;
6         albumid s1.Album.idalbum%TYPE;
7 BEGIN
8     open c1;
9     loop
10         fetch c1 into albumid;
11         exit when not found;
12
13         delete from s1.fotoalbum
14         where idalbum = albumid and idfoto = OLD.idfoto;
15
16     end loop;
17 end;
18 $$ language plpgsql;
19
20 create trigger t2
21     after update
22     on s1.foto
23     for each row
24 execute function s1.f2();
```

4.3 Definizione delle funzioni

4.3.1 Function - getsameplace

Implementazione di una funzione che recupera tutte le foto che sono state scattate nello stesso luogo

```
1 CREATE OR REPLACE FUNCTION s1.getsameplace(luogoid int)
2     returns table
3     (
4         idfoto      integer ,
5         stato       stato ,
6         idutente    integer ,
7         iddispositivo integer
8     )
9 as
10 $$
11 begin
12     return query select f.idfoto , f.stato , f.idutente , f.
13         iddispositivo
14         from s1.foto f
15         natural join s1.luogofoto lf
16         where lf.IdLuogo = luogoid;
17 end;
18 $$ language plpgsql;
```

4.3.2 Function - getsamesubject

Implementazione di una funzione che recupera tutte le fotografie che condividono lo stesso soggetto

```
1 CREATE OR REPLACE FUNCTION s1.getsamesubject(soggettoid int)
2     returns table
3     (
4         idfoto      integer ,
5         stato       stato ,
6         idutente    integer ,
7         iddispositivo integer
8     )
9 as
10 $$
11 begin
12     return query select f.idfoto , f.stato , f.idutente , f.
13         iddispositivo
14         from s1.foto f
15         natural join s1.soggettofoto sf
16         where sf.IdSoggetto = soggettoid;
17 end;
18 $$ language plpgsql;
```

```

1 CREATE OR REPLACE FUNCTION s1.getsameuser(utenteid int)
2     returns table
3     (
4         idfoto      integer ,
5         stato       stato ,
6         idutente    integer ,
7         iddispositivo integer
8     )
9 as
10 $$
11 begin
12     return query select f.idfoto , f.stato , f.idutente , f.
13         iddispositivo
14         from s1.foto f
15         natural join s1.tag t
16         where t.IdUtente = utenteid;
17 end;
18 $$ language plpgsql;

```

4.4 View

Classifica dei top tre luoghi immortalati

```

1 CREATE VIEW s1.Top3LuogiPi Immortalati AS
2 (
3     SELECT l.IdLuogo , l.Nome , l.Latitudine , l.Longitudine , COUNT(*) AS
4         NumeroFoto
5     FROM s1.Luogo l ,
6         s1.LuogoFoto lf
7     WHERE l.IdLuogo = lf.IdLuogo
8     GROUP BY l.IdLuogo , l.Nome , l.Latitudine , l.Longitudine
9     ORDER BY NumeroFoto DESC
10    LIMIT 3);

```

Capitolo 5

Popolamento della Base di Dati

```
1 insert into s1.utente (nome, cognome, email, password)
2 values ('Elisa', 'Tiberio', 'elisatiberio@live.it', 'nene26'),
3        ('Michela', 'Pollio', 'michelapollio19@icloud.com', 'baldusie19'),
4        ('Mario', 'Rossi', 'mario@email.com', 'password123'),
5        ('Luca', 'Bianchi', 'luca@email.com', 'securepassword'),
6        ('giovanni', 'fiume', 'giovannifiume2014@libero.it', 'fiume'),
7        ('andrea', 'dota', 'andreadota2000@gmail.com', 'dota'),
8        ('alan', 'autorino', 'alan_autorino@hotmail.it', 'autorino'),
9        ('emmanuel', 'manna', 'manumanna99@gmail.com', 'manna'),
10       ('ilaria', 'gilardi', 'ilariag@live.it', 'gilardi'),
11       ('ilaria', 'risimini', 'ilariarisimini@hotmail.it', 'risimini'),
12       ('andrea', 'tiberio', 'a.tiberio@gmail.com', 'tibe'),
13       ('noemi', 'spera', 'noemis@gmail.com', 'spera'),
14       ('rosa', 'liguori', 'rosaliguori05@live.it', 'liguori'),
15       ('lisa', 'liguori', 'lisaliguori04@hotmail.it', 'liguori2'),
16       ('angela', 'pollio', 'angelapollio@live.it', 'pollio'),
17       ('antonio', 'sisimbro', 'tonysisi@hotmail.it', 'sisimbro'),
18       ('lorenzo', 'tecchia', 'thewatcher@live.it', 'tecchia'),
19       ('tipo', 'frizzantino', 'tipofrizz@icloud.com', 'frizzantino'),
20       ('giovanni', 'zampetti', 'algebra.9@icloud.com', 'zampetti'),
21       ('alfredo', 'top', 'alfredotop@live.it', 'top'),
22       ('francesco', 'ilgemello', 'frageme00@hotmail.it', 'ilgemello'),
23       ('marco', 'pastore', 'pastorepecora@outlook.it', 'pastore'),
24       ('sabrina', 'amicamichela', 'sabri2001@gmail.com', 'amicamichela'),
25       ('alessandro', 'rossi', 'alerossi@libero.it', 'rossi'),
26       ('francesca', 'ferrari', 'fraferrari@libero.it', 'ferrari'),
27       ('matteo', 'spavone', 'matteospav1@outlook.it', 'spavone'),
28       ('sofia', 'bianchi', 'sofiabianchi99@hotmail.it', 'bianchi'),
29       ('giulia', 'gallo', 'g.gallo@libero.it', 'gallo'),
30       ('giulia', 'conti', 'giualiaconti@live.it', 'conti'),
31       ('leonardo', 'marino', 'leomarino@live.it', 'marino'),
32       ('emma', 'de luca', 'emmadeluca@libero.it', 'deluca'),
33       ('gabriele', 'esposito', 'gabriespo@libero.it', 'esposito'),
34       ('chiara', 'rizzo', 'chiararizzo@outlook.it', 'rizzo');
35
36 insert into s1.dispositivo (tipologia, idutente)
37 VALUES ('iphone', 2),
38        ('macchina fotografica', 5),
39        ('cellulare', 1),
40        ('cellulare', 16),
41        ('cellulare', 19),
42        ('cellulare', 18),
43        ('cellulare', 9),
44        ('macchina fotografica', 6),
45        ('macchina fotografica', 20),
```

```

46      ('macchina fotografica', 15),
47      ('macchina fotografica', 7),
48      ('ipad', 3),
49      ('ipad', 14),
50      ('ipad', 13);
51
52 insert into s1.foto(stato, idutente, iddispositivo)
53 VALUES ('privato', 2, 1);
54
55 insert into s1.Luogo(nome, latitudine, longitudine)
56 VALUES ('Complesso Studi Montesantangelo', 40.839026294160135,
57          14.184969766924183),
58          ('Stadio Diego Armando Maradona', 40.828123191020495, 14.193061095758704)
59          ,
60          ('O Murzillo', 40.8269492735944, 14.1956988245939),
61          ('Piazza Plebiscito', 40.83597168245228, 14.248550909253588),
62          (null, 40.838643293527596, 14.252676480418668),
63          ('Galleria Umberto Primo', 40.83847283745015, 14.249436372193905),
64          ('Colosseo', 41.89029802621207, 12.49223089581815),
65          ('Piazza di Spagna', 41.90584948385739, 12.482326995819047),
66          (null, 41.89929104233426, 12.473192213005643),
67          ('Duomo di Milano', 45.464217966695536, 9.191969411368579),
68          ('Duel Club', 40.828068108592305, 14.155206995758762),
69          ('Piazza Tasso', 40.626241621412944, 14.37565197856061),
70          ('Katar beach lounge bar', 40.63883580921175, 14.39993868410017),
71          ('Palazzo di Sch nbrum', 48.18595543300895, 16.3126888943512),
72          (null, 35.88852486514128, 14.40577501577596),
73          (null, 40.83679554625882, 14.189438780418527);
74
75 insert into s1.album(nome, idowner, privacy)
76 VALUES ('Roma', 10, true),
77          ('Padova2021', 18, false),
78          ('Galleria Univerit', 5, true),
79          ('Vacanza a Pestum', 18, true),
80          ('Milano2022', 7, false),
81          ('Gita a Vienna', 12, true),
82          ('Londra', 3, false),
83          ('Barcellona', 9, true),
84          ('Amsterdam', 20, false),
85          ('MSCGrandiosa', 15, true),
86          ('Madrid', 6, false),
87          ('Estate a Diamante', 14, true),
88          ('Atene', 4, true),
89          ('Partita16/05', 11, false);
90
91 insert into s1.albumutente(idalbum, idutente)
92 VALUES (3, 2),
93          (3, 5),
94          (16, 35),
95          (1, 41),
96          (1, 46),
97          (1, 59),
98          (5, 39),
99          (1, 56),
100         (4, 40),
101         (4, 55),
102         (2, 57),
103         (1, 53),
104         (4, 61),
105         (2, 43);

```

```
105
106 INSERT INTO sl.foto(idutente, stato, iddispositivo)
107 VALUES (1, 'pubblico', 14),
108          (2, 'privato', 6),
109          (3, 'pubblico', 6),
110          (4, 'privato', 3),
111          (5, 'pubblico', 2),
112          (6, 'privato', 2),
113          (7, 'pubblico', 1),
114          (8, 'privato', 5),
115          (9, 'pubblico', 6),
116          (10, 'privato', 7),
117          (11, 'pubblico', 8),
118          (12, 'privato', 9),
119          (13, 'pubblico', 4),
120          (14, 'privato', 2),
121          (15, 'pubblico', 9),
122          (16, 'privato', 3),
123          (17, 'pubblico', 1),
124          (18, 'privato', 3),
125          (19, 'pubblico', 3),
```