

ANALISI DEI REQUISITI

Nelle specifiche dateci è richiesta la progettazione di un Database che possa gestire le attività di una “Biblioteca” (essa è un gestore di catalogo bibliografico online).

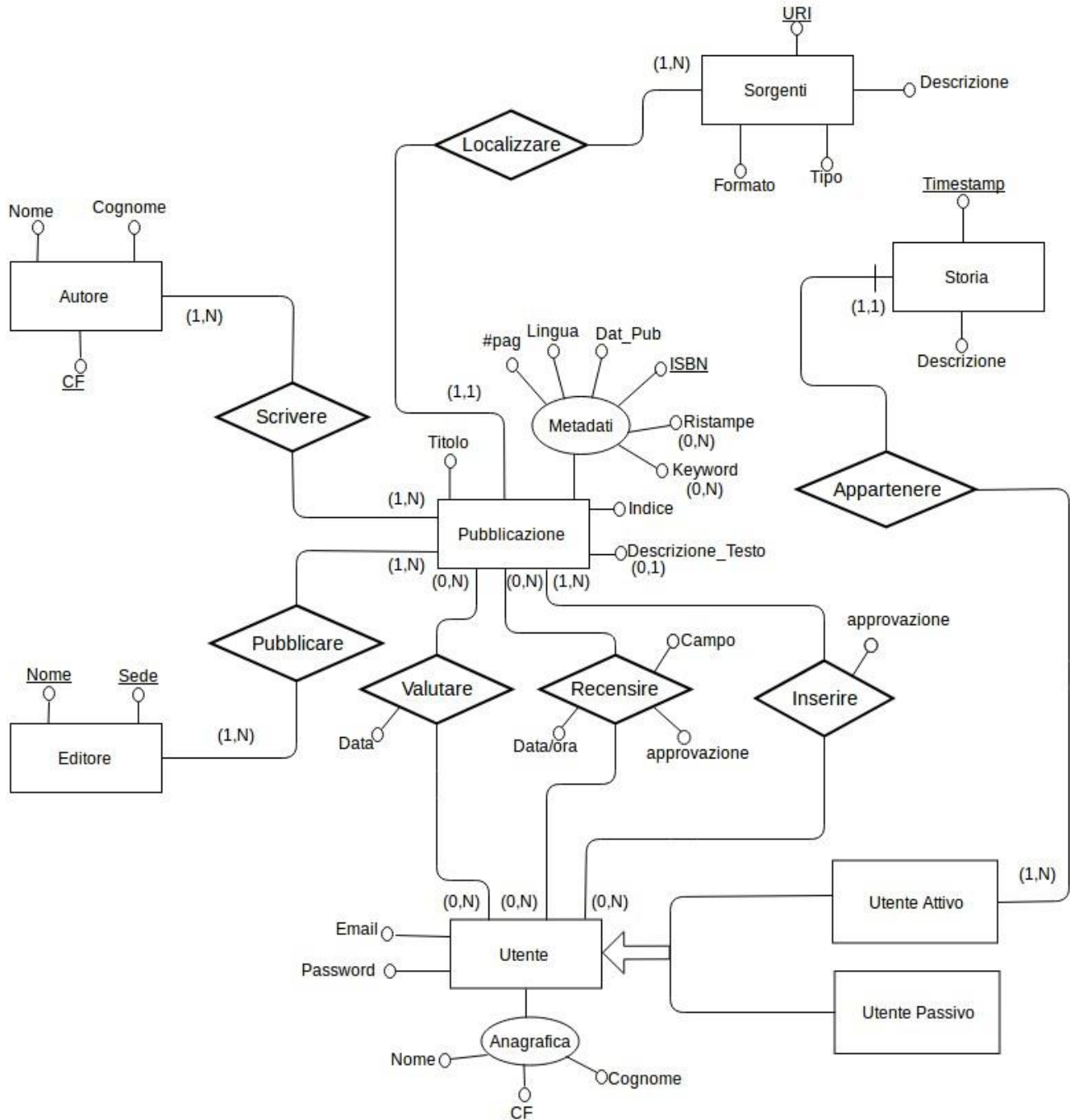
Il catalogo conterrà una serie di Pubblicazioni che saranno distinte dalle seguenti caratteristiche:

- Titolo
- Lista di autori
- Editore
- Descrizione testuale
- Metadati
- Indice
- Sorgenti

Le seguenti peculiarità serviranno agli utenti per poter accedere alle pubblicazioni; inoltre ogni pubblicazione è legata ad un certo numero di recensioni testuali (contrassegnate dall’utente che l’ha scritta e dalla data di inserimento) e da dei like (legati all’utente che ha dato il like e dalla data). Nel sistema gli utenti saranno registrati (con email e password) in base alla loro partecipazione nelle attività e potranno essere attivi o passivi.

Il Database da progettare deve essere in grado di sapere fornire all’utente, gestire in maniera efficiente e rispondere velocemente alle funzionalità che si trovano nella progettazione fisica.

PROGETTAZIONE CONCETTUALE



Note:

- Si evita di fare collegamenti tra le entità utente passivo e Storia perché si presume che se un utente sia nello stato passivo non abbia effettuato modifiche effettive nel sistema.
- Si sono aggiunte le entità “Editore” e “Autore” separatamente all’entità “Utente” perché si presume che questi non sono presenti nel sistema come “utenti” e per permettere una maggiore flessibilità

con le query e nel successivo mantenimento della base di dati ed evitare così anche possibili problemi di inconsistenza nei nomi degli Autori

- Le entità sopra descritte hanno cardinalità 1,N perché si presume che la loro presenza nel sistema sia possibile solo se essi hanno pubblicato o scritto una pubblicazione rispettivamente
- La generalizzazione nel modello E-R tra utente (entità padre) e utente attivo e passivo (entità figlie) è di tipo totale ed esclusiva

RISTRUTTURAZIONE MODELLO E-R

Nella ristrutturazione del modello e-r bisogna andare a considerare due concetti cardine cioè il numero di occorrenze di spazio e di tempo:

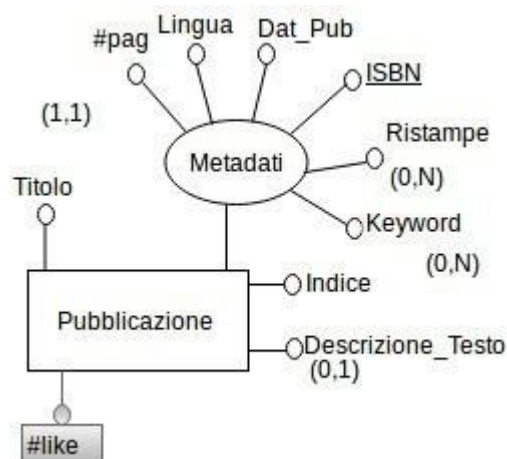
- quando andiamo ad analizzare lo spazio andiamo a considerare il numero di occorrenze previste- quando andiamo a considerare il tempo dobbiamo tener conto di quante occorrenze (entità e relazioni) andremmo a visitare per una cosiddetta operazione

Per la ristrutturazione dello schema E-R presente nella progettazione concettuale bisogna passare per varie fasi:

1)Analisi delle ridondanze:

Nello schema E-R della progettazione concettuale non è presente alcun tipo di ridondanza, anche se un tipo di ridondanza potrebbe essere aggiunto con le valutazioni:

Aggiungendo un attributo “#like” nell’entità pubblicazione potremmo riuscire ad accedere più velocemente al numero dei like che vengono assegnati ad una pubblicazione da più utenti, (contando però che aggiungerei anche se in minima parte una maggiore occupazione di memoria per tenere traccia del nuovo attributo), cosa che altrimenti sarebbe stata possibile solo contando attraverso la relazione “Valutazione” con una perdita di velocità da parte del sistema (il dato che stiamo evidenziando è molto importante all’interno del sistema ed è uno di quei dati che deve essere accessibile facilmente e velocemente per le numerose e frequenti modifiche che vengono fatte: es. ad ogni pubblicazioni vengono assegnate valutazioni da tanti utenti e questi possono anche scegliere di togliere una valutazione o di inserirla nuovamente)



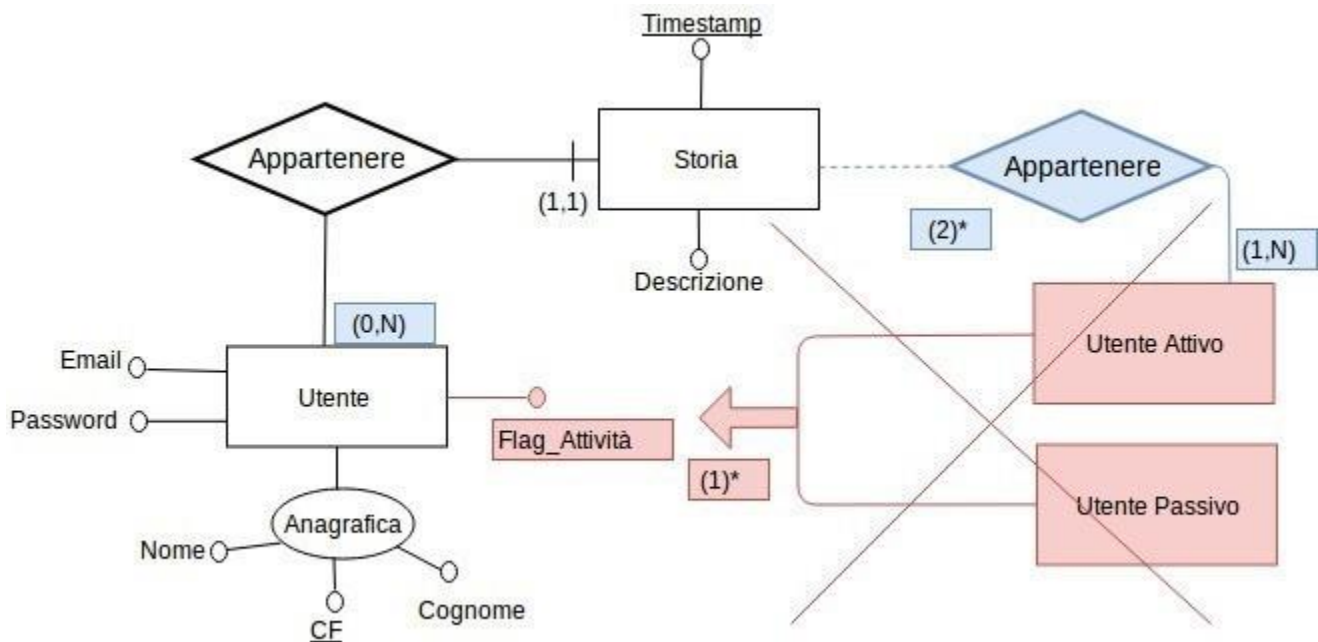
2)Eliminazione delle generalizzazioni:

Nello schema è presente una sola generalizzazione, totale ed esclusiva, tra Utente Attivo ed Utente Passivo in Utente. Abbiamo deciso di utilizzare l' accorpamento delle entità figlie nell'entità padre perché tra le ristrutturazioni possibili è quella che permette la minor perdita in senso di velocità e occupazione di memoria: le altre possibili soluzioni avrebbero aggiunto nuove relazioni e nuove entità che avrebbero appesantito e reso più lento l'accesso alle informazioni degli utenti.

La ristrutturazione è stata effettuata con le seguenti modifiche (illustrate nell'immagine):

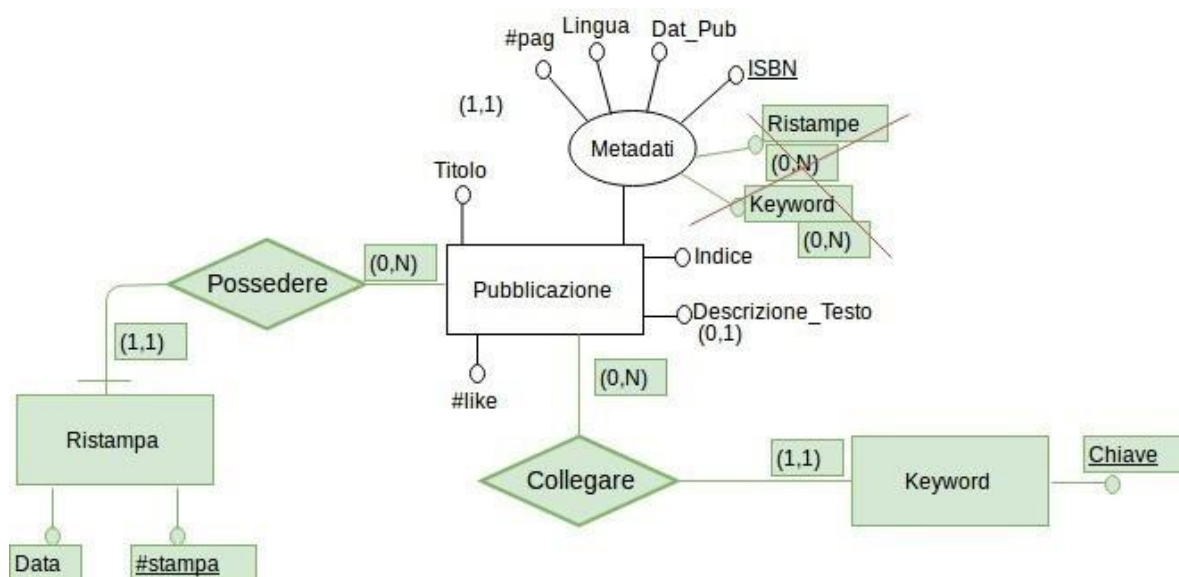
- (1) È stato aggiunto un booleano chiamato "Flag Attività" che indica lo stato in cui si trova l'utente, cioè se attivo o passivo
- (2) È stata modificata la cardinalità che va da "Utente" a "Storia", da (1,N) a (0,N) perché ora l'entità Utente rappresenta anche l'utenza passiva, a cui non appartiene una storia (anche se in casi speciali decisi dai moderatori sarebbe possibile avere una storia)

NB: Nel caso in cui il livello dell'utenza non fosse programmato in questo modo, ma la specifica intendeva che il livello di utenza dovesse essere gestito da un moderatore all'interno del sistema allora la cardinalità da utente poteva rimanere a (1,N)



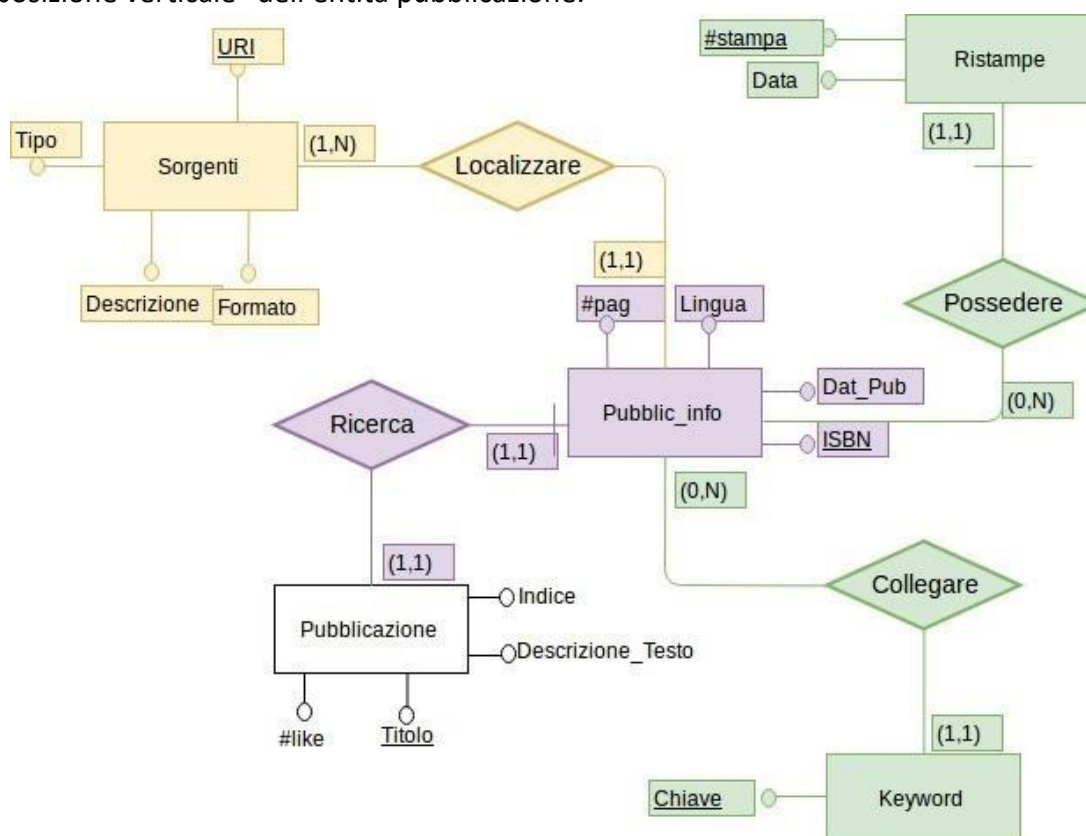
3) Eliminazione degli attributi multi-valore

Gli unici attributi multivalore presenti nel modello E-R sono le keyword e le ristampe (presenti nell'entità pubblicazione) e sono stati eliminati e sostituiti nel seguente modo:

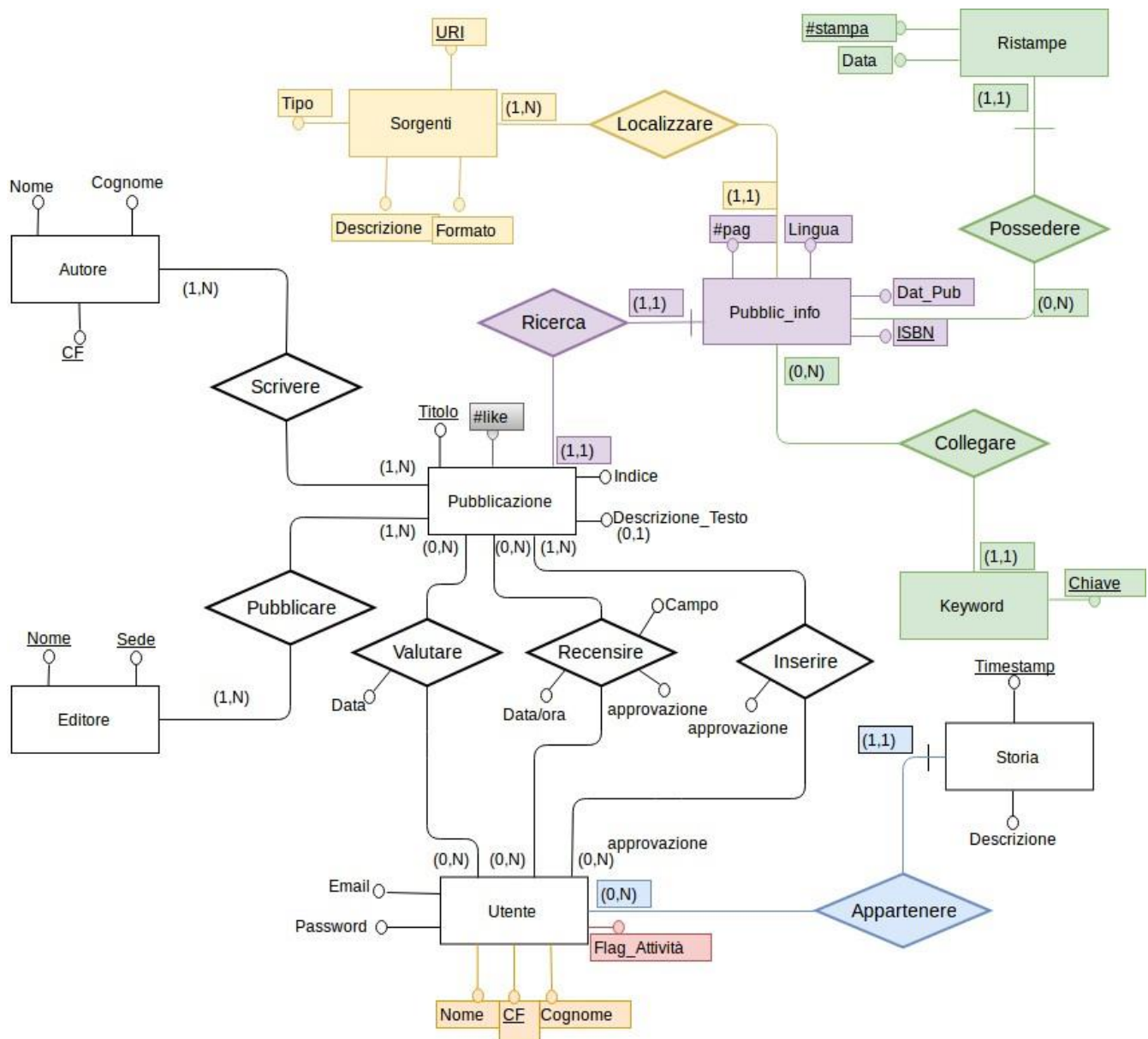


4) Partizionamento/accorpamento di concetti

Analizzando le query richieste dal progetto, si nota come l'operazione di ricerca tramite metadati (query#8), sia una delle principali (naturalmente, essendo il sistema una bibliografia online) ed anche che non sempre di ha bisogno di presentare all'utente tutte le informazioni subito, ma solo quelle importanti, cioè " Titolo, Editore e Descrizione"; per questo motivo abbiamo effettuato una "decomposizione verticale" dell'entità pubblicazione.



Il diagramma ristrutturato completo risulta il seguente:



Bisogna anche evidenziare che nel diagramma E-R non sono presenti dei vincoli impliciti:

- 1) Il numero delle recensioni che un utente può inserire è limitato a 1
- 2) Il numero dei like che un utente può inserire ad una pubblicazione è limitato a 1
- 3) Il codice fiscale di ogni utente è unico ed univoco

PROGETTAZIONE LOGICA

La progettazione logica deve tradurre i costrutti del modello E-R nei costrutti del modello relazionale, garantendo l'equivalenza dei modelli ed è stata eseguita in questo modo:

Traduzione delle entità:

AUTORE(ID, CF, Nome, Cognome)

EDITORE(ID, Nome, Sede)
 UTENTE(ID, CF, Nome, Cognome, Email, Password, Flag_Actività)
 STORIA(Timestamp, Descrizione, utentID, pubblicazioneID)
 KEYWORD(Chiave)
 PUBBLICAZIONE(ID, Titolo, #like, indice, Descrizione_testo, editorID, utentID)
 PUBBLIC_INFO(ID, ISBN, Dat_Pub, Lingua, #pag, pubblicazioneID)
 SORGENTI(ID, IDpublicinfo, URI, Descrizione, Formato, Tipo)
 RISTAMPE(ID, #stampa, Data, publicinfoID)

Traduzione relazioni molti a molti:

VALUTARE(pubblicazioneID, utentID, Data) [utente-pubblicazione]
 RECENSIRE(pubblicazioneID, utentID, Data, Campo, approvazione) [utente-pubblicazione]
 INSERIRE [utente-pubblicazione]
 SCRIVERE(pubblicazioneID, autorID) PUBBLICARE
 [editore-pubblicazione]

Traduzione uno a molti:

APPARTENERE [storia-utente]
 COLLEGARE(publicinfoID, chiave)
 [keyword-public_info]
 POSSEDERE [ristampe-public_info] LOCALIZZARE
 [Sorgenti-public_info]

Traduzione uno a uno:

RICERCA [public_info-pubblicazione]

NOTE:

- Gli attributi sottolineati ed evidenziati in giallo sono le chiavi esterne presenti nel database -
- All'interno delle parentesi quadre sono contenute le entità che sono coinvolte nelle relazioni.
- I colori indicano come sono state tradotte le relazioni.

PROGETTAZIONE FISICA

La creazione e l'inserimento di dati nel database rappresentano la struttura descritta nella richiesta del progetto è stata fatta tramite le seguenti query:

```

CREATE DATABASE IF NOT EXISTS `bibliografia`;
USE `bibliografia`;

```

```

drop table if exists `autore`;
create table `autore`{

```

```
`ID` int(15) NOT NULL AUTO_INCREMENT,  
`CF` char(16) DEFAULT NULL,  
`nome` varchar(30) DEFAULT NULL,  
`cognome` varchar(30) DEFAULT NULL,  
PRIMARY KEY (`ID`),  
UNIQUE KEY `CF` (`CF`);
```

lock tables `autore` write;

```
insert into `autore` values (1,'BNCOM2312ASFAF94','Omino','Bianco'),  
(2,'NRM243DSBB234C6','Maria','Neri'),  
(3,'GRGCRS235F235KLD','Cristiano','Grigio'),  
(4,'GRMHLL2342FSD34K','Grommash','Hellscream'),  
(5,'CRSGLD325FSE245G','Cristhie','Golden'); unlock table;
```

```
drop table if exists `editore`;  
create table `editore`(  
`ID` int(15) NOT NULL AUTO_INCREMENT,  
`nome` varchar(45) DEFAULT NULL,  
`sede` varchar(35) DEFAULT NULL,  
PRIMARY KEY (`ID`),  
UNIQUE KEY `nome` (`nome`,`sede`)  
);  
lock tables `editore` write;
```

```
insert into `editore` values (1,'DeAgostini','Milano'),(2,'Mondadori','Roma'),  
(3,'IBS','Bologna'),(4,'Blizzard','California');  
unlock table;
```

```
drop table if exists `utente`;  
create table `utente`(  
`ID` int(15) NOT NULL AUTO_INCREMENT,  
`CF` char(16) NOT NULL,  
`nome` varchar(25) NOT NULL,  
`cognome` varchar(25) NOT NULL,  
`email` varchar(30) NOT NULL,  
`password` varchar(20) NOT NULL,  
`flag_attività` tinyint(1) NOT NULL DEFAULT '0',  
PRIMARY KEY (`ID`),
```



```
UNIQUE KEY `email` (`email`),  
UNIQUE KEY `CF` (`CF`));
```

```
lock tables `utente` write;
```

```
insert into `utente` values
```

```
(1,'MRCZUC928DFSH45G','Marco','Zucconi','marcozucconi@gmail.com','zucco45',0),  
(2,'STFDCT38452UDJGN','Stefano','Diociaiuti','stefanodioci@gmail.com','diocisalvi',0),  
(3,'FBCPT238V4WBZVD6','Fabio','Capitanio','fabiocapitanio@gmail.com','fabio01',0);
```

```
unlock table;
```

```
drop table if exists `pubblicazione`;
```

```
create table `pubblicazione` (
```

```
`ID` int(15) NOT NULL AUTO_INCREMENT,
```

```
`titolo` varchar(50) DEFAULT NULL,
```

```
`descrizione_testo` varchar(400) NOT NULL,
```

```
`indice` varchar(500) DEFAULT NULL,
```

```
`#like` int(11) NOT NULL DEFAULT '0',
```

```
`UtentiID` int(15) NOT NULL,
```

```
`EditorID` int(15) NOT NULL,
```

```
PRIMARY KEY (`ID`),
```

```
KEY `edit_public` (`EditorID`), constraint `edit_public` foreign key (`EditorID`) references  
`editore`(`ID`) on update
```

```
cascade,
```

```
KEY `utente_public_ins` (`UtentiID`),
```

```
constraint `utente_public_ins` foreign key (`UtentiID`) references `utente`(`ID`) on  
update cascade
```

```
);
```

```
lock tables `pubblicazione` write;
```

```
insert into `pubblicazione` values (1,'Illidan','World of Warcraft Books',null,0,1,1), (2,'Le  
avventure di Peppa Pig','Il prosciutto senza grassi',null,0,3,1),(3,'Shining','Le avventure più  
divertenti...',null,0,2,2),  
(4,'Un caso bizzarro per il commissario Carra','Il professor Arbib ha scritto un libro??',null,0,1,2),  
(5,'50 sfumature di Grigio','La frusta fa male se non sai come usarla...',null,0,3,3);
```

```
unlock tables;
```

```

drop table if exists `storia_utente`; create
table `storia_utente` (
  `timestamp` timestamp not null default current_timestamp,
  `descrizione` varchar(500) not null,
  `utenteID` int(15), `pubbID` int(15), foreign key (`utenteID`) references `utente`(`ID`) on update
  cascade on delete cascade, foreign key (`pubbID`) references `pubblicazione`(`ID`) on update
  cascade on delete
  cascade
);
lock tables `storia_utente` write;

```

```

insert into `storia_utente` values ('18-07-10 20:04:34','ha recensito una opera...',3,3),
('2014-04-24 13:00:53','ha messo un mi piace...',1,4),
('1996-05-23 18:00:23','ha tolto mi piace...',2,1),
('2016-04-06 8:30:53','ha recensito una opera...',3,5);

```

```

unlock tables;

```

```

drop table if exists `pubblic_info`; create
table `pubblic_info` ( `ID` int(15) not null
  auto_increment,
  `ISBN` char(18) not null,
  `dat_pub` date Default null,
  `lingua` varchar (3) default null,
  `#pag` int (5) default null,
  `pubblicazioneID` int(15) not null,
  primary key (`ID`), unique key
  (`pubblicazioneID`), unique key
  (`ISBN`),
  key `pub_infodata`(`pubblicazioneID`), constraint `pub_infodata` foreign key
  (`pubblicazioneID`) references `pubblicazione`(`ID`) on update cascade on
  delete cascade
);
lock tables `pubblic_info` write;

```

```

insert into `pubblic_info` values (1,'978-3-4032-6634-5','1956-11-07','RUS',345,1),
(2,'979-5-2345-7436-9','1982-05-07','ITA',230,2),
(3,'979-3-5634-3464-7','2011-09-11','ENG',34,3),
(4,'978-4-5745-6336-6','1820-12-24','ITA',320,4),
(5,'978-7-2355-2323-6','2012-12-31','GER',453,5);

```

unlock table;

**drop table if exists`keyword`;
create table `keyword`(`chiave`
char(30) not null,
primary key `keyopera`(`chiave`)
);
lock tables `keyword` write;**

insert into `keyword` values ('Horror'),('Ragazzi'),('Fantasy'),('Giallo'); unlock tables;

**drop table if exists`ristampe`; create
table `ristampe`(
`ID` int(15) not null auto_increment,
`#stampa` int(7) default null,
`publicinfoID` int(15) not null,
`data_ristampa` date not null,
foreign key (`publicinfoID`) references `pubblic_info` (`ID`) on delete cascade on
update cascade,
primary key (`ID`),
unique key `numero` (`#stampa`, `data_ristampa`)
);
lock tables `ristampe` write;**

**insert into `ristampe` values (1,24,1,'2018-06-07'),
(2,2,4,'2016-02-03'),
(3,34,3,'1996-04-03');**

unlock table;

**drop table if exists`sorgenti`; create
table `sorgenti`(
`ID` int(15) not null auto_increment,
`URI` varchar(100) not null,
`descrizione` varchar(300) default null,**

```
`formato` varchar(25) default null,  
`tipo` varchar(25) default null, `IDpublicinfo` int(15) not null, primary key (`ID`), unique key `URI`  
(`URI`), key `localizzare` (`IDpublicinfo`), constraint `localizzare` foreign key (`IDpublicinfo`)  
references `pubblic_info` (`ID`) on  
delete cascade on update cascade  
);  
lock tables `sorgenti` write;
```

```
insert into `sorgenti` values  
(1,'https://www.google.it/immagini','copertina','jpeg','immagine',1),  
(2,'https://www.kindle.it/e-books.pdf','pdf di prova gratuito','pdf','download',2),  
(3,'https://www.ebay.com','gadget personaggio romanzo','oggetto','acquisto',3),  
(4,'https://www.amz.com','acquista online','copertina rigida','acquisto',4);
```

```
unlock table;
```

```
drop table if exists `valutare`;  
create table `valutare`(  
`pubblicazioneID` int(15) not null,  
`utenteID` int(15) NOT NULL,  
`data` timestamp not null default current_timestamp, unique key `like` (`utenteID`,  
`pubblicazioneID`), constraint `like-utente` foreign key (`utenteID`) references `utente` (`ID`)  
on delete  
cascade on update cascade,  
constraint `like-pubblicazione` foreign key (`pubblicazioneID`) references `pubblicazione` (`ID`)  
on update cascade on delete cascade  
);  
lock tables `valutare` write;
```

```
unlock table;
```

```
drop table if exists `recensire`; create  
table `recensire`(  
`pubblicazioneID` int(15) not null,  
`utenteID` int(15) NOT NULL,  
`data` timestamp not null default current_timestamp,  
`campo` varchar(400) not null,
```

```
`approvazione` tinyint(1) not null default '0', unique key `like` (`utenteID`, `pubblicazioneID`),
constraint `recensione-utente` foreign key (`utenteID`) references `utente` (`ID`) on delete
cascade on update cascade,
constraint `recensione-pubblicazione` foreign key (`pubblicazioneID`) references `pubblicazione`
(`ID`) on update cascade on delete cascade
);
lock tables `recensire` write;
```

```
insert into `recensire` values (1,2,'2017-06-10 12:10:55','Questo testo è davvero
bello!',0),
(2,3,'2008-01-06 21:14:56','Questo testo è davvero brutto',0),(3,1,'2002-07-11
07:30:45','Non è male, ma neanche bello',0),
(1,3,'2018-06-05 16:53:45','Opera tropo lunga!!!',0), (1,1,'2016-05-03 12:45:23', 'Scritto davvero
male',0);
```

```
unlock table;
drop table if exists `scrivere`; create
table `scrivere` (
`pubblicazioneID` int(15) not null,
`autorID` int(15) NOT NULL,
`approvazione` tinyint(1) not null default '0', key
`scrivere-utente` (`autorID`),
key `scrivere-pubblicazione` (`pubblicazioneID`), constraint `scrivere-autore` foreign key
(`autorID`) references `autore` (`ID`) on delete
cascade on update cascade,
constraint `scrivere-pubblicazione` foreign key (`pubblicazioneID`) references `pubblicazione`
(`ID`) on update cascade on delete cascade
);
```

```
lock tables `scrivere` write;
```

```
insert into `scrivere` values (1,2,'0'),(2,2,'0'),(5,3,'0');
```

```
unlock table;
```

```
drop table if exists `collegare`; create table `collegare` ( `publicinfoID` int(15) not null,
`chiave` char(30) not null, foreign key (`publicinfoID`) references `public_info` (`ID`) on
delete cascade on
update cascade,
foreign key (`chiave`) references `keyword` (`chiave`) on delete cascade on update
```

cascade

);

lock tables `collegare` write;

insert into `collegare` values (1,'Fantasy'), (2,'Ragazzi'), (4,'Giallo'), (5,'Giallo'); unlock tables;

Una volta creato il database, sono state create le query che il progetto richiede, ed esse sono qui elencate:

NB: -PER MOSTRARE COME LE QUERY MODIFICHINO LE TABELLE DEL DATABASE SONO STATE AGGIUNTE DELLE IMMAGINI DI RIFERIMENTO PRESE DA ESEMPI DI ESECUZIONE DELLE STESSE INTERROGAZIONI


- IN OGNI QUERY CON RICERCA TRAMITE UN ATTRIBUTO SCELTO (ricerca tramite titolo) SONO STATI USATI VALORI A CASO CHE FANNO RIFERIMENTO AI DATI INSERITI NEL DATABASE.

QUERY 1: Modifica del livello di un utente (da attivo a passivo e viceversa).

Nella base di dati, come detto in precedenza, è stato

La query effettiva che realizza la funzionalità richiesta è la seguente:

```
5      #(lo zero indica che lo stato dell'utente è passivo)
6      #query che serve a impostare il flag_Attività a 1
7      • update utente set flag_attività=1 where ID=1;
8
9
10
11      • select u.nome as `Nome Utente`, u.cognome as `Cognome Utente`,
12          u.flag_attività as `Stato Utente`
13      from utente u where ID=1;
```



Nome Utente	Cognome Utente	Stato Utente
Marco	Zucconi	1

(la seconda query nelle due immagini è stata usata solo per mostrare la modifica nella tabella)

```
15
16      #query che serve a impostare il flag_Attività a 1
17      #(l'1 indica lo stato attivo dell'utente)
18      • update utente set flag_attività=0 where ID=1;
19
20      • select u.nome as `Nome Utente`, u.cognome as `Cognome Utente`,
21          u.flag_attività as `Stato Utente`
22      from utente u where ID=1;
```



Nome Utente	Cognome Utente	Stato Utente
Marco	Zucconi	0

QUERY2 : Estrazione elenco delle ultime dieci pubblicazioni inserite.

6 • select pi.dat_pub as 'Data Pubblicazione Opera'
7 from public info pi

Result Grid





Data Pubblicazione Opera
1956-11-07
1982-05-07
2011-09-11
1820-12-24
2012-12-31

```

9 • select p.titolo as `Titolo Opera`, pi.dat_pub as `Data Pubblicazione Opera`
10 from public_info pi, pubblicazione p
11 where p.ID=pi.pubblicazioneID
12 order by dat_pub
13 desc limit 10;
14

```

<

Result Grid |   Filter Rows: | Export:  | Wrap Cell Content: 

	Titolo Opera	Data Pubblicazione Opera
	50 sfumature di Grigio	2012-12-31
	Shining	2011-09-11
	Le avventure di Peppa Pig	1982-05-07
	Ilidan	1956-11-07
	Un caso bizzarro per il commissario Carra	1820-12-24

```
1 select p.Titolo as `Titolo Opera`, r.data_ristampa as `Data Ristampa Opera`
2 from ristampe r, pubblicazione p, public_info pi
3 where r.publicinfoID=pi.ID and p.ID=pi.pubblicazioneID and r.data_ristampa
4 BETWEEN DATE_SUB(NOW(), INTERVAL 30 DAY) and now();
5
```

Result Grid | | Filter Rows: | Export: | Wrap Cell Content:

	Titolo Opera	Data Ristampa Opera

Quindi per controllare la correttezza assoluta della query, abbiamo cambiato prima visualizzato le ristampe presenti nel database,

```
6 • select p.Titolo as `Titolo Opera`, r.data_ristampa as `Data Ristampa Opera`
7   from ristampe r, pubblicazione p, public_info pi
8   where r.publicinfoID=pi.ID and p.ID=pi.pubblicazioneID
```

Titolo Opera	Data Ristampa Opera
Ilidan	2018-06-07
Un caso bizzario per il commissario Carra	2016-02-03
Shining	1996-04-03

per poi cambiare la data di una ristampa in modo tale da poter avere almeno una ristampa risalente ad almeno 30 giorni

```
5 • update ristampe set data_ristampa='2018-07-11' where publicinfoID=1;
```

Titolo Opera	Data Ristampa Opera
Ilidan	2018-07-11
Un caso bizzario per il commissario Carra	2016-02-03
Shining	1996-04-03

Infine, dimostrando la correttezza della query, abbiamo eseguito l'interrogazione con il seguente risultato:

```
1 • select p.Titolo as `Titolo Opera`, r.data_ristampa as `Data Ristampa Opera`
2   from ristampe r, pubblicazione p, public_info pi
3   where r.publicinfoID=pi.ID and p.ID=pi.pubblicazioneID and r.data_ristampa
4   BETWEEN DATE_SUB(NOW(), INTERVAL 30 DAY) and now();
5
```

Titolo Opera	Data Ristampa Opera
Ilidan	2018-07-11

(tabella contenente l'opera che avevamo modificato in precedenza)

QUERY4 : Estrazione elenco degli utenti più "collaborativi" (cioè quelli che hanno inserito più pubblicazioni).

Questa interrogazione è eseguita facendo un join su “utente” e “pubblicazione” con la clausola del collegamento tramite i due attributi ID (questo permette di capire le pubblicazioni collegate agli utenti, quindi quelle inserite da essa) es usando count per poter contare quante occorrenze di opere sono state inserite dai vari utenti.

La query effettiva che realizza la funzionalità richiesta è la seguente:

```
1 • select u.nome as `Nome Utente`, u.cognome as `Cognome Utente`,
2 count(*) as `Conteggio Attività`
3 from pubblicazione p join utente u on (p.UtentID=u.ID) group by u.ID
4 order by count(*) desc limit 10;
```

Nome Utente	Cognome Utente	Conteggio Attività
Marco	Zucconi	2
Fabio	Capitanio	2
Stefano	Diociaiuti	1

QUERY5: Estrazione elenco delle pubblicazioni inserite da un utente.

Come prima operazione viene mostrato come nel sistema sono collegate opere e utenti:

```
2 • select p.titolo as `Titolo Opera`, u.nome as `Nome Utente`,
3 u.cognome as `Cognome Utente`, u.ID as ID_UTENTE
4 from pubblicazione p, utente u where p.UtentID=u.ID ;
```

Titolo Opera	Nome Utente	Cognome Utente	ID_UTENTE
Ilidan	Marco	Zucconi	1
Le avventure di Peppa Pio	Fabio	Capitanio	3
Shinino	Stefano	Diociaiuti	2
Un caso bizzario per il commissario Carra	Marco	Zucconi	1
50 sfumature di Grigio	Fabio	Capitanio	3

Di questa query ne sono state fornite due versioni valide:

La prima sfrutta per la ricerca tramite uso di ID Utente per trovare le opere ad esso associato:

```
2 • select p.titolo as `Titolo Opera`, u.nome as `Nome Utente`,
3 u.cognome as `Cognome Utente`, u.ID as ID_UTENTE
4 from pubblicazione p, utente u where p.UtentID=u.ID and u.id=1;
```

Titolo Opera	Nome Utente	Cognome Utente	ID_UTENTE
Ilidan	Marco	Zucconi	1
Un caso bizzario per il commissario Carra	Marco	Zucconi	1

La seconda, d’altro canto, è una versione che utilizza il codice fiscale:

```

7 • select p.titolo as `Titolo Opera`, u.nome as `Nome Utente`,
8     u.cognome as `Cognome Utente`, u.ID as ID_UTENTE
9     from pubblicazione p, utente u where p.UtentID=u.ID and u.CF='MRCZUC928DFSH45G';

```

	Titolo Opera	Nome Utente	Cognome Utente	ID_UTENTE
	Illidan	Marco	Zucconi	1
	Un caso bizzario per il commissario Carra	Marco	Zucconi	1

Sono entrambe due alternative valide, che possono essere usate in base alla disponibilità di informazioni dell'utente.

QUERY6: Estrazione catalogo, cioè elenco di tutte le pubblicazioni con titolo, autori, editore e anno di pubblicazione, ordinato per titolo.

La query che rappresenta la funzionalità richiesta, è la seguente:

```

1 • select p.titolo as 'Titolo Opera', p.descrizione_testo AS 'Descrizione Opera',
2     e.nome as 'Nome Editore', a.nome as 'Nome Autore' , a.cognome AS 'Cognome Autore',
3     pi.dat_pub as 'Data Pubblicazione'
4     from pubblicazione p, autore a, editore e, pubblic_info pi, scrivere s
5     where p.editorID=e.ID and s.autorID=a.ID
6     and pi.pubblicazioneID=p.ID and s.pubblicazioneID=p.ID
7     group by p.titolo order by p.titolo;

```

	Titolo Opera	Descrizione Opera	Nome Editore	Nome Autore	Cognome Autore	Data Publi
	50 sfumature di Grigio	La frusta fa male se non sai come usarla...	IBS	Cristiano	Grigio	2012
	Illidan	World of Warcraft Books	DeAgostini	Maria	Neri	1956
	Le avventure di Peppa Pig	Il prosciutto senza grassi	DeAgostini	Maria	Neri	1982

Nella query non sono presenti tutte le opere inserite nel database per un errore negli inserimenti: come ovvio che sia, ogni opera ha un suo autore e negli inserimenti non sono state associate a tutte le opere un autore che le abbia scritte, quindi risultano presenti nel risultato, solo le opere di cui è stato inserito l'autore nel database.

QUERY 7: Estrazione dati completi di una pubblicazione specifica, dato il suo ID.

La query che rappresenta la funzionalità richiesta è la seguente:

```

1 • select p.titolo as 'Titolo Opera', p.descrizione_testo as 'Descrizione Testo',
2 pi.ISBN, pi.dat_pub as 'Data Pubblicazione Opera',
3 pi.lingua as 'Lingua Opera' , a.nome as 'Nome Autore' , a.cognome as 'Cognome Autore',
4 e.nome as Editore
5 from pubblicazione p, pubblic_info pi, autore a, editore e, scrivere s, utente u
6 where p.editorID=e.ID and p.UtentID=u.ID and pi.pubblicazioneID=p.ID
7 and s.autorID=a.ID and s.pubblicazioneID=p.ID and p.ID=2;
8 #query per l'estrapolazione dei dati della pubblicazione con ID 2
9
10
11

```

Titolo Opera	Descrizione Testo	ISBN	Data Pubblicazione Opera	Lingua Opera	Nome Autore	Cognome Autore	Editore
Le avventure d...	Il prosciutto senza grassi	979-5-2345-7436-9	1982-0...	ITA	Maria	Neri	DeAaostini

QUERY 8: Ricerca di pubblicazioni per ISBN, titolo, autore, e parole chiave. Di questa

query ne sono state fatte 5 versioni:

1)Versione in cui la ricerca viene effettuata solamente tramite il codice ISBN

```

1
2 • select p.titolo as 'Titolo Opera', pi.ISBN from pubblicazione p, pubblic_info pi
3 where pi.pubblicazioneID=p.ID and pi.ISBN='979-5-2345-7436-9';
4
5

```

Titolo Opera	ISBN
Le avventure di Peppa Pig	979-5-2345-7436-9

2)Versione in cui la ricerca viene effettuata solamente tramite il titolo dell'opera

(con questa versione potrebbero capitare casi di omonimia con il titolo della pubblicazione)

```

6 • select p.titolo as 'Titolo Opera', p.descrizione_testo as 'Descrizione Opera' from pubblicazione p
7 where p.titolo= 'Shining';
8
9

```

Titolo Opera	Descrizione Opera
Shining	Le avventure più divertenti...

3)Versione in cui la ricerca viene effettuata solamente tramite l'uso dell'ID dell'autore

```

10 • select p.titolo as 'Titolo Opera', p.descrizione_testo as 'Descrizione Opera',
11      a.nome as 'Nome Autore', a.cognome as 'Cognome Autore'
12 from pubblicazione p, autore a, scrivere s
13 where p.ID=s.pubblicazioneID and a.ID=s.autorID and a.ID=2;
14
15

```

	Titolo Opera	Descrizione Opera	Nome Autore	Cognome Autore
	Illidan	World of Warcraft Books	Maria	Neri
	Shinina	Le avventure più divertenti...	Maria	Neri

(nei casi 1,3 e 5 si evitano possibili occorrenze in più dovute ad omonimia di opere e di chiavi)

4) Versione in cui la ricerca viene effettuata tramite chiave

```

16 • select p.titolo as 'Titolo Opera', p.descrizione_testo as 'Descrizione Opera', k.chiave as 'Genere', pi.ISBN
17 from public_info pi, pubblicazione p, keyword k, collegare c
18 where p.ID=c.publicinfoID and k.chiave=c.chiave and pi.pubblicazioneID=p.ID
19 and k.chiave='Giallo';
20
21
22

```

	Titolo Opera	Descrizione Opera	Genere	ISBN
	Un caso bizzarro per il commissario Carra	Il professor Arbib ha scritto un libro??	Giallo	978-4-5745-6336-6
	50 sfumature di Griao	La frusta fa male se non sai come usarla...	Giallo	978-7-2355-2323-6

5) Versione in cui la ricerca viene effettuata tramite tutti gli attributi elencati sopra (titolo, autore, isbn e chiave)

```

20 • select p.titolo as 'Titolo Opera', p.descrizione_testo as 'Descrizione Opera', k.chiave as 'Genere', pi.ISBN
21 from public_info pi, pubblicazione p, keyword k,
22 collegare c, scrivere s, autore a
23 where p.ID=c.publicinfoID and k.chiave=c.chiave
24 and pi.pubblicazioneID=p.ID and a.ID=s.autorID and p.id=s.pubblicazioneID
25 and pi.ISBN='978-3-4032-6634-5' and a.ID=2 and k.chiave='Fantasy' and p.titolo='Illidan'
26
27
28

```

	Titolo Opera	Descrizione Opera	Genere	ISBN
	Illidan	World of Warcraft Books	Fantasv	978-3-4032-6634-5

Le seguenti due query sono state implementate nel database tramite due STORED PROCEDURE:

QUERY 9: Inserimento di una recensione relativa a una pubblicazione.

Per mostrare il funzionamento di questa query, abbiamo eseguito questa semplice interrogazione per mostrare cosa fosse presente nella tabella "recensire":

13 `select * from recensire`

	pubblicazioneID	utentID	data	campo	approvazione
1	1	1	2016-05-03 12:45:23	Scritto davvero male	1
3	1	1	2002-07-11 07:30:45	Non è male. ma neanche bello	0
1	2	2	2017-06-10 12:10:55	Questo testo è davvero bello!	1
1	3	3	2018-06-05 16:53:45	Opera troppo lunga!!!	1
2	3	3	2008-01-06 21:14:56	Questo testo è davvero brutto	0

La STORED PROCEDURE per questa query è stata costruita in questo modo ed è possibile eseguirla come se fosse una semplice funzione in C, richiamando il nome della funzione Recensire.

La procedure richiede in input l'ID utente, l'ID della pubblicazione e il campo della recensione da inserire.

```

2 DELIMITER $$
3 CREATE PROCEDURE Recensire(IN IDUt int, IN IDPub int, IN Cam Varchar(500))
4 BEGIN
5     insert into recensire(publicazioneID,utentID,Campo) values (IDPub,IDUt,Cam);
6     insert into storia_utente(Descrizione,utentID,PubbID) values (`utente recensisce l'opera`,IDUt,IDPub);
7     update utente set flag_attività=1 where ID=IDUt;
8 END $$
9 DELIMITER ;
10
11 CALL Recensire(1,2,'è bella');
12
13 select * from recensire

```

	pubblicazioneID	utentID	data	campo	approvazione
1	1	1	2016-05-03 12:45:23	Scritto davvero male	1
2	1	1	2018-07-12 17:08:44	è bella	0
3	1	1	2002-07-11 07:30:45	Non è male. ma neanche bello	0
1	2	2	2017-06-10 12:10:55	Questo testo è davvero bello!	1
1	3	3	2018-06-05 16:53:45	Opera troppo lunga!!!	1
2	3	3	2008-01-06 21:14:56	Questo testo è davvero brutto	0

QUERY11: Inserimento di un like relativo a una pubblicazione.

Per mostrare il funzionamento di questa query, abbiamo eseguito queste 3 interrogazione che ci servono rispettivamente per mostrare:

I dati nella tabella "valutare".

13 `select * from valutare`
14

	pubblicazioneID	utentID	data
	NULL	NULL	NULL

I dati nella tabella "pubblicazione".

13 `select * from pubblicazione`

Result Grid | Filter Rows: | Edit: | Export/Import: | Wrap Cell Content: |

ID	titolo	descrizione_testo	indice	#like	UtentiID	EditorID
1	Illidan	World of Warcraft Books	NULL	0	1	1
2	Le avventure di Peppa Pio	Il prosciutto senza orassi	NULL	0	3	1
3	Shining	Le avventure più divertenti...	NULL	0	2	2
4	Un caso bizzario per il commissario Carra	Il professor Arbib ha scritto un libro??	NULL	0	1	2
5	50 sfumature di Grigio	La frusta fa male se non sai come usarla...	NULL	0	3	3
NULL	NULL	NULL	NULL	NULL	NULL	NULL

I dati nella tabella "storia_utente".

13 `select * from storia_utente`

Result Grid | Filter Rows: | Export: | Wrap Cell Cont

timestamp	descrizione	utentiID	pubbID
2018-07-10 20:04:34	ha recensito una opera...	3	3
2014-04-24 13:00:53	ha messo un mi piace...	1	4
1996-05-23 18:00:23	ha tolto mi piace...	2	1
2016-04-06 08:30:53	ha recensito una opera...	3	5

Questa è la query effettiva che rappresenta la funzionalità richiesta:

```

2
3 DELIMITER $$
4 CREATE PROCEDURE AggiungiMiPiaci(IN IDUt int,IN IDPubb int, data_ timestamp)
5 BEGIN
6 insert into valutare(pubblicazioneID,utentiID,data) values (IDPubb,IDUt,data_);
7 update pubblicazione set '#like'='#Like'+1 where ID=IDPubb;
8 insert into storia_utente(descrizione,utentiID,pubbID) values ('Ha messo "mi piace" alla opera', IDUt,IDPubb);
9 update utente set Flag_Actività=1 where ID=IDUt;
10 END $$
11 DELIMITER ;
12

```

Di seguito sono mostrate le modifiche che la chiamata alla STORED PROCEDURE "aggiungiMiPiaci" (AggiungiMiPiaci(2,1,current_timestamp())) ha prodotto nelle tabelle "valutare", "pubblicazione" e "storia_utente" rispettivamente:

Result Grid | Filter Rows: | Edit: |

pubblicazioneID	utentiID	data
1	2	2018-07-12 17:30:21
NULL	NULL	NULL

Result Grid Filter Rows: <input type="text"/> Export: Wrap Cell Content				
	timestamp	descrizione	utentID	pubbID
	2014-04-24 13:00:53	ha messo un mi piace...	1	4
	1996-05-23 18:00:23	ha tolto mi piace...	2	1
	2016-04-06 08:30:53	ha recensito una opera...	3	5
	2018-07-12 17:56:00	Ha messo "mi piace" alla opera	2	1
	2018-07-12 18:02:51	Ha tolto "mi piace" alla opera	2	1

QUERY10 : Approvazione o di una recensione (da parte del moderatore).

Per mostrare il funzionamento di questa query abbiamo eseguita una dopo l'altra le seguenti interrogazioni:

La prima, per poter vedere quali recensioni sono in attesa di approvazione sulla opera con ID=1.

3 • `select * from recensire where pubblicazioneID=1`

<

Result Grid | Filter Rows: Edit: Export/Import:

	pubblicazioneID	utentID	data	campo	approvazione
	1	1	2016-05-03 12:45:23	Scritto davvero male	0
	1	2	2017-06-10 12:10:55	Questo testo è davvero bello!	0
	1	3	2018-06-05 16:53:45	Opera troppo lunga!!!	0

La seconda è la query vera e propria (la funzionalità che dobbiamo sviluppare noi deve avere anche la scelta dell'ID utente, cosa che nell'immagine è stata messa come commento)


```

1 • update recensire SET approvazione=true where pubblicazioneID=1; #and utentID=y;
2
3 • select * from recensire where pubblicazioneID=1

```

pubblicazioneID	utentID	data	campo	approvazione
1	1	2016-05-03 12:45:23	Scritto davvero male	1
1	2	2017-06-10 12:10:55	Questo testo è davvero bello!	1
1	3	2018-06-05 16:53:45	Opera troo lunaa!!!	1

La terza è servita a mostrare come le recensioni erano state modificate correttamente. La query effettiva sarebbe:

UPDATE recensire SET approvazione=true, Where pubblicazioneID=x and utentID=y; (dove x e y sono valori di ID generici).

QUERY12: Calcolo numero dei like per una pubblicazione.

Il funzionamento di questa query lo illustriamo utilizzando prima di tutto una generica interrogazione che mostra tutti i dati delle pubblicazione:

```

1 • select* from pubblicazione p where p.titolo='Le Avventure di Peppa Pig';

```

ID	titolo	descrizione_testo	indice	#like	UtentID	EditorID
2	Le avventure di Peppa Pig	Il prosciutto senza grassi	NULL	0	3	1

Di questa query ne forniamo due versioni:

1) Versione in cui la ricerca della pubblicazione avviene tramite titolo (potrebbe presentare casi di omonimia, ma l'informazione che richiede è più generica)

```

2 #versione con ricerca tramite titolo
3 • select p.titolo as 'Titolo Opera', p.`#like` as 'Numero di like'
4   from pubblicazione p where p.titolo='Le Avventure di Peppa Pig';
5

```

Titolo Opera	Numero di like
Le avventure di Peppa Pig	0

2) Versione in cui la ricerca della pubblicazione avviene tramite ID della pubblicazione (evita icasi di omonimia, ma l'informazione che richiede è più specifica)

```

7 #versione con ricerca tramite ID per evitare omonimie derivanti dall'uso del titolo
8 • select p.titolo as 'Titolo Opera', p.`#like` as 'Numero di like'
9   from pubblicazione p where p.ID=2

```

Titolo Opera	Numero di like
Le avventure di Peppa Pig	0

QUERY13: Estrazione elenco delle recensioni approvate per una pubblicazione.

Per mostrare il funzionamento di questa query, abbiamo utilizzato una interrogazione che mostri le recensioni e lo stato di queste recensioni (sono presenti ancora le modifiche effettuate dalla query #):

```

5 • select pubblicazioneID as 'ID Opera', utentID as 'ID Utente', 'data' as 'Data Recensione'
6   , campo as 'Testo Recensione', approvazione as 'Approvazione Recensione' from recensire
7

```

ID Opera	ID Utente	Data Recensione	Testo Recensione	Approvazione Recensione
1	1	data	Scritto davvero male	1
3	1	data	Non è male. ma neanche bello	0
1	2	data	Ouesto testo è davvero bello!	1
1	3	data	Opera troppa lunga!!!	1
2	3	data	Ouesto testo è davvero brutto	0

La query effettiva che realizza la funzionalità richiesta è la seguente:

```

1 • select pubblicazioneID as 'ID Opera', utenteID as 'ID Utente', 'data' as 'Data Recensione'
2   , campo as 'Testo Recensione', approvazione as 'Approvazione Recensione'
3   from recensire re
4   where re.approvazione=true and re.pubblicazioneID=1
5

```

	ID Opera	ID Utente	Data Recensione	Testo Recensione	Approvazione Recensione
	1	1	data	Scritto davvero male	1
	1	2	data	Questo testo è davvero bello!	1
	1	3	data	Opera troppo lunga!!!	1

QUERY14: Estrazione elenco delle recensioni in attesa di approvazione

Questa query è stata creata con la stessa struttura della precedente solo che nella clausola where è stata inserita la condizione che lo stato di approvazione sia 0

```

1 • select pubblicazioneID as 'ID Opera', utenteID as 'ID Utente', 'data' as 'Data Recensione'
2   , campo as 'Testo Recensione', approvazione as 'Approvazione Recensione'
3   from recensire re
4   where re.approvazione=0
5

```

	ID Opera	ID Utente	Data Recensione	Testo Recensione	Approvazione Recensione
	3	1	data	Non è male, ma neanche bello	0
	2	3	data	Questo testo è davvero brutto	0

NB: lo stato di approvazione delle recensioni può variare tra 0,1,2 (rispettivamente recensione in attesa di approvazione, recensione approvata e recensione scartata).

QUERY 15: Estrazione log delle modifiche effettuate su una pubblicazione.

Per mostrare il funzionamento di questa query, abbiamo utilizzato una interrogazione che mostri le opere in corrispondenza della storia dell'utente (cioè alle azioni che hanno compiuto gli utenti sulle opere, come recensione o mi piace):

```

1 select timestamp as 'Data Recensione', descrizione as 'Testo Recensione',
2 titolo as 'Titolo Opera', nome as 'Nome Utente', cognome as 'Cognome Utente'
3 from storia_utente sto, pubblicazione p, utente u
4 where sto.pubbID=p.ID and sto.utentID=u.ID;
5

```

	Data Recensione	Testo Recensione	Titolo Opera	Nome Utente	Cognome Utente
	2018-07-10 20:04:34	ha recensito una opera...	Shining	Fabio	Capitano
	2014-04-24 13:00:53	ha messo un mi piace...	Ilidan	Marco	Zucconi
	1996-05-23 18:00:23	ha tolto mi piace...	Ilidan	Stefano	Diocaiuti
	2016-04-06 08:30:53	ha recensito una opera...	Ilidan	Fabio	Capitano
	1998-10-23 10:00:01	ha messo un mi piace	Le avventure di Peppa Pig	Stefano	Diocaiuti
	2017-06-07 12:24:23	ha recensito una opera	50 sfumature di Grigio	Stefano	Diocaiuti

Di questa query ne sono state fornite due versioni:

- 1) Versione in cui la ricerca viene effettuata tramite l'ID della pubblicazione (evita i casi di omonimia, ma l'informazione che richiede è più specifica)

```

12 #versione che utilizza la ricerca tramite ID
13 #(per evitare omonimie che si potrebbero incontrare con il titolo)
14
15 select timestamp as 'Data Recensione', descrizione as 'Testo Recensione',
16 titolo as 'Titolo Opera', nome as 'Nome Utente', cognome as 'Cognome Utente', pubbID AS 'ID Opera'
17 from storia_utente sto, pubblicazione p, utente u
18 where sto.pubbID=p.ID and sto.utentID=u.ID and p.ID=1;
19

```

	Data Recensione	Testo Recensione	Titolo Opera	Nome Utente	Cognome Utente	ID Opera
	2014-04-24 13:00:53	ha messo un mi piace...	Ilidan	Marco	Zucconi	1
	1996-05-23 18:00:23	ha tolto mi piace...	Ilidan	Stefano	Diocaiuti	1
	2016-04-06 08:30:53	ha recensito una opera...	Ilidan	Fabio	Capitano	1

- 2) Versione in cui la ricerca viene effettuata tramite il Titolo della pubblicazione (potrebbe presentare casi di omonimia, ma l'informazione che richiede è più generica)

```

6 #Versione che utilizza la ricerca tramite il titolo della pubblicazione
7 select timestamp as 'Data Recensione', descrizione as 'Testo Recensione',
8 titolo as 'Titolo Opera', nome as 'Nome Utente', cognome as 'Cognome Utente', pubbID AS 'ID Opera'
9 from storia_utente sto, pubblicazione p, utente u
10 where sto.pubbID=p.ID and sto.utentID=u.ID and p.titolo='Ilidan';
11

```

	Data Recensione	Testo Recensione	Titolo Opera	Nome Utente	Cognome Utente	ID Opera
	2014-04-24 13:00:53	ha messo un mi piace...	Ilidan	Marco	Zucconi	1
	1996-05-23 18:00:23	ha tolto mi piace...	Ilidan	Stefano	Diocaiuti	1
	2016-04-06 08:30:53	ha recensito una opera...	Ilidan	Fabio	Capitano	1

QUERY 16: Estrazione elenco delle pubblicazioni per le quali è disponibile un download.

Per mostrare il funzionamento di questa query, abbiamo utilizzato una interrogazione che mostri le opere in corrispondenza alle date delle loro sorgenti (solo la "URI" ed il tipo di sorgente):

```
1 select p.titolo as 'Titolo Opera', sourc.URI, sourc.tipo as 'Tipo Sorgente'
2 from pubblicazione p, public_info pi, sorgenti sourc
3 where pi.pubblicazioneID=p.ID and sourc.IDpublicinfo=pi.ID;
4
```

Titolo Opera	URI	Tipo Sorgente
Ilidan	https://www.google.it/immagini	immagine
Le avventure di Peppa Pig	https://www.kindle.it/e-books.pdf	download
Shinina	https://www.ebay.com	acquisto
Un caso bizzarro per il commissario Carra	https://www.amz.com	acquisto
50 sfumature di Grigio	https://www.google.it/50sfumaturedownload	download

La query effettiva che realizza la funzionalità richiesta è la seguente:

```
6 select p.titolo as 'Titolo Opera', sourc.URI
7 from pubblicazione p, public_info pi, sorgenti sourc
8 where pi.pubblicazioneID=p.ID and sourc.IDpublicinfo=pi.ID and sourc.tipo= 'download';
```

Titolo Opera	URI
Le avventure di Peppa Pig	https://www.kindle.it/e-books.pdf
50 sfumature di Grigio	https://www.google.it/50sfumaturedownload

L'interrogazione è stata creata così pensando che non bisognasse controllare se un'opera avesse una "URI" disponibile, bensì pensando che nel database, più precisamente nell'attributo "Tipo" (nelle sorgenti) venisse indicato il tipo di sorgente usata.

QUERY17: Estrazione della lista delle pubblicazioni in catalogo, ognuna con la data dell'ultima ristampa.

Per mostrare il funzionamento di questa query, abbiamo utilizzato una interrogazione che mostri le opere in corrispondenza alle date delle loro ristampe (nel progetto si è pensato che le ristampe possano essere più di una):

```

1 • select p.titolo as 'Titolo Opera', r.data_ristampa as 'Data Ristampe'
2   from pubblicazione p, ristampe r
3  where r.publicinfoID=p.id order by p.titolo
4
5
6

```

Titolo Opera	Data Ristampe
50 sfumature di Grigio	2004-06-04
Ilidan	2015-12-03
Ilidan	2018-06-07
Shinina	1996-04-03
Un caso bizzario per il commissario Carra	2016-02-03
Un caso bizzario per il commissario Carra	2018-07-03

La query che realizza la funzionalità richiesta è la seguente:

```

4 • select p.titolo as 'Titolo Opera', max(r.data_ristampa) as 'Data Ultima Ristampa'
5   from pubblicazione p, ristampe r, public_info pi
6  where r.publicinfoID=pi.ID and pi.pubblicazioneID=p.ID group by p.titolo

```

Titolo Opera	Data Ultima Ristampa
Un caso bizzario per il commissario Carra	2018-07-03
Ilidan	2018-06-07
50 sfumature di Grigio	2004-06-04
Shinina	1996-04-03

QUERY 18: Data una pubblicazione, restituire tutte le pubblicazioni del catalogo aventi gli stessi autori.

Per mostrare il funzionamento di questa query abbiamo utilizzato una semplice interrogazione per mostrare le opere e i corrispondenti autori:

```

1 • select p.titolo as 'Titolo Opera', a.nome as 'Nome Autore', a.cognome as 'Cognome Autore' from pubblicazione p, autore a, scrivere s
2  where p.ID=s.pubblicazioneID and a.ID=s.autorID |
3

```

Titolo Opera	Nome Autore	Cognome Autore
Ilidan	Maria	Neri
Le avventure di Peppa Pig	Omino	Bianco
Shinina	Maria	Neri
Un caso bizzario per il commissario Carra	Cristiano	Grigio
50 sfumature di Grigio	Cristiano	Grigio

La query che realizza la funzionalità richiesta è quella nell'immagine seguente:

```

4 • select p.titolo as 'Titolo Opera', a.nome as 'Nome Autore'
5 from pubblicazione p, autore a, scrivere s
6 where p.ID=s.pubblicazioneID and a.ID=s.autorID
7 and a.nome=(select a.nome from autore a join scrivere s on (a.ID=s.autorID) join
8 pubblicazione p on (s.pubblicazioneID=p.ID) and p.titolo='Illidan');

```

Result Grid		Filter Rows:	Export:	Wrap Cell Content:
	Titolo Opera			
	Nome Autore			
	Illidan			
	Shinina			
	Maria			
	Maria			

Nome del progetto: BIBLIOGRAFIA

Nome	Cognome	Matricola	Contributo nel progetto/ruolo nello sviluppo
FRANCESCO	ZAPPACOSTA	249168	Tutti i membri del gruppo hanno contribuito ad ogni passo del progetto
GIUSEPPE	PAOLILLI	242096	Tutti i membri del gruppo hanno contribuito ad ogni passo del progetto
STEFANO	DIOCIAIUTI	246856	Tutti i membri del gruppo hanno contribuito ad ogni passo del progetto

Data di consegna del progetto: 2018-07-13