

BibliotecAQ

Progetto di Laboratorio di Basi di Dati 2017/2018

Daniele Campli, Tommaso Di Salle



Indice

1. Specifiche assegnate.....	3
1.1. Requisiti testuali.....	3
1.2. Operazioni da realizzare.....	4
2. Formalizzazione e analisi dei requisiti.....	5
2.1. Individuazione dei concetti rilevanti.....	5
2.2. Glossario dei termini.....	6
2.3. Ristrutturazione dei requisiti.....	7
2.3.1. Frasi di carattere generale.....	7
2.3.2. Frasi relative a "Pubblicazione".....	8
2.3.3. Frasi relative a "Sorgente".....	9
2.3.4. Frasi relative a "Metadati".....	10
2.3.5. Frasi relative a "Recensione".....	11
2.3.6. Frasi relative a "Like".....	12
2.3.7. Frasi relative a "Utente".....	12
2.3.8. Frasi relative a "Storico".....	14
2.3.9. Note finali sull'analisi dei requisiti.....	15
3. Progettazione concettuale mediante modello ER.....	16
3.1. Individuazione delle entita' e delle relazioni.....	16
3.1.1. Promozione del concetto "Utente" ad entita'.....	16
3.1.2. Promozione del concetto "Pubblicazione" e "Risorsa" a entita'.....	18
3.1.3. Relazioni tra le entita' "Pubblicazione" e "Utente".....	22
3.2. Schema concettuale risultante.....	24
4. Formalizzazione dei vincoli non espressi tramite il modello ER.....	26
5. Ristrutturazione e ottimizzazione del modello ER.....	27
5.1. Schema logico.....	27
5.2. Analisi dei volumi.....	30
5.3. Analisi delle operazioni.....	30
5.4. Analisi delle ridondanze e osservazioni sulle performance.....	33
5.5. Schema ER risultante ottimizzato.....	34
6. Traduzione del modello ER in modello relazionale.....	35
7. Note sull'implementazione fisica prodotta.....	36
7.1. Viste.....	36
7.2. Trigger.....	36
7.3. Struttura degli script SQL.....	37

1. Specifiche assegnate

1.1. Requisiti testuali

La Biblioteca rappresenta un semplice gestore di **catalogo bibliografico** online. Il catalogo conterrà una serie di **pubblicazioni**, ciascuna delle quali caratterizzata da un **titolo**, una **lista di autori**, un **editore**, una serie di **metadati** (vedi dopo) e, opzionalmente, una **descrizione testuale** (sunto del contenuto o presentazione della pubblicazione), un **indice** (composto dai titoli dei vari **capitoli/sezioni** della pubblicazione, numerati e ordinati) e una serie di **sorgenti** tramite le quali poter accedere alla pubblicazione o a parti di essa. Ciascuna sorgente sarà caratterizzata da un **tipo**, una **URI**, da un **formato** e da una **descrizione**. Esempi di sorgenti valide potrebbero essere i seguenti:

- tipo="immagine", URI="http://server.net/cover.jpg", formato="jpeg", descrizione="copertina"
- tipo="download", URI="http://server.net/book.pdf", formato="application/pdf", descrizione="versione elettronica gratuita"
- tipo="acquisto", URI="http://www.amazon.it/xyz", formato="cartaceo, copertina rigida", descrizione="acquista online"

Per quanto riguarda invece i metadati, questi saranno costituiti (almeno) dalle seguenti informazioni: codice **ISBN**, **numero di pagine**, **lingua**, **data di pubblicazione**, **lista delle ristampe** (numero e data) e **parole chiave** identificative (zero o più).

Ciascuna pubblicazione potrà essere associata a un certo numero di **recensioni** testuali, le quali dovranno anche indicare l'utente che le ha inserite e la data/ora dell'inserimento. Le recensioni saranno moderate, quindi è necessario prevedere un **flag** che indichi se la recensione è stata approvata oppure no dai moderatori. Infine, gli utenti potranno anche assegnare il loro **like** alle pubblicazioni, che quindi dovranno essere associate anche alla lista dei like ricevuti, con utente e data. Sarà inoltre necessario prevedere opportune strutture per immagazzinare i dati dell'utenza. Il sistema, infatti, prevede due tipologie di **utenza**: attiva e passiva. Gli utenti potranno registrarsi nel sito fornendo

i loro **dati anagrafici**, un indirizzo **email** (che verrà usato anche come username) e, ovviamente, la **password** scelta. Il livello di utenza iniziale sarà sempre quello passivo. Per tener traccia delle modifiche effettuate alla bibliografia dai vari utenti attivi, il

sistema dovrà mantenere una “**storia**” di ciascuna **scheda bibliografica**. Ogni entry di tale storia conterrà un timestamp, il nome dell’utente e una descrizione della modifica. Potete scegliere voi il dettaglio di questa descrizione: di base, sono accettabili anche descrizioni del tipo “ha modificato la pubblicazione”, “ha inserito la pubblicazione”, “ha approvato una recensione dell’utente X”, ecc.

1.2. Operazioni da realizzare

1. Modifica del **livello di un utente** (da attivo a passivo e viceversa).
2. Estrazione elenco delle ultime dieci pubblicazioni inserite.
3. Estrazione elenco delle pubblicazioni aggiornate di recente (ultimi 30 giorni).
4. Estrazione elenco degli utenti più “collaborativi” (cioè quelli che hanno inserito più pubblicazioni).
5. Estrazione elenco delle pubblicazioni inserite da un utente.
6. Estrazione catalogo, cioè elenco di tutte le pubblicazioni con titolo, autori, editore e anno di pubblicazione, ordinato per titolo.
7. Estrazione dati completi di una pubblicazione specifica dato il suo ID.
8. Ricerca di pubblicazioni per ISBN, titolo, autore, e parole chiave.
9. Inserimento di una recensione relativa a una pubblicazione.
10. Approvazione di una recensione (da parte del moderatore).
11. Inserimento di un like relativo a una pubblicazione.
12. Calcolo numero dei like per una pubblicazione.
13. Estrazione elenco delle recensioni approvate per una pubblicazione.
14. Estrazione elenco delle recensioni in attesa di approvazione.
15. Estrazione log delle modifiche effettuare su una pubblicazione.
16. Estrazione elenco delle pubblicazioni per le quali è disponibile un **download**.
17. Estrazione della lista delle pubblicazioni in catalogo, ognuna con la data dell’ultima ristampa.
18. Data una pubblicazione, restituire tutte le pubblicazioni del catalogo aventi gli stessi autori.

2. Formalizzazione e analisi dei requisiti

2.1. Individuazione dei concetti rilevanti

I concetti rilevanti sono stati evidenziati in grassetto nella sezione "1. Specifiche assegnate".

Come prima indagine, possiamo evidenziare quali siano i concetti fondamentali e quali quelli a questi ultimi associati. Per riuscire in questo scopo, e' sufficiente analizzare la quantita' di descrizione testuale che viene impiegata per descrivere un certo concetto.

Da tale analisi emergono i seguenti concetti principali (elencati senza un particolare ordinamento):

- Pubblicazione
- Metadati
- Sorgente
- Recensione
- Like
- Utente
- Storia delle schede bibliografiche (i.e. Storico)

Nella sezione "[2.3 Ristrutturazione dei requisiti](#)", effettuando una approfondita analisi della specifica testuale annessa al progetto, evidenzieremo quali sono i concetti associati a quelli principali sopra listati.

2.2. Glossario dei termini

Procediamo con la formalizzazione dei requisiti costruendo un glossario dei termini, basandoci sui concetti principali delineati nella sezione precedente.

TERMINE	DESCRIZIONE	SINONIMI	COLLEGAMENTI
Biblioteca	Collezione di pubblicazioni	Catalogo bibliografico	
Pubblicazione	Testo divulgativo		Sorgente, Recensione, Like, Utente
Metadati	Informazioni riguardanti una pubblicazione		Pubblicazione
Sorgente	Risorse collegate ad una pubblicazione		Pubblicazione, Metadati
Recensione	Opinione di un utente su una pubblicazione		Pubblicazione, Utente
Like	Gradimento di un utente per una pubblicazione		Pubblicazione, Utente
Utente	Persona che interagisce col sistema	Moderatore, utente attivo, utente passivo	Storico, Pubblicazione, Recensione, Like
Storia della bibliografia (Storico)	Registro delle modifiche effettuate sui metadati di una pubblicazione		Utente, Pubblicazione

2.3 Ristrutturazione dei requisiti

In questa fase, il focus sara' di ricostruire i requisiti in modo da eliminare le ambiguita' e definire quali siano i concetti rilevanti nel nostro sistema.

Preludio sulla notazione adottata

In questa sezione e' stata adottata -ove necessario- una sintassi specifica per definire quali valori possa assumere un certo termine o una certa caratteristica relativa ad un dato concetto, sia esso chiamato "C". Lo scopo di utilizzare una sintassi ad hoc e' sotteso al fatto di meglio riassumere visivamente quali siano i valori che il termine in esame puo' assumere. Al fine di descrivere la sintassi in questione, forniamo un esempio.

Il concetto C e' caratterizzato da:

- [...]
- *formato* ← *digitale* | *cartaceo*
- [...]

Possiamo tradurre la stenografia impiegata in linguaggio naturale come:

"il formato" del concetto C puo' essere "digitale" o "cartaceo".

2.3.1. Frasi di carattere generale

La biblioteca rappresenta un semplice gestore di catalogo bibliografico online.

Requisito ristrutturato

La base di dati da implementare rappresenta una biblioteca.

NOTA: Nel nostro dominio di interesse, "Catalogo bibliografico online" e' un sinonimo di biblioteca, per cui d'ora in avanti ci riferiremo al primo semplicemente con quest'ultimo termine, piu' breve e diretto.

2.3.2. Frasi relative a "Pubblicazione"

Il catalogo conterra' una serie di pubblicazioni, ciascuna delle quali caratterizzata da un titolo, una lista di autori, un editore, una serie di metadati, e opzionalmente, una descrizione testuale (sunto del contenuto o presentazione della pubblicazione), un indice (composto dai titoli dei vari capitoli/sezioni della pubblicazione, numerati e ordinati) e una serie di sorgenti tramite le quali poter accedere alla pubblicazione o a parti di essa.

(da operazione no. 6) Estrazione catalogo, cioe' l'elenco di tutte le pubblicazioni con titolo, autori, editore e anno di pubblicazione, ordinato per titolo.

Requisito ristrutturato

La biblioteca e' una collezione di pubblicazioni.

Ogni pubblicazione e' caratterizzata da:

1. titolo
2. lista di autori (vedi NOTE sottostanti)
3. descrizione testuale opzionale
4. indice, costituito dai titoli dei capitoli della pubblicazione numerati e ordinati (vedi NOTE sottostanti)
5. sorgenti, ossia risorse esterne collegate alla pubblicazione stessa
6. editore
7. anno di pubblicazione

NOTE: Dal concetto rilevante di "Pubblicazione" scaturiscono quelli associati di "Indice" e "Autore".

2.3.3. Frasi relative a “Sorgente”

Ciascuna sorgente sara' caratterizzata da un tipo, una URI, da un formato e da una descrizione. Esempi di sorgenti valide potrebbero essere i seguenti:

- tipo="immagine", URI="http://server.net/cover.jpg", formato="jpeg", descrizione="copertina"*
- tipo="download", URI="http://server.net/book.pdf", formato="application/pdf", descrizione="versione elettronica gratuita"*
- tipo="acquisto", URI="http://www.amazon.it/xyz", formato="cartaceo, copertina rigida", descrizione="acquista online"*

Requisito ristrutturato

Ogni sorgente e' caratterizzata da:

1. tipo (vedi NOTE sottostante)
2. URI
3. formato (vedi NOTE sottostante)
4. descrizione testuale

NOTE:

- il termine "sorgente" in questo contesto puo' essere meglio delineato con "risorsa", in quanto dalla specifica si evince che una sorgente non e' altro che un contenuto digitale associato ad una pubblicazione. A ragion di cio', d'ora in avanti ci riferiremo a "sorgente" con il lemma "risorsa", il quale e' in grado di esprimere piu' correttamente la semantica del suo stesso ruolo.
- Stando alla specifica, il "tipo" di una risorsa (i.e. di una sorgente) non e' definito con precisione, ma vengono forniti degli esempi che ne chiariscono l'uso. Al fine di disambiguare la specifica, restringeremo "tipo" ai valori seguenti:

tipo ← download | acquisto | applicazione | audio | immagine | video | testo

Per quanto “download” sia un termine generale e possa corrispondere -per esempio- ad un file audio o ad un’immagine, riteniamo che sia rilevante per il committente mantenere i riferimenti a tutte le risorse **direttamente** scaricabili attraverso l'applicativo di gestione della biblioteca, come dimostra la presenza

stessa della richiesta di operazione no. 16. Una motivazione analoga ci spinge a poter definire tipo come "acquisto", in quanto si potrebbero voler implementare successivamente delle operazioni che forniscano una lista delle pubblicazioni che dispongano di uno o piu' link per essere acquistate tramite siti ecommerce.

- Gli esempi dati, come nel caso di "tipo" forniscono un'idea di cosa il formato rappresenti. Anche qui e' necessaria una specificazione di quali siano i valori che un formato di risorsa possa assumere:
 - se "tipo" corrisponde a "acquisto", formato ← digitale | cartaceo
 - se "tipo" corrisponde ad un valore diverso da "acquisto", e.g. applicazione, verranno usati i [formati mediatype](#) specificati dagli RFC6838 e RFC4855. Quindi, per esempio, nel caso in cui "formato" sia "applicazione", valori ammessi saranno della forma "application/<nome_applicazione>" (e.g. "application/pdf"). La notazione "application/*" identifica tutti i formati standard definiti negli RFC precedentemente nominati che hanno come prefisso "application/"; analogo discorso vale per "audio/*", "video/*", "image/*", e "text/*".
 - tipo ← applicazione, formato ← application/*
 - tipo ← audio, formato ← audio/*
 - tipo ← immagine, formato ← image/*
 - tipo ← video, formato ← video/*
 - tipo ← testo, formato ← text/*
 - tipo ← download, formato ← application/* | audio/* | image/* | video/* | text/*

2.3.4. Frasi relative a “Metadati”

Per quanto riguarda invece i metadati, questi saranno costituiti (almeno) dalle seguenti informazioni: codice ISBN, numero di pagine, lingua, data di pubblicazione, lista delle ristampe (numero e data) e parole chiave identificative (zero o piu').

Requisito ristrutturato

I metadati relativi ad una pubblicazione sono l'insieme delle seguenti informazioni:

1. codice ISBN
2. numero di pagine
3. lingua
4. data di pubblicazione
5. lista delle ristampe, con numero e data
6. parole chiave (zero o piu')

NOTA: Dal concetto rilevante di "Metadati" scaturisce quello associato di "Ristampa" o, per essere piu' precisi, "Versione Stampa".

2.3.5. Frasi relative a "Recensione"

Ciascuna pubblicazione potra' essere associata ad un certo numero di recensioni testuali, le quali dovranno anche indicare l'utente che le ha inserite e la data/ora dell'inserimento. Le recensioni saranno moderate, quindi e' necessario prevedere un flag che indichi se la recensione e' stata approvata oppure no dai moderatori.

Requisito ristrutturato

Una determinata recensione e' associata ad un unico utente e ad una unica pubblicazione. Questo implica che un utente, data una certa pubblicazione, puo' scrivere al piu' una recensione su di essa. Zero o piu' recensioni possono essere associate ad una pubblicazione, cosi come zero o piu' recensioni possono essere state scritte da un utente.

Una recensione e' cosi costituita:

1. testo
2. stato (vedi NOTE sottostanti)
3. data e ora di inserimento

NOTE: lo stato di una recensione tiene traccia se quest'ultima sia stata approvata o meno da un moderatore. I valori che può assumere "stato" sono i seguenti:

stato ← in_attesa | approvata

Dal concetto rilevante di "Recensione" scaturisce quello associato di "Moderatore". Si veda la sezione "2.3.7. Frasi relative a "Utente"" per maggiori dettagli.

2.3.6. Frasi relative a "Like"

Infine, gli utenti potranno anche assegnare il loro like alle pubblicazioni, che quindi dovranno essere associate anche alla lista dei like ricevuti, con utente e data.

Requisito ristrutturato

Un dato utente può assegnare al più un like ad una data pubblicazione; una pubblicazione può avere associati zero o più like, così come un utente può assegnare in totale zero o più like. Le seguenti informazioni devono essere mantenute riguardo uno specifico like:

1. identificativo dell'utente che ha inserito il like
2. identificativo della pubblicazione a cui un dato utente ha assegnato un like
3. data di inserimento del like

NOTA: da un colloquio con il committente è stato stabilito da quest'ultimo che i like riguardano esclusivamente una pubblicazione, e non recensioni.

2.3.7. Frasi relative a "Utente"

Sarà inoltre necessario prevedere opportune strutture per immagazzinare i dati dell'utenza. Il sistema, infatti, prevede due tipologie di utenza: attiva e passiva. Gli utenti potranno registrarsi nel sito fornendo i loro dati anagrafici, un indirizzo email (che verrà usato anche come username) e, ovviamente, la password scelta. Il livello di utenza iniziale sarà sempre quello passivo.

Requisito ristrutturato

Le seguenti informazioni devono essere mantenute riguardo uno specifico utente:

1. indirizzo email
2. password
3. tipo (vedi NOTE sottostanti)
4. anagrafica (vedi NOTE sottostanti)

NOTE:

- Un utente puo' essere registrato o visitatore.
- Il "tipo" dell'utente registrato puo' essere "attivo" oppure "passivo". In riferimento alla note della sezione "[2.3.5. Frasi relative a "Recensione"](#)", un moderatore e' un tipo di utente attivo che si occupa di approvare recensioni. Ulteriori raffinamenti alla specifica assegnata riguardo il concetto di "Moderatore" verranno effettuati in seguito.
- Ogni utente ha associata un'anagrafica; il concetto rilevante "Utente" ha dunque associato un concetto "Anagrafica". Nella specifica non sono listate le informazioni da mantenere riguardo quest'ultima; come assunzione, quelle che verranno conservate nella base di dati saranno le seguenti:

Anagrafica

1. Nome
 2. Cognome
 3. Codice fiscale
 4. Data di nascita
 5. Luogo di nascita
 6. Nazionalita'
- Da un colloquio con il committente sono inoltre emersi i seguenti requisiti aggiuntivi:
 - la promozione da utente passivo ad attivo la possono effettuare solo altri utenti attivi;
 - un utente passivo non puo' inserire pubblicazioni;

- tutti gli utenti attivi possono moderare le recensioni di qualsiasi pubblicazione;
- un utente non registrato puo' vedere la lista delle pubblicazioni.
- Requisiti derivati: le operazioni che un utente passivo puo' effettuare sono le seguenti:
 - inserire una recensione per una pubblicazione
 - inserire un like ad una pubblicazione

Un utente attivo, in aggiunta, puo' anche:

- inserire pubblicazioni
- approvare una recensione

2.3.8. Frasi relative a "Storico"

Per tener traccia delle modifiche effettuate alla bibliografia dai vari utenti attivi, il sistema dovra' mantenere una storia di ciascuna scheda bibliografica. Ogni entry di tale storia conterra' un timestamp, il nome dell'utente e una descrizione della modifica. Potete scegliere voi il dettaglio di questa descrizione: di base, sono accettabili anche descrizioni del tipo "ha modificato la pubblicazione", "ha inserito la pubblicazione", "ha approvato una recensione dell'utente X", ecc.

Requisito ristrutturato

Lo storico e' una collezione di log in relazione alle operazioni che un dato utente effettua su una pubblicazione. La specifica asserisce che si ha necessita' di tenere traccia delle "modifiche effettuate alla bibliografia dai vari **utenti attivi**"; all'atto pratico dunque si deve tenere traccia delle operazioni di inserimento di una recensione, inserimento di un like, inserimento di una pubblicazione e cambiamento dello stato di una recensione (da "in attesa" ad "approvata").

Possiamo cosi riassumere i requisiti riguardanti "Storico": quest'ultimo e' una collezione di log, dove "log" e' un concetto associato definito dai seguenti elementi:

Log

1. timestamp

2. identificativo dell'utente coinvolto nell'operazione svolta
3. identificativo della pubblicazione coinvolta nell'operazione svolta
4. identificativo dell'operazione svolta

2.3.9. Note finali sull'analisi dei requisiti

Occorre definire una serie di considerazioni finali per completare la fase di analisi dei requisiti.

- Nella sezione "2.3.7. Frasi relative a "Utente"" e' emersa una prima suddivisione delle categorie di utenza: visitatori e utenti registrati. Dal momento che soltanto gli utenti registrati avranno un'interazione rilevante con la base di dati, d'ora in avanti con "Utente" ci riferiremo unicamente all'insieme di utenti registrati, siano essi attivi o passivi.
- Si puo' assumere in maniera deduttiva che si voglia poter modificare o cancellare pubblicazioni, e che gli utenti passivi siano gli unici a cui sia permesso di modificarne o cancellarne una, oltre che di inserirne. Possiamo dunque riassumere questo aspetto affermando che gli utenti attivi siano i soli incaricati della gestione delle pubblicazioni, dove per "gestione" indichiamo appunto le operazioni sopra esposte.
- Una pubblicazione puo' avere varie edizioni; possiamo dunque aggiungere ad edizione la caratteristica aggiuntiva "edizione".

3. Progettazione concettuale mediante modello ER

3.1. Individuazione delle entita' e delle relazioni

3.1.1. Promozione del concetto "Utente" ad entita'

Come deducibile dalla sezione ["2.3.7. Frasi relative a "Utente"'](#), il concetto di "Utente" si presta naturalmente ad essere promosso ad entita'.

Possiamo raffinare ulteriormente la specifica in questo modo:

- un utente puo' essere unicamente attivo o passivo; un moderatore e' particolare tipo di utente attivo;
- un utente e' caratterizzato dai seguenti attributi:
 - email
 - password
 - anagrafica, a sua volta costituita da:
 - nome
 - cognome
 - codice fiscale (cf)
 - data di nascita
 - luogo di nascita
 - nazionalita'

NOTE:

- anagrafica e' ovviamente un attributo composto;
- possiamo imporre una gerarchia nell'entita' utente: la gerarchia e' totale nel caso di utente attivo o passivo; al contrario, un moderatore e' un tipo di utente attivo, ma potrebbero eventualmente esserci differenti categorie di quest'ultimo, per cui in questa situazione preferiremo una gerarchia parziale ad una totale.

VINCOLI:

- L'email di un utente, dovendo assumere anche il ruolo di username -come asserito nella specifica-, identifica l'utente stesso, per cui e' univoca: non possono esserci due utenti diversi con la stessa email, in sostanza.
- Il codice fiscale di un utente e' univoco, per cui non possono essere registrati due utenti con stesso codice fiscale.

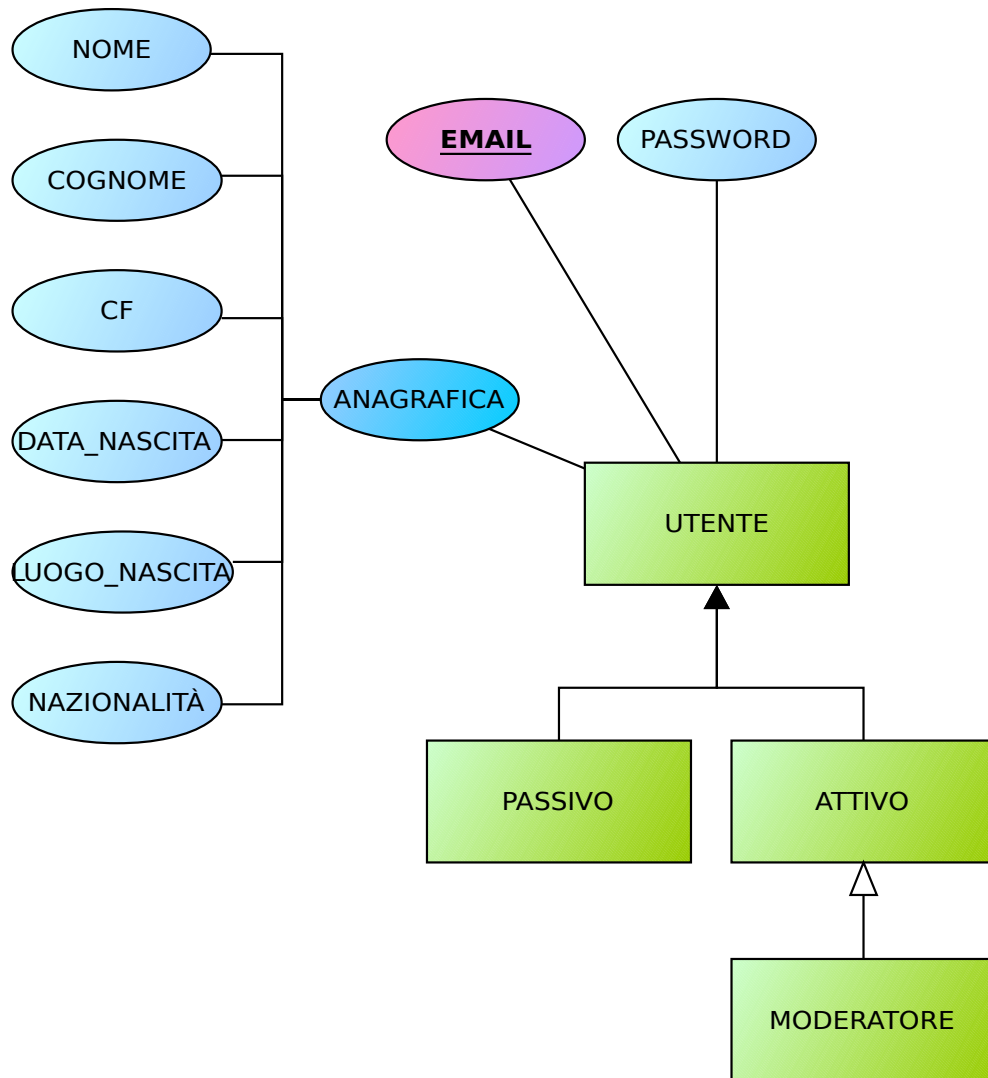


Figura 1: Bozza di schema concettuale dell'entita' "Utente"

Selezione della chiave primaria per l'entita' "Utente"

Poiche' e' stato imposto il vincolo di univocita' sull'email di un dato utente, possiamo utilizzare l'attributo "email" come chiave primaria dell'entita' "Utente".

3.1.2. Promozione del concetto "Pubblicazione" e "Risorsa" a entita'

La sezione ["2.3.2. Frasi relative a "Pubblicazione"'"](#) chiarisce che anche "Pubblicazione", come "Utente", si presta ad essere promossa ad entita'.

Una pubblicazione e' caratterizzata dai seguenti attributi:

- titolo
- autori
- indice
- metadati

E' naturale considerare che degli autori si vorra' tenere traccia almeno del nome e del cognome di ciascuno, per cui possiamo ritenere l'attributo "autore" sia composto che multi valore: composto perche' esso e' costituito da nome e cognome, multi valore perche' puo' assumere molteplici valori (i.e. una pubblicazione puo' essere attribuita a piu' autori).

Un discorso analogo si puo' estendere all'attributo "indice", anch'esso passibile di essere composto e multivalore. Un indice, infatti, e' un'insieme di capitoli; un capitolo e' costituito da un titolo, una descrizione ed un numero (per garantire un'ordinamento degli stessi all'interno di un indice). Ne deduciamo che "indice" puo' assumere piu' valori (considerandolo come un insieme di capitoli), e ha degli attributi ad esso associati (listati sopra), per cui e' composto.

I "metadati" di una pubblicazione sono certamente un attributo composto, infatti sono costituiti da lingua, numero delle pagine, codice isbn, data di pubblicazione e da un insieme di parole chiave (rif. sezione ["2.3.4. Frasi relative a "Metadati"'"](#)).

Le parole chiave possono essere molteplici: siamo dunque in presenza di un attributo multivalore.

Anche l'attributo "ristampa", puo' assumere un insieme di valori, ed e' inoltre composto da due attributi, numero (versione) della ristampa, e data di ristampa.

Infine, per quanto detto in ["2.3.3. Frasi relative a "Sorgente"'"](#) promuoviamo ad entita' anche il concetto di "Risorsa", costituita dai seguenti attributi:

- formato

- uri
- descrizione
- tipo

OSSERVAZIONI:

- I "metadati" sono per definizione dati che descrivono altri dati; i metadati di una pubblicazione dunque sono tutti quei dati che la definiscono nelle suoi dettagli. Possiamo dunque assimilare a metadati tutti quegli attributi inizialmente associati direttamente a pubblicazione (per esempio titolo e lista di autori); in fondo, metadati e' comunque un attributo di pubblicazione, per cui rimangono validi tutti i ragionamenti sviluppati fino a questo punto.
- Per il punto sopra esposto, la versione di ristampa che identifica una determina pubblicazione e' un'informazione che puo' essere considerata come facente parte dei metadati della stessa.
- Un indice e' assimilabile a far parte dei metadati di una pubblicazione, tuttavia scegliamo di tenerlo distinto da "metadati" e direttamente connesso a "Pubblicazione". La scelta e' basata sul buon senso, in quanto possiamo immaginare i metadati come una collezione *minima* di dati realmente utili a fornire un'idea di cosa la pubblicazione tratti al suo interno, congiuntamente a tutte le informazioni che descrivono la produzione logica e fisica della stessa (si pensi all'attributo "Autore", ad esempio). Un indice aggiungerebbe uno strato di informazioni secondario, pertanto separabile dal contesto dei metadati.
- Possiamo aggiungere degli ulteriori attributi a "metadati" per consentire al sistema di conservare informazioni il piu' complete possibile; ad esempio, e' possibile aggiungere i seguenti attributi:
 - categoria (vedi NOTA sottostante)
 - edizione
 - editore
 - sinossi

NOTA: per quanto riguarda l'attributo "categoria", potra' avere valori in:

Categoria:

- report
- articolo
- libro
- tesi

Se necessario potranno essere aggiunti possibili valori in seguito.

Selezione delle chiavi primarie per le entita' "Pubblicazione" e "Risorse"

L'unico attributo che si presta ad essere una chiave primaria per l'entita' "Utente" e' "isbn", in quanto *"[...] Ogni codice ISBN identifica in modo univoco ogni specifica edizione di un libro (non però le semplici ristampe, che mantengono lo stesso codice dell'edizione cui si riferiscono) e, una volta assegnato, non può più essere riutilizzato."* (rif. [Wikipedia](#)).

Per quanto riguarda l'entita' "Risorse", nessun attributo e' adatto ad assumere il ruolo di chiave primaria, per cui aggiungeremo un attributo univoco "id_risorsa" che ne faccia le veci.

Individuazione di relazioni tra "Pubblicazione" e "Risorse"

Come emerge da "2.3.3. Frasi relative a "Sorgente"", una pubblicazione puo' avere zero o piu' risorse associate: possiamo quindi stabilire una relazione "Link" tra le due entita', con le seguenti cardinalita':

- una risorsa potra' essere associata almeno ad una e al piu' ad un numero finito n di pubblicazioni;
- una pubblicazione potra' essere associata a zero o piu' risorse (come gia' detto sopra).

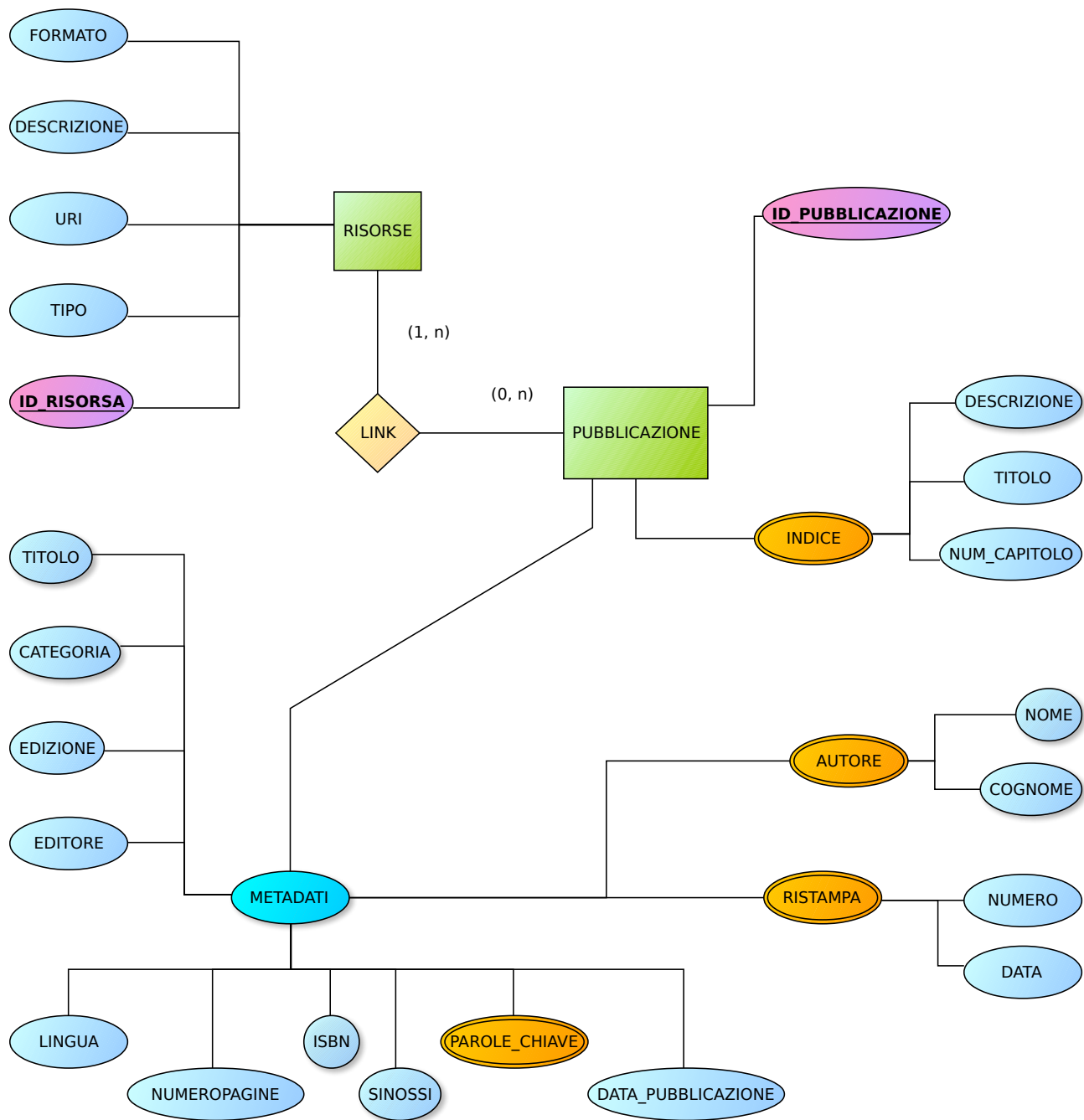


Figura 2: Bozza di schema concettuale per le entita' "Risorsa" e "Pubblicazione"

3.1.3. Relazioni tra le entita' "Pubblicazione" e "Utente"

Le sezioni "2.3.5. Frasi relative a "Recensione"", "2.3.6. Frasi relative a "Like"", "2.3.8. Frasi relative a "Storico"" portano alla luce la presenza di concetti rilevanti che sono in associazione sia con l'entita' "Utente" che con "Pubblicazione".

Osservazioni sui concetti "Recensione" e "Like"

Considerando la possibilita' di promuovere "Recensione" ad entita', abbiamo constatato le seguenti conseguenze:

- poiche' un utente puo' scrivere una singola recensione per una pubblicazione e, viceversa, una recensione puo' essere attribuita ad un solo utente, saremmo costretti ad inserire una relazione tra Utente e Recensione dove le cardinalita' sono (1, 1) in entrambi i versi.
- Inoltre, una eventuale entita' "Recensione" potra' essere associata ad una sola pubblicazione e, viceversa, una pubblicazione potra' avere zero o piu' recensioni annesse. Questo implicherebbe una cardinalita' della relazione tra Recensione e Pubblicazione pari a (1, 1) nel primo verso e (0, n) nel secondo.
- Possiamo sintetizzare le stesse informazioni inserendo "Recensione" come relazione, migliorando la leggibilita' dello schema ER risultante e riducendo il numero totale di entita' e relative relazioni coinvolte.

Un ragionamento analogo puo' essere sostenuto per "Like", pertanto anche quest'ultima potremo classificarla come relazione.

Riguardo le cardinalita' di "Recensione" e "Like":

- un utente puo' inserire al piu' una recensione per pubblicazione (0, 1), mentre ad una pubblicazione sono associate al piu' un numero finito n di recensioni (0, n);
- un utente puo' inserire al piu' un like per pubblicazione (0, 1), mentre una pubblicazione puo' ricevere al piu' n like (0, n).

Per quanto riguarda eventuali attributi di Recensione e Like, per la prima dobbiamo conservare stato ("approvata" o "in attesa"), timestamp e testo, mentre per la seconda esclusivamente la data in cui e' stato assegnato il like.

Osservazioni sul concetto di "Storico"

Per capire se assegnare a "Storico" il ruolo di entita' oppure di relazione occorre definire con precisione di cosa lo storico dovra' tener traccia. La sezione ["2.3.8. Frasi relative a "Storico"'"](#) delinea che lo storico e' una collezione di log delle operazioni che gli utenti attivi effettuano sulle pubblicazioni. A questo punto possiamo aggiungere delle caratteristiche utili al nostro sistema, e al contempo definire delle limitazioni sul concetto di storico.

- E' bene notare che lo storico conserva esclusivamente informazioni sulla cronologia delle operazioni svolte sulle pubblicazioni, e non tiene traccia di altre informazioni (es. una modifica da parte di un utente dei suoi dati anagrafici).
- Possiamo ampliare il set di operazioni effettuabili su una pubblicazione: saranno consentite operazioni di modifica e cancellazione di una pubblicazione, oltre alla possibilita' di inserirne una, recensirla o applicargli un like. Lo storico dovra' - naturalmente- conservare il tipo dell'operazione svolta.
- Estenderemo il monitoraggio delle operazioni svolte su pubblicazioni anche agli utenti passivi, in modo da mantenere una collezione completa delle operazioni svolte da *tutti* gli utenti registrati nel sistema. Questo potra' risultare utile poiche', qualora si vogliano stimare le cronologie dei livelli di gradimento (dove il gradimento e' assegnato con dei like, ad esempio) sara' possibile farlo.

I punti precedenti, in particolare il primo, suggeriscono di definire il concetto "Storico" come relazione tra Utente e Pubblicazione.

Relativamente alla cardinalita', notiamo che:

- un utente potra' generare al piu' n log riguardanti una pubblicazione (cardinalita' $(0, n)$)
- una pubblicazione avra' associati un minimo di un log e un massimo di n (cardinalita' $(1, n)$), dove il minimo e' fissato ad 1 in quanto al momento dell'inserimento di una pubblicazione nella base di dati, verra' inserito un log nello storico che annotera' l'atto di inserimento della stessa da parte di un preciso utente con un preciso timestamp.

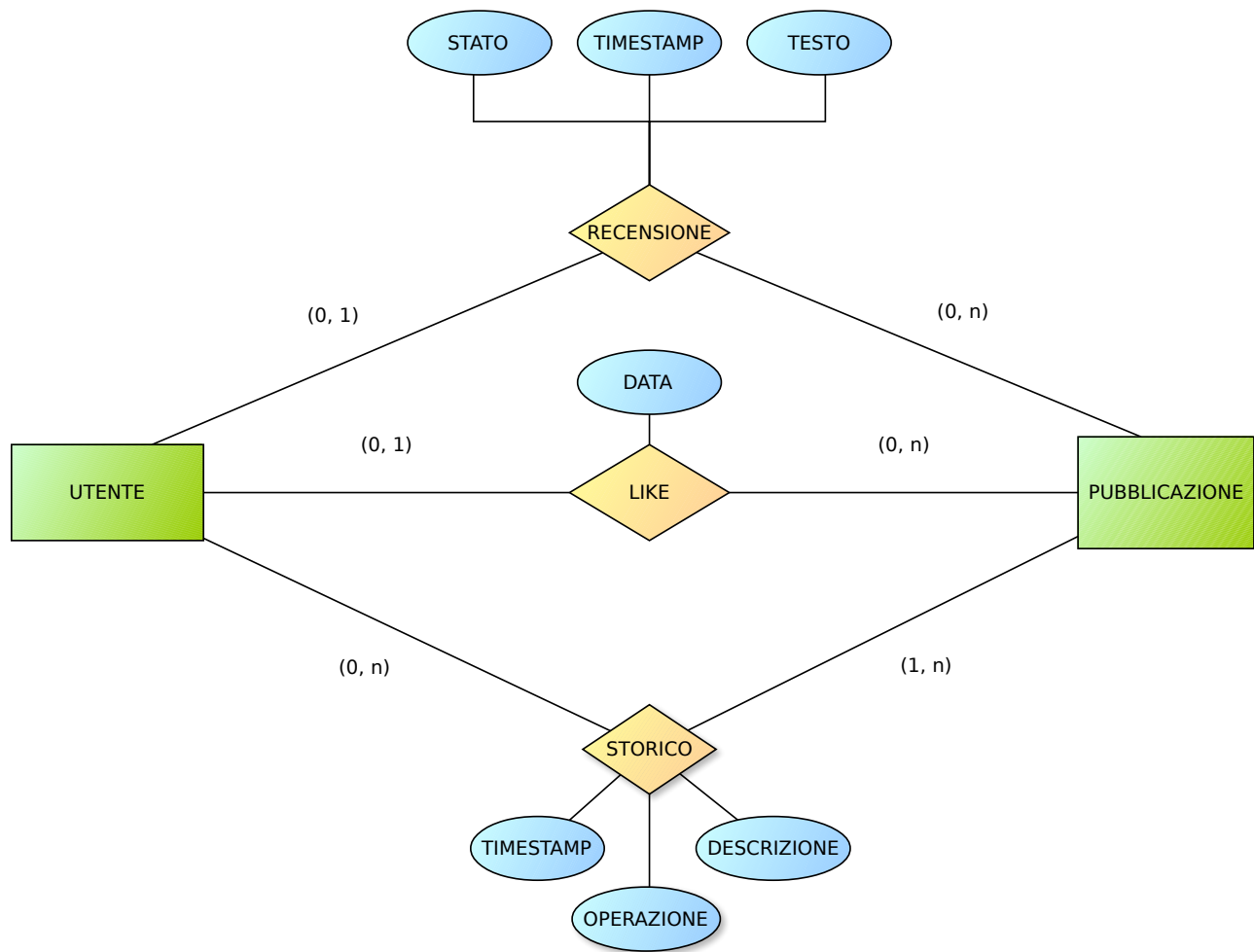


Figura 3: Bozza di schema concettuale rappresentanti le relazioni tra "Utente" e "Pubblicazione"

3.2. Schema concettuale risultante

Lo schema concettuale completo e' ottenuto unendo gli schemi concettuali parziali sviluppati nelle sezioni precedenti.

NOTA: rimane non specificato dal diagramma che le operazioni di gestione di pubblicazioni (inserimenti, cancellazioni, modifiche) possono essere effettuati soltanto da utenti attivi.

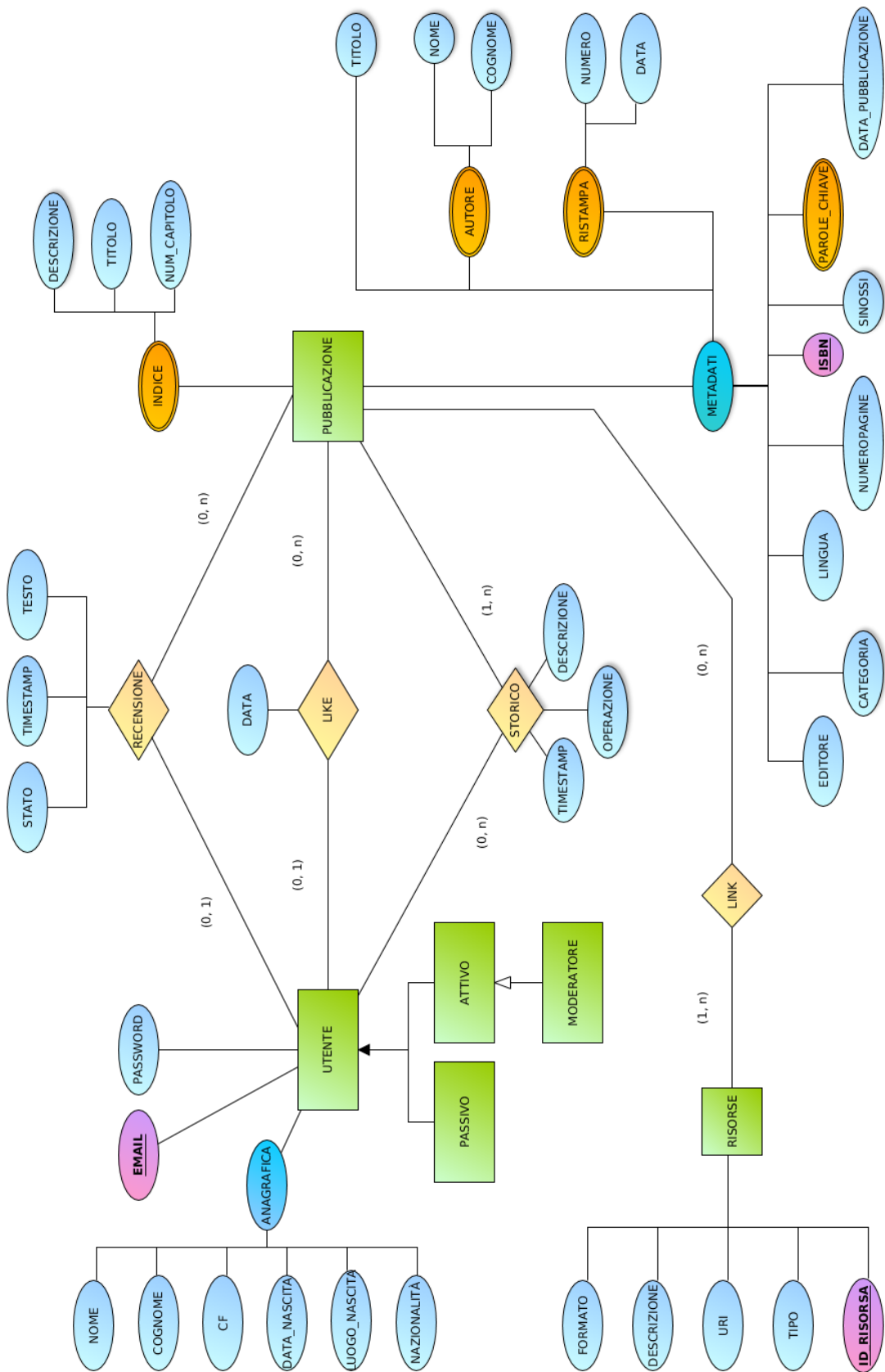


Figura 4: Schema concettuale completo

4. Formalizzazione dei vincoli non espressi tramite il modello ER

Vincoli su:

- Anagrafica: la data di nascita deve essere minore della data al momento della registrazione;
- Storico: le operazioni registrate in storico sono le seguenti:
 - inserimento/cancellazione/modifica pubblicazione
 - inserimento/cancellazione/modifica recensione
 - inserimento/cancellazione/modifica risorsa
 - inserimento/cancellazione like
- Pubblicazione: i possibili valori di categoria sono:
 - tesi
 - report
 - articolo
 - libro
- Risorse: vincoli su formato e tipo e le loro relazioni, definite nella sezione ["2.3.3. Frasi relative a "Sorgente""](#)
- Versione Stampa: la data di stampa di una pubblicazione non puo' essere precedente alla data di pubblicazione della stessa (rif. attributo "data_pubblicazione" appartenente a "Metadati"); inoltre la data dell'ultima ristampa non puo' essere precedente alla data di stampa delle versioni che la antecedono.

5. Ristrutturazione e ottimizzazione del modello ER

5.1. Schema logico

Dallo schema concettuale completo (rif. sezione ["3.2. Schema concettuale risultante"](#)), applichiamo le seguenti trasformazioni:

- Aggiungiamo per praticità l'attributo "id_utente", che useremo come chiave primaria al posto dell'attributo "email".
- L'attributo composto "Anagrafica" è promosso ad entità; inseriremo una relazione "Identificazione" tra Anagrafica e Utente. Poiché "Anagrafica" non ha attributi adatti a rivestire il ruolo di chiave primaria (per via delle [omocodie](#) sul codice fiscale), essa è un'entità debole; tuttavia, siccome un singolo utente è associato ad una ed una sola anagrafica, possiamo includere la relazione "Identificazione" nella chiave.
- Rimuoviamo la gerarchia di generalizzazione su Utente, inserendo un semplice attributo "tipo" che identifichi se sia attivo oppure passivo.
- L'attributo multivalore "Indice" è sostituito da una relazione, che chiameremo in maniera omonima, la quale lega Pubblicazione e una nuova entità "Capitolo"; quest'ultima erediterà tutti gli attributi precedentemente associati all'attributo composto "Indice". Le cardinalità saranno (0, n) tra Pubblicazione e Indice, in quanto ad una pubblicazione può avere associati al più n capitoli (anche 0, se essa non presenta un indice!), e (1, 1) tra Indice e Capitolo poiché uno specifico capitolo sarà relativo unicamente ad una pubblicazione.
- L'attributo multivalore "Autore" è sostituito da una nuova entità omonima, collegata a "Metadati" attraverso una relazione "Attribuzione". Gli attributi su autori saranno "nome", "cognome" e "id_autore", quest'ultimo fungerà da chiave primaria dell'entità. Le cardinalità saranno (1, n) tra Metadati e Attribuzione (in quanto un set di metadati di una pubblicazione potrà avere come riferimenti da uno ad n autori), e (0, n) tra Attribuzione e Autore (poiché un autore potrebbe essere riferito da più set di metadati, i.e. a più pubblicazioni).
- L'attributo composto "Metadati" è sostituito da una nuova entità omonima che eredita tutti gli attributi del primo. L'attributo composto "Ristampa" è sostituito da una relazione "Ristampa" e da una nuova entità "Versione Stampa", caratterizzata

dagli attributi "numero_copie", "data_stampa" e "id_versione_stampa", con quest'ultimo chiave primaria dell'entita'. Le cardinalita' saranno (1, 1) tra "Versione Stampa" e Ristampa (poiche' una versione corrisponde univocamente ad un set di metadati, i.e. ad una pubblicazione) e (1, n) tra Ristampa e "Metadati" (poiche' un set di metadati potra' mantenere riferimenti ad una o piu' versioni di ristampa di una stessa pubblicazione). Infine, creiamo una nuova entita' "Mediatype" che descriva le associazioni tra formato e tipo, avente appunto come attributi "formato", "tipo" e "id_mediatype" (nel ruolo di chiave primaria). "Mediatype" sara' associata all'entita' "Risorse" (adesso contenente gli attributi "descrizione", "uri" e "id_risorsa" come chiave primaria) attraverso una relazione "Classificazione". Questa modifica dello schema ER e' dettata dal fatto che si vogliono distinguere i concetti di "Risorsa" e "Tipo digitale della Risorsa". Dei semplici attributi "formato" e "tipo", non avrebbe potuto soddisfare i requisiti nella sezione ["2.3.3. Frasi relative a "Sorgente""](#), in cui vengono delineate delle correlazioni precise tra "formato" e "tipo".

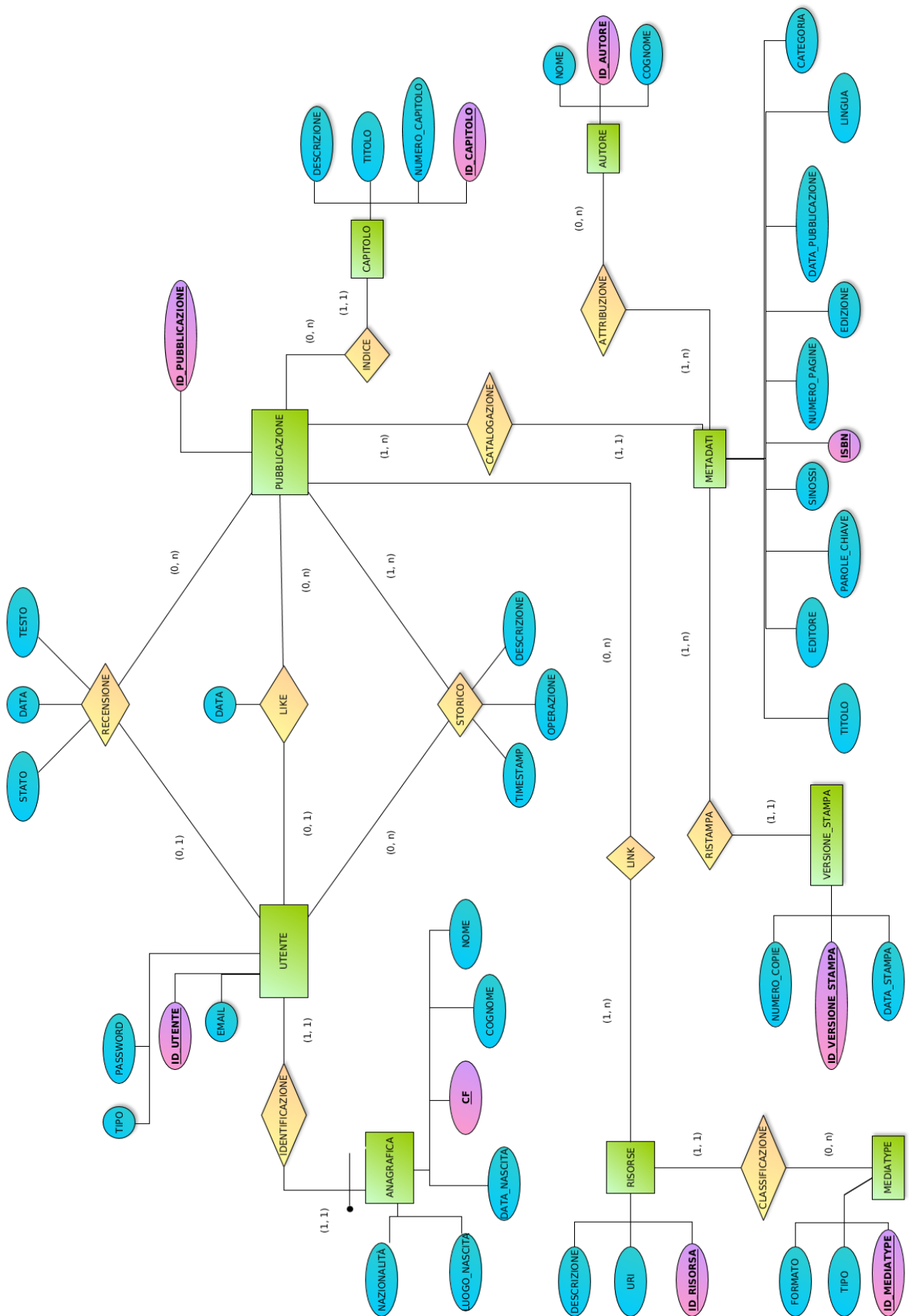


Figura 5: Schema logico completo

5.2. Analisi dei volumi

Non sono stati forniti dati sui volumi, e purtroppo e' abbastanza difficile stimare -anche approssimativamente- quali possano essere; dunque, per cercare di ottimizzare al meglio il sistema, cercheremo di effettuare un'analisi approfondita sulle operazioni richieste, in modo tale da mitigare tale mancanza.

5.3. Analisi delle operazioni

Non disponendo di valori di riferimento approssimativi per le operazioni da eseguire, assegneremo una priorita' a ciascuna di esse, selezionando tra tre livelli progressivi di importanza (priorita' **bassa**, **media**, **alta**). Ovviamente, le operazioni con priorita' piu' alta saranno quelle che verranno richiamate piu' volte, e via via a scendere per quelle con priorita' media e bassa. Sono usati dei colori per evidenziare la classificazione risultante, ed e' fornita una breve spiegazione a supporto.

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

Si prevede che il numero degli utenti attivi rimanga pressoché costante nel tempo oppure variera' di una quantita' non significativa, o almeno non tale da assegnare una priorita' alta a quest'operazione.

2. **Estrazione elenco delle ultime dieci pubblicazioni inserite.**

Potrebbe essere un'operazione richiamata piu' volte, ad esempio per riempire una pagina di benvenuto nel profilo di un utente; e' chiaro che l'importanza di questa operazione e' elevata.

3. **Estrazione elenco delle pubblicazioni aggiornate di recente (ultimi 30 giorni).**

Come per l'operazione no. 2, anche questa si stima poter essere eseguita spesso.

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

Questa operazione puo' avere volumi variabili, in quanto potrebbe dover essere richiamata per scopi differenti. Una stima accurata non e' possibile, per cui considereremo la relativa priorita' come media.

5. **Estrazione elenco delle pubblicazioni inserite da un utente.**

Un flusso di interazione possibile e' dato da un utente che vuole avere i dettagli di un utente attivo, e tra quest'ultimi si vorranno certamente annettere le

pubblicazioni inserite a nome di tale utente. Questo tipo di operazioni e' abbastanza comune, per cui l'importanza e' media.

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

Questa operazione si stima essere richiamata molto spesso, nella pratica ogni qual volta si un utente voglia avere una visione d'insieme delle pubblicazioni presenti nella base di dati. La priorita' e' chiaramente alta.

7. **Estrazione dati completi di una pubblicazione specifica dato il suo ID.**

In comunione con l'operazione no. 6, un flusso di azione comune per un utente e' costituito dall'osservare una lista delle pubblicazioni presenti nel catalogo (magari filtrate attraverso degli specifici parametri di ricerca) per poi selezionare una pubblicazione di interesse per poterne vedere i dettagli.

8. **Ricerca di pubblicazioni per ISBN, titolo, autore, e parole chiave.**

Questa operazione e' forse quella piu' importante tra le interazioni che l'utente effettua col sistema. Priorita' massima.

9. **Inserimento di una recensione relativa a una pubblicazione.**

Soltanto un sottoinsieme di utenti recensira' un data pubblicazione -solo quelli interessati ad essa, nello specifico-; inoltre, un utente non puo' inserire piu' di una recensione per pubblicazione, quindi il numero massimo di inserimenti totali possibili (a meno di modifiche e cancellazioni di recensioni, che comunque sono molto piu' rare degli inserimenti) sara' al piu' (numero di utenti registrati)*(numero di pubblicazioni inserite nella base di dati). Queste considerazioni conseguono nel porre l'operazione ad un livello di priorita' piuttosto basso.

10. **Approvazione di una recensione (da parte del moderatore).**

La scelta dell'importanza di questa operazione e' derivata direttamente dal basso volume di esecuzioni stimato per l'operazione no. 9.

11. **Inserimento di un like relativo a una pubblicazione.**

Puo' essere fatto un ragionamento analogo all'operazione no. 9.

12. **Calcolo numero dei like per una pubblicazione.**

Dovendo mostrare spesso una pagina mostrante una specifica pubblicazione (ossia l'esecuzione dell'operazione no. 7), e' assai probabile che si vorra' mostrare con evidenza l'ammontare di gradimento che questa suscita.

13. Estrazione elenco delle recensioni approvate per una pubblicazione.

Come conseguenza dell'alto volume stimato per l'operazione no. 7 (visualizzazione di una pubblicazione nei suoi dettagli), e' probabile che la lista di recensioni (approvate) associate alla pubblicazione stessa dovra essere di sovente richiamata.

14. Estrazione elenco delle recensioni in attesa di approvazione.

Segue dalle considerazioni fatte per le operazioni no. 9 e 10.

15. Estrazione log delle modifiche effettuare su una pubblicazione.

Questa operazione potrebbe sottostare ad una fase di verifica per sapere chi abbia -ad esempio- modificato una pubblicazione e commesso degli errori, siano essi refusi o causati per dolo. Si puo' dunque supporre una priorita' bassa.

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

Per un ragionamento in linea con quanto detto per l'operazione no. 6 e 8, dobbiamo considerare l'importanza di questa operazione come alta.

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

Segue dalle considerazioni fatte per l'operazione no. 6 di cui la no. 17 e' una variante.

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

Se si e' interessati ad una pubblicazione specifica, e' altamente probabile che si abbia un interesse nei confronti dei contenuti della stessa. Visionare altre pubblicazioni attribuite agli stessi autori e' un'operazione abbastanza comune da considerare la sua priorita' come media.

5.4. Analisi delle ridondanze e osservazioni sulle performance

Alcune informazioni possono essere ridondate per incrementare le prestazioni.

- Per via dell'importanza dell'operazione no. 12, e' doveroso inserire un attributo "num_like" su Pubblicazione che tracci il numero di like ad essa relativi, evitando di coinvolgere la relazione Like troppo spesso durante le interrogazioni.
- Inserendo un attributo "num_inserimenti" su Utente che tracci il numero di inserimenti effettuati, e' possibile velocizzare le interrogazioni basate sull'operazione no. 4.
- Possiamo rendere il sistema piu' efficiente nell'eseguire le operazioni no. 2 e 3 (che rappresentano l'estrazione di un elenco di pubblicazione dati alcuni criteri) muovendo alcuni attributi da Metadati a Pubblicazione; tra tutti scegliamo "titolo" e "categoria" perche' certamente appariranno negli elenchi sopra citati.
- L'operazione no. 3 puo' ulteriormente essere velocizzata aggiungendo un ulteriore attributo "data_ultima_mod" su Pubblicazione.

5.5. Schema ER risultante ottimizzato

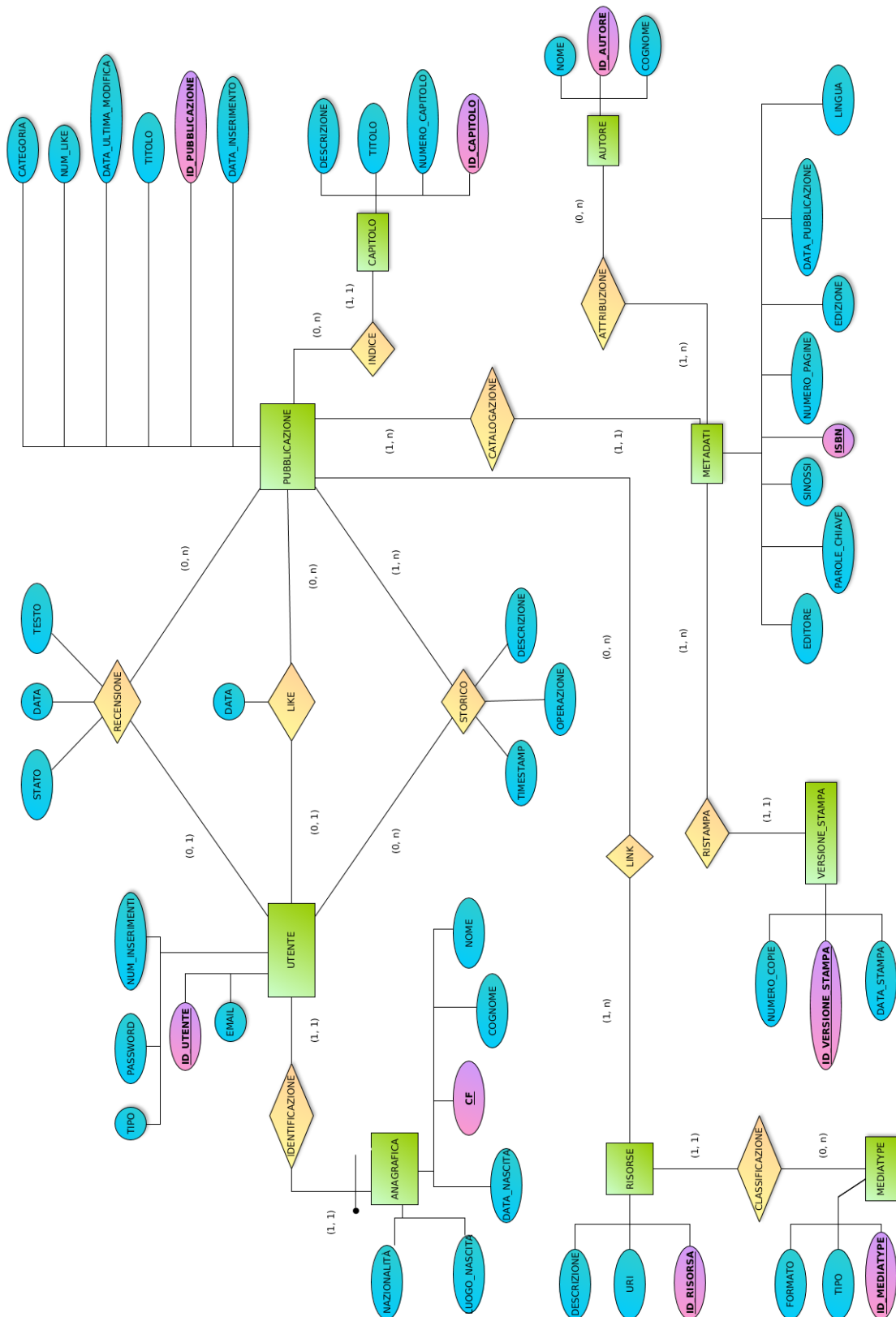


Figura 6: Schema logico completo ottimizzato

6. Traduzione del modello ER in modello relazionale

Il modello relazionale risultante dai passi di design precedenti e' il seguente:

Utente(**id utente**, email, password, tipo, num_inserimenti);

Anagrafica(**id utente**, nome, cognome, cf, data_nascita, luogo_nascita, nazionalita);

Pubblicazione(**id pubblicazione**, titolo, data_inserimento, data_ultima_modifica, categoria, num_like);

Metadati(**isbn**, id_pubblicazione, edizione, editore, data_pubblicazione, parole_chiave, num_pagine, lingua, sinossi);

Autore(**id autore**, nome, cognome);

Capitolo(**id capitolo**, id_pubblicazione, titolo, descrizione, num_capitolo);

Versione_Stampa(**id versione stampa**, id_pubblicazione, num_copie, data_stampa);

Risorsa(**id risorsa**, id_mediatype, uri, descrizione);

Mediatype(**id mediatype**, tipo, formato);

Recensione(**id utente**, **id pubblicazione**, stato, data, testo);

Gradimento(**id utente**, **id pubblicazione**, data); (vedi NOTE sottostanti)

Storico(**id utente**, **id pubblicazione**, data, descrizione, operazione);

Attribuzione(**isbn**, **id autore**);

Link(**id pubblicazione**, **id risorsa**);

NOTA:

- le chiavi primarie sono evidenziate in grassetto e sottolineate; il colore rosso denota chiavi esterne.
- "Gradimento" sostituisce "Like", poiche' quest'ultima costiuisce una keyword in SQL.

7. Note sull'implementazione fisica prodotta

7.1. Viste

- L'operazione no. 18 ricerca l'elenco delle pubblicazione che hanno esattamente gli stessi autori della pubblicazione riferita in input. Per eseguirla abbiamo implementato una vista (view_pubblicazione_autore), che abbiamo sfruttato anche per le operazione no. 6, 7.
- Le viste possono aumentare le performance del sistema; tale motivo e' sotteso all'impiego di una vista ad hoc per le operazioni no. 6 e 7 che, come documentato nella sezione "[5.3. Analisi delle operazioni](#)", possono essere considerate di importanza primaria.
- L'operazione no 16 e' un'operazione ad alta priorita', e anche qui e' stata impiegata una vista.

7.2. Trigger

- Sono stati implementati alcuni trigger:
 - su Storico: per verificare che gli inserimenti nello stesso abbiano valori consentiti sull'attributo "operazione" (vedi "[4. Formalizzazione dei vincoli non espressi tramite il modello ER](#)")
 - su Recensione: una recensione inserita da un utente attivo viene automaticamente approvata
 - su Anagrafica: per verificare che la data di nascita inserita sia precedente a quella corrente
 - su Versione_Stampa: per verificare che la data dell'ultima versione di una pubblicazione inserita sia precedente a quella delle versioni antecedenti la stessa; contemporaneamente, si controlla che la data di stampa non sia precedente la data di pubblicazione
 - su Pubblicazione: ogni volta che viene eseguita una modifica ad una pubblicazione viene automaticamente aggiornata la data di ultima modifica
 - NOTA: per come e' strutturata la nostra base di dati, una modifica ad una pubblicazione puo' corrispondere ad una

modifica di molteplici tabelle (e.g. Metadati, Autori, Indice, etc); la data di ultima modifica, teoricamente, dovrebbe essere automaticamente aggiornata per ogni possibile modifica ai dettagli di una pubblicazione. Al fine di sperimentare differenti soluzioni, e con l'obiettivo di capire meglio il funzionamento dei trigger, ne abbiamo implementato uno specifico che consente l'aggiornamento automatico sopra descritto quando viene effettuato un aggiornamento sulla tabella Pubblicazione.

- ancora su Pubblicazione: per verificare che i valori sull'attributo "categoria" siano quelli appropriati

7.3. Struttura degli script SQL

Alleghiamo infine una breve descrizione del contenuto dei vari script SQL.

- 1-creazione_db.sql → crea le strutture del database
- 2-procedure_crud.sql → come da nome, raccoglie le procedure di creazione, aggiornamento e cancellazione
- 3-funzioni_helper.sql → procedure di supporto
- 4-procedure_composite → collezione di procedure aggiuntive come login, registrazione, ed altro
- 5-operazioni_richieste → procedure riguardanti le operazioni richieste dal progetto
- 6-trigger.sql → raccolta dei trigger sviluppati documentati nella sezione precedente
- 7-procedure_modifica.sql → raccolta delle procedure di modifica su Metadati con aggiornamento su Storico
- 8-viste.sql → raccolta delle viste sviluppate
- 9-inserimento_dataset.sql → procedure di inserimento dei dataset di esempio