



Documentazione del progetto di Basi di Dati a.a. 2022-2023

Paolo Tedesco Giulio Ruopolo Fabrizio Quaranta
N86004408 N86004285 N86004300

Id Gruppo: OOB_D_T1G22

Data: 23/01/2023

Indice

1	Requisiti identificati	3
1.1	Analisi della realtà di interesse	3
2	Progettazione concettuale	6
2.1	Schema Concettuale	6
2.2	Dizionario delle entità e delle associazioni	7
3	Ristrutturazione dello schema concettuale	14
3.1	Analisi delle Ridondanze	14
3.2	Eliminazione delle Generalizzazioni	14
3.3	Eliminazione degli attributi multi valore	16
3.4	Eliminazione degli attributi strutturati	16
3.5	Vincoli	17
3.6	Identificazione delle chiavi primarie	19
3.7	Schema ristrutturato ER	21
3.8	Schema ristrutturato UML	22
4	Mapping relazionale nel modello logico	23
5	Descrizione Funzioni e trigger	25

Capitolo 1

Requisiti identificati

1.1 Analisi della realtà di interesse

Per la creazione e gestione del database relativo alla realtà di interesse proposta, iniziamo con la prima fase:

L'analisi dei requisiti, ossia l'individuazione delle entità e le associazioni presenti tra esse.

«Gli elementi che possono essere inclusi nella biblioteca digitali sono di due tipi: articoli scientifici (o pubblicazioni) e libri (didattici o romanzi). »

Gli elementi cardine riscontrati nella base di dati sono le Entità "pubblicazioni", le quali derivazioni rappresentano le entità "libri" e "articoli scientifici" della biblioteca digitale.

«Per le pubblicazioni, andare a definire in quale rivista (nome, argomento, anno di pubblicazione, responsabile della rivista) o in quale conferenza (luogo della conferenza, data di inizio e data fine conferenza, struttura organizzatrice e responsabile) è stato pubblicato. »

Per gli "articoli scientifici" andrà ad essere indicata tramite le entità "piattaforma" la modalità in cui essi sono stati presentati, da cui derivano le due uniche modalità, rappresentate con le entità "conferenza" o "rivista".

«libri (didattici o romanzi) »

A loro volta i libri possono essere di due tipologie: didattico o romanzo che verranno indicate come entità specializzanti di libro.

Per i "libri didattici" viene indicato il loro argomento tramite l'attributo "materia" oltre a gli altri attributi che saranno presenti anche per un romanzo.

«Un romanzo può avere anche uno o più seguiti. »

La gestione dell'appartenenza di un libro ad una serie è effettuata tramite l'entità "serie", inoltre per indicare la successione dei libri è presente un'associazione ricorsiva "seguito" sull'entità libro.

«Non appena una serie sarà disponibile per l'acquisto da almeno una libreria, il sistema notificherà la disponibilità all'utente. »

Il concetto di utente e la preferenza di una serie da parte di esso viene indicata tramite l'introduzione dell'entità "utente" la quale attraverso l'associazione "preferiti" con l'entità serie, permetterà la gestione della notifica.

«Per i libri è importante definire la data di uscita del libro e la sala/libreria in cui è fatta una eventuale presentazione »

L'entità "libro" sarà supportata dall'entità "presentazione" per indicare l'uscita di esso in una "sala" o in una "libreria" le quali saranno entità derivate dall'Entità "presentazione".

«Un libro può anche far parte di una collana, la quale può raggruppare tutte le pubblicazioni che condividono una determinata caratteristica »

Il raggruppamento di libri simili ma non appartenenti alla stessa serie, è tracciato dall'entità "collana"

«per ogni libro, è necessario specificare dove può essere acquistato (libreria, online). »

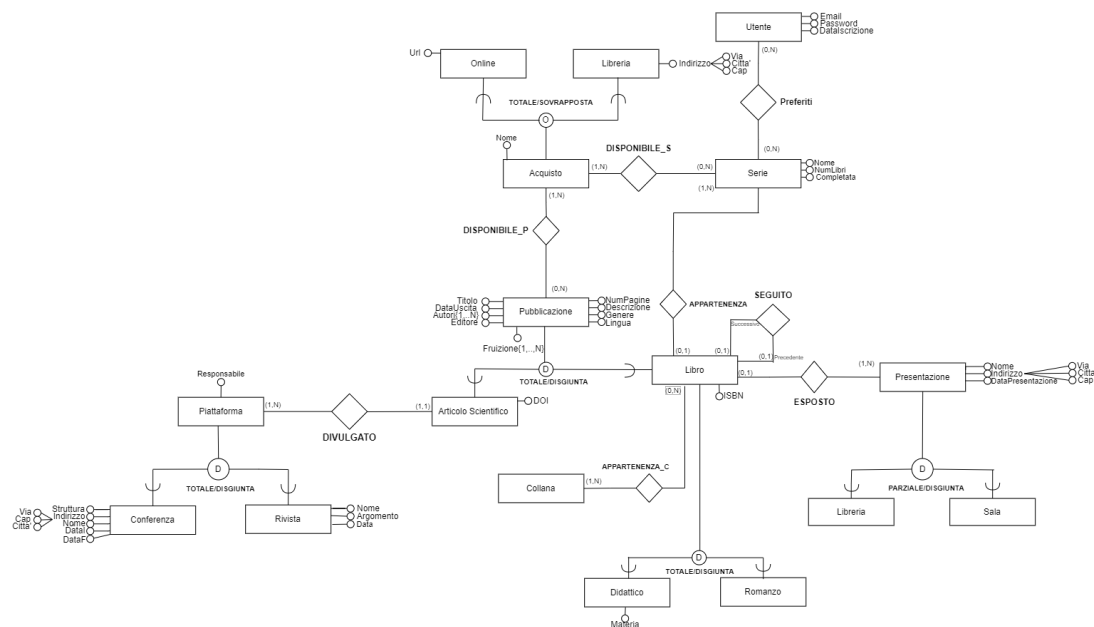
In fine per ogni libro presente nella base di dati per mostrare le modalità di acquisto disponibili viene introdotta l'entità "acquisto", la quale specifica se acquistabile da un sito online indicato dall'entità "online" e/o da una libreria fisica rappresentato dall'entità "libreria".

Tale processo è stato applicato anche per gli "articoli scientifici", anche se non espressamente indicato, dato che anche un "articolo scientifico" può essere acquistato negli stessi modi di un "libro".

Capitolo 2

Progettazione concettuale

2.1 Schema Concettuale



2.2 Dizionario delle entità e delle associazioni

Entità	Descrizione	Attributi: descrizione
Pubblicazione	Rappresentazione generale delle pubblicazioni, entità padre di "Articolo Scientifico" e "Libro".	Titolo: Rappresenta il titolo della pubblicazione. DataUscita: Indica l'anno di uscita della pubblicazione. Autori[1..n]: Attributo multiplo che rappresenta gli autori. Editore: Rappresenta il nome dell'editoria. NumPagine: Indica il numero delle pagine della pubblicazione. Descrizione: Definisce la descrizione posta dall'autore. Genere: Rappresenta il genere della pubblicazione. Lingua: Indica in che lingua è scritta la pubblicazione. Fruizione[Cartaceo, Digitale, AudioLibro]: Attributo multiplo che definisce la modalità con la quale può essere fruito.

Libro	Rappresentazione specifica di una pubblicazione, che traccia i libri, entità figlia dell'entità Pubblicazione e entità padre delle entità "Romanzo" e "Didattico".	ISBN: Identificativo del libro.
Articolo Scientifico	Rappresentazione specifica di una pubblicazione che traccia gli Articoli Scientifici, entità figlia dell'entità "Pubblicazione".	DOI: Identificatore dell'articolo scientifico.
Didattici	Rappresenta la specializzazione dell'entità "Libro" composta dai libri didattici che hanno come attributo distintivo la materia.	Materia: Indica la materia del libro didattico.
Romanzo	Rappresentazione della specializzazione dell'entità "Libro" che contiene i Romanzi	
Collana	Rappresenta l'insieme di libri con una caratteristica comune	

Piattaforma	Rappresentazione generale delle piattaforme dove vengono pubblicati gli articoli scientifici, entità padre di "Conferenza" e "Rivista".	Responsabile: Rappresenta il coordinatore della piattaforma.
Conferenza	Specializzazione di "Piattaforma", entità figlio che traccia le conferenze	Struttura: Indica il luogo in cui si è svolta o si svolgerà la conferenza. Indirizzo[Via,Città,CAP]: Definisce la locazione della struttura Nome: Rappresenta il nome della struttura DataI: Indica la data in cui inizia l'evento DataF: Indica la data in cui finisce l'evento
Rivista	Specializzazione di "Piattaforma", entità figlio che traccia le riviste	Nome: Delinea il nome della rivista Argomento: Definisce l'argomento della rivista Data: Indica la data in cui è stata pubblicata la rivista

Presentazione	Rappresentazione generale delle Presentazioni dove vengono pubblicati i libri, entità padre di "Libreria" e "sala".	Nome: Raffigura il nome della presentazione. Indirizzo[Via,Città,CAP]: Indica il luogo dove si terra la presentazione. DataPresentazione : Definisce il luogo della presentazione.
Libreria	Specializzazione di "Presentazione", entità figlio che traccia le librerie.	
Sala	Specializzazione di "Presentazione", entità figlio che traccia le sale.	
Serie	Raccolta di libri facente parte della stessa sequenza	Nome: Indica il nome della serie. NumLibri: Rappresenta il numero dei libri della serie. Completata: Identifica se la serie è completa o meno.
Acquisto	Entità generale che indica dove è possibile acquistare una pubblicazione, entità padre dell'entità "Online" e "libreria".	Nome: Definisce il nome della libreria

Online	Specializzazione dell'entità "Acquisto" che rappresenta i siti web dove potere acquistare i libri	URL: Indica l'indirizzo web del sito.
Libreria	Specializzazione dell'entità "Acquisto" che rappresenta le librerie fisiche dove potere acquistare i libri.	Indirizzo [Via,Città,CAP]: Attributo composto, indica la posizione della libreria fisica.
Utente	Rappresenta gli utenti che accederanno al database.	Email: Identificativo dell'utente. Password: Indica la password del profilo utente. DataIscrizione: Rappresenta la data d'iscrizione dell'utente.

Associazione	Descrizione	Regole di lettura
Divulgato	Associazione 1 a N tra "Piattaforma" e "Articolo scientifico", indica la piattaforma dove l'articolo scientifico è stato pubblicato.	Una "Piattaforma" deve pubblicare uno o più "Articoli scientifici", un "Articolo scientifico" deve essere pubblicato da una e una sola "Piattaforma".
Esposto	Associazione 1 a N tra "Presentazione" e "Libro", indica il luogo in cui un libro è stato presentato.	Una "Presentazione" deve esporre uno o più "Libri", Un "Libro" può essere esposto in nessuna o una "Presentazione".
Seguito	Associazione ricorsiva 1 a 1 tra "Libro" e "Libro", esprime il concetto di successione di libri in una serie.	Un "Libro" può essere il successivo di nessuno o un "Libro" e un "Libro" può essere precedente di nessuno o un "Libro".
Appartenenza	Associazione 1 a N tra "Serie" e "Libro", rappresenta l'appartenenza di un libro in una serie.	Una "Serie" deve avere uno o più "Libri", un "Libro" può appartenere a nessuna o una "Serie".
Preferiti	Associazione N a N tra "Utente" e "Serie", utilizzata per rappresentare le preferenze di un utente.	Un "Utente" può avere preferita nessuna o più "Serie", una "Serie" può essere preferita da nessuno o più "Utenti".
Disponibile_S	Associazione N a N tra "Serie" e "Acquisto", indica la possibilità di una serie di essere acquistata e dove.	Una "Serie" può essere disponibile in nessuno o più "Acquisti", un "Acquisto" può avere disponibile nessuna o più "Serie".

Disponibile_P	Associazione N a N tra "Pubblicazione" e "Acquisto".	Una "Pubblicazione" può essere disponibile in nessuno o più "Acquisti", un "Acquisto" può avere disponibile nessuna o più "Pubblicazioni".
Appartenenza_C	Associazione N a N tra "Collana" e "Libro".	Una "Collana" deve possedere uno o più "Libri", un "Libro" può appartenere a nessuna o più "Collane".

Capitolo 3

Ristrutturazione dello schema concettuale

3.1 Analisi delle Ridondanze

In questa fase abbiamo rilevato che l'entità "Collana", potesse essere gestita da una "View" in quanto la "Collana" in se può essere rappresentata da una interrogazione sulla base di dati che rappresenta una collezione di "Libri" con caratteristiche comuni. Inoltre dato l'utilizzo frequente dell'informazione della quantità di libri in una serie, si è optato per l'introduzione dell'attributo "NumLibri" sull'entità "Serie", anche essendo esso ridondante, in quanto derivabile da una interrogazione.

3.2 Eliminazione delle Generalizzazioni

Le generalizzazioni riscontrate nello schema concettuale sono: "Pubblicazione", "Libro", "Presentazione", "Piattaforma" e "Acquisto". Ognuna di esse dovrà essere opportunamente ristrutturata.

- La generalizzazione "Pubblicazione" verrà risolta accorpendo il padre nelle figlie, in quanto è necessaria una marcata distinzione per

le entità "Libro" e "Articolo scientifico" che saranno caratterizzate dai loro attributi propri e acquisiranno gli attributi di "Pubblicazione". Per l'associazione "Disponibile_P", la quale definita tra "Pubblicazione" e "Acquisto", verrà rimpiazzata con "Disponibile_L" tra "Libro" e "Acquisto" e "Disponibile_A" tra "Articolo scientifico" e "Acquisto".

- La generalizzazione "Libro" verrà risolta accorpendo le entità figlie nel padre, in quanto non ci interessa trattare separatamente le entità figlie "Didattico" e "Romanzo". L'entità "Libro" acquisirà gli attributi delle sue entità figlie con l'aggiunta dell'attributo tipo per differenziare le due possibili istanze di "Libro".
- La generalizzazione "Presentazione" verrà risolta accorpendo le figlie nel padre, in quanto non sarà necessaria una distinzione per le entità figlie "Libreria" e "Sala", dunque i loro attributi saranno acquisiti da "Presentazione" con l'aggiunta dell'attributo tipo per differenziare le due possibili istanze di "Presentazione". L'associazione tra "Presentazione" e "Libro" rimarrà invariata.
- La generalizzazione "Piattaforma" verrà risolta accorpendo le figlie nel padre, in quanto non sarà necessaria una distinzione per le entità "Conferenza" e "Rivista", dunque i loro attributi saranno acquisiti da "Piattaforma" con l'aggiunta dell'attributo tipo per differenziare le due possibili istanze di "Piattaforma". L'associazione tra "Piattaforma" e "Articolo scientifico" rimarrà invariata.
- La generalizzazione "Acquisto" verrà risolta accorpendo le figlie nel padre, in quanto non sarà necessaria una distinzione per le entità "Online" e "Libreria", dunque i loro attributi saranno acquisiti da "Acquisto" con l'aggiunta dell'attributo tipo per differenziare le due possibili istanze di "Acquisto". L'associazione tra "Acquisto" e "Libro" rimarrà invariata.

3.3 Eliminazione degli attributi multi valore

Gli attributi multi valore presenti nello schema concettuale sono: "Autore" e "Fruizione" entrambi presenti nell'entità "Libro".

- L'attributo multiplo "Autore" può rappresentare nel diagramma ER uno o più autori, dunque abbiamo deciso di trasformarlo in un attributo semplice di tipo stringa in cui tutti gli autori vengono rappresentati separati da una virgola. Questa scelta è stata adoperata in quanto non ci interesserà tenere traccia ed utilizzare gli autori singolarmente, inoltre i libri spesso vengono scritti da un singolo autore, ma nell'eventualità di altri autori, essi saranno rappresentati dopo l'autore e saranno intesi come co-autori.
- L'attributo multiplo "Fruizione" assume le modalità di consumazione di un libro (es cartaceo, digitale e audiolibro). Esso verrà trasformato in una stringa contenente le modalità disponibili per il libro separate da una virgola, in modo da non vincolarci sulle sue modalità ma avere la possibilità di inserire in futuro una nuova modalità non presente in questo momento.

3.4 Eliminazione degli attributi strutturati

L'attributo strutturato presente nello schema concettuale è l'attributo "Indirizzo". Esso è presente nell'entità "Conferenza", nell'entità "Acquisto" e nell'entità "Presentazione".

- L'attributo "Indirizzo" è composto dagli attributi "Via", "Città" e "Cap". Sarà trasformato in un'unica stringa in quanto nella gestione della nostra realtà d'interesse non è importante tenere traccia separatamente di ogni attributo ma unicamente del singolo indirizzo.

3.5 Vincoli

I vincoli individuati della nostra realtà di interesse sono :

- Vincoli di integrità impliciti
 - Vincoli di chiave primaria: Spiegati in *"Identificazione delle chiavi primarie"*
 - Vincoli di chiave esterna: Spiegati in *"Mapping"*
- Vincoli di integrità espliciti
 - **Controllo_Date_Conferenza (Conferenza):**
(DataI <= DataF)
Vincolo che impedisce ad una conferenza di avere una data di inizio precedente alla sua data di fine.
 - **Numero_Pagine_ArticoloS (Articolo Scientifico) :**
(NumPagine > 0 and
(Fruizione LIKE '%Cartaceo%' or Fruizione LIKE '%Digitale%'))or
(NumPagine = 0 and (Fruizione LIKE 'AudioLibro')))
Vincolo che impedisce ad un Articolo Scientifico di avere un numero di pagine negativo e una fruizione "Cartacea" e/o "Digitale" oppure un numero di pagine diverso da zero e la sua unica fruizione non è "Audiolibro".
 - **Fruizione (Acquisto):**
((url IS NULL) != (indirizzo IS NULL)) and
((Tipo = 'Sito web') and (url is not null) or
(Tipo = 'Libreria') and (Indirizzo is not null))
Vincolo che deriva dalla generalizzazione, il valore del tipo deve essere "Libreria" o "Online" e di conseguenza avere un "indirizzo" per la "Libreria" e un "url" per "Online" ed il contrario non valorizzabile.

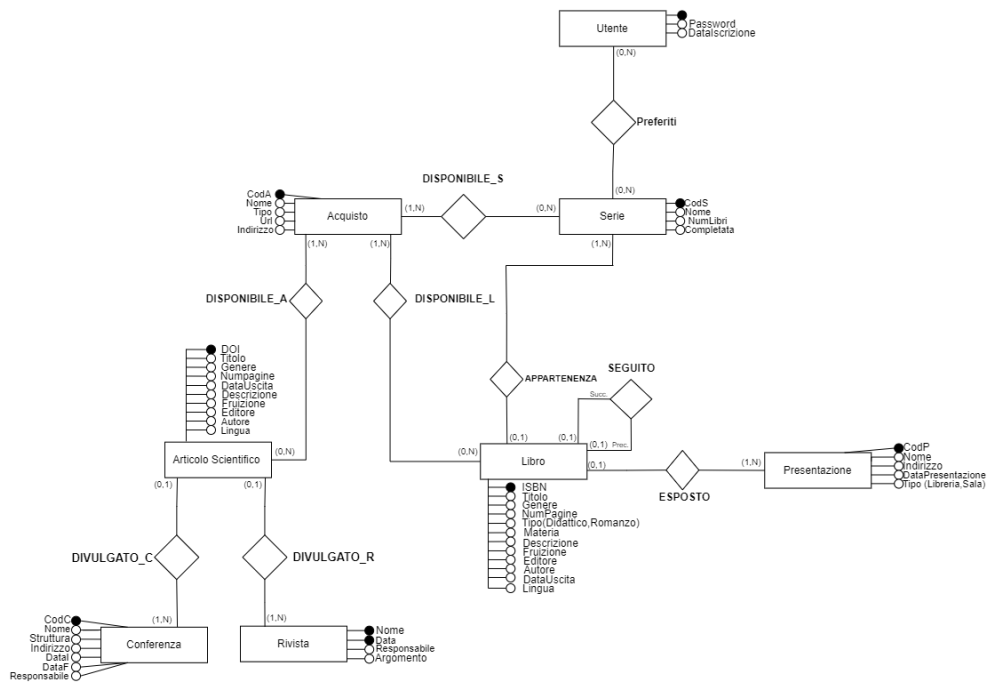
- **Controllo_Completata (Serie):**
 ((Complemata = TRUE and NumLibri > 1) or Complemata = FALSE),
Vincolo che impone ad una serie completa di avere almeno due libri altrimenti dovrà essere non completa.
- **Controllo_NumLibri (Serie);**
 NumLibri >= 0
Vincolo che impone ad una serie di avere un numero non negativo di libri
- **Tipo_Libro (Libro):**
 ((tipo = 'Didattico') and (Materia is not null)) or
 ((tipo = 'Romanzo') and (materia is null)),
Vincolo che deriva dalla generalizzazione, il valore del tipo deve essere "Didattico" o "Romanzo", se il libro è "didattico" deve possedere una materie, altrimenti un libro "romanzo" non deve possedere una materia.
- **Numero_Pagine_Libro (Libro):**
 (NumPagine > 0 and
 (Fruizione LIKE '%Cartaceo%' or Fruizione LIKE '%Digitale%')) or
 (NumPagine = 0 and (Fruizione LIKE 'AudioLibro'))
Vincolo che impedisce ad un Libro di avere un numero di pagine negativo e una fruizione "Cartacea" e/o "Digitale" oppure un numero di pagine diverso da zero e la sua unica fruizione non è "Audiolibro".

3.6 Identificazione delle chiavi primarie

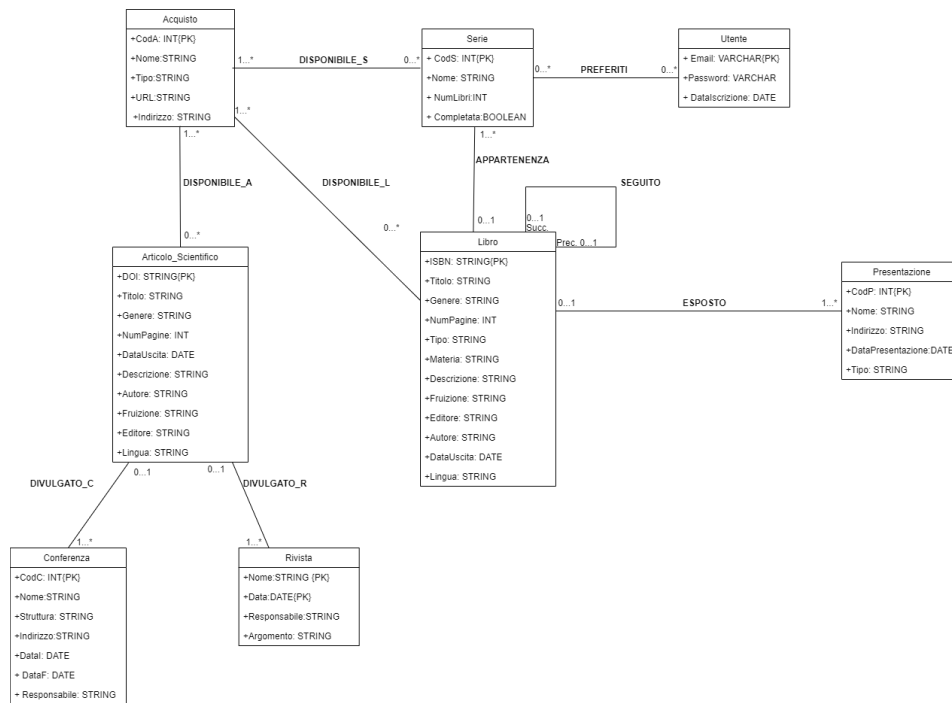
Entità	Chiave Primaria	Descrizione pk
Libro	ISBN	Codice di 13 cifre che identifica univocamente un "Libro".
Articolo Scientifico	DOI	Codice di caratteri che identifica univocamente un "Articolo scientifico".
Conferenza	CodC	Codice di cifre che identifica univocamente una "Conferenza", aggiunto data la mancanza in "Conferenza" di un attributo univoco.
Rivista	Nome , Data	Nome e data di uscita della "Rivista" che insieme identificano univocamente una "Rivista".

Presentazione	CodP	Codice di cifre che identifica univocamente una "Presentazione", aggiunto data la mancanza in "Presentazione" di un attributo univoco.
Serie	CodS	Codice di cifre che identifica univocamente una "Serie", aggiunto data la mancanza in "Serie" di un attributo univoco.
Acquisto	CodA	Codice di cifre che identifica univocamente un' "Acquisto", aggiunto data la mancanza in "Acquisto" di un attributo univoco.
Utente	Email	Email dell'"Utente" unica, lo identifica univocamente.

3.7 Schema ristrutturato ER



3.8 Schema ristrutturato UML



Capitolo 4

Mapping relazionale nel modello logico

Dopo la fase di ristrutturazione possiamo passare alla fase di mapping delle entità e delle associazioni in relazioni.

Le entità verranno trasformate in relazioni aventi stesso nome, attributi e chiave primaria. Le associazioni invece verranno gestite in base alla loro molteplicità portando alla sostituzione di esse con nuove relazioni o aggiungendo alle relazioni eventuali chiavi esterne.

- **CONFERENZA** (CodC, Nome, Struttura, Indirizzo, DataI, DataF, Responsabile)
- **RIVISTA** (Nome, Data, Responsabile, Argomento);
- **ARTICOLO_SCIENTIFICO** (Doi, Titolo, Genere, NumPagine, DataUscita, Descrizione, Fruizione, Editore, Autore, Lingua, Conferenza \uparrow , NomeR \uparrow , DataR \uparrow);
 - L'Attributo "Conferenza" è chiave esterna sull'attributo "Conferenza.CodC", data l'associazione "DIVULGATO_C" tra "Articolo_Scientifico" e "Conferenza" di molteplicità N a 1.
 - L'Attributo "NomeR" è chiave esterna sull'attributo "Rivista.Nome", data l'associazione "DIVULGATO_R" tra "Articolo_Scientifico" e "Rivista" di molteplicità N a 1.
 - L'Attributo "DataR" è chiave esterna sull'attributo "Rivista.Data", data l'associazione "DIVULGATO_R" tra "Articolo_Scientifico" e "Rivista" di molteplicità N a 1.
- **ACQUISTO** (CodA, Nome, Tipo, Url, Indirizzo);

- **PRESENTAZIONE** (CodP, Nome, Indirizzo, DataPresentazione, Tipo);
- **SERIE** (CodS, Nome, NumLibri, Completata);
- **LIBRO** (ISBN, Titolo, Genere, NumPagine, Tipo, Materia, Descrizione, Fruizione, Editore, Autore, DataUscita, Lingua, Successivo \uparrow , Serie \uparrow , Presentazione \uparrow);
 - L'Attributo "Successivo" è chiave esterna sull'attributo "Libro.ISBN", data l'associazione ricorsiva "SEGUITO" con "Libro" di molteplicità 1 a 1.
 - L'Attributo "Serie" è chiave esterna sull'attributo "Serie.CodS", data l'associazione "APPARTENENZA" tra "Libro" e "Serie" di molteplicità N a 1.
 - L'Attributo "Presentazione" è chiave esterna sull'attributo "Presentazione.CodP", data l'associazione "ESPOSTO" tra "Libro" e "Presentazione" di molteplicità N a 1.
- **UTENTE** (Email, Password, DataIscrizione);
- **DISPONIBILE_L** (CodA, ISBN);

Relazione derivata dalla associazione "DISPONIBILE_L" di molteplicità N a N, tra "Libro" e "Acquisto".

 - L'Attributo "CodA" è chiave esterna sull'attributo "Acquisto.CodA"
 - L'Attributo "ISBN" è chiave esterna sull'attributo "Libro.ISBN"
- **DISPONIBILE_S** (CodA \uparrow , CodS \uparrow);

Relazione derivata dalla associazione "DISPONIBILE_S" di molteplicità N a N, tra "Serie" e "Acquisto".

 - L'Attributo "CodA" è chiave esterna sull'attributo "Acquisto.CodA",
 - L'Attributo "CodS" è chiave esterna sull'attributo "Serie.CodS"
- **DISPONIBILE_A** (CodA \uparrow , Doi \uparrow);

Relazione derivata dalla associazione "DISPONIBILE_A" di molteplicità N a N, tra "Articolo_Scientifico" e "Acquisto".

 - L'Attributo "CodA" è chiave esterna sull'attributo "Acquisto.CodA",
 - L'Attributo "Doi" è chiave esterna sull'attributo "Articolo_Scientifico.DoI"
- **PREFERITI** (Email \uparrow , CodS \uparrow);

Relazione derivata dalla associazione "PREFERITI" di molteplicità N a N, tra "Utente" e "Serie".

 - L'Attributo "Email" è chiave esterna sull'attributo "Utente.Email"
 - L'Attributo "CodS" è chiave esterna sull'attributo "Serie.CodS".

Capitolo 5

Descrizione Funzioni e trigger

- create function ADD_Disponibile_S_funz() returns trigger as \$\$
declare
 f_codSerie libro.serie%type;
 f_numLibriS serie.numLibri%type;
 f_codSerieApp libro.serie%type;
 f_numLibriAcq serie.numLibri%type;

begin
 select serie into f_codSerie
 from libro
 where isbn = new.isbn;

 select numLibri into f_numLibriS
 from serie
 where codS = f_codSerie;

 select codS into f_codSerieApp
 from disponibile_S
 where codS = f_codSerie and codA = new.codA;

 if f_codSerieApp is null then
 select count(*) into f_numLibriAcq
 from disponibile_L
 where codA = new.codA and isbn in
 (select isbn from libro where serie = f_codSerie);

```

        if f_numLibriAcq = f_numLibriS then
            insert into disponibile_S(CodA,CodS) values
            (new.codA, f_codSerie);
        end if;
    end if;
return new;
end;
$$
language plpgsql;

```

Descrizione :

Funzione che controlla la disponibilità della serie di un libro reso acquistabile in una piattaforma.

Nel caso in cui non sia segnata come acquistabile la serie nella piattaforma indicata, ma il numero dei libri di quella serie acquistabili sulla piattaforma coincide al numero di libri della serie, verrà aggiunta la disponibilità della serie ad essere acquistata nella specifica piattaforma.

- Create Trigger ADD_Disponibile_S after insert on Disponibile_L
for each row
execute function ADD_Disponibile_S_funz();

Descrizione :

Trigger attivato all'atto dell'inserimento di un libro disponibile in una piattaforma, esso manda in esecuzione la funzione "ADD_Disponibile_S_funz()";

- create function Controllo_Succ_funz() returns trigger as
\$\$
declare
f_codSerie libro.serie%type;
f_libro_noSucc libro.isbn%type;
f_numLibri serie.NumLibri%type;
begin
 f_codSerie := new.serie;
 select isbn into f_libro_noSucc
 from libro where successivo is null and serie = f_codSerie and isbn <>
 new.isbn;

 if new.Serie is not null then
 delete from disponibile_S
 where codS = new.Serie;
 select numLibri into f_numLibri

```

from serie
where codS = new.Serie;

f_numLibri := f_numLibri +1;
update serie set numLibri = f_numLibri
where codS = new.Serie;
if f_libro_noSucc is not null then
    update libro set successivo = new.isbn
    where isbn = f_libro_noSucc;
end if;
end if;
return new;
end;
$$
language plpgsql;

```

Descrizione :

Funzione che all'aggiunta di un libro, appartenente ad una Serie, cerca nella serie un libro senza il successore ad eccezione del libro stesso, il quale diventerà il predecessore del libro inserito. Di conseguenza saranno eliminate le disponibilità della serie ad essere acquistate in ogni piattaforma.

- create trigger Controllo_Succ after insert on Libro
for each row
execute function Controllo_Succ_funz();

Descrizione :

Trigger attivato all'atto dell'inserimento di un libro e manda in esecuzione la funzione Controllo_Succ_funz() per effettuare il possibile collegamento al precedente;

- create function Remove_Disponibile_S_funz() returns trigger as
\$\$
declare
 f_codSerie libro.serie%type;
begin
 select serie into f_codSerie
 from libro
 where isbn = old.isbn;
 if f_codSerie is not null then
 Delete from Disponibile_S
 Where codA = old.codA AND codS = f_codSerie;

```

        end if;
    return new;
end;
$$
language plpgsql;

```

Descrizione : Funzione che elimina la disponibilità di una serie da una piattaforma nell'eventualità che il libro appartenete alla serie non sia più disponibile sulla stessa piattaforma.

- Create Trigger InDisponibile_S after delete on Disponibile_L
for each row
execute function Remove_Disponibile_S_funz();

Descrizione : Trigger che al momento di una indisponibilità di un libro, manda in esecuzione Remove_Disponibile_S_funz() per vedere se il libro appartiene a una serie.

- create procedure ADD_Disponibile_L
(in f_ISBN_in libro.ISBN%type, in f_acqStrng_in varchar(255)) as
\$\$

```

declare    /*f_acqStrng_in e una stringa che contine più codA*/
    f_acqStrngAppo VARCHAR(255);
    f_pos INTEGER;
    f_codACntrl disponibile_l.codA%type;
    f_codA disponibile_l.codA%type;
    f_acqStrngAppoINT INTEGER;

```

```
begin
```

```

    f_acqStrngAppo := f_acqStrng_in;
    while f_acqStrngAppo is not null loop

```

```

        f_pos := position('+' in f_acqStrngAppo);
        if f_pos <> 0 then

```

```

            f_codA := cast(substring(f_acqStrngAppo, 1, f_pos - 1) as INTEGER);
            f_acqStrngAppo := replace(f_acqStrngAppo, concat(f_codA,'+'),'');

```

```

                select codA into f_codACntrl
                from disponibile_L
                where ISBN = f_ISBN_in and codA = f_codA;

```

```

        if f_codACntrl is null then
            insert into disponibile_L(codA,ISBN) values
                (f_codA,f_ISBN_in);
        end if;
    else

        f_acqStrngAppoINT := cast(f_acqStrngAppo as INTEGER);

        select codA into f_codACntrl
        from disponibile_L
        where ISBN = f_ISBN_in and codA = f_acqStrngAppoINT;

        if f_codACntrl is null then
            insert into disponibile_L(codA,ISBN) values
                (f_acqStrngAppoINT,f_ISBN_in);
        end if;
        f_acqStrngAppo := NULL;
    end if;
end loop;
end;
$$
language plpgsql;

```

Descrizione : Funzione che dato in ingresso un libro e una stringa composta da codici di Acquisti, separati da un '+', estrapola dalla stringa gli acquisti e crea la disponibilità del libro negli Acquisti indicati.

- create procedure ADD_Disponibile_A
(in f_DOI_in Articolo_Scientifico.DOI%type, in f_acqStrng_in varchar(255)) as
\$\$
declare
 f_acqStrngAppo VARCHAR(255);
 f_pos INTEGER;
 f_codACntrl disponibile_l.codA%type;
 f_codA disponibile_l.codA%type;
 f_acqStrngAppoINT INTEGER;

begin
 f_acqStrngAppo := f_acqStrng_in;
 while f_acqStrngAppo is not null loop

```

f_pos := position('+ ' in f_acqStrngAppo);
if f_pos <> 0 then

    f_codA := cast(substring(f_acqStrngAppo, 1, f_pos - 1) as INTEGER);
    f_acqStrngAppo := replace(f_acqStrngAppo, concat(f_codA, '+'), '');

    select codA into f_codACntrl
    from disponibile_A
    where DOI = f_DOI_in and codA = f_codA;

    if f_codACntrl is null then
        insert into disponibile_A(codA,DOI) values
        (f_codA,f_DOI_in);
    end if;
else

    f_acqStrngAppoINT := cast(f_acqStrngAppo as INTEGER);

    select codA into f_codACntrl
    from disponibile_A
    where DOI = f_DOI_in and codA = f_acqStrngAppoINT;

    if f_codACntrl is null then
        insert into disponibile_A(codA,DOI) values
        (f_acqStrngAppoINT,f_DOI_in);
    end if;
    f_acqStrngAppo := NULL;
end if;
end loop;
end;
$$
language plpgsql;

```

Descrizione : Funzione che dato in ingresso un Articolo scientifico e una stringa composta da codici di Acquisti, separati da un '+', estrapola dalla stringa gli acquisti e crea la disponibilità di un Articolo scientifico per gli Acquisti indicati.

- create procedure DEL_Disponibile_L
(in f_ISBN_in libro.ISBN%type, in f_acqStrng_in varchar(255)) as
\$\$

```

declare
    f_acqStrngAppo VARCHAR(255);
    f_pos INTEGER;
    f_codACntrl disponibile_l.codA%type;
    f_codA disponibile_l.codA%type;
    f_acqStrngAppoINT INTEGER;

begin
    f_acqStrngAppo := f_acqStrng_in;
    while f_acqStrngAppo is not null loop

        f_pos := position('+' in f_acqStrngAppo);
        if f_pos <> 0 then

            f_codA := cast(substring(f_acqStrngAppo, 1, f_pos - 1) as INTEGER);
            f_acqStrngAppo := replace(f_acqStrngAppo, concat(f_codA,'+'),'');

            select codA into f_codACntrl
            from disponibile_L
            where ISBN = f_ISBN_in and codA = f_codA;

            if f_codACntrl is not null then
                Delete from disponibile_L
                Where (CodA=f_codA AND ISBN=f_ISBN_in);
            end if;
        else

            f_acqStrngAppoINT := cast(f_acqStrngAppo as INTEGER);

            select codA into f_codACntrl
            from disponibile_L
            where ISBN = f_ISBN_in and codA = f_acqStrngAppoINT;

            if f_codACntrl is not null then
                Delete from disponibile_L
                Where (CodA=f_acqStrngAppoINT AND ISBN=f_ISBN_in);
            end if;
            f_acqStrngAppo := NULL;
        end if;
    end loop;

```

```
end;
$$
language plpgsql;
```

Descrizione : Funzione che data in ingresso un libro e una stringa composta da codici di Acquisti, separati da un '+', estrapola dalla stringa gli acquisti e elimina la disponibilità del libro per gli Acquisti specificati.

- create procedure DEL_Disponibile_A
(in f_DOI_in articolo_scientifico.DOI%type, in f_acqStrng_in varchar(255)) as
\$\$
declare
 f_acqStrngAppo VARCHAR(255);
 f_pos INTEGER;
 f_codACntrl disponibile_A.codA%type;
 f_codA disponibile_A.codA%type;
 f_acqStrngAppoINT INTEGER;

begin
 f_acqStrngAppo := f_acqStrng_in;
 while f_acqStrngAppo is not null loop
 f_pos := position('+ ' in f_acqStrngAppo);
 if f_pos <> 0 then

 f_codA := cast(substring(f_acqStrngAppo, 1, f_pos - 1) as INTEGER);
 f_acqStrngAppo := replace(f_acqStrngAppo, concat(f_codA, '+ '), '');

 select codA into f_codACntrl
 from disponibile_A
 where DOI = f_DOI_in and codA = f_codA;

 if f_codACntrl is not null then
 Delete from disponibile_A
 Where (CodA=f_codA AND DOI=f_DOI_in);
 end if;
 else

 f_acqStrngAppoINT := cast(f_acqStrngAppo as INTEGER);

 select codA into f_codACntrl
 from disponibile_A


```

        where DOI = f_DOI_in and codA = f_acqStrngAppoINT;

        if f_codACntrl is not null then
            Delete from disponibile_A
            Where (CodA=f_acqStrngAppoINT AND DOI=f_DOI_in);
        end if;
        f_acqStrngAppo := NULL;
    end if;
end loop;
end;
$$
language plpgsql;

```

Descrizione : Funzione che data in ingresso un Articolo scientifico e una stringa composta da codici di Acquisti, separati da un '+', estrapola dalla stringa gli acquisti e elimina la disponibilità di un Articolo scientifico dagli Acquisti indicati.

- create function ADD_view_procedure() returns trigger as

```

$$
declare
    querySqlGenere varchar(1000) := '_collana as select * from libro
    where genere =';
    querySqlAutore varchar(1000) := '_collana as select * from libro
    where autore =';
    querySqlEditore varchar(1000) := '_collana as select * from libro
    where editore =';
begin
    execute 'create or replace view ' ||
    replace(replace(new.genere,' ','_'),'.','_')
    || querySqlGenere || '''' || new.genere || '''';

    execute 'create or replace view '
    || replace(replace(new.autore,' ','_'),'.','_')
    || querySqlAutore || '''' || new.autore || '''';

    execute 'create or replace view ' ||
    replace(replace(new.editore,' ','_'),'.','_')
    || querySqlEditore || '''' || new.editore || '''';

```

```

return new;
end;
$$
language plpgsql;

```

Descrizione : Funzione che dato un libro genera le collane riguardanti i suoi attributi: specificamente gli attributi di genere, autore e editore.

- create trigger ADD_view after insert on libro
for each row
execute function ADD_view_procedure();

Descrizione : Trigger che al momento di una inserimento di un libro, manda in esecuzione ADD_view_procedure(), che genera le collane relative agli attributi di quel libro.

- create function show_Preferiti(in f_email_in utente.email%type)
returns table (NomeSerie varchar(1000),
NomeLuogoAcq varchar(1000))
as
\$\$
begin
return query select se.nome,ac.Nome
from preferiti as pr
join serie as se on pr.codS = se.codS
join disponibile_S as ds on se.codS = ds.codS
join acquisto as ac on ds.codA = ac.codA
where pr.email = f_email_in;
end;
\$\$
language plpgsql;

Descrizione : Funzione che data un' email di un utente mostra i suoi libri preferiti che sono disponibili indicandone il luogo dove è possibile acquistarli.