



Progetto Object Orientation e Basi di Dati

Lorenzo Tecchia N86004446

Mirko Marciano N86004019

10/12/2022

Progettazione e sviluppo di un applicativo per la gestione di una
Biblioteca Digitale



Figura 1: Logo della Libreria

Indice

1	Descrizione e Analisi del Progetto	6
1.1	Descrizione e risoluzione sintetica	6
2	Progettazione concettuale	7
2.1	Class Diagram	7
2.2	Analisi della ristrutturazione del Class Diagram	8
2.2.1	Analisi delle ridondanze	8
2.2.2	Analisi degli identificativi	8
2.2.3	Rimozione degli attributi multipli	8
2.2.4	Rimozione degli attributi composti	8
2.2.5	Partizione/Accorpamento delle associazioni	8
2.2.6	Rimozione delle gerarchie, delle composizioni	8
2.3	Class Diagram ristrutturato	9
2.4	Dizionario delle Classi	11
2.5	Dizionario delle associazioni	11
3	Schema logico	12
3.1	Spiegazione Schema Logico	12
3.1.1	Schema	12
4	Progettazione Fisica	13
4.1	Creazione dei domini	13
4.2	Creazione delle Tabelle	13
4.3	Creazione delle View	16
4.4	Creazione dei Controlli	16
4.4.1	Creazione Trigger per il controllo di validità degli ISBN	16
4.4.2	Creazione Trigger per il controllo di validità degli ISSN delle Riviste	17
4.4.3	Creazione Trigger per il controllo di validità degli ISSN delle Collane	17
4.4.4	Creazione Trigger per l'inserimento automatico della tabella associativa tra Autori e Libri	18
4.4.5	Creazione Trigger per l'inserimento automatico della tabella associativa tra Autori e Articoli	19
4.4.6	Creazione Trigger per il controllo di coerenza delle date di pubblicazione tra Articoli e Riviste	19

Elenco delle figure

1	Logo della Libreria	1
2.1	Class Diagram	7
2.2	Class Diagram Ristrutturato	9
2.3	Diagramma Entity Relation	10

Elenco delle tabelle

2.1	Dizionario delle Classi	11
2.2	Tabella delle Associazioni	11

Listings

4.1	Domini	13
4.2	CreazioneTabelle	14
4.3	View	16
4.4	Trigger per ISBN	16
4.5	Trigger per ISSN_M	17
4.6	Trigger per ISSN_S	17
4.7	Trigger per inserimento Autori	18
4.8	Trigger per inserimento Autori	19
4.9	Trigger per date Magazine-Article	20

Capitolo 1

Descrizione e Analisi del Progetto

1.1 Descrizione e risoluzione sintetica

Il progetto consiste in una base di dati relazionale unito ad un'interfaccia grafica costruita in Java per la gestione di una libreria digitale. Si è pensato di implementare una base di dati relazionali con server in locale tramite il dialetto PostgreSQL.

Si è pensato di articolare la base dati in cinque classi di oggetti per tenere traccia di articoli, libri, autori, collane e riviste.

Sono state implementate delle viste: **Conference** e **Presentation** per mettere in risalto luoghi di presentazioni di libri o discussioni di articoli con rispettivi autori e titoli.

Si è pensato di controllare gli inserimenti delle tuple nel database tramite implementazioni di trigger e domini che controllano ad ogni inserimento la validità degli elementi caratteristici degli oggetti consumabili all'interno della biblioteca, quali ISBN e ISSN per libri e collane rispettivamente. Per articoli e riviste vengono controllati gli inserimenti corretti di DOI e ISSN rispettivamente.

Inoltre vengono attuati controlli per verificare la correlazione tra articoli e le riviste in cui vengono pubblicati.

Tutti i tipi di codici menzionati sono stati scelti come vincoli di chiavi primarie per i rispettivi oggetti.

Capitolo 2

Progettazione concettuale

2.1 Class Diagram

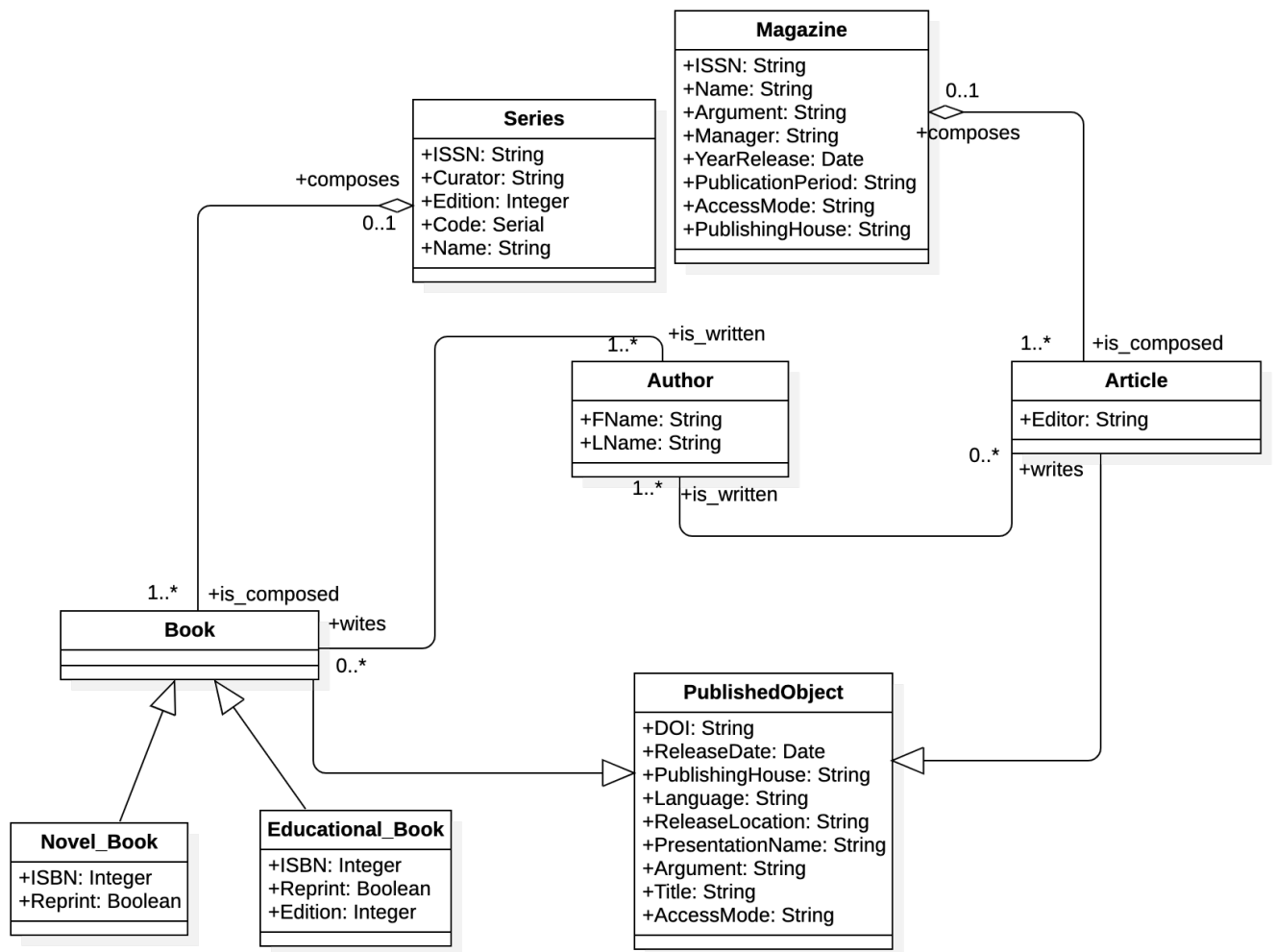


Figura 2.1: Class Diagram

2.2 Analisi della ristrutturazione del Class Diagram

2.2.1 Analisi delle ridondanze

Le ridondanze individuate nel class diagram sono presenti nella specializzazione della classe `Book` in: `Novel_Book` e `Educational_Book`.

Gli attributi: `ISBN_B` e `Reprint` di entrambe le specializzazioni vengono inserite all'interno della classe `Book`.

2.2.2 Analisi degli identificativi

Identificativo primario per la classe `Book` è l'attributo `ISBN_B`, viene attribuito all'attributo `DOI_B` il vincolo di chiave candidata. Per la classe `Author` viene implementato un attributo aggiuntivo: `ID_Author` che sarà chiave primaria. Per la classe `Article` viene usato l'attributo `DOI_A` come chiave primaria. Per le classi `Series` e `Magazine` viene usato come chiave primaria l'attributo `ISSN_S` e `ISSN_M` rispettivamente.

2.2.3 Rimozione degli attributi multipli

Non sono presenti attributi multipli all'interno del Class Diagram.

2.2.4 Rimozione degli attributi composti

Non sono presenti attributi composti all'interno del Class Diagram.

2.2.5 Partizione/Accorpamento delle associazioni

In questo Class Diagram non sono presenti associazioni 1..1 da eliminare.

2.2.6 Rimozione delle gerarchie, delle composizioni

Nel Class Diagram vengono incorporati nella classe `Book` le specializzazioni `Novel_Book` e `Educational_Book` assieme ai rispettivi attributi. All'attributo `Argument` nel Class Diagram ristrutturato, viene specificato l'argomento educativo (Filosofia, Economia ecc.) oppure all'attributo viene assegnato il valore: `Romanzo`. Dato l'interesse di tracciamento delle classi `Book` e `Article` in modo separato, viene eliminata la classe `PublishedObject` e i suoi attributi vengono inseriti all'interno delle classi menzionate precedentemente. Le composizioni che riguardano le classi `Series` e `Magazine`, vengono eliminate e sostituite da una semplice associazione.

2.3 Class Diagram ristrutturato

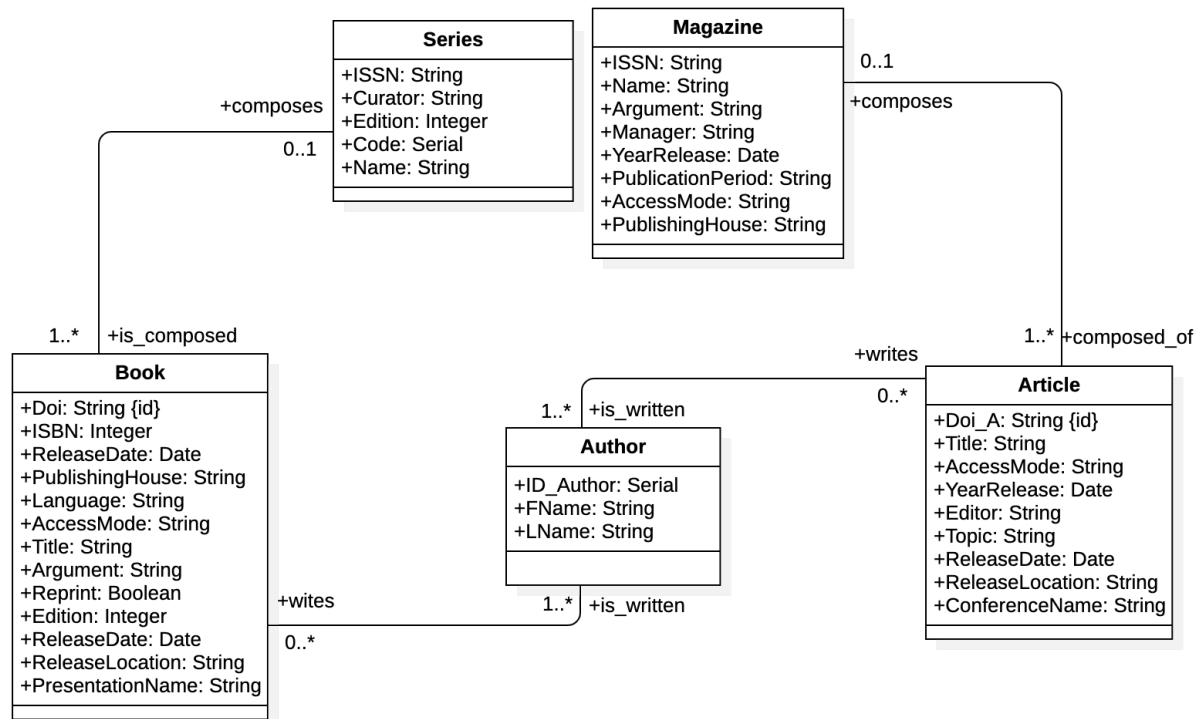


Figura 2.2: Class Diagram Ristrutturato

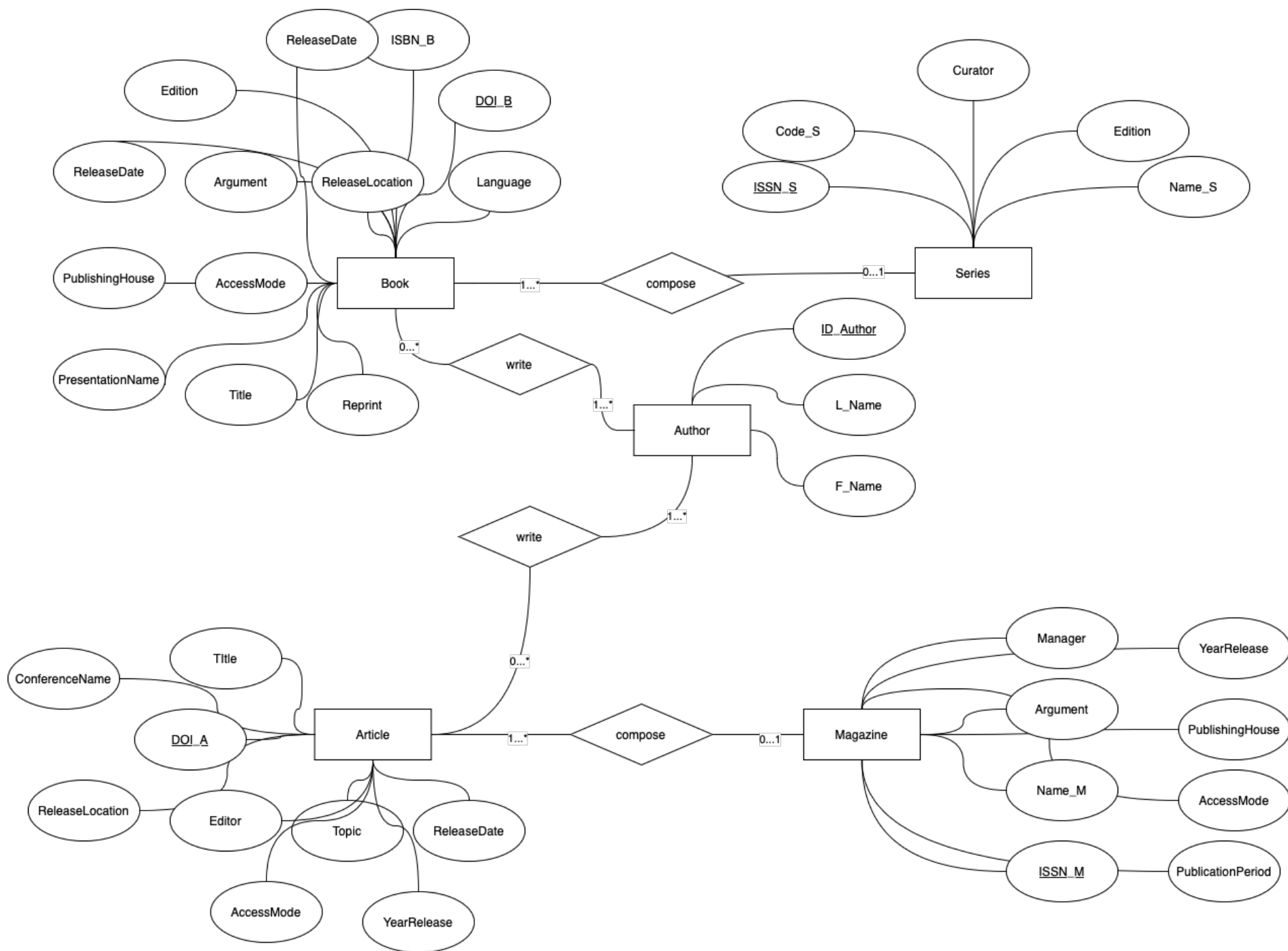


Figura 2.3: Diagramma Entity Relation

2.4 Dizionario delle Classi

Tabella 2.1: Dizionario delle Classi

Classe	Spiegazione	Attributi
Author	Autore di libri o articoli	ID_Author (Serial): Identificazione dell'autore. FirstName (String): Nome dell'autore. LastName (String): Cognome dell'autore.
Books	Oggetti leggibili, romanzi o d'educazione	ISBN_B (String): Classificazione numerica di un libro. Edition (Integer): Numero d'edizione. DOI_B (String): Identificativo Digitale AccessMode (AccessMode): Modo di fruizione. ReleaseDate (Date): Data di pubblicazione. PublishingHouse (String): Casa editrice che ha stampato il libro. ReleaseLocation (String): Luogo di pubblicazione. Language (String): Lingua in cui è scritto il libro. Title (String): Titolo del libro. Argument (String): Argomento del libro. Reprint (Boolean): Parametro che identifica se il libro è una ristampa. PresentationName (String): Nome della presentazione alla quale il libro è presentato.
Series	Insieme di libri	ISSN_S (String): Numero internazionale che identifica le collane. Edition (Integer): Numero dell'edizione. Curator (String): Curatore della collana. Code (Serial): Codice affidato alla collana. Name (String): Nome della collana.
Magazine	Insieme di Articoli	ISSN_M (String): numero internazionale che identifica le riviste. Name (String): Nome della rivista. Argument (String): Argomento della rivista. Manager (String): Manager della rivista. YearRelease (Timestamp): Anno di pubblicazione. PublicationPeriod (String): Periodicità della rivista. AccessMode (AccessMode): Modo di fruizione.
Article	Articoli di ricerca Scientifica	DOI_A (String): Digital object Identifier dell'articolo. Title (String): Titolo dell'articolo. AccessMode (AccessMode): Metodo di fruizione. Editor (String): Editore dell'articolo. ReleaseDate (Timestamp): Data di pubblicazione. ReleaseLocation (String): Luogo di pubblicazione. ConferenceName (String): Nome di conferenza in cui è presentato/discusso l'articolo.

2.5 Dizionario delle associazioni

Tabella 2.2: Tabella delle Associazioni

Nome	Descrizione	Classi Coinvolte
composes/is_composed	Una collana è composta da uno o più libri/ Un libro può comporre oppure no una collana	Series/Book
writes/is_written	Un libro è scritto da uno o più autori/ Un autore scrive molti oppure nessun libro	Book/Author
is_written/writes	Un autore scrive molti oppure nessun articolo/ Un articolo è scritto da uno o più autori	Author/Article
composes/is_composed	Un articolo può comporre oppure no una rivista/ Una rivista è composta da uno o più articoli	Article/Magazine

Capitolo 3

Schema logico

3.1 Spiegazione Schema Logico

Il seguente schema logico riassume la composizione del database, evidenziando le chiavi primarie (in grassetto) e le chiavi esterne per il collegamento di una tabella con l'altra (sottolineate).

3.1.1 Schema

- Author
(**ID_Author**, FName, LName)
- Book
(**DOI_B**, ISBN_B, PublishingHouse, Language, AccessMode, Title, Argument, Reprint, Edition, ReleaseDate, ReleaseLocation, PresentationName, FK_Series)
 $FK_Series \rightarrow Series(ISSN_S)$
- Article
(**DOI_A**, Title, AccessMode, YearRelease, Editor, Topic, ReleaseDate, ReleaseLocation, ConferenceName, FK_Magazine)
 $FK_Magazine \rightarrow Series(ISSN_M)$
- Series
(**ISSN_S**, Curator, Edition, Code, Name)
- Magazine
(**ISSN_M**, Name, Argument, Manager, YearRelease, PublicationPeriod, AccessMode, PublishingHouse)
- Author_Books
(AuthorsFK, BooksFK)
 $AuthorsFK \rightarrow Author(CodAuthor)$
 $BooksFK \rightarrow Book(ISBN_B)$
- Author_Article
(AuthorsFK, ArticlesFK)
 $AuthorsFK \rightarrow Author(CodAuthor)$
 $ArticlesFK \rightarrow Article(DOI_A)$

Capitolo 4

Progettazione Fisica

4.1 Creazione dei domini

La seguente sezione di codice permette la creazione di domini per la verifica dell'inserimento corretto degli attributi all'interno del database. Tali domini fungono da primo controllo per l'inserimento di valori all'interno del database. Nel caso dei domini dell'ISSN, per esempio, non possono essere inseriti caratteri che non siano numeri e il caratteristico trattino tra le due coppie di quattro numeri che servono a comporlo.

Si è scelto di implementare un ragionamento simile per altri tipi sensibili all'interno della base dati, quali ISBN e DOI. Implementando delle restrizioni anche sul formato di inserimento della stringa stessa, andando a specificare quanti e quali caratteri inserire all'interno dell'attributo.

```
1 create domain mtl.issn as varchar(9)
2     check ( value like '%-%'
3           and value not similar to '%[a-z]+'
4           and value not similar to '%[A-Z]+'
5           and value not similar to '%[@!# $ %&]+' );
6
7 create domain mtl.isbn as varchar(30)
8     check ( value like '%-_%-_%-' );
9
10 create domain mtl.doi as varchar(30)
11     check ( value like '10.%%/%' );
12
13 create domain mtl.access as varchar(20)
14     check ( value <> '' and value not similar to '%[0-9]+'
15           and value not similar to '%[@!# $ %&]+' );
16
17 create domain mtl.names as varchar(30)
18     check ( value not similar to '%[@!# $ %&]+' );
19
20 create domain mtl.location as varchar(50)
21     check ( value like '%,%,%,%,%'
22           and value not similar to '%[@!# $ %&]+' );
```

Listing 4.1: Domini

4.2 Creazione delle Tabelle

Sezione di codice atta alla creazione dello schema nel DB e alla creazione di tutte le tabelle necessarie.

Si noti che la sezione di codice riguardante i domini, andrebbe inserita subito dopo le prime due righe della sezione di codice qui rappresentata. Le due sono state separate solo a scopo dimostrativo.

```

1 drop schema if exists mtl cascade;
2 create schema mtl;
3
4 create table mtl.author
5 (
6     CodAuthor serial,
7     FName      varchar(20),
8     LName      varchar(20)
9 );
10
11 CREATE INDEX idx_name
12     ON mtl.author (LName, FName);
13
14 ALTER TABLE IF EXISTS mtl.author
15     ADD CONSTRAINT AuthorPK primary key (CodAuthor);
16
17 create table mtl.series
18 (
19     ISSN_S mtl.issn,
20     Curator mtl.names,
21     Edition int,
22     Code_S varchar(10),
23     Name_S mtl.names
24 );
25
26 ALTER TABLE IF EXISTS mtl.series
27     ADD CONSTRAINT SeriesPK primary key (ISSN_S);
28
29 CREATE INDEX IF NOT EXISTS SeriesIndex ON mtl.series (Code_S);
30
31 create table mtl.magazine
32 (
33     ISSN_M      mtl.issn,
34     Name_M      mtl.names,
35     Argument    mtl.names,
36     Manager     mtl.names,
37     YearRelease timestamp,
38     PublicationPeriod mtl.names,
39     AccessMode  mtl.access,
40     PublishingHouse mtl.names
41 );
42
43 ALTER TABLE IF EXISTS mtl.magazine
44     ADD CONSTRAINT MagazinePk primary key (ISSN_M);
45
46 create table mtl.book
47 (
48     ISBN_B      mtl.isbn,
49     DOI_B       mtl.doi,
50     PublishingHouse mtl.names,
51     Language    mtl.names,
52     AccessMode  mtl.access,
53     Title       varchar(30),
54     Argument    mtl.names,
55     Reprint     boolean,
56     Edition     int,
57     ReleaseDate timestamp,
58     ReleaseLocation mtl.location,

```

```

59     PresentationName mtl.names,
60     FK_Series        mtl.issn,
61
62     constraint BookFK_3 foreign key (FK_Series) references mtl.Series (ISSN_S) on delete set null
63 );
64
65 ALTER TABLE IF EXISTS mtl.book
66     ADD CONSTRAINT BookPk primary key (ISBN_B);
67
68 ALTER TABLE IF EXISTS mtl.book
69     ADD CONSTRAINT BookUnique unique (DOI_B);
70
71
72 create table mtl.article
73 (
74     Doi_A            mtl.doi,
75     Title            varchar(40),
76     AccessMode       mtl.access,
77     Editor           mtl.names,
78     Topic            mtl.names,
79     ReleaseDate       timestamp,
80     ReleaseLocation   mtl.location,
81     ConferenceName    varchar(50),
82     FK_Magazine       mtl.issn,
83
84     constraint ArticleFK_2 foreign key (FK_Magazine) references mtl.Magazine (ISSN_M) on delete set
85         null,
86     check ((FK_Magazine is not null and (ConferenceName is null and ReleaseLocation is null)) OR
87         (FK_Magazine is null and (ConferenceName is not null and ReleaseLocation is not null)))
88 );
89
90 ALTER TABLE IF EXISTS mtl.article
91     ADD CONSTRAINT ArticlePK primary key (Doi_A);
92
93 create table mtl.author_books
94 (
95     AuthorsFK serial,
96     BooksFK    mtl.isbn,
97
98     constraint autori_libriFK_1 foreign key (AuthorsFK) references mtl.author (codauthor) on delete
99         cascade,
100    constraint autori_libriFK_2 foreign key (BooksFK) references mtl.book (ISBN_B) on delete cascade
101 );
102
103 create table mtl.author_article
104 (
105     AuthorsFK serial,
106     ArticlesFK mtl.doi,
107
108     constraint autori_libriFK_1 foreign key (AuthorsFK) references mtl.author (codauthor) on delete
109         cascade,
110    constraint autori_libriFK_2 foreign key (ArticlesFK) references mtl.article (Doi_A) on delete
111         cascade
112 );

```

Listing 4.2: CreazioneTabelle

4.3 Creazione delle View

Vengono implementate due view, per mettere in risalto le conferenze di articoli e le presentazioni di libri. Mettendo anche i nomi degli autori delle rispettive, assieme a data e luogo di ognuna.

```
1 create or replace view mtl.presentation as
2 select b.title, a.fname, a.lname, b.presentationname, b.releaselocation, b.releasedate
3 from (mtl.book b join mtl.author_books au on au.BooksFK = b.ISBN_B)
4      join mtl.author a on a.codauthor = au.AuthorsFK
5 where b.presentationname is not null
6      and b.releaselocation is not null
7 order by lname;
8
9 create or replace view mtl.conference as
10 select a.title, ar.fname, ar.lname, a.conferencename, a.releaselocation, a.releasedate
11 from (mtl.article a join mtl.author_article au on au.ArticlesFK = a.doi_a)
12      join mtl.author ar on ar.AuthorsFK = au.codauthor
13 where a.conferencename is not null
14      and a.releaselocation is not null
15 order by lname;
```

Listing 4.3: View

4.4 Creazione dei Controlli

4.4.1 Creazione Trigger per il controllo di validità degli ISBN

Il seguente trigger implementa un controllo sulla validità dell'ISBN. Non si pretende di entrare nei particolari di tale algoritmo, ma si rimanda alla pagina [Wikipedia](#) per una approfondita spiegazione.

```
1 create or replace function mtl.function_1() returns trigger as
2 $$
3 declare
4     stringa_in varchar(19) = new.isbn_b;
5     sum        integer    := 0;
6     var_appoggio integer;
7     resto      integer;
8 begin
9     stringa_in := replace(stringa_in, '-', '');
10    for i in 1..13
11        loop
12            var_appoggio = cast(substring(stringa_in from i for 1) as int);
13            if (i % 2 = 0) then
14                sum := sum + var_appoggio * 3;
15            else
16                sum := sum + var_appoggio;
17            end if;
18        end loop;
19    resto = sum % 10;
20    if (resto != 0) then
21        delete from mtl.book where isbn_b = new.isbn_b;
22    end if;
23    return new;
24 end
25 $$
26 language plpgsql;
27
28 create trigger validity_isbn
```



```

29     after insert
30     on mtl.book
31     for each row
32 execute procedure mtl.function_1();

```

Listing 4.4: Trigger per ISBN

4.4.2 Creazione Trigger per il controllo di validità degli ISSN delle Riviste

I seguenti trigger implementa un controllo sulla validità dell'ISSN. Non si pretende di entrare nei particolari di tale algoritmo, ma si rimanda alla pagina della [Library of Congress](#) per una approfondita spiegazione.

```

1  create or replace function mtl.function_2() returns trigger as
2  $$
3  declare
4      stringa_in varchar(13) = new.issn_s;
5      sum        integer    := 0;
6      var_appoggio integer;
7      resto      integer;
8  begin
9      stringa_in := replace(stringa_in, '-', '');
10     for i in 1..8
11         loop
12             if substr(stringa_in, 8, 1) = 'X' then
13                 sum = sum + 10;
14                 var_appoggio = cast(substring(stringa_in from i for 1) as int);
15                 if (i = 8) then
16                     sum = sum + 0;
17                 else
18                     sum := sum + var_appoggio * (9 - i);
19                 end if;
20             else
21                 var_appoggio = cast(substring(stringa_in from i for 1) as int);
22                 sum := sum + var_appoggio * (9 - i);
23             end if;
24         end loop;
25     resto = sum % 11;
26     if (resto != 0) then
27         delete from mtl.series where issn_s = new.issn_s;
28     end if;
29     return new;
30 end
31 $$
32 language plpgsql;
33
34 create trigger validity_issn_s
35 after insert
36 on mtl.series
37 for each row
38 execute procedure mtl.function_2();

```

Listing 4.5: Trigger per ISSN_M

4.4.3 Creazione Trigger per il controllo di validità degli ISSN delle Collane

```

1  create or replace function mtl.function_3() returns trigger as
2  $$

```

```

3 declare
4     stringa_in varchar(13) = new.issn_m;
5     sum        integer    := 0;
6     var_appoggio integer;
7     resto      integer;
8 begin
9     stringa_in := replace(stringa_in, '-', '');
10    for i in 1..8
11        loop
12            if substr(stringa_in, 8, 1) = 'X' then
13                sum = sum + 10;
14                var_appoggio = cast(substring(stringa_in from i for 1) as int);
15                if (i = 8) then
16                    sum = sum + 0;
17                else
18                    sum := sum + var_appoggio * (9 - i);
19                end if;
20            else
21                var_appoggio = cast(substring(stringa_in from i for 1) as int);
22                sum := sum + var_appoggio * (9 - i);
23            end if;
24        end loop;
25    resto = sum % 11;
26    if (resto != 0) then
27        delete from mtl.magazine where issn_m = new.issn_m;
28    end if;
29    return new;
30 end
31 $$
32 language plpgsql;
33
34 create trigger validity_issn_m
35 after insert
36 on mtl.magazine
37 for each row
38 execute procedure mtl.function_3();

```

Listing 4.6: Trigger per ISSN_S

4.4.4 Creazione Trigger per l'inserimento automatico della tabella associativa tra Autori e Libri

I seguenti Trigger implementano una procedura di controllo e inserimento per le tabelle di collegamento tra autori e libri, e tra autori e articoli.

Si è scelto di fare ciò per una questione di velocità e facilità nella creazione e popolazione del Database.

```

1 CREATE OR REPLACE FUNCTION mtl.function_4()
2     returnS TRIGGER AS
3 $$
4 declare
5     isbn_app      mtl.book.isbn_b%TYPE;
6     autori_record mtl.author%ROWTYPE;
7     cod_author_app mtl.author.codauthor%TYPE;
8     cod_author CURSOR FOR SELECT codauthor
9                        FROM mtl.author;
10 begin
11     isbn_app := new.isbn_b;
12     FOR autori_record IN cod_author

```

```

13     loop
14         cod_author_app := autori_record.codauthor;
15         INSERT INTO mtl.author_books(AuthorsFK, BooksFK) VALUES (cod_author_app, isbn_app);
16     end loop;
17     return NULL;
18 end;
19 $$
20 LANGUAGE plpgsql;
21
22
23 create trigger insert_autori_libri
24     after insert
25     on mtl.book
26     for each row
27 execute procedure mtl.function_4();

```

Listing 4.7: Trigger per inserimento Autori

4.4.5 Creazione Trigger per l'inserimento automatico della tabella associativa tra Autori e Articoli

```

1  CREATE OR REPLACE FUNCTION mtl.function_5() returnS TRIGGER AS
2  $$
3  declare
4      doi_app      mtl.article.doi_a%TYPE;
5      autori_record mtl.author%ROWTYPE;
6      cod_author_app mtl.author.codauthor%TYPE;
7      cod_author CURSOR FOR SELECT codauthor
8                          FROM mtl.author;
9  begin
10     doi_app := new.doi_a;
11     for autori_record in cod_author
12     loop
13         cod_author_app := autori_record.codauthor;
14         insert into mtl.author_article (AuthorsFK, ArticlesFK) values (cod_author_app, doi_app);
15     end loop;
16     return NULL;
17 end;
18 $$
19 LANGUAGE plpgsql;
20
21
22 create trigger insert_autori_articoli
23     after insert
24     on mtl.article
25     for each row
26 execute procedure mtl.function_5();

```

Listing 4.8: Trigger per inserimento Autori

4.4.6 Creazione Trigger per il controllo di coerenza delle date di pubblicazione tra Articoli e Riviste

Il seguente trigger viene implementato per il controllo di coerenza tra la data di pubblicazione di una rivista e la data di pubblicazione di un articolo pubblicato in tale rivista, andando ad eliminare la riga che viola tale controllo.

```

1  create or replace function mtl.function_6() returns trigger as
2  $$
3  declare
4      articoli_record record;
5      articoli_cursore cursor for (select *
6                                  from mtl.article a
7                                  where a.fk_magazine = new.issn_m);
8  begin
9      for articoli_record in articoli_cursore
10         loop
11             delete
12             from mtl.article
13             where new.yearrelease <> articoli_record.releasedate;
14         end loop;
15     return new;
16 end;
17 $$
18     language plpgsql;
19
20
21 create trigger date_coerenti
22     after insert
23     on mtl.magazine
24     for each row
25 execute procedure mtl.function_6();

```

Listing 4.9: Trigger per date Magazine-Article