Progetto Object Orientation e Basi di Dati

Lorenzo Tecchia N86004446
Mirko Marciano N86004019
10/12/2022

Progettazione e sviluppo di una base di dati relazionale per la gestione di una biblioteca digitale

Indice

1 Descrizione e Analisi del Progetto **5**
1.1 Descrizione e risoluzione sintetica 5

2 Progettazione concettuale **6**
2.1 Class Diagram 6
2.2 Analisi della ristrutturazione del Class Diagram 7
2.2.1 Analisi delle ridondanze 7
2.2.2 Analisi degli identificativi 7
2.2.3 Rimozione degli attributi multipli 7
2.2.4 Rimozione degli attributi composti 7
2.2.5 Partizione/Accorpamento delle associazioni 7
2.2.6 Rimozione delle gerarchie 7
2.3 Class Diagram ristrutturato 7
2.4 Dizionario delle classi 8
2.5 Dizionario delle associazioni 10

3 Schema logico **11**

4 Progettazione Fisica **12**
4.1 Creazione delle Tabelle 12
4.2 Creazione dei domini 14
4.3 Creazione delle viste 14
4.4 Creazione di funzioni e trigger 15

5 Caso d'uso e manuale **18**

Elenco delle figure

2.1 Class Diagram 6
2.2 Class Diagram Ristrutturato 7

Elenco delle tabelle

2.1 Dizionario delle Classi 8

Capitolo 1

Descrizione e Analisi del Progetto

1.1 Descrizione e risoluzione sintetica

Capitolo 2

Progettazione concettuale

2.1 Class Diagram

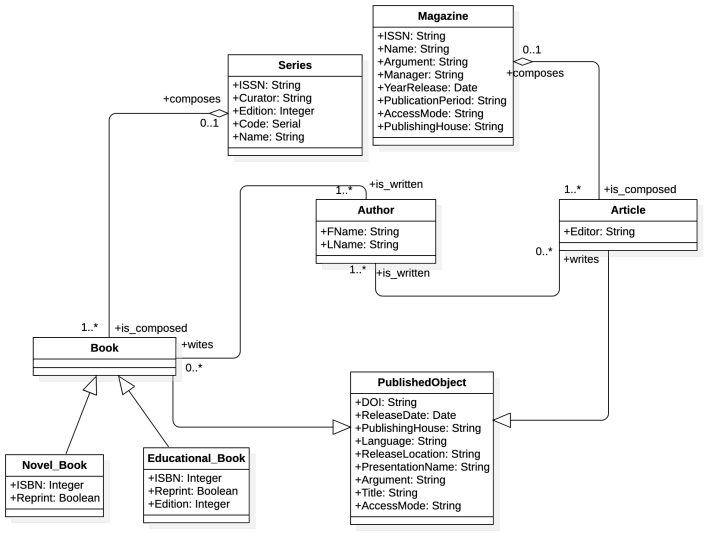

Figura 2.1: Class Diagram

2.2 Analisi della ristrutturazione del Class Diagram

2.2.1 Analisi delle ridondanze

2.2.2 Analisi degli identificativi

2.2.3 Rimozione degli attributi multipli

2.2.4 Rimozione degli attributi composti

2.2.5 Partizione/Accorpamento delle associazioni

2.2.6 Rimozione delle gerarchie

2.3 Class Diagram ristrutturato

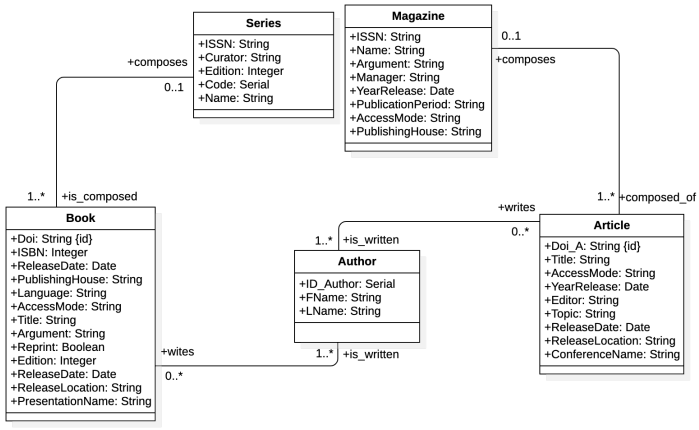

Figura 2.2: Class Diagram Ristrutturato

2.4 Dizionario delle classi

Tabella 2.1: Dizionario delle Classi

Class	Explanation	Attributes
Authors	Authors of books or articles	ID_Author (Serial): Author's identification code FName (String): Author's first name LName (String): Author's last name
Book	Books that can be novels or educational	DOI (String): Digital object Identifier of the book. ISBN (Integer): Numerical classification sequence of the book. Edition (Integer): Edition number. AccessMode (AccessMode): Fruition method. ReleaseDate (Date): Publication date. PublishingHouse (String): Publishing house that printed the book. ReleaseLocation (String): Place of publication of the book. Language (String): Language in which the book is written. Title (String): Book title. Argument (String): Book topic. Reprint (Boolean): Parameter that identifies if the book is a reprint or not. PresentationName (String): Name of pesentation in which books are presented

Series	Set of books	ISSN (Integer): International number that identifies serial publications. Edition (Integer): Edition number. Curator (String): Curator of the series. Code (Serial): Code assigned to the series. Name (String): Series' name .
Magazine	Set of articles	ISSN (Integer): International number that identifies serial publications. Name (String): Magazine's name Argument (String): Magazine topic. Manager (String): Event organizer. YearRelease (Date): Publication year. PublicationPeriod (String): Periodicity of publication. AccessMode (AccessMode): Fruition method.
Article	Articles of scientific research	DOI (String): Digital object Identifier of the book. Title (String): Book title. AccessMode (AccessMode): Fruition method. YearRelease (Date): Publication year. Editor (String):Article editor. ReleaseDate (Date): Publication date. ReleaseLocation (String): Place of publicationof the book. ConferenceName (String): Name of pesentation in which books are presented

2.5 Dizionario delle associazioni

Capitolo 3

Schema logico

- Author

  (**IDAuthor**, FName, LName)

- Book

  (**DOI**, ISBN, PublishingHouse, Language, AccessMode, Title, Argument, Reprint, Edition, ReleaseDate, ReleaseLocation, PresentationName, AuthorID)

- Article

  (**DOI** , Title, AccessMode, YearRelease, Editor, Topic, ReleaseDate, ReleaseLocatione, ConferenceName , AuthorID)

- Series

  (**ISSN**, Curator. Edition, Code, Name)

- Magazine

  (**ISSN**, Name , Argument, Manager, YearRelease, PublicationPeriod, AccessMode, PublishingHouse)

Capitolo 4

Progettazione Fisica

4.1 Creazione delle Tabelle

```
drop schema mtl cascade;
create schema mtl;

--create table mtl.author
create table mtl.author
(
    CodAuthor serial
        primary key,
    FName    varchar(20),
    LName    varchar(20)
);

-- create table mtl.series
create table mtl.series
(
    ISSN_S issn primary key,
    Curator names,
    Edition int,
    Code_S varchar(10),
    Name_S names
);
--create table mtl.magazine
create table mtl.magazine
(
    ISSN_M          issn primary key,
    Name_M          names,
    Argument        names,
    Manager         names,
    YearRelease     timestamp,
    PublicationPeriod names,
```

```
    AccessMode      access,
    PublishingHouse names
);


--create table mtl.book
create table mtl.book
(
    Doi_B          doi
        primary key,
    ISBN_B         varchar(13)
        unique,
    PublishingHouse names,
    Language       names,
    AccessMode     access,
    Title          varchar(30),
    Argument       names,
    Reprint        boolean,
    Edition        int,
    ReleaseDate    timestamp,
    ReleaseLocation location,
    PresentationName names,
    FK_Author      serial,
    FK_Series      issn,

    constraint BookFK_2 foreign key (Fk_Author) references mtl.Author
        (CodAuthor) on delete cascade,
    constraint BookFK_3 foreign key (FK_Series) references mtl.Series
        (ISSN_S) on delete set null
);

--create table mtl.article
create table mtl.article
(
    Doi_A          doi
        primary key,
    Title          varchar(40),
    AccessMode     access,
    YearRelease    timestamp,
    Editor         names,
    Topic          names,
    ReleaseDate    timestamp,
    ReleaseLocation location,
    ConferenceName varchar(50),
    FK_Author      serial,
    FK_Magazine    issn,

    constraint ArticleFK_1 foreign key (FK_Author) references mtl.Author
        (CodAuthor) on delete cascade,
```

```
    constraint ArticleFK_2 foreign key (FK_Magazine) references
        mtl.Magazine (ISSN_M) on delete set null
);
```

4.2 Creazione dei domini

```
   create domain issn as varchar(9)
check ( value like '%-%' );

create domain isbn as varchar(17)
check ( value like '%-_-%-%-_');

create domain doi as varchar(30)
check ( value like'10.%/%');

create domain access as varchar(20)
check ( value <> '' and value not similar to '%[0-9]+%'
       and value not similar to '%[@!# $ %&]+%');

create domain names as varchar(30)
check ( value not similar to '%[@!# $ %&]+%');

create domain location as varchar(50)
check ( value like '%,[0-9],%,[0-9],%');
```

4.3 Creazione delle viste

```
create view mtl.bibliography as
select distinct b.Title,b.ReleaseDate,a.lname
from mtl.book b join mtl.author a on b.fk_author = a.codauthor
order by b.releasedate desc;

create view mtl.history as
select distinct a.fname, a.lname, ar.title,ar.yearrelease,ar.editor
from mtl.author a join mtl.article ar on a.codauthor = ar.fk_author
order by ar.yearrelease desc;

create view mtl.digital_goods as
select distinct b.title from mtl.book b where accessmode = 'Digital'
union
select distinct a.title from mtl.article a where accessmode = 'Digital'
union
select distinct m.name_m from mtl.magazine m where accessmode = 'Digital'
union
```

```
select distinct s.name_s from mtl.series s join mtl.book b on s.issn_s =
    b.fk_series where b.accessmode='Digital';

create view mtl.paper_goods as
select distinct b.title from mtl.book b where accessmode = 'Paper'
union
select distinct a.title from mtl.article a where accessmode = 'Paper'
union
select distinct m.name_m from mtl.magazine m where accessmode = 'Paper'
union
select distinct s.name_s from mtl.series s join mtl.book b on s.issn_s =
    b.fk_series where b.accessmode='Digital';

create view mtl.audio_goods as
select distinct b.title from mtl.book b where accessmode = 'Audio'
union
select distinct a.title from mtl.article a where accessmode = 'Audio'
union
select distinct m.name_m from mtl.magazine m where accessmode = 'Audio'
union
select distinct s.name_s from mtl.series s join mtl.book b on s.issn_s =
    b.fk_series where b.accessmode='Digital';

create view mtl.presentation as
select b.title,a.fname,a.lname, b.presentationname, b.releaselocation,
    b.releasedate
from mtl.book b join mtl.author a on a.codauthor = b.fk_author;

create view mtl.discussion as
select ar.title,a.fname,a.lname, ar.conferencename, ar.releaselocation,
    ar.releasedate
from mtl.article ar join mtl.author a on a.codauthor = ar.fk_author
order by a.lname;
```

4.4 Creazione di funzioni e trigger

```
   create or replace function mtl.function_1() returns trigger as
$$
declare
    stringa_in  varchar(13) = new.isbn_b;
    sum         integer    := 0;
    var_appoggio integer;
    resto       integer;
begin
    stringa_in := replace(stringa_in, '-', '');
    for i in 1..13
        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 doi_b = new.doi_b;
22      end if;
23      return new;
24  end
25  $$
26      language plpgsql;
27
28  create trigger validity_isbn
29      after insert
30      on mtl.book
31      for each row
32  execute procedure mtl.function_1();
33
34  create or replace function mtl.function_2() returns trigger as
35  $$
36  declare
37      stringa_in  varchar(13) = new.issn_s;
38      sum         integer    := 0;
39      var_appoggio integer;
40      resto       integer;
41  begin
42      stringa_in := replace(stringa_in, '-', '');
43      for i in 1..8
44          loop
45              if substr(stringa_in, 8, 1) = 'X' then
46                  sum = sum + 10;
47              end if;
48              var_appoggio = cast(substring(stringa_in from i for 1) as
                    int);
49              if (i = 8) then
50                  sum = sum + 0;
51              else
52                  sum := sum + var_appoggio * (9 - i);
53              end if;
54          end loop;
55      resto = sum % 11;
56      if (resto != 0) then
57          delete from mtl.series where issn_s = new.issn_s;
58      end if;
59      return new;
```

```
60  end
61  $$
62      language plpgsql;
63
64  create trigger validity_issn_s
65      after insert
66      on mtl.series
67      for each row
68  execute procedure mtl.function_2();
69
70  create or replace function mtl.function_3() returns trigger as
71  $$
72  declare
73      stringa_in  varchar(13) = new.issn_m;
74      sum         integer    := 0;
75      var_appoggio integer;
76      resto       integer;
77  begin
78      stringa_in := replace(stringa_in, '-', '');
79      for i in 1..8
80          loop
81              if substr(stringa_in, 8, 1) = 'X' then
82                  sum = sum + 10;
83              end if;
84              var_appoggio = cast(substring(stringa_in from i for 1) as
                    int);
85              if (i = 8) then
86                  sum = sum + 0;
87              else
88                  sum := sum + var_appoggio * (9 - i);
89              end if;
90          end loop;
91      resto = sum % 11;
92      if (resto != 0) then
93          delete from mtl.magazine where issn_m = new.issn_m;
94      end if;
95      return new;
96  end
97  $$
98      language plpgsql;
99
100 create trigger validity_issn_m
101     after insert
102     on mtl.magazine
103     for each row
104 execute procedure mtl.function_3();
```

Capitolo 5

Caso d'uso e manuale