



DOCUMENTAZIONE DEL PROGETTO DI BASI DI DATI A.A. 2022-2023

Giulio Pianese Giuseppe Sindoni
N86004335 N86004107
ID GRUPPO: OODB_T1G25

Indice

1	Analisi dei Requisiti	3
1.1	Analisi della realtà di interesse	3
2	Progettazione Concettuale	5
2.1	Schema Concettuale	5
2.2	Dizionario delle Entità e delle Associazioni	6
3	Ristrutturazione dello Schema concettuale	9
3.1	Analisi delle Ridondanze	9
3.2	Ristrutturazione delle generalizzazioni	9
3.3	Eliminazione degli attributi Multivalore	10
3.4	Eliminazione degli attributi strutturati	10
3.5	Individuazione delle chiavi primarie	11
3.6	Schema Concettuale ristrutturato	11
	3.6.1 Modello E/R	11
	3.6.2 Modello UML	12
4	Progettazione Logica	13
4.1	Schema logico	13
5	Progettazione Fisica	15
5.1	Definizione dei domini	15
5.2	Definizione Tabelle	16
5.3	Trigger e Funzioni	20

Capitolo 1

Analisi dei Requisiti

1.1 Analisi della realtà di interesse

La prima fase da affrontare è quella di esaminare le richieste da parte del cliente.
La realtà che ci è stata chiesta di gestire è quella di una **Biblioteca digitale**.

*"Si sviluppi un sistema informativo, composto da una base di dati relazionale e da un applicativo Java dotato di GUI (Swing o JavaFX), per la gestione di una biblioteca digitale.
Gli elementi che possono essere inclusi nella biblioteca digitale sono di due tipi: articoli scientifici (o pubblicazioni) e libri (didattici o romanzi)."*

Le entità principali del **Miniworld** in questione sono : **Pubblicazioni** (o Articoli scientifici) e **Libri**. Una prima precisazione è la suddivisione dei libri in **Libri didattici** e **Romanzi**. Non essendo specificato deduciamo che un libro può far parte strettamente ad una delle due categorie.

Un libro può anche far parte di una collana, la quale può raggruppare tutte le pubblicazioni che condividono una determinata caratteristica (non tutti i libri fanno parte di collane)".

Qui ci viene specificata l'esistenza di **Collane**, che possiamo vedere come un raggruppamento di libri secondo un particolare criterio. Come specificato un libro non fa parte necessariamente di una collana e nel caso opposto deduciamo possa far parte di una o più di quest'ultime.

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.

Altra informazione importante relativa ad una **Pubblicazione** è la sua relativa divulgazione, che può avvenire tramite una **Conferenza** o tramite una **Rivista** .

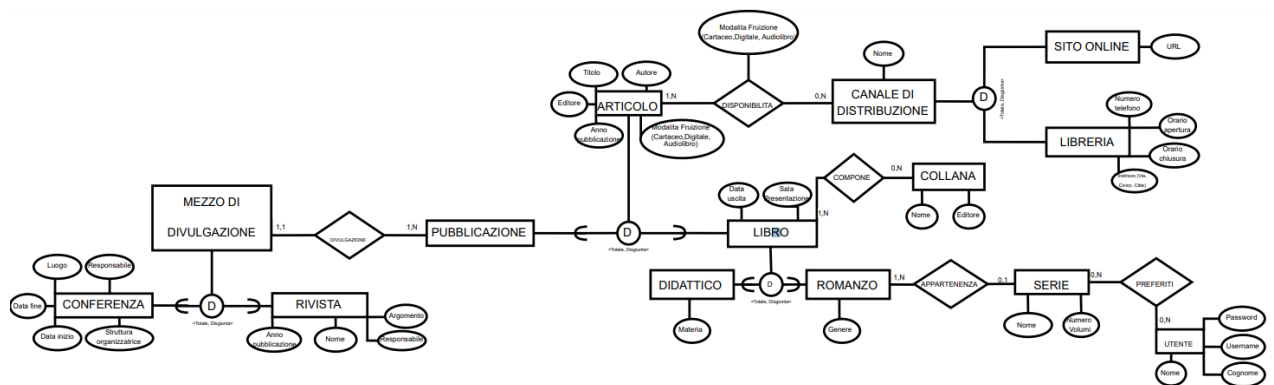
Per ogni libro, è necessario specificare dove può essere acquistato (libreria, online). Un romanzo può avere anche uno o più seguiti. In tal caso, è importante prevedere un'interrogazione che permette di recuperare tutte le librerie (o siti internet) dai quali è possibile acquistare l'intera serie dei libri.

Oltre alla '**Disponibilità**' di un libro mediante **Siti** o **Librerie**, ci vien specificato che un Romanzo può far parte di una Serie. La richiesta prevede che il sistema notifichi l'**utente** ogni qualvolta un sito o una libreria dispongano di un'intera serie di libri (più precisamente Romanzi). Per soddisfare a pieno la richiesta definiamo la lista dei **Preferiti**, che non è altro un elenco di Serie per il quale un utente è interessato.

Capitolo 2

Progettazione Concettuale

2.1 Schema Concettuale



2.2 Dizionario delle Entità e delle Associazioni

Entità	Descrizione	Attributi
Articolo	Rappresentazione generica dei vari articoli, entità padre di "Libro" e "Pubblicazione"	Titolo: rappresenta il titolo dell'articolo. Editore: l'ente o l'organizzazione responsabile della pubblicazione. Anno pubblicazione: anno in cui è stato pubblicato l'articolo. Modalità fruizione: (Cartaceo, Digitale, Audiolibro), attributo multiplo che rappresenta i formati nei quali l'articolo è stato realizzato. Autore: attributo multiplo che indica l'autore o gli autori che hanno contribuito alla scrittura dell'articolo.
Pubblicazione	Rappresentazione specifica di un articolo, entità figlia di Articolo	Nessun attributo specifico.
Libro	Rappresentazione specifica di un articolo, entità figlia di Articolo	Data uscita: si riferisce alla data in cui il libro è stato pubblicato e reso disponibile per l'acquisto e la distribuzione al pubblico. Sala presentazione: sala o libreria in cui è stata effettuata l'eventuale presentazione del libro.
Didattico	Genere specifico di libro, entità figlia di Libro	Materia: argomento principale o il campo di conoscenza che viene trattato nel libro.
Romanzo	Genere specifico di libro, entità figlia di Libro	Genere: genere del romanzo.
Collana	Insieme di libri accomunati da uno specifico criterio	Nome: nome della collana. Editore: l'ente o l'organizzazione responsabile della pubblicazione.
Serie	Insieme di Romanzi correlati che condividono personaggi, ambientazioni o una trama continua.	Nome, nome della serie. Numero volumi, rappresenta il numero di Romanzi che compongono l'intera serie.

Entità	Descrizione	Attributi
Utente	Rappresenta l'utente che interagisce con il sistema	Nome: nome dell'utente. Cognome: cognome dell'utente. Username: identificatore univoco o un nome scelto dall'utente stesso per identificarsi.
Mezzo di divulgazione	Rappresenta i mezzi tramite i quali una Pubblicazione viene divulgata.	
Conferenza	Rappresenta una conferenza in cui una pubblicazione scientifica è stata presentata, entità figlia di Mezzo di divulgazione.	Luogo, luogo in cui si è svolta la conferenza. Responsabile, la persona o l'ente responsabile della gestione e supervisione generale della Conferenza. Struttura organizzatrice, struttura che ha ospitato la conferenza. Data inizio, data di inizio della conferenza. Data fine, data di termine della conferenza.
Rivista	Rappresenta una rivista in cui una pubblicazione scientifica è stata presentata, entità figlia di Mezzo di divulgazione.	Nome, il nome della rivista. Anno pubblicazione, anno in cui è stata pubblicata la rivista. Argomento, l'argomento principale o l'area di interesse della rivista. Responsabile, la persona o l'ente responsabile della gestione e supervisione generale della rivista.
Canale di distribuzione	Piattaforma generica che si occupa della vendita di articoli.	Nome, nome del canale di distribuzione.
Sito online	una piattaforma o un'entità virtuale accessibile tramite Internet, entità figlia di Canale di distribuzione.	URL, indirizzo web del sito.
Libreria	Luogo dedicato alla raccolta e fruizione di articoli, , entità figlia di Canale di distribuzione.	Indirizzo attributo composto che specifica la posizione geografica della libreria. Numero telefono, numero di recapito telefonico della libreria. Orario apertura, Orario di apertura della libreria. Orario chiusura, Orario di chiusura della libreria.

Associazione	Descrizione	Cardinalità
Disponibilità	Associazione tra Articolo e Canale di distribuzione. Rappresenta la disponibilità di un' articolo mediante un canale di distribuzione in una specifica modalità di fruizione.	M:N Un articolo può esser disponibile tramite 0 o più canali di distribuzione, mentre un canale di distribuzione deve disporre di uno o più articoli.
Divulgazione	Associazione tra Pubblicazione e Mezzo di divulgazione. Rappresenta la divulgazione di una pubblicazione mediante uno specifico mezzo di divulgazione.	1:N Una pubblicazione è pubblicata da uno e un solo mezzo di divulgazione mentre un mezzo di divulgazione può pubblicare uno o più pubblicazioni.
Compone	Associazione tra Libro e Collana. Associazione che rappresenta di un libro ad una collana.	N:M Un libro può far parte di zero o più collane, mentre ad una collana devono far parte uno o più libri.
Appartenenza	Associazione tra Romanzo e serie. Associazione che rappresenta l'appartenenza di un Romanzo ad una Serie.	1:N Un romanzo può appartenere al più ad una serie, mentre una serie devono appartenere uno o più libri.
Preferiti	Associazione tra Serie e Utente. Tale associazione rappresenta l'interesse dell'utente verso quella serie.	N:M Un utente può avere zero o più serie preferite e allo stesso tempo una serie può esser la preferita di zero o più utenti.

Capitolo 3

Ristrutturazione dello Schema concettuale

3.1 Analisi delle Ridondanze

Le ridondanze che abbiamo riscontrato sono:

- **Entità Collana:** in quanto può esser riprodotta mediante una view sulla tabella libro.
- **L'attributo Numero volumi dell'associazione Serie :** in quanto è un attributo il cui valore è il risultato di una semplice interrogazione. Abbiamo deciso di tenerlo conseguentemente al numero di volte cui ci accediamo.

3.2 Ristrutturazione delle generalizzazioni

Le generalizzazioni all'interno del nostro schema concettuale sono:

- **Articolo**
- **Libro**
- **Mezzo di divulgazione**
- **Canale di distribuzione**

Partendo dalla generalizzazione "Articolo" abbiamo seguito le normali regole di ristrutturazione. Quindi essendo totale-disgiunta la abbiamo ristrutturata verso il basso (dal padre verso i figli), quindi Pubblicazione e Libro (entità figlie) ereditano tutti gli attributi e associazioni di Articolo (entità padre).

Per quanto riguarda l'associazione **Disponibilità** ora avremo l'associazione '**Disponibilita_Pubblicazione**' e '**Disponibilita_Libro**' rispettivamente da Pubblicazione e Libro verso Canale di Distribuzione.

Riguardo la generalizzazione **Libro** abbiamo deciso di ristrutturarla verso l'alto in quanto la maggior parte delle informazioni è contenuta nell'entità padre anch'essa coinvolta in una generalizzazione. Conseguentemente alla ristrutturazione verso l'alto l'entità **Libro** (entità padre) erediterà tutti gli attributi e associazioni di **Didattico** e **Romanzo** (entità figlie). Inoltre sarà aggiunto un attributo **'Tipo'**, all'entità padre, che specificherà se la tupla in questione sarà un **Libro Didattico** o un **Romanzo**.

Riguardo la generalizzazione **Mezzo di divulgazione** l'abbiamo ristrutturata verso il basso. L'Associazione **Divulgazione** sarà ereditata da **Conferenza** e **Rivista**, quindi avremo due distinte associazioni: **Divulgazione_Conferenza** e **Divulgazione_Rivista**.

Riguardo la generalizzazione **Canale di distribuzione** abbiamo deciso di ristrutturarla verso l'alto. Quindi l'entità padre **Canale di distribuzione** erediterà tutti gli attributi di **Sito Online** e **Libreria**, oltre ad un attributo **Tipo** che specificherà se la tupla in questione sarà un sito o una libreria.

3.3 Eliminazione degli attributi Multivalore

Come attributi multivalore abbiamo riscontrato:

- **Autore** (di **Libro**), che abbiamo deciso di renderlo una stringa di autori separati da virgole, in quanto non ci interessa memorizzare informazioni sugli specifici autori.
- **Modalità di fruizione** (di **Articolo**), che abbiamo deciso di rendere anch'esso una stringa di modalità di fruizione separati da virgole, in quanto è facile da gestire anche all'interno di interrogazioni al database e inoltre perché tale scelta ci permette di essere tolleranti nel caso di aggiunte future di nuove modalità di fruizione.

3.4 Eliminazione degli attributi strutturati

Come attributi strutturati abbiamo riscontrato:

- **Indirizzo** (di **Libreria**) composto da (Via, Civico, Città). Abbiamo deciso di renderlo semplicemente una stringa.

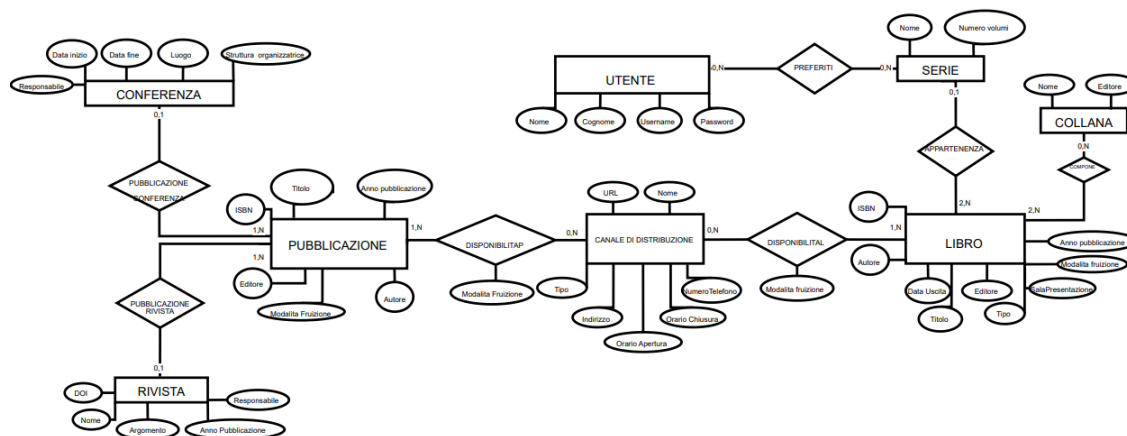
3.5 Individuazione delle chiave primarie

Entità	Chiave Primaria
Utente	IdUtente
Pubblicazione	ISBN *
Libro	ISBN *
Conferenza	CodConferenza
Rivista	CodRivista
CanaleDistribuzione	CodCanale
Collana	CodCollana
Serie	CodSerie

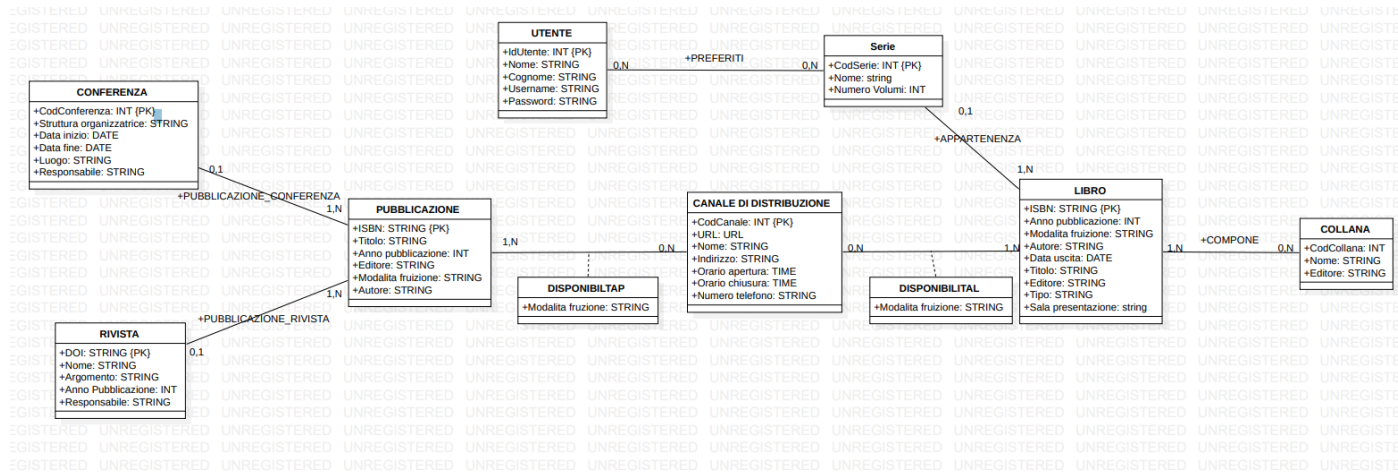
* L'ISBN è un codice identificativo univoco utilizzato per identificare in modo univoco un libro o una pubblicazione

3.6 Schema Concettuale ristrutturato

3.6.1 Modello E/R



3.6.2 Modello UML



Capitolo 4

Progettazione Logica

Successivamente alla progettazione concettuale passiamo alla fase di Mapping delle entità e associazioni in **relazioni**. Le entità saranno trasformate in relazioni con il proprio nome i propri attributi e la propria chiave primaria. Mentre le associazioni verranno gestite diversamente in base alla loro molteplicità.

4.1 Schema logico

Legenda: Chiave Primaria Chiave esterna*

Pubblicazione(ISBN, Titolo, Editore, AnnoPubblicazione, ModalitaFruizione, Autore, Conferenza*, Rivista*)

In tale relazione l'attributo **Conferenza** referencia la relazione Conferenza riferendosi alla sua chiave primaria **CodConferenza**. Ciò perchè tra le due entità all'interno dello schema concettuale vi era l'associazione DivulgazioneConferenza di tipo 1:N. Mentre l'attributo **Rivista** funge anch'esso da chiave esterna referenziando la relazione Rivista riferendosi alla sua chiave primaria **CodRivista**.

Libro(ISBN, Titolo, Editore, SalaPresentazione, DataUscita, AnnoPubblicazione, ModalitaFruizione, Autore, Serie*)

In questo caso l'attributo **Serie** è una chiave esterna che referencia la relazione Serie riferendosi alla sua chiave primaria **CodSerie**. E' stato realizzato ciò, in quanto all'interno dello schema concettuale, l'associazione **Appartenenza** tra Libro e Serie è un' associazione di tipo 1:N

Conferenza(CodConferenza, DataInizio, DataFine, Responsabile, AnnoPubblicazione)

Rivista(DOI, Nome, Argomento, Responsabile, AnnoPubblicazione)

CanaleDistribuzione(CodCanale, Nome, Tipo, URL, Indirizzo, NumeroTelefono, OrarioApertura, OrarioChiusura)

DisponibilitaP(ISBN*, CodCanale*, ModalitaFruizione)

Essendo DisponibilitaP, all'interno del nostro schema concettuale, un'associazione N:M viene realizzata una vera e propria relazione, dove la chiave primaria è composta da ISBN, CodCanale e ModalitaFruizione. A sua volta ISBN funge da chiave esterna referenziando la relazione Pubblicazione, riferendosi alla sua chiave primaria ISBN. Invece CodCanale referencia la relazione CanaleDistribuzione riferendosi alla sua chiave primaria CodCanale. Nel complesso tale relazione rappresenta la disponibilità una specifica Pubblicazione (indicata dalla chiave esterna ISBN) presso uno specifico Canale di Distribuzione (indicato dalla chiave esterna CodCanale) in una specifica modalità di fruizione (indicata dall'attributo ModalitaFruizione).

DisponibilitaL(ISBN*, CodCanale*, ModalitaFruizione)

Caso identico alla relazione DisponibilitaP, l'unica differenza è che la chiave esterna ISBN referencia la tabella Libro, in quanto tale relazione nasce dall'associazione DisponibilitaL.

Collana(CodCollana, Nome, Editore)

Compone(ISBN*, CodCollana*)

Tale relazione nasce dall'associazione Compone tra Libro e Collana. Essendo un'associazione N:M viene realizzata una vera e propria relazione. ISBN funge da chiave esterna, referencia la relazione Libro, mentre la chiave esterna CodCollana referencia la relazione Collana.

Serie(CodSerie, Nome, NumVolumi)

Preferiti(IdUtente*, codSerie*)

Tale relazione nasce dall'associazione Preferiti tra Utente e Serie. Essendo un'associazione N:M viene realizzata una vera e propria relazione. La chiave primaria è composta da idUtente e CodSerie. IdUtente funge da chiave esterna e referencia la tabella Utente, mentre la chiave esterna Serie referencia la tabella Serie.

Capitolo 5

Progettazione Fisica

5.1 Definizione dei domini

I domini di attributi sono regole o vincoli che specificano il tipo di dato e le restrizioni associate a un attributo all'interno di una tabella del database. Definiscono i valori validi che un attributo può assumere, contribuendo a garantire l'integrità dei dati.

```
CREATE DOMAIN tipo_canale AS VARCHAR(20)
CHECK (VALUE IN ('Sito Online', 'Libreria'));
```

Tale dominio viene definito come un dato VARCHAR al più di venti caratteri, grazie alla clausola check viene verificato che il valore dell'attributo sia uno dei due: "Sito Online" o "Libreria". Esso verrà utilizzato per l'attributo **Tipo** della tabella CanaleDistribuzione.

```
CREATE DOMAIN URL varchar(100)
CHECK (VALUE LIKE 'http://%' OR VALUE LIKE 'https://%' );
```

Tale dominio viene definito come un dato di tipo VARCHAR con una lunghezza massima di 100 caratteri, viene verificato che il valore dell'attributo inizi con "http://" o "https://%". Tale dominio sarà utilizzato per l'attributo URL della tabella SitoOnline.

```
CREATE DOMAIN ISBN AS VARCHAR(17)
CHECK (VALUE ~ '^((97(8|9))?\d{9})(\d|X)$');
```

Il dominio "ISBN" viene definito come un dato di tipo VARCHAR con una lunghezza massima di 17 caratteri. L'espressione regolare utilizzata nella clausola check controlla le seguenti condizioni:

- Deve iniziare con "97" seguito da "8" o "9" (per gli ISBN-10 e ISBN-13).
- Successivamente, deve seguire una sequenza di 9 cifre.

- Può terminare con una cifra o la lettera "X" (usata come cifra speciale per rappresentare 10).

In sostanza, questa definizione di dominio "ISBN" indica che l'attributo associato deve rappresentare un ISBN valido.

```
CREATE DOMAIN tipo_libro AS VARCHAR(10)
CHECK (VALUE IN ('Romanzo', 'Didattico'));
```

Il dominio "tipo_libro" viene definito come un dato di tipo VARCHAR al più di 10 caratteri. La clausola check verifica che il valore dell'attributo sia 'Romanzo' o 'Didattico'. Tale dominio verrà utilizzato per l'attributo **tipo** della tabella **Libro**.

```
CREATE DOMAIN modalita_fruizione AS VARCHAR(10)
CHECK (VALUE IN ('Cartaceo', 'Digitale', 'Audiolibro'));
```

Il dominio "modalita_fruizione" viene definito come un dato VARCHAR al più di 10 caratteri. La clausola check verifica che il valore dell'attributo sia "Cartaceo", "Digitale" o "Audiolibro". Tale dominio verrà utilizzato per l'attributo modalitaFruizione delle tabelle **DisponibilitaP** e **DisponibilitaL**.

```
CREATE DOMAIN modalita_fruizione_libro AS VARCHAR(50)
CHECK (VALUE LIKE '%Cartaceo%' OR VALUE LIKE '%Digitale%' OR VALUE LIKE '%Audiolibro%');
```

Il dominio "modalita_fruizione_libro" viene definito come un dato VARCHAR al più di 50 caratteri. La clausola check verifica che il valore dell'attributo contenga almeno uno tra "Cartaceo", "Digitale" e "Audiolibro". Tale dominio verrà utilizzato per definire l'attributo modalitaFruizione delle tabelle **Pubblicazione** e **Libro**.

5.2 Definizione Tabelle

```
1  /*DEFINIZIONE DELLA TABELLA CONFERENZA.*/
2  CREATE TABLE Conferenza(
3      CodConferenza serial,
4      DataInizio date NOT NULL,
5      DataFine date NOT NULL,
6      Luogo varchar(50) NOT NULL,
7      Responsabile varchar(50) NOT NULL,
8      StrutturaOrganizzatrice varchar(50) NOT NULL,
9
10     CONSTRAINT Data_conferenza CHECK (DataInizio < DataFine),
11     CONSTRAINT PK_Conferenza PRIMARY KEY (CodConferenza)
12 )
```

Il vincolo **Data_conferenza** impone che la data di inizio della conferenza sia precedente alla data fine.


```

1  /*DEFINIZIONE DELLA TABELLA SERIE.*/
2  CREATE TABLE Serie(
3      CodSerie serial,
4      Nome varchar(50) NOT NULL,
5      NumeroVolumi int NOT NULL,
6
7      CONSTRAINT PK_Serie PRIMARY KEY(CodSerie))
8
9
10 /*DEFINIZIONE DELLA TABELLA CANALE DISTRIBUZIONE.*/
11 CREATE TABLE CanaleDistribuzione(
12     CodCanale serial,
13     Nome varchar(50) NOT NULL,
14     TipoCanale tipo_canale NOT NULL,
15
16     URL URL,
17
18     Indirizzo varchar(50),
19     NumeroTelefono varchar(11) ,
20     OrarioApertura time,
21     OrarioChiusura time,
22
23     CONSTRAINT PK_canale_distribuzione PRIMARY KEY (CodCanale),
24     CONSTRAINT orario_libreria CHECK (OrarioApertura < OrarioChiusura),
25     CONSTRAINT CK_NumeroTelefono CHECK (NumeroTelefono ~ '[0-9]+$'))
26
27
28

```

Il vincolo **orario_libreria** impone che l'orario di apertura di una libreria preceda sempre l'orario di chiusura.

```

1  /*DEFINIZIONE DELLA TABELLA RIVISTA.*/
2  CREATE TABLE Rivista(
3      DOI varchar(100) NOT NULL ,
4      Nome varchar(30) NOT NULL,
5      Argomento varchar(50) NOT NULL,
6      AnnoPubblicazione int NOT NULL,
7      Responsabile varchar(30) NOT NULL,
8
9      CONSTRAINT PK_Rivista PRIMARY KEY (DOI))
10
11
12
13
14 /*DEFINIZIONE TABELLA SERIE*/
15 CREATE TABLE Preferiti(
16     Utente int NOT NULL,
17     Serie int NOT NULL,
18
19     CONSTRAINT PK_Preferiti PRIMARY KEY (Utente,Serie),
20
21     CONSTRAINT FK_Preferiti_Utente FOREIGN KEY (Utente) REFERENCES Utente(idUtente),
22
23     CONSTRAINT FK_Preferiti_Serie FOREIGN KEY (Serie) REFERENCES Serie(CodSerie)
24 )
25
26
27

```

```

1  /*DEFINIZIONE DELLA TABELLA UTENTE.*/
2  CREATE TABLE Utente (
3      idUtente serial,
4      Nome varchar(32) NOT NULL,
5      Cognome varchar(32) NOT NULL,
6      Username varchar(32) NOT NULL,
7      Pwd varchar(50) NOT NULL,
8
9
10     CONSTRAINT PK_Utente PRIMARY KEY(idUtente),
11
12     CONSTRAINT Username_Unique UNIQUE(Username),
13
14     CONSTRAINT CK_Password
15     CHECK (length(Pwd) >= 10 AND
16         Pwd ~ '[0-9]' AND
17         Pwd ~ '[a-z]' AND
18         Pwd ~ '[A-Z]'))
19

```

Il vincolo **CK_password** impone che il valore dell'attributo password contenga almeno: un numero, una lettera minuscola, una lettera maiuscola e che abbia una lunghezza almeno di 10 caratteri.

```

1  /*DEFINIZIONE DELLA TABELLA COLLANA */
2  CREATE TABLE Collana(
3      CodCollana serial,
4      Nome varchar(50) NOT NULL,
5      Editore varchar(32) NOT NULL,
6
7      CONSTRAINT PK_Collana PRIMARY KEY(CodCollana)
8  )
9

```

```

1  /*DEFINIZIONE TABELLA PUBBLICAZIONE*/
2  CREATE TABLE Pubblicazione (
3      ISBN ISBN NOT NULL,
4      Titolo varchar(50) NOT NULL,
5      Editore varchar(32) NOT NULL,
6      AnnoPubblicazione int NOT NULL,
7      ModalitaFruizione modalita_fruizione_libro NOT NULL,
8      Conferenza int,
9      Rivista varchar(100),
10
11     CONSTRAINT PK_Pubblicazione PRIMARY KEY(ISBN),
12
13     CONSTRAINT FK_Pubblicazione_Conferenza FOREIGN KEY (Conferenza) REFERENCES
14         Conferenza(CodConferenza),
15
16     CONSTRAINT FK_Pubblicazione_Rivista FOREIGN KEY(Rivista) REFERENCES Rivista(DOI)
17 )

```

```

1  /*DEFINIZIONE TABELLA LIBRO */
2  CREATE TABLE Libro (
3      ISBN ISBN NOT NULL,
4      Titolo varchar(50) NOT NULL,
5      Tipo tipo_libro NOT NULL,
6      Editore varchar(32) NOT NULL,
7      SalaPresentazione varchar(32),
8      DataUscita date NOT NULL,
9      Genere varchar(50),
10     Materia varchar(50),
11     AnnoPubblicazione int NOT NULL,
12     ModalitaFruizione modalita_fruizione_libro NOT NULL,
13     Serie int,
14
15     CONSTRAINT PK_Libro PRIMARY KEY(ISBN),
16     CONSTRAINT FK_Libro_Serie FOREIGN KEY(Serie) REFERENCES Serie(CodSerie))
17
18
19  /*CREAZIONE TABELLA DISPONIBILITAL*/
20  CREATE TABLE DisponibilitaL(
21      ISBN ISBN NOT NULL,
22      Canale int NOT NULL ,
23      ModalitaFruizione modalita_fruizione NOT NULL,
24
25     CONSTRAINT PK_DisponibilitaPL PRIMARY KEY (Canale, ISBN,ModalitaFruizione),
26
27     CONSTRAINT FK_DisPL_Libreria FOREIGN KEY (Canale) REFERENCES CanaleDistribuzione(
28         CodCanale),
29
30     CONSTRAINT FK_DisPL_Pubblicazione FOREIGN KEY (ISBN) REFERENCES Libro(ISBN)
31
32 )
33
34
35  /*CREAZIONE TABELLA DISPONIBILITAP*/
36  CREATE TABLE DisponibilitaP(
37      ISBN ISBN NOT NULL,
38      Canale int NOT NULL,
39      ModalitaFruizione modalita_fruizione NOT NULL,
40
41     CONSTRAINT PK_DisponibilitaLLibreria PRIMARY KEY(Canale,ISBN,ModalitaFruizione),
42
43     CONSTRAINT FK_DisPLL_Canale FOREIGN KEY (ISBN) REFERENCES Pubblicazione(ISBN),
44
45     CONSTRAINT FK_DisPLL_Libro FOREIGN KEY (Canale) REFERENCES CanaleDistribuzione(
46         CodCanale))
47
48
49  /*CREAZIONE TABELLA COMPONE */
50  CREATE TABLE Compone(
51      ISBN ISBN NOT NULL,
52      Collana int NOT NULL,
53
54     CONSTRAINT PK_Compone PRIMARY KEY (ISBN,Collana),
55     CONSTRAINT FK_Compone_Libro FOREIGN KEY (ISBN) REFERENCES Libro(ISBN),
56     CONSTRAINT FK_Compone_Collana FOREIGN KEY (Collana) REFERENCES Collana(CodCollana)
57 )
58
59

```

5.3 Trigger e Funzioni

```
1 CREATE OR REPLACE FUNCTION check_tipo_canale()
2 RETURNS TRIGGER AS
3 $$
4 BEGIN
5 IF NEW.tipocanale = 'Sito Online' THEN
6 -- Imposta gli attributi della Libreria a NULL
7 NEW.Indirizzo := NULL;
8 NEW.NumeroTelefono := NULL;
9 NEW.OrarioApertura := NULL;
10 NEW.OrarioChiusura := NULL;
11
12 -- Verifica che gli attributi del SitoOnline siano diversi da NULL
13 IF NEW.URL IS NULL THEN
14 RAISE EXCEPTION 'Gli attributi del Sito Online non possono essere NULL';
15 END IF;
16
17 ELSIF NEW.tipocanale = 'Libreria' THEN
18 -- Imposta gli attributi del SitoOnline a NULL
19 NEW.URL := NULL;
20
21 -- Verifica che gli attributi della Libreria siano diversi da NULL
22 IF NEW.NumeroTelefono IS NULL OR NEW.OrarioApertura IS NULL OR NEW.OrarioChiusura
23 IS NULL OR NEW.Indirizzo IS NULL THEN
24 RAISE EXCEPTION 'Gli attributi della Libreria non possono essere NULL';
25 END IF;
26
27 RETURN NEW;
28 END;
29 $$
30 LANGUAGE plpgsql;
31
32
33
34 CREATE TRIGGER before_insert_canale
35 BEFORE INSERT ON CanaleDistribuzione
36 FOR EACH ROW
37 EXECUTE FUNCTION check_tipo_canale();
38
```

Il trigger `before_insert_canale` e la funzione `check_tipo_canale()` assicurano che vengano rispettate le regole di integrità dei dati quando si inseriscono nuove tuple nella tabella `CanaleDistribuzione`. Se i valori degli attributi non corrispondono al tipo di canale specificato, vengono sollevate eccezioni per impedire l'inserimento di dati non validi.

```
1 ALTER TABLE Libro
2 ADD CONSTRAINT CK_Libro_Serie CHECK (
3 (Tipo = 'Didattico' AND Serie IS NULL) OR
4 (Tipo <> 'Didattico'));
5
```

Tale vincolo sulla tabella `Libro` verifica, come nel caso precedente, che vengano rispettate le regole di integrità dei dati. Più precisamente se un libro è un **Libro Didattico** l'attributo `Serie` deve essere posto a `NULL` poichè non essendo un **Romanzo** non può appartenere a **Serie**.

```

1 ALTER TABLE libro
2 ADD CONSTRAINT chk_tipo
3 CHECK (
4 (tipo = 'Romanzo' AND Materia IS NULL AND Genere IS NOT NULL)
5 OR (tipo = 'Didattico' AND Genere IS NULL AND Materia IS NOT NULL)
6 );
7

```

Il vincolo **chk_tipo** controlla che i valori delle colonne 'Materia' e 'Genere' nella tabella **Libro** siano coerenti con il valore della colonna **Tipo**.

- Se il valore di 'tipo' è "Romanzo", allora 'Materia' deve essere vuota (NULL) e 'Genere' non può essere vuota.
- e il valore di 'tipo' è "Didattico", allora 'Genere' deve essere vuota (NULL) e 'Materia' non può essere vuota.

```

1 ALTER TABLE Pubblicazione
2 ADD CONSTRAINT CK_RivistaConferenzaNotNull
3 CHECK (
4 ( Conferenza IS NOT NULL AND Rivista IS NULL) OR
5 ( Conferenza IS NULL AND Rivista IS NOT NULL) OR
6 (Conferenza IS NULL AND Rivista IS NULL)
7 );
8

```

Tale vincolo gestisce l'esclusività riguardo la divulgazione di una **Pubblicazione** che può esser effettuata mediante una **Rivista** o mediante una **Conferenza**.

```

1 CREATE OR REPLACE FUNCTION chk_libro()
2 RETURNS TRIGGER AS $$
3 BEGIN
4 IF (NOT EXISTS (
5 SELECT 1 FROM libro
6 WHERE libro.ISBN = NEW.ISBN
7 AND libro.ModalitaFruizione LIKE '%' || NEW.ModalitaFruizione || '%')
8 )) THEN
9 RAISE EXCEPTION 'La modalita di fruizione selezionata non \e disponibile per il
10 libro selezionato.';
11 END IF;
12 RETURN NEW;
13 END;
14 $$ LANGUAGE plpgsql;
15
16 CREATE TRIGGER chk_DisponibilitaL
17 AFTER INSERT OR UPDATE ON DisponibilitaL
18 FOR EACH ROW
19 EXECUTE FUNCTION chk_libro();

```

I trigger e la funzione verificano se la modalità di fruizione selezionata per un libro nella tabella **DisponibilitaL** è effettivamente disponibile per quel libro nella tabella **Libro**. Se non lo è, viene sollevata un'eccezione per impedire l'inserimento o l'aggiornamento del record.

```

1 CREATE OR REPLACE FUNCTION chk_pubblicazione()
2 RETURNS TRIGGER AS $$
3
4 BEGIN
5 IF (NOT EXISTS (
6 SELECT 1 FROM pubblicazione
7 WHERE pubblicazione.ISBN = NEW.ISBN
8 AND Pubblicazione.ModalitaFruizione LIKE '%' || New.ModalitaFruizione || '%'
9 )) THEN
10 RAISE EXCEPTION 'La modalita di fruizione selezionata non e disponibile per la
    pubblicazione selezionata.';
11
12 END IF;
13 RETURN NEW;
14 END;
15 $$ LANGUAGE plpgsql;
16
17 CREATE TRIGGER chk_DisponibilitaP
18 AFTER INSERT OR UPDATE ON DisponibilitaP
19 FOR EACH ROW
20 EXECUTE FUNCTION chk_pubblicazione();
21

```

Come nel caso precedente il trigger e la funzione verificano che non possa accadere che una **Pubblicazione** sia disponibile (presente nella tabella **DisponibilitaP**) in un formato per il quale non esiste. Se non lo è, viene sollevata un'eccezione per impedire l'inserimento o l'aggiornamento del record.

```

1 CREATE OR REPLACE FUNCTION trova_Librerie_Siti(utente_input int)
2 RETURNS TABLE (Nome VARCHAR(50), Cod int, Tipo tipo_canale) AS $$
3
4 BEGIN
5
6 RETURN QUERY (
7 SELECT cd.nome, cd.codCanale, cd.tipocanale
8 FROM (
9 SELECT pref.Serie, s.NumeroVolumi
10 FROM preferiti pref
11 JOIN serie s ON pref.Serie = s.CodSerie
12 WHERE pref.Utente = utente_input
13 ) AS v
14 JOIN Libro AS l ON v.Serie = l.Serie
15 JOIN disponibilitaL AS disp ON l.ISBN = disp.ISBN
16 JOIN CanaleDistribuzione AS cd ON disp.Canale = cd.codCanale
17 GROUP BY cd.Nome, cd.CodCanale, v.Serie, v.NumeroVolumi
18 HAVING COUNT(*) = v.NumeroVolumi);
19
20 END;
21 $$ LANGUAGE plpgsql;
22

```

Tale funzione restituisce tutte le librerie o i siti in cui sono disponibili tutte le serie preferite dell'utente specificato in input.

```

1 CREATE OR REPLACE FUNCTION aggiorna_num_volumi_inserimento()
2 RETURNS TRIGGER AS $$
3 BEGIN
4 IF NEW.serie IS NOT NULL THEN
5 UPDATE serie
6 SET numerovolumi = numerovolumi + 1
7 WHERE CodSerie = NEW.serie;
8
9 END IF;
10 RETURN NEW;
11 END;
12 $$ LANGUAGE plpgsql;
13
14 CREATE TRIGGER trigger_aggiorna_num_volumi
15 BEFORE INSERT ON libro
16 FOR EACH ROW
17 EXECUTE FUNCTION aggiorna_num_volumi_inserimento();
18
19

```

La funzione e il trigger in questione gestiscono la ridondanza dell'attributo **numerovolumi** della tabella **Serie**, andando ad incrementare tale attributo quando viene inserita una nuova tupla all'interno della tabella **Libro** che referencia tale serie.

```

1 CREATE OR REPLACE FUNCTION aggiorna_num_volumi_cancellazione()
2 RETURNS TRIGGER AS $$
3 BEGIN
4 IF OLD.serie IS NOT NULL THEN
5 UPDATE serie
6 SET NumeroVolumi = NumeroVolumi - 1
7 WHERE CodSerie =OLD.serie;
8
9 END IF;
10 RETURN OLD;
11 END;
12 $$ LANGUAGE plpgsql;
13
14 CREATE TRIGGER trigger_aggiorna_num_volumi_cancellazione
15 BEFORE DELETE ON libro
16 FOR EACH ROW
17 EXECUTE FUNCTION aggiorna_num_volumi_cancellazione();
18
19

```

La funzione e il trigger in questione gestiscono la ridondanza dell'attributo **numerovolumi** della tabella **Serie**, andando ad decrementare tale attributo quando viene eliminata una nuova tupla all'interno della tabella **Libro** che referenziava tale serie.

```

1 CREATE OR REPLACE FUNCTION set_NumeroVolumi_to_zero()
2 RETURNS TRIGGER AS
3 $$
4 BEGIN
5 IF NEW.NumeroVolumi <> 0 THEN
6 NEW.NumeroVolumi := 0;
7 END IF;
8 RETURN NEW;
9 END;
10 $$
11 LANGUAGE plpgsql;

```

```
12
13  -- Creazione del trigger
14  CREATE TRIGGER trg_SetNumeroVolumiToZero
15  BEFORE INSERT ON serie
16  FOR EACH ROW
17  EXECUTE FUNCTION set_NumeroVolumi_to_zero();
18
```

La seguente funzione insieme al trigger non permette al momento dell'inserimento di una nuova serie di attribuire all'attributo **numerovolumi** un valore diverso da 0, in quanto, essendo l'attributo **numerovolumi** incrementato e decrementato in base agli inserimenti e alle cancellazioni delle tuple della tabella **Libro**, assegnare un valore iniziale diverso da 0 sarebbe sbagliato.