



UNIVERSITÀ DEGLI STUDI
DI NAPOLI FEDERICO II

SCUOLA POLITECNICA E DELLE SCIENZE DI BASE
DIPARTIMENTO DI INGEGNERIA ELETTRICA E
TECNOLOGIE DELL'INFORMAZIONE

CORSO DI LAUREA IN INFORMATICA

INSEGNAMENTO DI BASI DI DATI E SISTEMI
OPERATIVI

PROGETTAZIONE DI UNA BASE DI DATI RELAZIONALE
PER LA GESTIONE DI BIBLIOGRAFIE

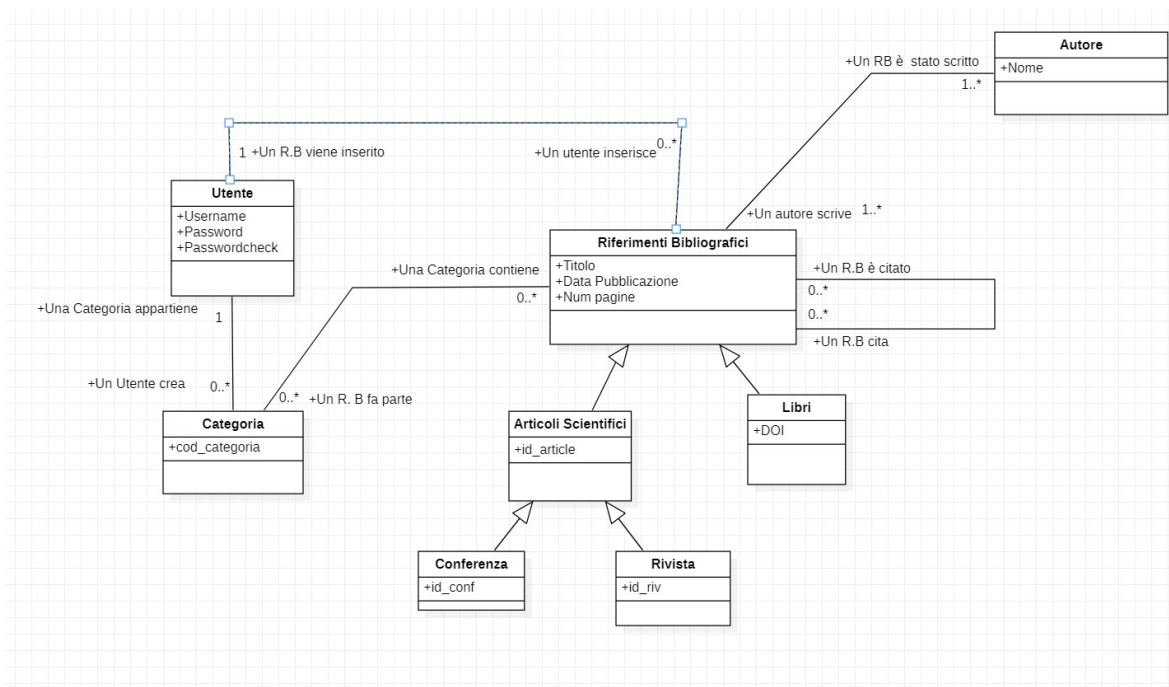
Emiliana Genzini N86002860
ANNO ACCADEMICO 2021-2022

1 Introduzione

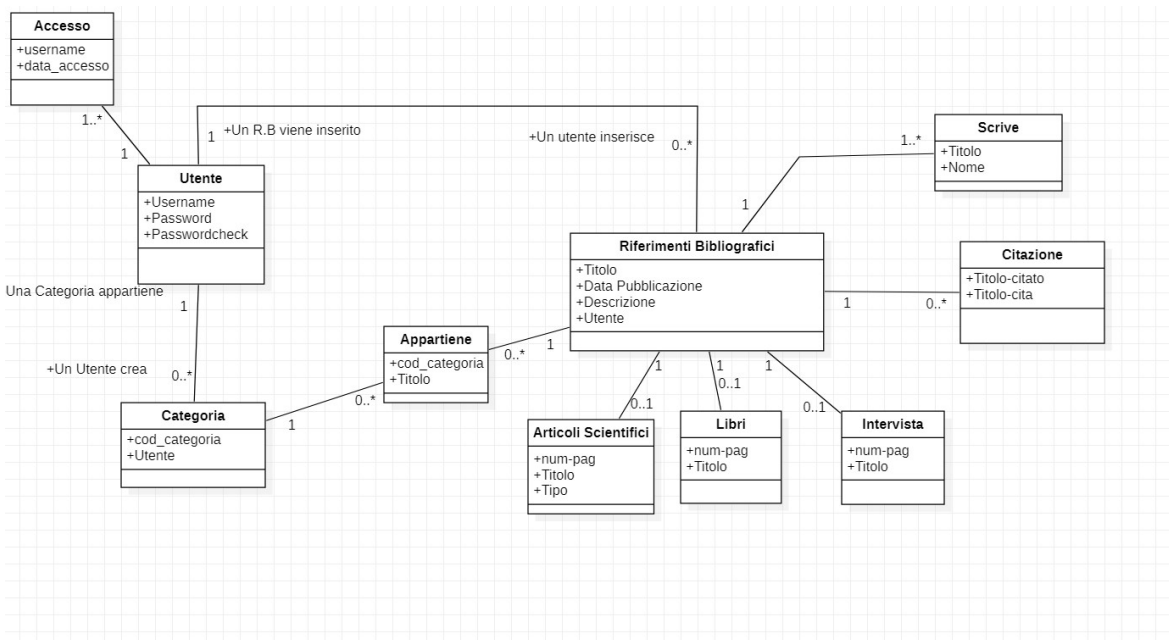
La base di dati sviluppata, gestita da un applicativo Java dotato di GUI, si occupa della gestione di bibliografie. L'utente, identificato dall'username, per accedere all'applicativo deve registrarsi. In fase di registrazione vengono fatti dei controlli per garantire la validità dei dati inseriti. Soltanto dopo la fase di registrazione l'utente potrà accedere all'applicativo, durante la fase di accesso i dati inseriti verranno confrontati con quelli precedentemente registrati, in caso di esito positivo si passerà alla schermata di home. In questa schermata sono disponibili le funzioni di inserimento e rimozione di riferimenti bibliografici divisi per tipo, inoltre è possibile aprire le sezioni "Profilo" e "Pubblicazioni". La sezione Profilo permette all'utente di creare ed eliminare categorie personalizzate e di inserire e rimuovere riferimenti bibliografici già presenti nel sistema, inoltre è possibile visualizzare tutte le categorie create dall'utente e gli elementi che fanno parte di una data categoria. La sezione Pubblicazioni è dedicata al filtraggio dei riferimenti bibliografici per tipo, per autore e per descrizione, inoltre è possibile ordinarli per il numero di citazioni ricevute sia in maniera crescente che decrescente. Ogni riferimento bibliografico inserito e ogni categoria creata vengono associati ad un determinato utente, ciò è possibile grazie ad una tabella "Accessi" presente nel database che permette di tenere traccia dell'ultimo utente ad aver effettuato l'accesso attraverso un campo data accesso che è di tipo "timestamp without time zone". Dalla schermata di home è possibile disconnettersi e ritornare alla schermata di accesso. La base di dati è stata modellata mediante Class Diagram realizzato con l'applicativo "StarUML" ed è stata gestita attraverso l'applicativo "PostgreSQL". La seguente documentazione è stata sviluppata in $L_A T_E X$.

2 Class Diagram

2.1 Non Ristrutturato



2.2 Ristrutturato



2.3 Modifiche di Ristrutturazione

Il class diagram presentato è composto da 8 classi, 4 delle quali sono specializzazioni della classe "Riferimenti Bibliografici" che rappresenta la parte principale dell'applicazione.

La classe Riferimenti Bibliografici è collegata in maniera diretta a tutte le altre classi presenti nel C.D.

E' necessario apportare delle modifiche di ristrutturazione poiché sono presenti specializzazioni e molteplicità (associazione molti a molti). Tutte le specializzazioni presenti sono Total - Disjoint in quanto rappresentano tutte le istanze separate possibili del caso principale. Essendo total - disjoint verranno sostituite con associazioni 1-0..1 ovvero un Riferimento Bibliografico o è un Libro o è un Articolo Scientifico. Nell'ultimo caso è presente un'ulteriore specializzazione, total - disjoint, in cui un articolo scientifico o è su una conferenza o è su una rivista. Nel Class Diagram ristrutturato è stata aggiunta la classe intervista pensata sempre come specializzazione della classe principale Riferimenti Bibliografici e le specializzazioni Rivista e Conferenza sono state rimosse ed è stato aggiunto il campo "Tipo" nella classe Articoli Scientifici che ,attraverso un vincolo di dominio, può assumere solo i valori 'Rivista' e 'conferenza'. E' stata inserita la classe "Accesso" che permette di tenere traccia dell'ultimo utente ad aver effettuato l'accesso attraverso un campo data accesso che è di tipo "timestamp without time zone". In questo modo è possibile determinare quale utente è svolge le operazioni di inserimento o rimozione di riferimenti bibliografici o categorie. Durante la ristrutturazione anche la classe Autore è stata rimossa ed è stata introdotta la classe Scrive

Per gestire le molteplicità presenti, in totale 3, verranno create delle tabelle apposite per ogni associazione e conterranno

le chiavi primarie delle tabelle che partecipano all'associazione.

Consideriamo le varie associazioni tra le classi:

·*RiferimentiBibliografici* — *Scrive* :

Un riferimento bibliografico può essere stato scritto da uno o più autori mentre la coppia Titolo-Autore può essere associata ad un unico riferimento bibliografico. Nella classe scrive sono presenti gli attributi Titolo che è una chiave esterna e fa riferimento alla chiave primaria della classe Riferimenti Bibliografici e il campo Autore che indica .

·*RiferimentiBibliografici* — *Categoria* :

Un riferimento bibliografico può appartenere a nessuna o più categorie e una categoria può contenere nessuno o più riferimenti bibliografici. Quest'associazione viene divisa in Riferimenti Bibliografici-Appartiene e Appartiene-Categoria in modo da poter avere due associazioni 1-0..* e quindi rispettare i vincoli di molteplicità richiesti.

L'associazione ricorsiva cita - è citato di Riferimenti Bibliografici è anch'essa un'associazione molti a molti in quanto un riferimento bibliografico può citare o nessun altro riferimento o altri riferimenti e una citazione può appartenere a un unico riferimento bibliografico. Per rispettare i vincoli di molteplicità richiesti verrà creata la tabella Cita collegata alla tabella Riferimenti Bibliografici attraverso un'associazione 1-0..*.

Inoltre i vincoli di ristrutturazione richiedono di modificare/eliminare attributi strutturati, in questo caso assenti già in principio.

3 Dizionario delle classi:

Nel Class Diagram ristrutturato sono presenti 10 classi. Le classi Utente , Categoria, Riferimenti Bibliografici, Scrive e Accessi sono entità forti e rappresentano gli aspetti principali dell'applicazione. L'utente è colui che svolge le operazioni principali, ovvero quelle di inserimento e rimozione di un Riferimento Bibliografico. Inoltre l'Utente può creare apposite directory (categorie) personalizzate. Le classi Citazione e Appartiene sono classi che derivano da associazioni molti a molti. Le classi Articoli Scientifici, Libri , Intervista sono specializzazioni.

Nella classe **Utente** sono presenti tre attributi: Username , Password e PasswordCheck. L'attributo Username è la chiave primaria mentre gli attributi Password e Passwordcheck sono usati in java in fase di accesso e registrazione.

Nella classe **Riferimenti Bibliografici** sono presenti cinque attributi: Titolo , Utente , Descrizione , Data Pubblicazione e Data Inserimento. L'attributo Titolo rappresenta la chiave primaria mentre Utente rappresenta la chiave esterna e fa riferimento alla chiave primaria della tabella Utente, username , ottenuto grazie alla tabella Accessi. L'attributo descrizione rappresenta a descrizione testuale in cui l'utente può indicare aspetti significativi. Il campo Data Inserimento ha il vincolo di essere uguale alla data corrente mentre il campo Data Pubblicazione ha il vincolo di non essere successiva alla Data di Inserimento

Nelle classi **Libri e Intervista** sono presenti due attributi : Titolo e num - pagine. L'attributo Titolo, in entrambi i casi, rappresenta la chiave esterna e fa riferimento alla chiave primaria della tabella Riferimenti Bibliografici mentre l'attributo num - pag è di tipo integer e fa da discriminante tra i vari tipi di riferimenti bibliografici, il suo dominio è controllato da specifici check.

Nella classe **Articolo Scientifico** oltre agli attributi Titolo e num - pagine è presente un terzo attributo "Tipo" il cui dominio è controllato da un check che gli permette di assumere i valori "Rivista" o "Conferenza".

Nella classe **Categoria** sono presenti due attributi: nome - categorie e Username. L'attributo Username è la chiave esterna e fa riferimento alla chiave primaria della tabella Utente, serve per tenere traccia di quale utente crea quale categoria. L'attributo nome - categoria è la chiave primaria. La classe **Appartiene** è formata da due chiavi esterne nome-categoria e Titolo che fanno rispettivamente riferimento alle chiavi primarie delle tabelle Categoria e Riferimenti Bibliografici. La classe Appartiene ci permette di tenere traccia delle componenti di ogni categoria. La classe **Accesso** permette di mantenere la cronologia di tutti gli accessi e di capire quale utente svolge determinate operazioni. E' composta da due attributi: username e data - accesso. L'attributo username è la chiave esterna e fa riferimento alla chiave primaria della tabella Utente. La classe **Scrivo** è formata da due attributi: Titolo e Nome. Il primo è una chiave esterna e fa riferimento alla tabella Riferimenti Bibliografici mentre l'attributo Nome si riferisce a quale autore ha scritto quel riferimento bibliografico. La classe **Citazione** ci permette di ordinare i riferimenti per numero di citazioni ricevute, ovvero per il numero di volte in cui il riferimento è presente nei rimandi di altri riferimenti, è formata da due attributi : Titolo - cita e Titolo - citato. Titolo - cita è una chiave esterna e fa riferimento alla tabella Riferimenti Bibliografici.

4 Dizionario delle associazioni:

- Utente-Categoria
- Utente-Riferimenti Bibliografici
- Utente-Accesso
- Categoria-Appartiene
- Appartiene-Riferimenti Bibliografici
- Riferimenti Bibliografici-Articoli Scientifici
- Riferimenti Bibliografici - Libri
- Riferimenti Bibliografici - Intervista
- Riferimenti Bibliografici - Citazione
- Riferimenti Bibliografici - Scrive

Analizziamo le più importanti :

4.1 Utente-Categoria:

L'associazione Utente-Categoria è un'associazione 1-0..*. Un Utente può definire nessuna o più categorie, mentre una categoria è specifica di un utente.

4.2 Utente-Accesso:

L'associazione Utente-Riferimenti Bibliografici è un'associazione 1-1..*. Ogni volta che un utente accede all'applicativo si inserisce una nuova riga nella tabella accesso, e ogni riga rimanda univocamente ad un dato accesso.

4.3 Utente-Riferimenti Bibliografici:

L'associazione Utente-Riferimenti Bibliografici è un'associazione 1-0..*. Un Utente può inserire nessuno o più riferimenti bibli-

ografici e un riferimento bibliografico può essere inserito da un unico utente

4.4 Riferimenti Bibliografici-Articoli Scientifici:

4.5 Riferimenti Bibliografici - Libri:

4.6 Riferimenti Bibliografici - Intervista:

Le associazioni 4.4, 4.5, 4.6 sono associazioni 1-0..1 . Un riferimento bibliografico o è un articolo Scientifico o è un libro oppure è un'intervista mentre un articolo scientifico , un libro e un'intervista sono riferimenti bibliografici

4.7 Riferimenti Bibliografici - Citazione:

L'associazione Riferimenti Bibliografici - Citazione è un'associazione 1-0..*. Un riferimento può essere associato a un insieme di rimandi, ovvero di altri riferimenti presenti nel sistema che vengono menzionati nel testo.

5 Schema Logico

- Utente(Username, Password, Passwordcheck)
- Categoria(nome-categoria,Utente)
- Appartiene(Titolo- Categoria,Titolo-Riferimento)
- Rif.Bibliografici(Titolo,Data Pubblicazione,Data Inserimento,
Utente)
- Articoli Scientifici(num-pag , Titolo)
- Libri(num-pag , Titolo)
- Intervista(num-pag, Titolo)
- Scrive(Titolo,Nome)
- Citazione(Titolo-citato,Titolo- cita)
- Accesso(Username,data - accesso)

6 Dizionario dei Vincoli:

Un database non deve solamente memorizzare i dati, ma garantire che i dati memorizzati siano corretti; se i dati sono imprecisi o incoerenti, l'integrità di tali dati può essere violata, compromettendo l'affidabilità del database. L'integrità dei dati, praticamente, impone delle restrizioni sui valori assunti da colonne e tabelle in un database.

I vincoli possono essere:

- Di dominio (Su un attributo di una tabella);
- Di n-upla (Su più attributi di una stessa tabella);
- Intrarelazionali (Unique, Primary key);
- Interrelazionali (Foreign key);

- Vincoli di dominio :

I vincoli CHECK impongono l'integrità dei dati limitando i valori che gli utenti possono immettere in una colonna. Nella base di dati definita sono presenti dei controlli per gli attributi num-pag nelle tabelle Articoli Scientifici , Libri e Intervista:

- Un Libro non può avere meno di 50 pagine e più di 1000
- Un Articolo Scientifico può avere un numero di pagine compreso tra 0 e 100.
- Un'intervista può avere un numero di pagine compreso tra 0 e 30.

Inoltre, è presente un altro controllo di validità sull'attributo Tipo della tabella Articoli Scientifici:

- Un Articolo Scientifico può essere o su una Rivista o su una Conferenza.

- Vincoli di n-upla :
Assenti.

- Vincoli Intrarelazionali:
- Ogni utente è identificato dall'username che funge da chiave primaria.
- Ogni Riferimento Bibliografico è identificato dal titolo che funge da chiave primaria.
- Ogni Categoria è identificata dal nome-categoria che funge da chiave primaria.

- Vincoli Interrelazionali:
- Un riferimento bibliografico può essere specializzato solo in un Libro o un Articolo Scientifico o un'Intervista. In queste tabelle è presente il campo Titolo che fa riferimento alla chiave primaria della tabella Riferimenti Bibliografici, funge quindi da chiave esterna.
- Ogni volta che un utente accede all'applicativo viene inserita una nuova riga nella tabella accesso, ogni riga quindi si riferisce ad un determinato utente. Il campo username nella tabella accessi fa riferimento al campo username nella tabella Utente, ha quindi funzione di chiave esterna.
- Ogni Categoria è composta da nessuno o più riferimenti bibliografici e ogni riferimento bibliografico può appartenere a nessuna o più categorie. Per rappresentare questa associazione è stata creata la tabella Appartiene composta dalle sole chiavi esterne Titolo e nome categoria che fanno rispettivamente riferimento alle chiavi primarie delle tabelle Riferimenti Bibliografici e Categoria.

- Ogni Riferimento bibliografico è inserito da un unico utente. Il campo Utente nella tabella Riferimenti bibliografici ha funzione di chiave esterna e fa riferimento alla chiave primaria della tabella Utente.

- La tabella Citazione è stata inserita come sviluppo di un'associazione ricorsiva molti a molti in quanto bisognava permettere al sistema di associare un riferimento a un insieme di rimandi, ovvero di altri riferimenti presenti nel sistema che vengono menzionati nel testo. Un riferimento bibliografico doveva poter citare altri riferimenti e essere citato da altri riferimenti, per questo motivo la tabella Citazione è formata da due chiavi esterne Titolo - Cita e Titolo - Citato che fanno entrambe riferimento alla chiave primaria della tabella Riferimenti Bibliografici.

- Nella classe Scrive è presente il campo Titolo che è una chiave esterna e fa riferimento alla chiave primaria della tabella Riferimenti Bibliografici.

7 SQL

In questa sezione sono riportati gli aspetti principali del codice SQL

7.1 Trigger e Procedure:

```
1  -- FUNCTION: public.currentdate()
2
3  -- DROP FUNCTION IF EXISTS public.currentdate();
4
5  CREATE OR REPLACE FUNCTION public.currentdate()
6      RETURNS trigger
7      LANGUAGE 'plpgsql'
8      COST 100
9      VOLATILE NOT LEAKPROOF
10 AS $BODY$
11 BEGIN
12     new.data_accesso := CURRENT_TIMESTAMP;
13     RETURN NEW;
14 END;
15 $BODY$;
16
17 ALTER FUNCTION public.currentdate()
18     OWNER TO postgres;
19
```

```
1  -- Trigger: currentdate
2
3  -- DROP TRIGGER IF EXISTS currentdate ON public.accessi;
4
5  CREATE TRIGGER currentdate
6      BEFORE INSERT
7      ON public.accessi
8      FOR EACH ROW
9      EXECUTE FUNCTION public.currentdate();
```

7.2 DAO:

```
public boolean checkriferimento(String titolo, JButton btnRimuovi) {
    ResultSet rs = null;
    String tit;
    Statement st;
    try {
        st = connection.createStatement();
        String check = "SELECT COUNT(\"Titolo\") AS num FROM"
            + " public.\"riferimenti_bibliografici \""
            + " WHERE \"Titolo\" = '" + titolo + "'";

        rs = st.executeQuery(check);
        if (rs != null) {
            while(rs.next()) {
                int contr = (rs.getInt("num"));
                System.out.println(contr);
                if(contr != 0) {

                    return true;

                } else {
                    JOptionPane.showMessageDialog(btnRimuovi, "Inserire un titolo valido");
                    return false;
                }
            }
        }
    }
}
```

```
33 public String getUtente() {
34
35     ResultSet rs = null;
36     Statement st;
37     try {
38         st = connection.createStatement();
39         String check = " SELECT username FROM public.accessi "
40             + "ORDER BY data_accesso DESC LIMIT 1";
41         rs = st.executeQuery(check);
42         if (rs != null) {
43             while(rs.next()) {
44                 String utente = (rs.getString("username"));
45                 return utente;
46             }
47         }
48     } catch (SQLException e) {
49         // TODO Auto-generated catch block
50         e.printStackTrace();
51     }
52     return null;
53
54
55 }
```

```

572•public void filtradescrizione (JList list , String descrizione) {
573    try {
574        String sql ="SELECT  \"Titolo\", \"Descrizione\", \"Utente\", \"
575        + \"data_inserimento, data_publicazione FROM public.\"\"riferimenti_bibliografici \"
576        + \"\" WHERE \"Descrizione\" LIKE '%" + descrizione + "%'";
577        PreparedStatement pst = connection.prepareStatement(sql);
578        ResultSet rs = pst.executeQuery();
579        DefaultListModel DLM = new DefaultListModel();
580
581        while(rs.next()) {
582            DLM.addElement(rs.getString("titolo"));
583            DLM.addElement(rs.getString("descrizione"));
584        }
585        list.setModel(DLM);
586
587    } catch (SQLException e) {
588        // TODO Auto-generated catch block
589        e.printStackTrace();
590    }
591 }
592 }

```

```

•public void ordinadecrescente(JList list) {
    try {
        String sql = "SELECT \"Titolo_Citato\"\\r\\n\" +
        \" FROM public.\"\"Citazione\" \"
        + \" GROUP BY \"\"Titolo_Citato\"\"
        + \"ORDER BY COUNT(\"\"Titolo_Citato\") DESC\";
        PreparedStatement pst = connection.prepareStatement(sql);
        ResultSet rs = pst.executeQuery();
        DefaultListModel DLM = new DefaultListModel();

        while(rs.next()) {
            DLM.addElement(rs.getString("Titolo_Citato"));
        }
        list.setModel(DLM);
    } catch (SQLException e) {
        // TODO Auto-generated catch block
        e.printStackTrace();
    }
}

```