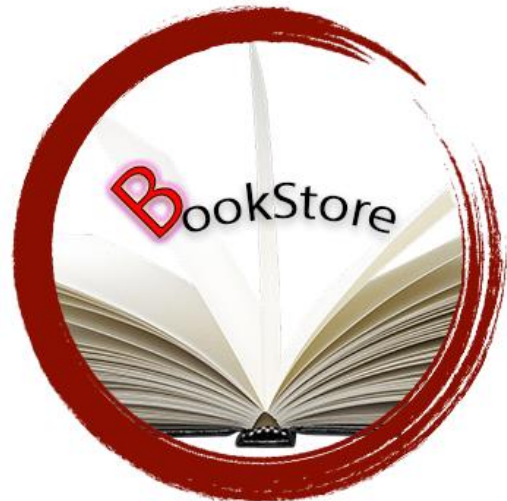




Laurea Magistrale in informatica-Università di Salerno
Corso di Gestione dei Progetti Software- Prof. Andrea DE LUCIA



Database Design Document

Riferimento	
Versione	1.0
Data	08/03/2021
Destinatario	Prof Andrea DE LUCIA
Presentato da	Vincenzo Raia (VR) - 0512106140 Raffaele Scarpa (RS) - 0512105708 Giuseppe Pepe (GP) - 0512105930



Revision History

Data	Versione	Descrizione	Autori
14/01/2020	0.1	Prima stesura	VR
08/03/2020	1.0	Revisione Finale	VR



Sommario

Revision History	2
1. Scelta del DBMS.....	4
2. Progettazione del database.....	5
2.1 Gestione dati persistenti	5
2.1.1 Modello ER iniziale	5
2.1.2 Modello ER ristrutturato	6
2.1.3 Dizionario dei dati.....	6
2.1.4 Tavola dei volumi.....	7
2.1.5 Tavola delle operazioni.....	7
2.1.6 Tavole degli accessi.....	8
2.1.6 Modello logico	9
3.Descrizione del database.....	10
Account.....	10
Indirizzo	10
Ordine.....	10
Libro ordinato	11
Libro	11
Autore.....	11
Categoria	11
4.Codice di creazione de database	12



1. Scelta del DBMS

Il problema della persistenza nasce dalla necessità di rendere permanenti le informazioni inserite nel sistema anche quando questo è spento.

Per tenere traccia dei dati utili anche quando l'applicazione che li ha creati ha smesso di funzionare utilizziamo la tecnologia MySQL.

MySQL è il DBMS (DataBase Management System) Open Source più' diffuso e utilizzato al mondo. La sigla SQL presente all'interno del nome sta ad indicare che la tecnologia MySQL è in grado di interpretare le istruzioni del linguaggio SQL.

MySQL permette la creazione di database relazionali, ossia consente la conservazione dei dati in tabelle separate anziché in un'unica grande entità.

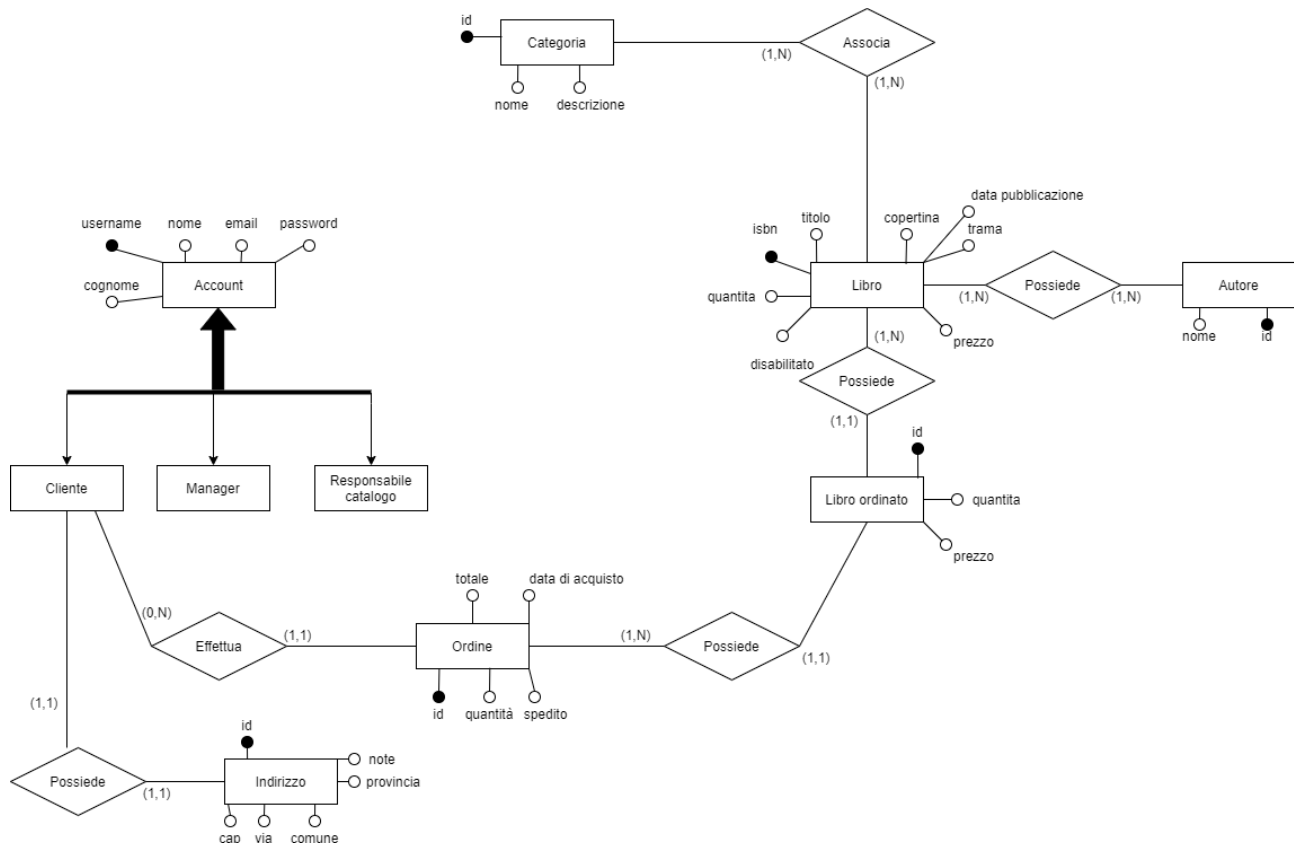
Questo permette di raggiungere un'ottima velocità e flessibilità di accesso ai dati ed una maggiore modellazione delle basi di dati.

2. Progettazione del database

2.1 Gestione dati persistenti

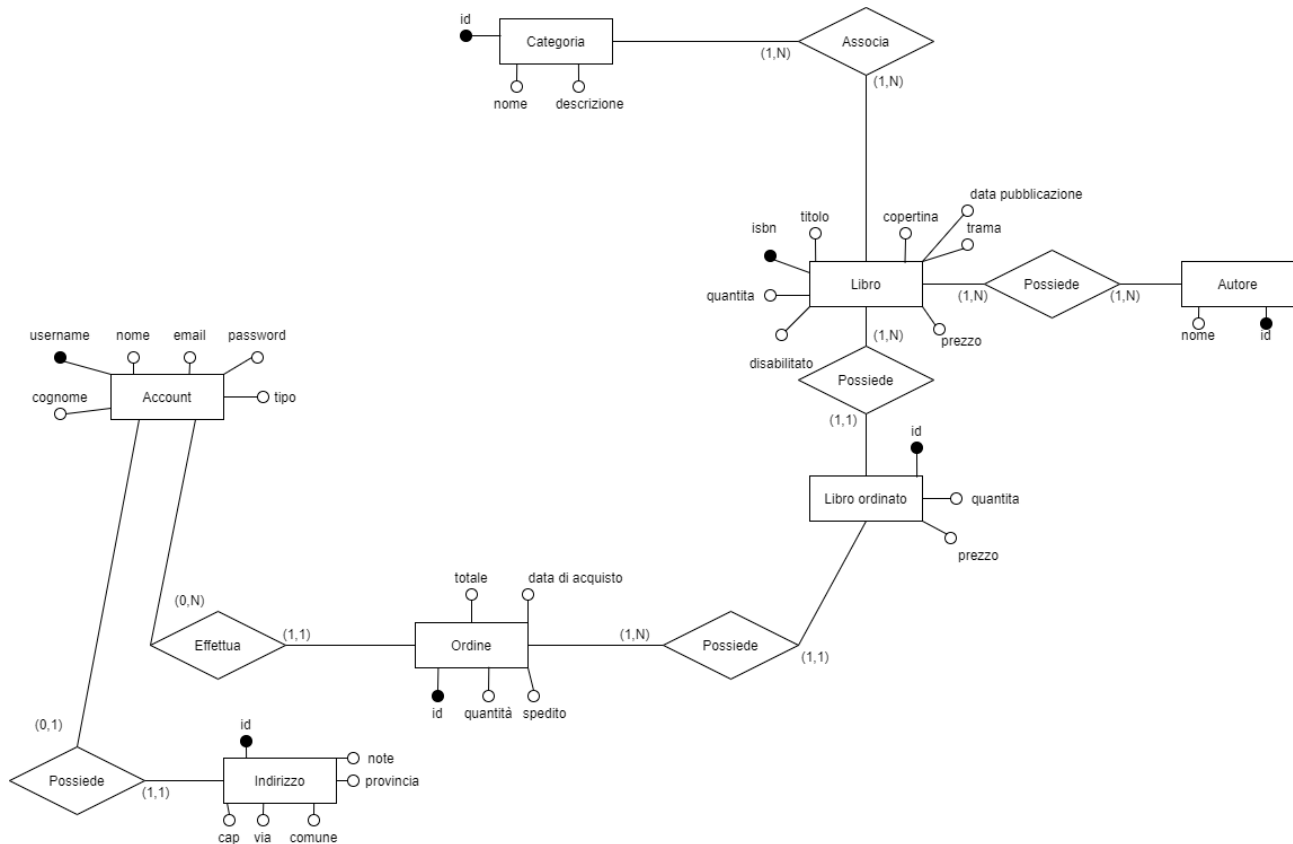
2.1.1 Modello ER iniziale

Di seguito è riportato il diagramma Entità-Relazioni (ER) del sistema BookStore. In una prima fase, verrà presentata la modellazione iniziale della base di dati. Questa sarà seguita dalla ristrutturazione, in modo da poter meglio definire le entità del sistema.



Di seguito è riportato il diagramma Entità-Relazioni (ER) ristrutturato del sistema “BookStore”

2.1.2 Modello ER ristrutturato



2.1.3 Dizionario dei dati

Entità	Descrizione	Attributi	Identificatori
Account	Dati di accesso per il sistema per gli utenti registrati e l'amministrazione	Nome,cognome, email, password, tipo	USERNAME
Indirizzo	Informazioni riguardo alla locazione di una cliente	Via, comune, provincia, cap, note	ID
Ordine	Informazioni riguardo all'ordine di un libro	Totale,data di acquisto,quantità,stato	ID
Libro ordinato	Informazioni riguardanti un libro ordinato da un cliente	Quantita,prezzo	ID
Libro	Informazioni riguardanti un libro presente nel sistema	Titolo, copertina, trama, quantità, disabilitato,prezzo,data pubblicazione	ISBN
Autore	Informazioni che riguardano lo scrittore del libro	Nome	ID



Categoria	Informazioni riguardanti una categoria presente nel sistema	Nome, descrizione	ID
-----------	---	-------------------	----

Relazione	Descrizione	Entità coinvolte	Attributi
Account/Indirizzo	Associa account a indirizzo	Account (0,1) Indirizzo (1,1)	
Account/Ordine	Associa account a ordine	Account (0,N) Ordine(1,1)	
Ordine/Libro ordinato	Associa ordine a libro ordinato	Ordine(1,N) Libro ordinato(1,1)	
Libro/ Libro ordinato	Associa libro a libro ordinato	Libro(1,N) Libro ordinato(1,1)	
Libro/Autore	Associa libro a autore	Libro(1,N) Autore(1,N)	
Libro/Categoria	Associa libro a categoria	Libro(1,N) Categoria(1,N)	

2.1.4 Tavola dei volumi

Tavola dei volumi		
Concetto	Tipo	Volume
Account	E	1000
Libro	E	100
Categoria	E	200
Ordine	E	1000
Autore	E	1000
Libro ordinato	E	1000
Account/Indirizzo	R	100
Account/Ordine	R	1000
Ordine/Libro ordinato	R	1000
Libro/ Libro ordinato	R	1000
Libro/Autore	R	100
Libro/Categoria	R	100

2.1.5 Tavola delle operazioni

Tavola delle operazioni		
Operazione	Tipo	Frequenza
Op1	I	2/settimana
Op2	I	1/mese
Op3	I	2/giorno
Op4	I	1/mese
Op5	I	1/settimana
Op6	I	1/anno
Op7	I	1/mese
Op8	I	2/giorno
Op9	I	2/anno



Glossario operazioni

- Op1. Aggiungere un libro nel database
- Op2. Modificare un libro nel database
- Op3. Visualizzare delle informazioni di un libro scelto
- Op4. Visualizzare dei dati legati agli utenti registrati
- Op5. Ricerca di tutti gli ordini di uno specifico cliente
- Op6. Modificare i dati di un utente registrato
- Op7. Registrazione di cliente nella piattaforma
- Op8. Visualizzazione di una categoria
- Op9. Modificare l'indirizzo di spedizione di un cliente

2.1.6 Tavole degli accessi

OP1			
Concetto	Costrutto	Accessi	Tipo
Libro	E	1	S
Libro/Autore	R	1	S
Autore	E	1	S
Costo OP1:12 accessi/settimana			

OP2			
Concetto	Costrutto	Accessi	Tipo
Libro	E	1	L
Libro	E	1	S
Libro/Autore	R	1	L
Autore	E	1	L
Autore	E	1	S
Costo OP2:7 accessi/mese			

OP3			
Concetto	Costrutto	Accessi	Tipo
Libro	E	1	L
Libro/Autore	R	1	L
Autore	E	1	L
Costo OP3:6 accessi/giorno			

OP4			
Concetto	Costrutto	Accessi	Tipo
Account	E	1000	L
Costo OP4:1000 accessi/mese			

OP5			
Concetto	Costrutto	Accessi	Tipo
Account	E	1	L
Account/Indirizzo	R	1	L
Indirizzo	E	1	L



Account/Ordine	R	1	L
Ordine	E	1	L
Costo OP5:5 accessi/settimana			

OP6			
Concetto	Costrutto	Accessi	Tipo
Account	E	1	L
Account	E	1	S
Costo OP6:3 accessi/anno			

OP7			
Concetto	Costrutto	Accessi	Tipo
Account	E	1	S
Account/Indirizzo	R	1	S
Indirizzo	E	1	S
Costo OP7: 6 accessi/mese			

OP8			
Concetto	Costrutto	Accessi	Tipo
Categoria	E	1	L
Categoria/Libri	R	1	L
Libri	E	1	L
Costo OP8:6 accessi/giorno			

OP9			
Concetto	Costrutto	Accessi	Tipo
Account	E	1	L
Account	E	1	S
Account/Indirizzo	R	1	S
Indirizzo	E	1	L
Indirizzo	E	1	S
Costo OP9: 16 accessi/anno			

2.1.6 Modello logico

Account(**username**,nome,cognome,email,password,tipo)

Indirizzo(**id**,note,provincia,cap,via,comune,username)

Ordine(**id**,totale,quantita,datadiacquisto,username,spedito)

Libroordinato(**id** ,quantita,prezzo,isbn,id.ordine)

Libro(**isbn**,quantita,disabilitato,titolo,copertina,trama,prezzo,datapubblicazione)

Libro_Autore(**isbn**,**id**)

Autore(**id**,nome)



Categoria(**id**,nome,descrizione)

Libro_Categoria(**isbn**,**id**)

3.Descrizione del database

Account

Colonna	Tipo	Null	Commenti
username	varchar(32)	NO	Identifica un account dall'altro.
nome	varchar(128)	NO	Indica la nome dell'account.
cognome	varchar(128)	NO	Indica la cognome dell'account.
email	varchar(128)	NO	Indica la mail dell'account.
password	varchar(32)	NO	Indica la password dell'account.
tipo	varchar(1)	NO	indica il tipo dell'account.
sex	varchar(1)	NO	indica il sesso dell'account.

Indirizzo

Colonna	Tipo	Null	Commenti
id	int	NO	Identifica un indirizzo dall'altro.
via	varchar(128)	NO	Indica la via di un indirizzo.
comune	varchar(128)	NO	Indica il comune di un indirizzo.
provincia	varchar(128)	NO	Indica la provincia di un indirizzo.
cap	int	NO	Indica il cap di un indirizzo.
note	text	SI	Indica le note da lasciare al corriere di un indirizzo.

Ordine

Colonna	Tipo	Null	Commenti
id	int	NO	Identifica un ordine dall'altro.
quantity	int	NO	Indica la quantità di un ordine.



totale	float	NO	Indica il totale di un ordine.
data di acquisto	date	NO	Indica la data di acquisto di un ordine.
spedito	boolean	NO	Indica l'ordine è stato spedito o meno

Libro ordinato

Colonna	Tipo	Null	Commenti
id	int	NO	Identifica un libro ordinato dall'altro.
quantita	int	NO	Indica la quantità di un libro ordinato.
prezzo	float	NO	Indica il prezzo di un libro ordinato.

Libro

Colonna	Tipo	Null	Commenti
isbn	varchar(13)	NO	Identifica un libro dall'altro.
titolo	varchar(100)	NO	Indica il titolo di un libro.
trama	varchar(500)	NO	Indica la trama di un libro.
prezzo	float	NO	Indica il prezzo di un libro.
disabilitato	boolean	NO	Indica se un libro è disabilitato o meno.
copertina	varchar(400)	NO	Indica il percorso di una copertina di un libro
quantita	int	NO	Indica la quantità di un libro
datadipubblicazione	date	NO	Indica la data di pubblicazione di un libro

Autore

Colonna	Tipo	Null	Commenti
id	int	NO	Identifica un autore dall'altro.
nome	varhchar(100)	NO	Indica il nome di un autore

Categoria

Colonna	Tipo	Null	Commenti
---------	------	------	----------



id	int	NO	Identifica una categoria dall'altra.
nome	varchar(100)	NO	Indica il nome di una categoria
descrizione	text	NO	Indica la descrizione di una categoria.

4.Codice di creazione de database

```
-- Author Vincenzo Raia
```

```
DROP DATABASE IF EXISTS bookstoreDB;
```

```
CREATE DATABASE bookstoreDB;
```

```
USE bookstoreDB;
```

```
-- Database: `bookStoreDB`
```

```
-- Table structure for table `Account`
```

```
CREATE TABLE `account`
```

```
(
```

```
  `username` varchar(32) NOT NULL,
```

```
  `password` varchar(32) NOT NULL,
```

```
  `nome`   varchar(128) NOT NULL,
```

```
  `cognome` varchar(128) NOT NULL,
```

```
  `email`  varchar(128) NOT NULL,
```

```
  `tipo`   varchar(128) NOT NULL,
```

```
  `abilitato` boolean default true,
```

```
  PRIMARY KEY (`username`)
```

```
);
```

```
-- Table structure for table `Indirizzo`
```

```
CREATE TABLE `indirizzo`
```

```
(
```

```
  `id`   int      NOT NULL AUTO_INCREMENT,
```

```
  `via`  varchar(128) NOT NULL,
```



```
`comune` varchar(128) NOT NULL,  
`provincia` varchar(128) NOT NULL,  
`cap` int NOT NULL,  
`note` text,  
`username` varchar(32),  
FOREIGN KEY (`username`) REFERENCES account (`username`) on update cascade on delete cascade,  
PRIMARY KEY (`id`)  
);
```

-- Table structure for table `Libro`

```
CREATE TABLE `libro`  
(  
  `ISBN` varchar(14),  
  `prezzo` float not null,  
  `quantita` int not null,  
  `trama` text not null,  
  `titolo` varchar(100) not null,  
  `copertina` varchar(400) not null unique,  
  `disabilitato` float not null,  
  `datapubblicazione` date not null,  
  FULLTEXT KEY (`titolo`),  
  FULLTEXT KEY (`titolo`, `trama`),  
  primary key (`ISBN`)  
);
```

-- Table structure for table `Ordine`

```
CREATE TABLE `ordine`  
(  
  `id` int AUTO_INCREMENT,  
  `quantita` int NOT NULL,  
  `totale` float NOT NULL,  
  `datadiacquisto` date NOT NULL,
```



```
`username`    varchar(32) NOT NULL,
`spedito`     boolean default false,
PRIMARY KEY (`id`),
FOREIGN KEY (`username`) REFERENCES account (`username`) on update cascade on delete cascade
);
-- Table structure for table `Libro ordinato`
CREATE TABLE `libroordinato`
(
  `id`      int    NOT NULL AUTO_INCREMENT,
  `quantita` int    NOT NULL,
  `prezzo`  float  NOT NULL,
  `ISBN`    varchar(14),
  `idordine` int   not null,
  PRIMARY KEY (`id`),
  FOREIGN KEY (`ISBN`) REFERENCES libro (`ISBN`) on update cascade on delete cascade,
  FOREIGN KEY (`idordine`) REFERENCES ordine (`id`) on update cascade on delete cascade
);
-- Table structure for table `Autore`
CREATE TABLE `autore`
(
  `id`      int    NOT NULL AUTO_INCREMENT,
  `nomecompleto` varchar(100) NOT NULL,
  PRIMARY KEY (`id`)
);
-- Table structure for table `Categoria`
CREATE TABLE `categoria`
(
  `id`      int    NOT NULL AUTO_INCREMENT,
  `nome`     varchar(100) NOT NULL unique,
  `descrizione` text    NOT NULL,
```



PRIMARY KEY (`id`)

);

-- Table structure for relation `Libro` and `Autore`

CREATE TABLE `libroautore`

(

 `ISBN` varchar(14),

 `id` int,

 primary key (`id`, `isbn`),

 FOREIGN KEY (`ISBN`) REFERENCES libro (`ISBN`) on update cascade on delete cascade,

 FOREIGN KEY (`id`) REFERENCES autore (`id`) on update cascade on delete cascade

);

-- Table structure for relation `Libro` and `Categoria`

CREATE TABLE `librocategoria`

(

 `ISBN` varchar(14),

 `id` int NOT NULL,

 PRIMARY KEY (`ISBN`,`id`),

 FOREIGN KEY (`id`) REFERENCES categoria (`id`) on update cascade on delete cascade,

 FOREIGN KEY (`ISBN`) REFERENCES libro (`ISBN`) on update cascade on delete cascade

);