

Esame di Stato Istituto Tecnico Industriale CORSO DI ORDINAMENTO Indirizzo: INFORMATICA

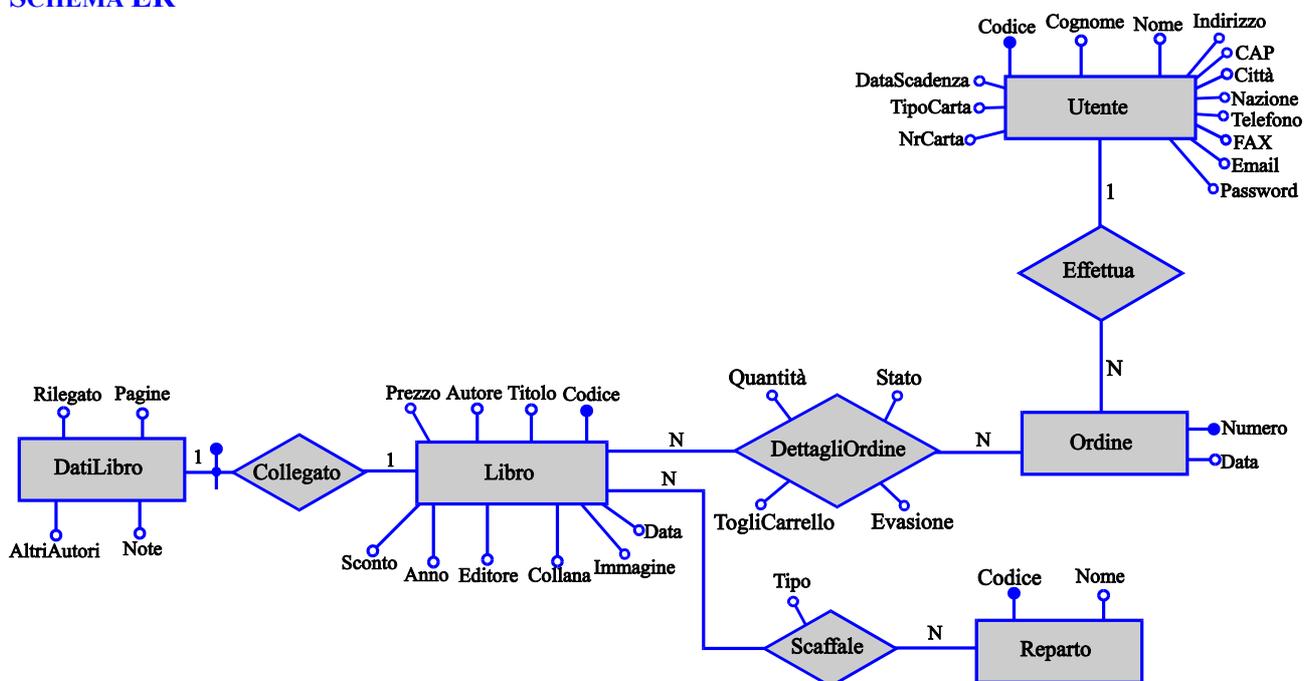
Proposta di soluzione per il tema di: INFORMATICA GENERALE E APPLICAZIONI TECNICO-SCIENTIFICHE

La progettazione di un database non ha un'unica soluzione. Quella proposta può essere quindi variata in base all'esperienza del programmatore.

■■■ PROGETTO CONCETTUALE

Nella stesura dello schema ER si è ipotizzato che, come avviene spesso nella realtà, i libri siano inseriti in più reparti virtuali (un libro può essere un fumetto, ma anche un fumetto di fantascienza). Un reparto virtuale è organizzato, a sua volta, in scaffali virtuali. Un libro può essere inserito in più scaffali virtuali, ad esempio, può essere un libro da non perdere e anche uno dei più venduti.

SCHEMA ER



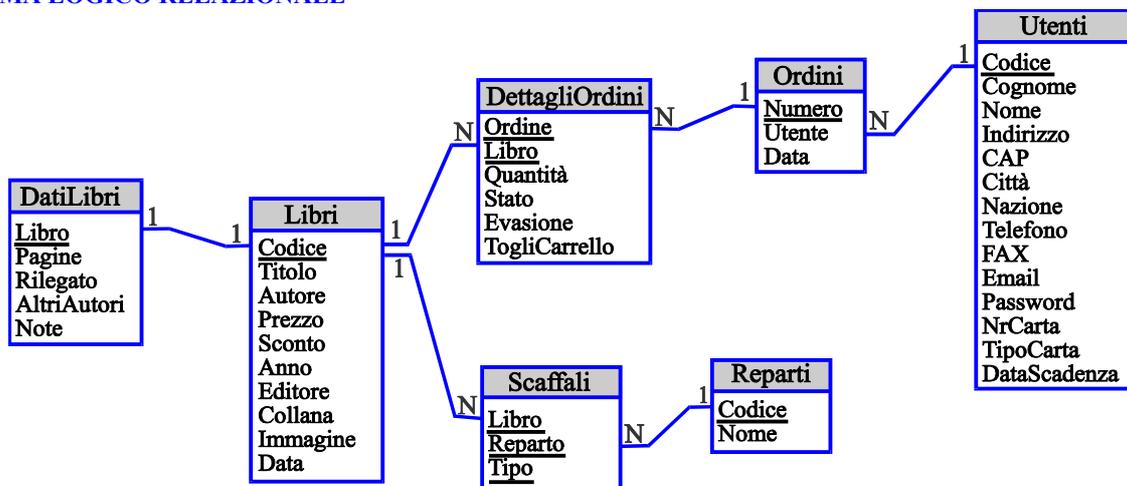
Lo schema ER deve seguire queste **regole**.

- R1.** L'attributo *Codice* di un *Libro* può assumere il formato L-XXXX.
- R2.** L'attributo *Immagine* di un *Libro* memorizza l'URL del file con l'immagine della copertina.
- R3.** L'attributo *Codice* di un *Reparto* può assumere il formato R-XXX.
- R4.** L'attributo *Tipo* di uno *Scaffale* può assumere solo i valori: "Da non perdere", "I più venduti", "Ultimi arrivi", "Offerte speciali", "Remainders".
- R5.** L'attributo *Codice* di un *Utente* può assumere il formato U-XXXX.
- R6.** L'attributo *Email* di un *Utente* rappresenta anche lo user name dell'account.
- R7.** L'attributo *Stato* di un *DettagliOrdine* indica lo stato dell'ordine (*Stato=FALSE*, libro non consegnato).
- R8.** L'attributo *Evasione* di un *DettagliOrdine* rappresenta il numero medio di giorni di attesa per la Consegna (o disponibilità).

Tutti i dati del carrello acquisti sono memorizzati nel database e quindi può essere costruito con più query.

■ ■ ■ REALIZZAZIONE

SCHEMA LOGICO RELAZIONALE



Lo schema logico relazionale **rispetta le forme normali** ed è soggetto a questi **vincoli di integrità referenziale**.

- V1. La chiave esterna *Libro* della tabella *DatiLibri* è in relazione con la tabella *Libri* mediante la chiave primaria *Codice*.
- V2. La chiave esterna *Libro* della tabella *DettagliOrdini* è in relazione con la tabella *Libri* mediante la chiave primaria *Codice*.
- V3. La chiave esterna *Ordine* della tabella *DettagliOrdini* è in relazione con la tabella *Ordini* mediante la chiave primaria *Codice*.
- V4. La chiave esterna *Utente* della tabella *Ordini* è in relazione con la tabella *Utenti* mediante la chiave primaria *Codice*.
- V5. La chiave esterna *Reparto* della tabella *Scaffali* è in relazione con la tabella *Reparti* mediante la chiave primaria *Codice*.
- V6. La chiave esterna *Libro* della tabella *Scaffali* è in relazione con la tabella *Libri* mediante la chiave primaria *Codice*.

CODIFICA

Per tradurre nel DBMS lo **schema logico del database** si deve eseguire (una sola volta) questo codice sorgente SQL.

```

CREATE DATABASE DBPortaleLibri

CREATE TABLE Libri
(
  Codice    CHAR(6) CHECK(Codice LIKE 'L-____'),
  Titolo    CHAR(50) NOT NULL,
  Autore    CHAR(60) NOT NULL,
  Prezzo    DECIMAL(7,3) NOT NULL,
  Sconto    DECIMAL(7,3) NOT NULL,
  Anno      INTEGER,
  Editore   CHAR(30),
  Collana   CHAR(30),
  Immagine  CHAR(50),
  Data      DATE,
  PRIMARY KEY(Codice)
)

CREATE TABLE DettagliLibri
(
  Libro     CHAR(6),

```

```
Pagine          INTEGER NOT NULL,
Rlegato         BIT,
AltriAutori    CHAR(50),
Note           CHAR(255),
PRIMARY KEY(Libro),
FOREIGN KEY(Libro) REFERENCES Libri(Codice)
)

CREATE TABLE Reparti
(
  Codice CHAR(5) CHECK(Codice LIKE 'R-____'),
  Nome   CHAR(20) NOT NULL,
  PRIMARY KEY(Codice)
)

CREATE TABLE Scaffali
(
  Libro   CHAR(6),
  Reparto CHAR(5),
  Tipo   CHAR(20) CHECK(Tipo IN ('Da non perdere', 'I più venduti',
                                'Ultimi arrivi', 'Offerte speciali', 'Remainders')) NOT NULL,
  PRIMARY KEY(Libro, Reparto, Tipo),
  FOREIGN KEY(Libro) REFERENCES Libri(Codice),
  FOREIGN KEY(Reparto) REFERENCES Reparti(Codice)
)

CREATE TABLE Utenti
(
  Codice      CHAR(6) CHECK(Codice LIKE 'U-____'),
  Cognome    CHAR(15) NOT NULL,
  Nome       CHAR(15) NOT NULL,
  Indirizzo  CHAR(60) NOT NULL,
  CAP        CHAR(5) NOT NULL,
  Città      CHAR(20) NOT NULL,
  Nazione    CHAR(20) NOT NULL,
  Telefono   CHAR(15) NOT NULL,
  FAX        CHAR(15) NOT NULL,
  Password   CHAR(14) NOT NULL,
  NrCarta    CHAR(16),
  TipoCarta  CHAR(10),
  DataScadenza DATE,
  PRIMARY KEY(Codice)
)

CREATE TABLE Ordini
(
  Numero INTEGER,
  Utente CHAR(6),
  Data   DATE NOT NULL,
  PRIMARY KEY(Numero),
  FOREIGN KEY(Utente) REFERENCES Utenti(Codice)
)

CREATE TABLE DettagliOrdini
(
  Ordine      INTEGER,
  Libro       CHAR(6),
  Quantità    INTEGER NOT NULL,
  Stato       BIT NOT NULL,
  Evasione    INTEGER NOT NULL,
  TogliCarrello BIT NOT NULL,
```

```
PRIMARY KEY(Ordine, Libro),  
FOREIGN KEY(Ordine) REFERENCES Ordini(Numero),  
FOREIGN KEY(Libro) REFERENCES Libri(Codice)  
)
```

Le **interrogazioni sul database** sono realizzate mediante questi codici sorgente SQL che devono essere eseguiti in un DBMS.

Interrogazione: ricerca per reparto

Numero libri per scaffale

```
SELECT COUNT(*) AS 'Numero di libri per scaffale virtuale'  
FROM Reparti, Scaffali, Libri  
WHERE Reparti.Codice = Scaffali.Reparto AND  
Scaffali.Libro = Libri.Codice AND  
Reparti.Nome = [Digita nome del reparto] AND  
Scaffali.Tipo = [Digita scaffale, ad esempio, "I più venduti"]
```

Dettagli libri per scaffale

```
SELECT Libri.*  
FROM Reparti, Scaffali, Libri  
WHERE Reparti.Codice = Scaffali.Reparto AND  
Scaffali.Libro = Libri.Codice AND  
Reparti.Nome = [Digita nome del reparto] AND  
Scaffali.Tipo = [Digita scaffale, ad esempio, "I più venduti"]
```

Interrogazione: ordini

```
SELECT DettagliOrdini.*  
FROM Utenti, Ordini, DettagliOrdini  
WHERE Utenti.Codice = Ordini.Utente AND  
Ordini.Numero = DettagliOrdini.Ordine AND  
DettagliOrdini.Stato = FALSE AND  
Utenti.Codice = [Digita il codice utente]
```