

Progetto del database di una rete di biblioteche



Lorenzo Gasparini

Università degli Studi di Trieste

7 giugno 2015

Analisi dei requisiti

- Requisiti espressi in linguaggio naturale
- Glossario dei termini
- Strutturazione dei requisiti

Requisiti espressi in linguaggio naturale

“ *Si vuole realizzare una base di dati per gestire la rete delle biblioteche della regione Friuli Venezia-Giulia.*
Ci sono diverse biblioteche in tutte le province della regione, ed esse possono fare parte di diversi gruppi (e.g. Biblioteche Universitarie, Biblioteche Scolastiche).
Le biblioteche gestiscono libri cartacei; lo stesso libro, identificato dal codice ISBN, si può trovare in diverse biblioteche e nella stessa biblioteca ci possono essere più copie dello stesso. Del libro si vogliono memorizzare anche il nome e l'anno.
Ogni libro ha una lista di autori e fa parte di una o più categorie. Gli autori sono caratterizzati da nome e cognome, mentre le categorie hanno solo un nome. ”

Requisiti espressi in linguaggio naturale

“ Per richiedere un prestito le persone si possono registrare come membri presso una qualsiasi biblioteca fornendo nome, cognome e data di nascita. Bisogna essere maggiorenni per registrarsi. I prestiti durano 30 giorni. Ogni membro può prendere in prestito al massimo 5 libri alla volta. Dei membri si vuole memorizzare inoltre lo stato dell'iscrizione (attiva, sospesa) e il numero di ammonizioni. Se si riconsegna un libro in ritardo, si viene ammoniti. Alla quarta ammonizione, l'iscrizione viene sospesa per 30 giorni. Si vuole inoltre rendere disponibile la lista dei 10 libri più noleggiati in assoluto ad un utente esterno.

”

Glossario dei termini

Termine	Descrizione	Sinonimi	Collegamenti logici
Libro	Libro come entità astratta, che può essere presente in diverse biblioteche delle regione, anche in più copie		Copia libro, Autore, Categoria
Autore	Persona che ha scritto un libro, anche insieme ad altri. Ogni libro ne ha almeno uno	Scrittore	Libro
Categoria	Categoria di un libro. Una categoria può avere molti libri e un libro può appartenere a molte categorie	Genere	Libro
Copia libro	Copia fisica di un libro presente in una biblioteca, data in prestito ai membri		Libro, Membro, Biblioteca
Membro	Persona iscritta che può richiedere libri in prestito	Iscritto	Indirizzo, Prestito

Glossario dei termini

Termine	Descrizione	Sinonimi	Collegamenti logici
Biblioteca	Biblioteca situata presso un certo indirizzo		Indirizzo, Copia libro, Gruppo
Gruppo	Insieme di biblioteche accomunate da una certa caratteristica		Biblioteca
Indirizzo	Indicazione di posizione geografica di un luogo		Membro, Biblioteca
Comune	Ente territoriale di base		Indirizzo, Provincia
Provincia	Ente locale con competenza su un gruppo di comuni		Comune
Prestito	Noleggio da parte di un membro di una copia di un libro presso una biblioteca	Noleggio	Membro, Copia libro

Strutturazione dei requisiti

- Frasi di carattere generale:
 - Si vuole realizzare una base di dati per gestire la rete delle biblioteche della regione Friuli Venezia-Giulia
- Frasi relative alle biblioteche e ai gruppi:
 - Ci sono diverse biblioteche in tutte le province della regione, ed esse possono fare parte di diversi gruppi (e.g. Biblioteche Universitarie, Biblioteche Scolastiche)
- Frasi relative ai libri e alle copie:
 - Le biblioteche gestiscono libri cartacei; lo stesso libro, identificato dal codice ISBN, si può trovare in diverse biblioteche e nella stessa biblioteca ci possono essere più copie dello stesso. Del libro si vogliono memorizzare anche il nome e l'anno
- Frasi relative agli autori e alle categorie:
 - Ogni libro ha una lista di autori e fa parte di una o più categorie. Gli autori sono caratterizzati da nome e cognome, mentre le categorie hanno solo un nome

Strutturazione dei requisiti

- Frasi relative ai membri:
 - Per richiedere un prestito le persone si possono registrare come membri presso una qualsiasi biblioteca fornendo nome, cognome e data di nascita. Bisogna essere maggiorenni per registrarsi
- Frasi relative ai prestiti:
 - I prestiti durano 30 giorni. Ogni membro può prendere in prestito al massimo 5 libri alla volta. Dei membri si vuole memorizzare inoltre lo stato dell'iscrizione (attiva, sospesa) e il numero di ammonizioni
 - Se si riconsegna un libro in ritardo, si viene ammoniti. Alla quarta ammonizione, l'iscrizione viene sospesa per 30 giorni
- Frasi aggiuntive:
 - Si vuole inoltre rendere disponibile la lista dei 10 libri più noleggiati in assoluto ad un utente esterno

Progettazione concettuale

- Analisi delle entità e delle relazioni
- Modello ER
- Vincoli non esprimibili

Analisi delle entità e delle relazioni

È stata scelta la metodologia di progetto *bottom-up*.

Questa metodologia consiste nello sviluppare singolarmente in dettaglio le diverse parti del database, per poi connetterle.

Vengono quindi presentate di seguito le parti che compongono il modello ER, analizzandone gli attributi insieme alla cardinalità delle relazioni.

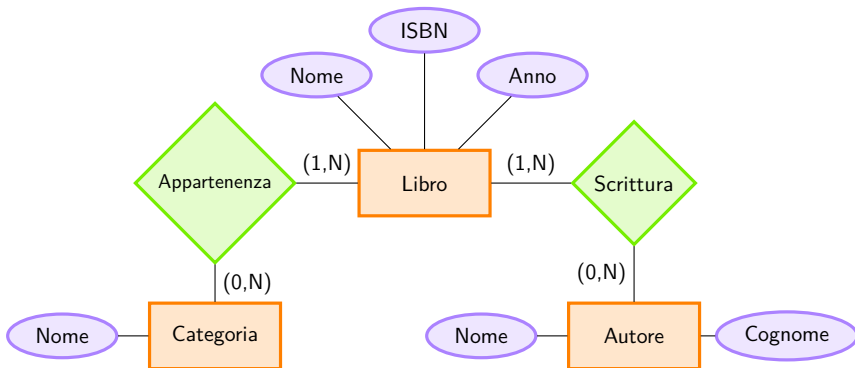
Analisi modello Libro/Autore/Categoria

- Analisi delle entità
 - *Libro*
 - Nome: il nome di copertina del libro
 - ISBN: il codice ISBN del libro
 - Anno: anno di pubblicazione del libro
 - *Categoria*
 - Nome: il nome della categoria
 - *Autore*
 - Nome: il nome dell'autore
 - Cognome: il cognome dell'autore

Analisi modello Libro/Autore/Categoria

- Analisi delle relazioni
 - *Appartenenza* (Libro - Categoria)
 - Un libro appartiene ad una o più categorie
 - Una categoria contiene zero (categoria vuota) o più libri
 - Relazione $(1, N) \leftrightarrow (0, N)$
 - *Scrittura* (Libro - Autore)
 - Un libro è scritto da uno o più autori
 - Un autore ha scritto zero (nessuno) o più libri
 - Relazione $(1, N) \leftrightarrow (0, N)$

Modello ER Libro/Autore/Categoria



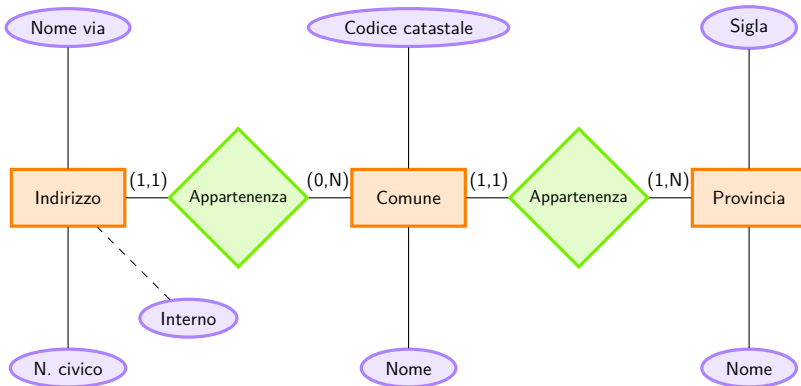
Analisi modello Indirizzo/Comune/Provincia

- Analisi delle entità
 - *Indirizzo*
 - Nome via: il nome della via privo di numero civico
 - Numero civico: il numero civico della via
 - Interno (opzionale): l'interno in caso di appartamento
 - *Comune*
 - Codice catastale: codice catastale del comune
 - Nome: il nome del comune
 - *Provincia*
 - Sigla: sigla di due lettere della provincia
 - Nome: nome della provincia

Analisi modello Indirizzo/Comune/Provincia

- Analisi delle relazioni
 - *Appartenenza* (Indirizzo - Comune)
 - Un indirizzo appartiene ad un solo comune
 - Ad un comune appartengono zero o più indirizzi
 - Relazione $(1, 1) \leftrightarrow (0, N)$
 - *Appartenenza* (Comune - Provincia)
 - Un comune appartiene ad una sola provincia
 - Ad una provincia appartengono uno o più comuni
 - Relazione $(1, 1) \leftrightarrow (1, N)$

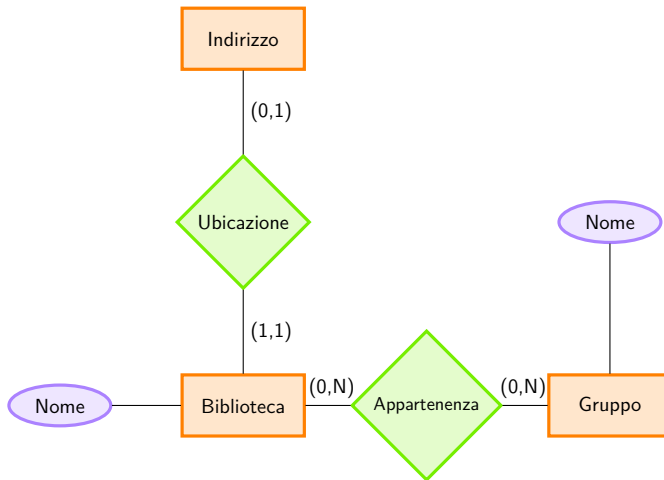
Modello ER Indirizzo/Comune/Provincia



Analisi modello Biblioteca/Gruppo

- Analisi delle entità
 - *Biblioteca*
 - Nome: nome della biblioteca
 - *Gruppo*
 - Nome: nome del gruppo
- Analisi delle relazioni
 - *Ubicazione* (Biblioteca - Indirizzo)
 - Una biblioteca è ubicata presso un solo indirizzo
 - Ad un indirizzo può o può non essere ubicata una biblioteca
 - Relazione $(1, 1) \leftrightarrow (0, 1)$
 - *Appartenenza* (Biblioteca - Gruppo)
 - Una biblioteca appartiene a zero o più gruppi
 - Ad un gruppo appartengono zero o più biblioteche
 - Relazione $(0, N) \leftrightarrow (0, N)$

Modello ER Biblioteca/Gruppo



Analisi modello Membro

- Analisi delle entità

- *Membro*

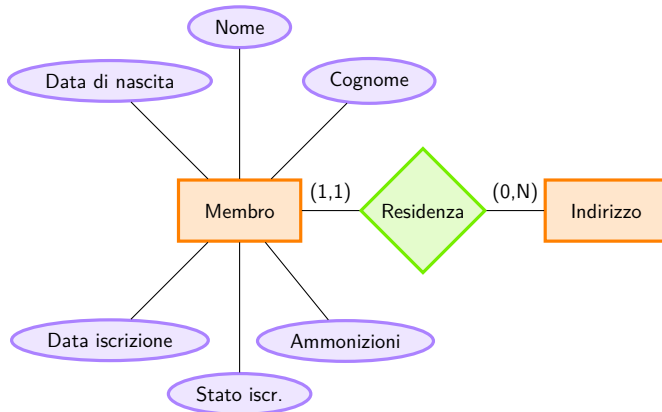
- Data di nascita: data di nascita del membro
 - Nome: nome del membro
 - Cognome: cognome del membro
 - Ammonizioni: numero di ammonizioni del membro
 - Data iscrizione: data di iscrizione del membro
 - Stato iscrizione: stato dell'iscrizione del membro

- Analisi delle relazioni

- *Residenza* (Membro - Indirizzo)

- Un membro risiede presso un solo indirizzo
 - Ad un indirizzo possono risiedere zero o più membri
 - Relazione $(1, 1) \leftrightarrow (0, N)$

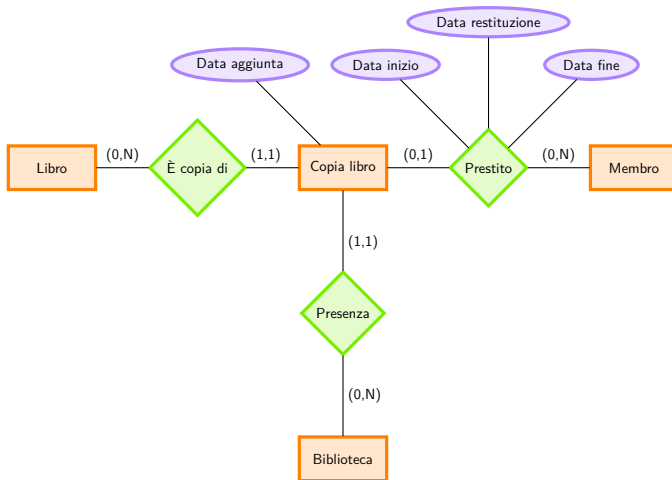
Modello ER Membro



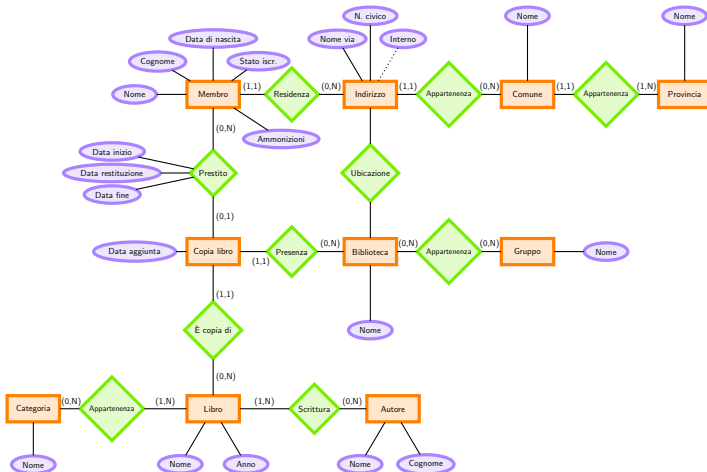
Analisi modello Copia libro/Prestito

- Analisi delle entità
 - *Copia libro*
 - Data aggiunta: data di aggiunta della copia al catalogo della biblioteca
- Analisi delle relazioni
 - *È copia di* (Copia libro - Libro)
 - Una copia fisica è tale di uno e di un solo libro
 - Un libro ha zero o più copie fisiche
 - Relazione $(1, 1) \leftrightarrow (0, N)$
 - *Presenza* (Copia libro - Biblioteca)
 - Una copia fisica è presente in una ed una sola biblioteca
 - In una biblioteca sono presenti zero o più copie fisiche di libri
 - Relazione $(1, 1) \leftrightarrow (0, N)$
 - *Prestito* (Copia libro - Membro)
 - Una copia fisica può essere o non essere in prestito ad un membro
 - Un membro può avere in prestito zero o più copie fisiche di libri
 - Relazione $(0, 1) \leftrightarrow (0, N)$
 - Attributi: data inizio, data fine, data restituzione

Modello ER Copia libro/Prestito



Modello ER



Vincoli non esprimibili

- I membri devono essere maggiorenni
- I membri possono prendere in prestito al massimo 5 libri alla volta
- La data di restituzione deve essere precedente alla data di fine prestito per evitare l'ammonizione
- Dopo 3 ammonizioni, l'iscrizione viene sospesa per 30 giorni durante i quali non è possibile prendere in prestito libri
- Il valore di "Stato iscrizione" può essere solo "Attiva" e "Sospesa"

Progettazione logica

- Analisi delle prestazioni
 - Tavola dei volumi
 - Tabella delle operazioni
- Analisi delle ridondanze
- Partizionamento/Accorpamento di entità e relazioni
- Scelta identificatori principali
- Modello ER ristrutturato
- Schema logico

Analisi delle prestazioni - Tavola dei volumi

Elemento	Volume	Commento
Biblioteca	264	Numero attuale preso dal sito ufficiale
Gruppo	15	Numero attuale preso dal sito ufficiale
Membro	250000	Stimato come il 20% della popolazione
Copia libro	5544000	Stimato come patrimonio librario medio in Friuli (21247) per il numero di biblioteche
Libro	1848000	Stima di 3 copie per libro
Comune	216	Numero esatto
Provincia	4	Numero esatto
Categoria	100	Stima
Autori	616000	Stima di 3 libri per autore
Prestito	2208000	Stimato come numero di prestiti nel 2012 in Friuli (8366) per il numero di biblioteche

Analisi delle prestazioni - Tabella delle operazioni

- Molto frequenti
 - Ricerca dei libri dato
 - il codice ISBN
 - e/o uno specifico autore
 - e/o una specifica categoria
 - e/o l'anno di pubblicazione
 - e/o una stringa che deve essere contenuta nel titolo
 - in
 - tutte le biblioteche della regione
 - o in un gruppo di biblioteche
 - o in una specifica biblioteca
- Poco frequenti
 - Iscrizione nuovo membro
 - Inserimento nuova copia in catalogo
- Frequenti
 - Prestito libro
 - inizio
 - termine
 - Inserimento nuova copia in catalogo
 - eventuale creazione *libro* se non presente
- Molto poco frequenti
 - Creazione nuova biblioteca
 - creazione indirizzo biblioteca
 - Creazione nuova categoria

Analisi delle ridondanze

È presente un solo ciclo, ma non è una ridondanza perchè gli indirizzi possono essere di un membro o di una biblioteca, e non si può ricavare l'uno dall'altro.

Non sono presenti altre ridondanze.

Partizionamento/Accorpamento di entità e rel.

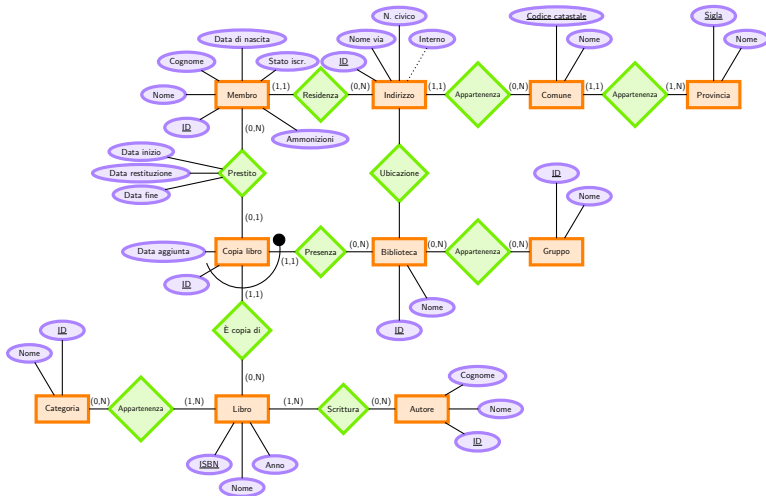
Non sono presenti attributi composti nè multivalore.

Si potrebbe salvare l'entità Indirizzo nella forma di attributi di
Membro e Biblioteca rispettivamente, ma è stato scelto fin dall'inizio
di mantenere un'entità separata per evitare ridondanze, dal momento
che più membri possono risiedere presso lo stesso indirizzo.

Scelta identificatori principali

- Membro: ID membro (aggiunto)
- Indirizzo: ID indirizzo (aggiunto)
- Comune: codice catastale
- Provincia: sigla
- Biblioteca: ID biblioteca (aggiunto)
- Gruppo: ID gruppo (aggiunto)
- Libro: codice ISBN
- Autore: ID autore (aggiunto)
- Categoria: ID categoria (aggiunto)
- Copia libro: ID copia (aggiunto), ISBN libro (esterno), ID biblioteca (esterno)

Modello ER ristrutturato



Schema logico

- Membro (ID¹, *IDindirizzo*, Nome, Cognome, DataDiNascita, Statolscrizione, Ammonizioni)
- Indirizzo (ID, *CodiceCatastaleComune*, NomeVia, Ncivico, Interno)
- Comune (*CodiceCatastale*, *SiglaProvincia*, Nome)
- Provincia (Sigla, Nome)
- Biblioteca (ID, *IDindirizzo*, Nome)
- BibliotecaGruppo (*IDbiblioteca*, *IDgruppo*)
- Gruppo (ID, Nome)
- Libro (ISBN, Nome)
- LibroAutore (*ISBN*, *IDautore*)
- Autore (ID, Nome, Cognome)
- LibroCategoria (*ISBN*, *IDcategoria*)
- Categoria (ID, Nome)
- CopiaLibro (ID, *ISBNlibro*, *IDbiblioteca*, DataAggiunta)
- Prestito (ID, *IDcopia*, *IDmembro*, DataInizio, DataFine, DataRestituzione)

¹Le PK sono sottolineate, le FK sono in corsivo.

Progettazione fisica

- Scelta degli indici
- Definizione dati
- Definizione trigger
- Definizione viste
- User defined functions

Progettazione fisica - Scelta degli indici

È buona norma definire gli indici sulle colonne che corrispondono a chiavi esterne, ma InnoDB li definisce in automatico.

Dalla versione 5.6 di MySQL, questo motore supporta gli indici di tipo FULLTEXT. Scegliamo quindi di definire questo indice sulla colonna *nome* della tabella *libro* data la frequenza e la tipologia delle ricerche che la coinvolgono.

Aggiungiamo inoltre un indice sulla colonna *anno* della stessa tabella.

Progettazione fisica - Definizione dati

```
-- Creazione database
CREATE DATABASE 'bibfv'
    CHARACTER SET utf8 -- Usiamo UTF-8 come codifica
    COLLATE utf8_general_ci;

-- Definizione database predefinito
USE 'bibfv';

-- Usiamo InnoDB (che supporta le FK) come default engine
-- per la creazione di tabelle nella sessione corrente
SET default_storage_engine=InnoDB;
```

Progettazione fisica - Definizione dati

```
-- Creazione tabella 'provincia'
CREATE TABLE 'provincia' (
  -- La sigla e' sempre lunga 2
  'sigla' CHAR(2) NOT NULL,
  'nome' VARCHAR(40) NOT NULL,

  PRIMARY KEY ('sigla')
);
```

```
-- Creazione tabella 'comune'
CREATE TABLE 'comune' (
  -- Il cod. cat. e' sempre lungo 4
  'codice_catastale' CHAR(4) NOT NULL,
  'nome' VARCHAR(40) NOT NULL,
  'sigla_prov' CHAR(2) NOT NULL,

  PRIMARY KEY ('codice_catastale'),

  FOREIGN KEY ('sigla_prov')
    REFERENCES 'provincia'('sigla')
    ON DELETE CASCADE
    ON UPDATE CASCADE
);
```

Progettazione fisica - Definizione dati

```
-- Creazione tabella 'indirizzo'
CREATE TABLE 'indirizzo' (
  'id' INT NOT NULL AUTO_INCREMENT,
  'codice_catastale_com' CHAR(4) NOT NULL,
  -- N. civico e interno possono
  -- contenere lettere
  'nome_via' VARCHAR(40) NOT NULL,
  'ncivico' VARCHAR(5) NOT NULL,
  'interno' VARCHAR(5) DEFAULT NULL,

  PRIMARY KEY ('id'),

  FOREIGN KEY ('codice_catastale_com')
    REFERENCES 'comune'('codice_catastale')
    ON DELETE CASCADE
    ON UPDATE CASCADE
);
```

```
-- Creazione tabella 'membro'
CREATE TABLE 'membro' (
  'id' INT NOT NULL AUTO_INCREMENT,
  'id_indirizzo' INT NOT NULL,
  'nome' VARCHAR(40) NOT NULL,
  'cognome' VARCHAR(40) NOT NULL,
  'data_di_nascita' DATE NOT NULL,
  -- Lo stato puo' assumere solo 2 valori
  'stato_iscrizione' ENUM('Attiva',
                          'Sospesa'),
  'ammonizioni' INT NOT NULL DEFAULT 0,

  PRIMARY KEY ('id'),

  FOREIGN KEY ('id_indirizzo')
    REFERENCES 'indirizzo'('id')
    ON DELETE CASCADE
    ON UPDATE CASCADE
);
```

Progettazione fisica - Definizione dati

```
-- Creazione tabella 'gruppo'
CREATE TABLE 'gruppo' (
    'id' INT NOT NULL AUTO_INCREMENT,
    'nome' VARCHAR(40) NOT NULL,

    PRIMARY KEY ('id')
);
```

```
-- Creazione tabella 'biblioteca'
CREATE TABLE 'biblioteca' (
    'id' INT NOT NULL AUTO_INCREMENT,
    'id_indirizzo' INT NOT NULL,
    'nome' VARCHAR(40) NOT NULL,

    PRIMARY KEY ('id'),

    FOREIGN KEY ('id_indirizzo')
        REFERENCES 'indirizzo'('id')
        ON DELETE CASCADE
        ON UPDATE CASCADE
);
```

```
-- Creazione tabella 'biblioteca_gruppo'
CREATE TABLE 'biblioteca_gruppo' (
    'id_biblioteca' INT NOT NULL,
    'id_gruppo' INT NOT NULL,

    PRIMARY KEY ('id_biblioteca', 'id_gruppo'),

    FOREIGN KEY ('id_biblioteca')
        REFERENCES 'biblioteca'('id')
        ON DELETE CASCADE
        ON UPDATE CASCADE,

    FOREIGN KEY ('id_gruppo')
        REFERENCES 'gruppo'('id')
        ON DELETE CASCADE
        ON UPDATE CASCADE
);
```

Progettazione fisica - Definizione dati

```
-- Creazione tabella 'libro'
CREATE TABLE 'libro' (
  -- Usiamo l'ISBN13 che e' sempre lungo 13
  'isbn' CHAR(13) NOT NULL,
  'nome' VARCHAR(80) NOT NULL,
  'anno' YEAR(4) NOT NULL,

  PRIMARY KEY ('isbn'),
  FULLTEXT ('nome'),
  INDEX ('anno')
);
```

```
-- Creazione tabella 'autore'
CREATE TABLE 'autore' (
  'id' INT NOT NULL AUTO_INCREMENT,
  'nome' VARCHAR(40) NOT NULL,
  'cognome' VARCHAR(40) NOT NULL,

  PRIMARY KEY ('id')
);

-- Creazione tabella 'categoria'
CREATE TABLE 'categoria' (
  'id' INT NOT NULL AUTO_INCREMENT,
  'nome' VARCHAR(40),

  PRIMARY KEY ('id')
);
```

Progettazione fisica - Definizione dati

```
-- Creazione tabella 'libro_autore'
CREATE TABLE 'libro_autore' (
  'isbn_libro' CHAR(13) NOT NULL,
  'id_autore' INT NOT NULL,

  PRIMARY KEY ('isbn_libro', 'id_autore'),

  FOREIGN KEY ('isbn_libro')
    REFERENCES 'libro'('isbn')
    ON DELETE CASCADE
    ON UPDATE CASCADE,

  FOREIGN KEY ('id_autore')
    REFERENCES 'gruppo'('id')
    ON DELETE CASCADE
    ON UPDATE CASCADE
);
```

```
-- Creazione tabella 'libro_categoria'
CREATE TABLE 'libro_categoria' (
  'isbn_libro' CHAR(13) NOT NULL,
  'id_categoria' INT NOT NULL,

  PRIMARY KEY ('isbn_libro', 'id_categoria'),

  FOREIGN KEY ('isbn_libro')
    REFERENCES 'libro'('isbn')
    ON DELETE CASCADE
    ON UPDATE CASCADE,

  FOREIGN KEY ('id_categoria')
    REFERENCES 'gruppo'('id')
    ON DELETE CASCADE
    ON UPDATE CASCADE
);
```


Progettazione fisica - Definizione dati

```
-- Creazione tabella 'copia_libro'
CREATE TABLE 'copia_libro' (
    'id' INT NOT NULL AUTO_INCREMENT,
    'isbn_libro' CHAR(13) NOT NULL,
    'id_biblioteca' INT NOT NULL,

    PRIMARY KEY ('id',
        'isbn_libro',
        'id_biblioteca'),

    FOREIGN KEY ('isbn_libro')
        REFERENCES 'libro'('isbn')
        ON DELETE CASCADE
        ON UPDATE CASCADE,

    FOREIGN KEY ('id_biblioteca')
        REFERENCES 'biblioteca'('id')
        ON DELETE CASCADE
        ON UPDATE CASCADE
);
```

```
-- Creazione tabella 'prestito'
CREATE TABLE 'prestito' (
    'id' INT NOT NULL AUTO_INCREMENT,
    'id_copia' INT NOT NULL,
    'id_membro' INT NOT NULL,
    'data_inizio' DATE NOT NULL,
    'data_fine' DATE DEFAULT NULL,
    'data_restituzione' DATE DEFAULT NULL,

    PRIMARY KEY ('id'),

    FOREIGN KEY ('id_copia')
        REFERENCES 'copia_libro'('id')
        ON DELETE CASCADE
        ON UPDATE CASCADE,

    FOREIGN KEY ('id_membro')
        REFERENCES 'membro'('id')
        ON DELETE CASCADE
        ON UPDATE CASCADE
);
```

Progettazione fisica - Definizione trigger

```

DELIMITER //
-- Trigger per la verifica della possibilità di richiedere un prestito
CREATE TRIGGER check_prestito BEFORE INSERT ON 'prestito'
FOR EACH ROW
BEGIN
    DECLARE NUMPRESTITI INT;
    DECLARE STATOISCRIZIONE VARCHAR(7);

    SET STATOISCRIZIONE = (SELECT 'stato_iscrizione' FROM 'membro' WHERE 'id' = NEW.id_membro);
    IF STATOISCRIZIONE = 'Sospesa' THEN -- Se l'iscrizione è sospesa
        IF (SELECT TIMEDIFF(DAY, 'data_restituzione', CURDATE()) FROM 'prestito'
            WHERE 'id' = NEW.id_membro ORDER BY 'data_restituzione' DESC LIMIT 1) > 30 THEN
            -- Se sono passati > 30gg dall'ultima restituzione, riattivo l'iscrizione
            UPDATE 'membro'
            SET 'stato_iscrizione' = 'Attiva';
        ELSE -- Altrimenti impedisco l'operazione
            SIGNAL SQLSTATE '45000' SET MESSAGE_TEXT = "L'iscrizione è sospesa";
        END IF;
    END IF;

    SET NUMPRESTITI = (SELECT COUNT(*) FROM 'prestito' WHERE 'id_membro' = NEW.id_membro
                        AND 'data_fine' IS NULL);
    IF NUMPRESTITI > 4 THEN -- Se il membro ha già preso in prestito 5 libri, impedisco l'operazione
        SIGNAL SQLSTATE '45000' SET MESSAGE_TEXT = 'Si possono prendere in prestito'
            ' al massimo 5 libri alla volta';
    END IF;
END //
DELIMITER ;

```

Progettazione fisica - Definizione trigger

```
DELIMITER //
```

```
-- Trigger per la segnalazione di ritardi/ammonizioni
```

```
CREATE TRIGGER check_ritardo_restituzione BEFORE UPDATE ON 'prestito'
```

```
  FOR EACH ROW
```

```
  BEGIN
```

```
    IF NEW.data_restituzione > OLD.data_fine THEN -- Se la restituzione è in ritardo
```

```
      IF (SELECT 'ammonizioni' FROM 'membro' WHERE 'id_membro' = NEW.id_membro) > 2 THEN
```

```
        -- Se il membro ha già almeno 3 ammonizioni sospendo l'iscrizione, azzerandole
```

```
        UPDATE 'membro' SET 'ammonizioni' = 0, 'stato_iscrizione' = 'Sospesa'
```

```
        WHERE id = NEW.id_membro;
```

```
      ELSE -- Altrimenti incremento il numero di ammonizioni
```

```
        UPDATE 'membro' SET 'ammonizioni' = 'ammonizioni' + 1
```

```
        WHERE id = NEW.id_membro;
```

```
      END IF;
```

```
    END IF;
```

```
  END //
```

```
-- Trigger per permettere solo ai maggiorenni di iscriversi
```

```
CREATE TRIGGER check_maggiorenne BEFORE INSERT ON 'membro'
```

```
  FOR EACH ROW
```

```
  BEGIN
```

```
    IF TIMESTAMPDIFF(YEAR, NEW.data_di_nascita, CURDATE()) < 18 THEN
```

```
      -- Se il membro non è maggiorenne, impedisco l'iscrizione
```

```
      SIGNAL SQLSTATE '45000' SET MESSAGE_TEXT = 'I membri devono essere maggiorenni';
```

```
    END IF;
```

```
  END //
```

```
DELIMITER ;
```

Progettazione fisica - Definizione viste

```
-- Creiamo la vista contenente i 10 libri più noleggiati
CREATE VIEW 'top10_libri' AS
    SELECT 'l'.'nome' AS 'nome_libro',
           'l'.'isbn' AS 'isbn',
           COUNT(*) AS 'numero_prestiti'
    FROM 'prestito' AS 'p'
        INNER JOIN 'copia_libro' AS 'c' ON 'p'.'id_copia' = 'c'.'id'
        INNER JOIN 'libro' AS 'l' ON 'c'.'isbn_libro' = 'l'.'isbn'
    WHERE 'data_restituzione' IS NOT NULL
    GROUP BY 'isbn'
    ORDER BY 'numero_prestiti' DESC
    LIMIT 10;

-- Diamo all'utente esterno la possibilità di leggere la vista
-- contenente la classifica
GRANT SELECT ON 'bibfv'.'top10_libri'
TO 'utente_esterno'@'localhost';
```

Progettazione fisica - User defined functions

```
DELIMITER //
-- Funzione per la conversione dell'ISBN10 a ISBN13
CREATE FUNCTION ISBN10TO13 (ISBN10 CHAR(10))
RETURNS CHAR(13)
DETERMINISTIC
BEGIN
    DECLARE ISBN13 CHAR(13);
    DECLARE CHECKSUM, I INT;

    SET ISBN13 = CONCAT('978' , LEFT(ISBN10, 9));

    SET I = 1, CHECKSUM = 0;
    WHILE I < 12 DO
        -- Sommo al checksum le cifre dispari, e quelle pari moltiplicate per 3
        SET CHECKSUM = CHECKSUM
            + SUBSTRING(ISBN13, I, 1)
            + SUBSTRING(ISBN13, I+1, 1) * 3;
        SET I = I + 2;
    END WHILE;

    SET CHECKSUM = (10 - (CHECKSUM % 10)) % 10;

    -- ISBN13 = '978' + prime 9 cifre dell'ISBN10 + checksum digit
    RETURN CONCAT(ISBN13, CONVERT(CHECKSUM, CHAR(1)));
END //
DELIMITER ;
```

Progettazione fisica - User defined functions

```

DELIMITER //
-- Funzione per la conversione dell'ISBN13 a ISBN10
CREATE FUNCTION ISBN13TO10 (ISBN13 CHAR(13))
RETURNS CHAR(10)
DETERMINISTIC
BEGIN
    DECLARE ISBN10 CHAR(10);
    DECLARE CHECKSUM, I INT;

    IF LEFT(ISBN13, 3) <> '978' THEN -- Ha senso convertire solo gli ISBN13 che iniziano con '978'
        SIGNAL SQLSTATE '45000' SET MESSAGE_TEXT = 'ISBN13 non convertibile';
    END IF;

    SET ISBN10 = SUBSTRING(ISBN13, 4, 9), I = 1, CHECKSUM = 0;
    WHILE I < 10 DO
        -- Sommo al checksum le cifre dispari, e quelle pari moltiplicate per 3
        SET CHECKSUM = CHECKSUM + SUBSTRING(ISBN10, I, 1) * (11-I), I = I + 1;
    END WHILE;

    SET CHECKSUM = (11 - (CHECKSUM % 11)) % 11;
    -- ISBN10 = ISBN13 senza le prime 3 e l'ultima cifra + checksum digit
    IF CHECKSUM = 10 THEN
        RETURN CONCAT(ISBN10, 'X');
    ELSE
        RETURN CONCAT(ISBN10, CONVERT(CHECKSUM, CHAR(1)));
    END IF;
END //
DELIMITER ;

```