Progetto Database – Biblioteca Comunale

Università degli studi di Napoli Parthenope

Biblioteca Comunale Grazia Deledda

A.A. 2022/2023

Data di consegna: 16/07/2023

Progetto e Relazione a cura di:

Romeo Gaetano 0124002667
Picardi Giovanni 0124002641
Arcopinto Lorenzo 0124002626

INDICE

PREMESSA

Con la seguente relazione, si cerca di descrivere al meglio il problema da risolvere e il modo in cui ciò è stato realizzato.
Come realtà di riferimento è stata scelta una biblioteca comunale, un problema reale presso la quale uno dei candidati svolge servizio di volontariato.
Si premette che l'obiettivo del progetto è gestire esclusivamente il funzionamento della biblioteca, senza considerare gli aspetti burocratici e finanziari, che sono di competenza dell'ente pubblico di riferimento.
Non avendo avuto la possibilità di interagire con una figura che facesse da direttore della biblioteca ed esponesse tutte le problematiche e le necessità riferite al caso, le informazioni da gestire sono state determinate in seguito ad un'attenta riflessione e collaborazione tra i candidati.
Detto ciò, si può procedere con la descrizione del problema.

PROGETTAZIONE

ENTITA' DI RIFERIMENTO

Si vuole realizzare un database per gestire il funzionamento di una biblioteca comunale.
I clienti, dei quali interessa memorizzare i dati anagrafici e l'email, potranno accedere alla biblioteca dal lunedì al venerdì dalle ore 8:00 alle ore 20:00 per trovare un luogo insonorizzato in cui poter studiare in tranquillità.
Per poter ottenere il diritto di prendere in prestito copie di libri, sarà necessaria una registrazione, la quale fornirà una tessera.
La tessera, identificata univocamente da un numero, avrà una scadenza annuale e andrà rinnovata con un costo simbolico stabilito e fisso, che servirà all'ente pubblico di riferimento per effettuare periodicamente operazioni di ristrutturazione della biblioteca.
La tessera di un cliente verrà bloccata se in possesso di almeno tre multe non ancora pagate.
Si fa presente che il prestito di un libro avverrà al momento della richiesta di un cliente e che non si avrà la possibilità di prenotare copie da prendere in prestito in un periodo successivo.
Inoltre, non sarà permesso ad un cliente di ottenere in prestito più copie dello stesso libro allo stesso istante.
Una volta registrato, il cliente avrà anche diritto a partecipare ad eventi periodicamente organizzati dalla biblioteca senza costi aggiuntivi e a lasciare recensioni riguardanti i libri che ha letto.
Gli eventi saranno descritti da un nome, dalla data e dall'ora di inizio e da una durata.
La necessità di memorizzare i dati anagrafici del cliente sorge dal momento che una restituzione avvenuta in ritardo rispetto alla scadenza prefissata al momento di un prestito porterà alla generazione di una multa.
Per ogni libro saranno memorizzati l'ISBN, il titolo, l'anno di pubblicazione, il genere letterario di appartenenza e gli autori.
Per la conservazione dei libri si è scelta una soluzione che permetta ai bibliotecari di risalire facilmente alla posizione di una copia.
Considerando che la biblioteca possiede tre piani di altezza, si avrà su ognuno di essi uno scaffale per ogni genere letterario del quale è possibile trovare copie.
Inoltre, gli scaffali saranno composti da mensole, su ognuna delle quali andranno posizionate le copie la cui lettera iniziale del titolo corrisponderà a quella indicata dalla mensola.

Si avrà pertanto la seguente disposizione: ad ogni piano ci sarà uno scaffale per ogni genere di libro ed ogni scaffale conterrà un numero stabilito di mensole, contraddistinte dalla lettera che indicherà i libri che potranno essere posizionati su di essa.
Ad esempio, al primo piano si avrà uno scaffale nel quale andranno conservate le copie che appartengono al genere "Matematica" e il cui titolo inizierà per una lettera compresa tra "A" e "H". Per trovare le restanti copie appartenenti al genere "Matematica" bisognerà recarsi agli altri piani.
Questa disposizione, seppur apparentemente complicata, permetterà ai bibliotecari di trovare una copia semplicemente effettuando una ricerca alfabetica nello scaffale e nella mensola appropriati, evitando sprechi di tempo e grattacapi.
È importante che, per far sì che questa soluzione funzioni, le copie siano sempre memorizzate nella posizione corretta.
Per ogni scaffale verrà memorizzata la data di acquisto per far sì che questi possano essere sostituiti quando troppo vecchi, il numero del piano sulla quale è situato e il genere di libri che contiene.
Per le mensole, invece, occorrerà memorizzare la capienza, così da evitare che vengano ordinate più copie di quante la biblioteca possa effettivamente contenere.
Le copie potranno essere fornite alla biblioteca tramite ordini da case editrici, oppure gratuitamente da parte di donatori.
È necessario memorizzare i dettagli sulle spedizioni e i contatti dei fornitori, ad esempio la casa editrice che ha effettuato la spedizione e il numero di copie ordinate, per controllare le giacenze ed evitare truffe.
I bibliotecari, dei quali verranno memorizzati i dati anagrafici e il numero di telefono, avranno il compito di assistere i clienti, rinnovare le tessere e posizionare correttamente le copie sulle apposite mensole, secondo la disposizione precedentemente descritta.
Siccome sono volontari, ad essi non sarà assegnato uno stipendio ma, in compenso, avranno turni assegnati, a durata variabile, che potranno rispettare anche con leggere infrazioni.
Come premesso, non ci si occuperà dell'effettivo pagamento delle multe, della tassa per l'ottenimento/aggiornamento delle tessere e degli ordini per l'ottenimento delle copie dalle case editrici, per questo motivo per le multe verranno memorizzate solamente la data di generazione e quella in cui avverrà l’effettivo pagamento, che si presuppone avvenga prima o poi.

GLOSSARIO

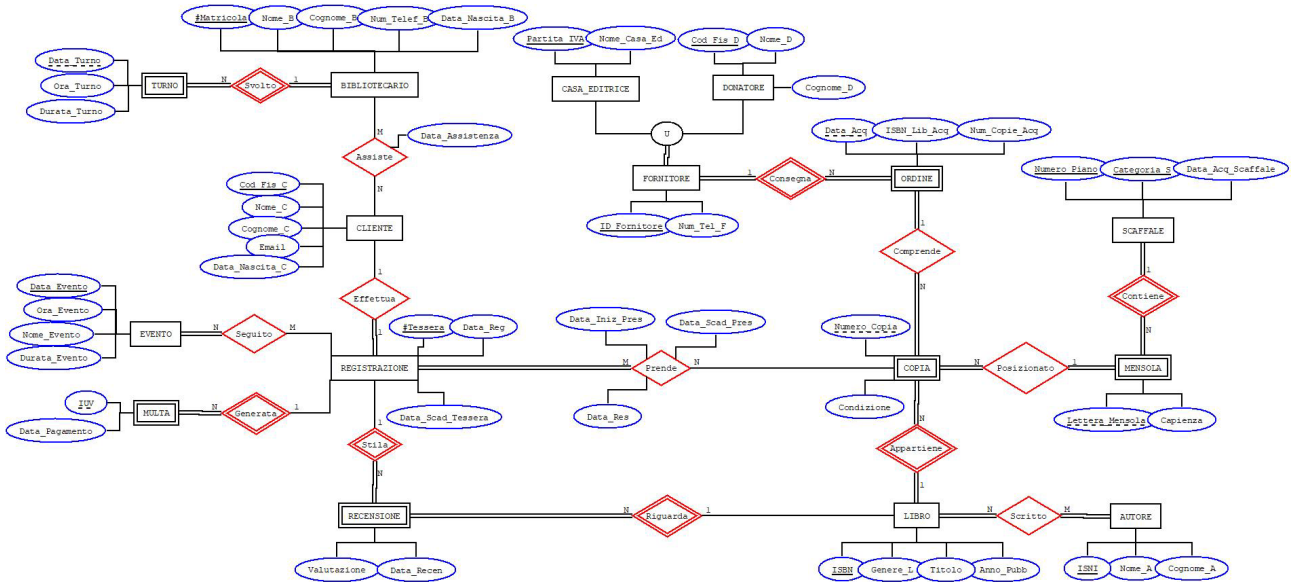
Il seguente glossario ha lo scopo di chiarire eventuali termini tecnici o non facilmente comprensibili.

TERMINE	DEFINIZIONE	SINONIMO
Durata Turno	Durata espressa in ore del turno svolto dal bibliotecario	
Bibliotecario	Colui che svolge le proprie mansioni all'interno della biblioteca	Volontario, Addetto
Numero Matricola	Codice univoco assegnato dalla biblioteca per il riconoscimento dei bibliotecari	
Cliente	Frequentatore della biblioteca, non ancora registrato	Utente, Studente, Frequentatore
Registrato	Cliente che ha effettuato la registrazione alla biblioteca ottenendo la tessera	Tesserato
Registrazione	Tesseramento da parte del cliente attraverso il quale può usufruire dei servizi offerti dalla biblioteca	Iscrizione
Numero Tessera	Codice univoco assegnato dalla biblioteca per il riconoscimento dei clienti registrati	
Scadenza Tessera	Data entro la quale il cliente è tenuto a rinnovare la propria tessera per continuare ad usufruire dei benefici offerti dalla biblioteca	
Scadenza Prestito	Data entro la quale il cliente è tenuto a restituire la copia di un libro presa in prestito onde evitare una multa	
Multa	Contravvenzione da parte del comune in caso di restituzione avvenuta oltre la scadenza del prestito da parte di un cliente	
Ente Pubblico	Istituzione di riferimento della biblioteca	Comune
Evento	Eventi periodicamente organizzati dalla biblioteca al fine di presentare libri ed effettuare laboratori didattici	

Durata Evento	Durata di un evento espressa in ore	
Valutazione	Voto assegnato ad un libro da parte di un cliente	Voto
Copia	Copia singola di un determinato libro	
Condizione	Condizione della copia al momento dell'acquisto	
Ordine	Acquisto effettuato dalla biblioteca da parte di una casa editrice o libro ceduto gratuitamente da parte di un donatore	Spedizione, Donazione, Approvvigionamento
ID Fornitore	Codice univoco utilizzato dalla biblioteca per il riconoscimento dei fornitori	
Donatore	Persona esterna alla biblioteca che fornisce in dono copie di determinati libri	
ISBN	Codice univoco e riconosciuto che identifica un determinato libro	
ISNI	Codice univoco e riconosciuto che identifica un determinato autore	
Lettera Mensola	Iniziale delle copie posizionate su di una mensola	Iniziale

DIAGRAMMA EE/R

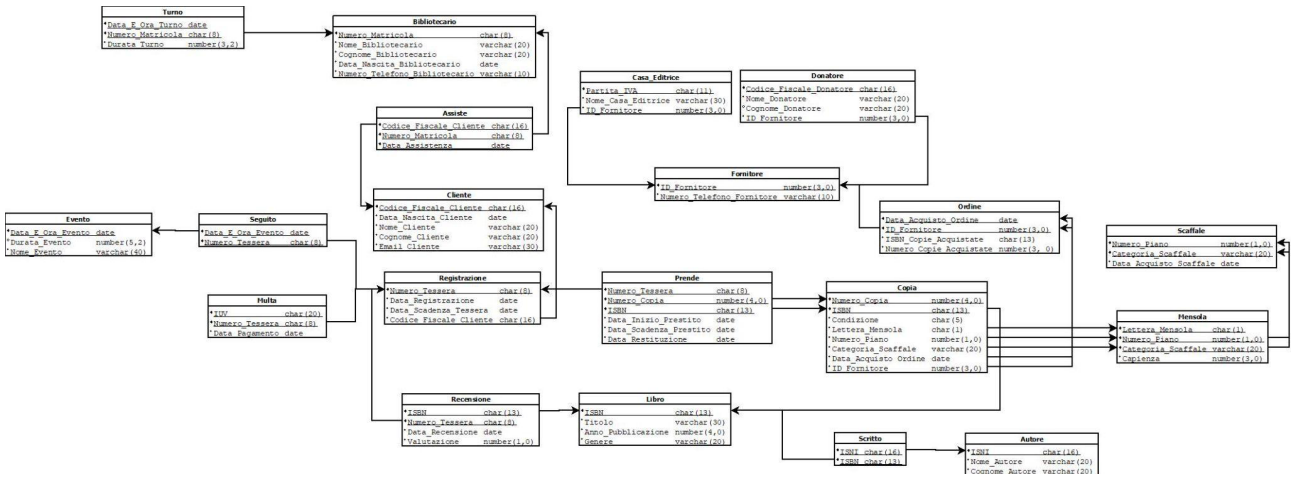
Dopo un'attenta analisi del problema, è stato deciso di impostare il diagramma ee/r nel seguente modo:



Come è possibile notare non sono indicate le cardinalità massime e minime delle associazioni, ma solamente le molteplicità.
Per una questione di leggibilità, i nomi degli attributi nel diagramma sono stati abbreviati.

DIAGRAMMA RELAZIONALE

Di conseguenza il diagramma relazionale che ne deriva è il seguente:



Come è possibile notare, al contrario del modello ee/r, nel modello relazionale gli attributi indicanti la data e l'ora degli eventi e dei turni, sono rappresentati come un unico attributo.
Il motivo di questa scelta è dato dal fatto che in Oracle SQL, il tipo date contiene anche l'ora.
Inoltre, nel modello relazionale sono indicate anche le chiavi esterne.

SCELTA DELLE CHIAVI PRIMARIE

Per identificare univocamente ogni tupla delle tabelle, sono state decise le seguenti chiavi primarie:

- “Data_Turno” e “Numero_Matricola” per la tabella Turno, siccome ogni bibliotecario svolge al più un turno al giorno;
- “Data_Evento” per la tabella Evento, siccome ogni giorno si tiene al più un evento;
- “IUV” e “Numero_Tessera” per la tabella Multa, siccome ad un cliente possono essere assegnate più multe e non basta l'IUV di una multa per risalire al cliente multato;
- “Numero_Tessera” e “ISBN” per la tabella Recensione, siccome ogni cliente registrato può recensire una sola volta un determinato libro. Ciò non vieta al cliente di modificare la propria recensione di un libro;
- “ISBN” e “Numero_Copia” per la tabella Copia, ovvero un numero che permette alla biblioteca di differenziare le singole copie di un determinato libro;
- “Numero_Piano” e “Categoria_Scaffale” per la tabella Scaffale, siccome all’interno della biblioteca è presente un solo scaffale per ogni genere per ogni piano;
- “Lettera_Mensola”, “Numero_Piano” e “Categoria_Scaffale” per la tabella Mensola, siccome ogni mensola appartiene ad un determinato scaffale e può contenere solo copie che iniziano per una determinata lettera dell'alfabeto;
- “ID_Fornitore” e “Data_Acquisto_Ordine” per la tabella Ordine, siccome ogni fornitore spedisce al più un ordine al giorno;
- Per le tabelle Fornitore, Casa Editrice e Donatore non è stato possibile trovare un attributo in comune che facesse da chiave e per questo è risultato necessario l’utilizzo di una chiave artificiale (ID_Fornitore), che prende il nome di chiave surrogata.

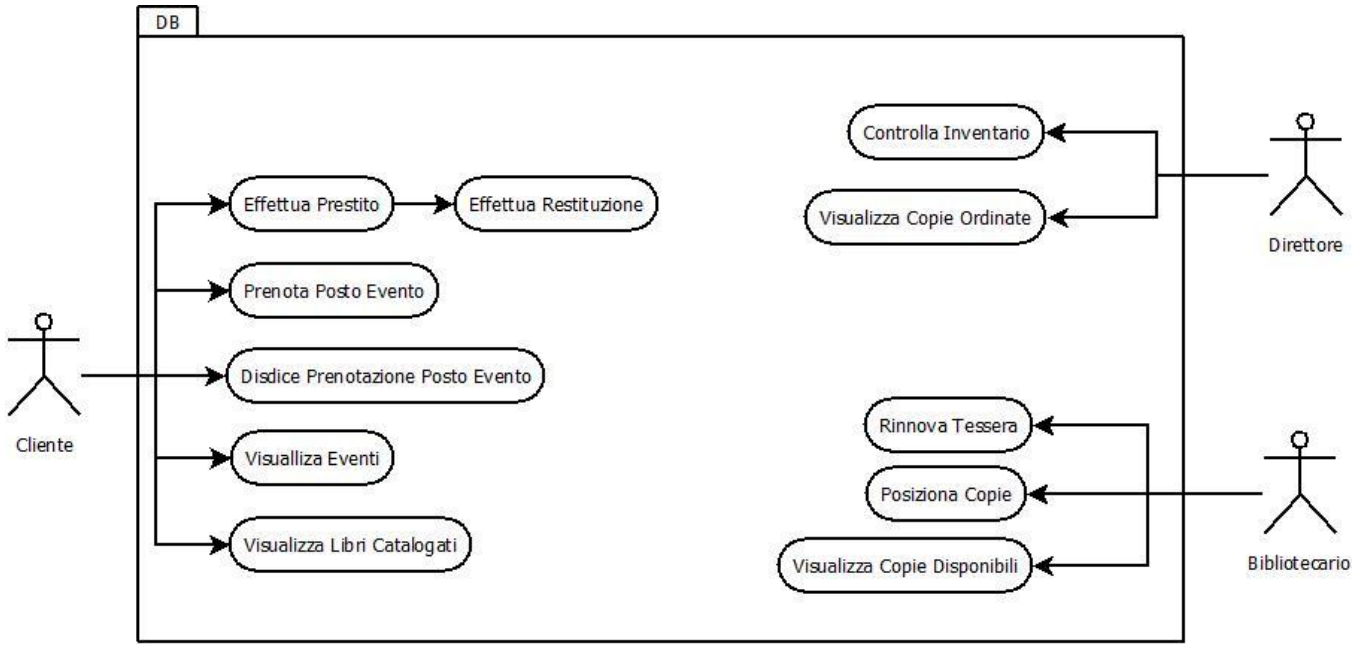
UTENTI E CATEGORIE

È possibile individuare i seguenti utenti:

- Registrato: colui che può prendere in prestito copie di libri e prenotarsi agli eventi. Ha la possibilità di visualizzare i libri catalogati nella biblioteca, le relative recensioni, gli eventi a cui può prenotarsi, i libri che ha preso in prestito e la data di scadenza entro la quale è tenuto a restituirli. Può inoltre visualizzare la valutazione che ha attribuito ai libri letti e le multe che gli sono state assegnate;
- Addetto: colui che si occupa di posizionare correttamente le copie e di rinnovare le tessere dei clienti. Ha la possibilità di visualizzare la posizione delle copie e il numero di copie di ogni libro presenti in biblioteca. Può inoltre visualizzare i turni per capire quando dovrà essere presente in biblioteca, assistere i clienti durante lo svolgimento del suo turno e rinnovare le tessere dei clienti;
- Direttore: colui che gestisce l’arrivo e l’acquisto di ordini da parte delle case editrici e l’inventario della biblioteca. Il suo scopo è quello di controllare che non ci siano incongruenze tra gli ordini di copie effettuati e il numero di copie effettivamente presenti nella biblioteca. Può anche assumere o licenziare bibliotecari, assegnare loro turni, modificarne gli orari e aggiungere, annullare o modificare gli orari degli eventi. Si occupa inoltre della sostituzione degli scaffali quando vecchi;
- Admin: colui che ha pieni poteri gestionali sul database. Possiede ogni privilegio, ma li utilizza solo in casi straordinari o di emergenza.

OPERAZIONI DEGLI UTENTI

Le operazioni fanno riferimento alle azioni che gli utenti possono effettuare mediante procedure. Di seguito è riportato uno schema che rappresenta le operazioni effettuabili e i relativi utenti coinvolti.



Operazione	Effettua_Prestito
Scopo	Permette ad un cliente registrato di prendere in prestito la copia di un libro se disponibile
Argomenti	Numero tessera del cliente e ISBN del libro
Risultato	Aggiornamento di Prende se la copia è disponibile e il cliente la ottiene in prestito / Errore se la copia non è disponibile
Errori	Libro_Gia_Prestato se il cliente possiede già una copia del libro, Multe_Non_Pagate se il cliente ha tre multe non ancora pagate, Tessera_Scaduta se la tessera del cliente è scaduta
Usa	Multa per controllare il numero di multe non ancora pagate assegnate al cliente, Registrazione per controllare se la tessera del cliente non è scaduta, Prende per stabilire la data di scadenza del prestito e indicare la copia prestata al cliente, Copia per controllare prestare al cliente la pima copia disponibile del libro richiesto
Modifica	Effettua un inserimento in Prende
Prima	Il cliente registrato non ha preso in prestito la copia, che quindi è disponibile
Poi	Il cliente registrato ha preso in prestito la copia, che quindi non sarà disponibile fino alla restituzione

Operazione	Effettua_Restituzione
Scopo	Permette ad un cliente di restituire una copia presa in prestito
Argomenti	Numero tessera del cliente, ISBN e Numero copia del libro restituito
Risultato	Aggiornamento di Prende ed inserimento in Multa se la data di restituzione è successiva a quella di scadenza del prestito / Errore se il cliente cerca di restituire una copia non presa in prestito
Errori	Copia_Non_Prestata se il cliente tenta di restituire una copia che non ha preso in prestito
Usa	Prende per aggiornare la data di restituzione, Multa per assegnare una multa al cliente se la data di restituzione è successiva a quella di scadenza del prestito
Modifica	Prende.Data_Restituzione ed effettua eventualmente un inserimento in Multa
Prima	La copia non può essere presa in prestito da altri clienti
Poi	La copia può essere presa in prestito

Operazione	Prenota_Posto_Evento
Scopo	Permette a un cliente registrato di partecipare a un evento se ci sono posti disponibili
Argomenti	Numero tessera del cliente e Data dell'evento
Risultato	Inserimento in Seguito se l'utente può prenotarsi per l'evento / Errore altrimenti
Errori	Evento_Pieno se i posti disponibili sono terminati, Multe_Non_Pagate se il cliente ha tre multe non ancora pagate, Tessera_Scaduta se la tessera del cliente è scaduta, Nessun_Evento se il cliente tenta di prenotarsi ad un evento non programmato
Usa	Seguito per controllare i posti disponibili all'evento, Multa per controllare il numero di multe non ancora pagate assegnate al cliente, Registrazione per controllare che la tessera del cliente non sia scaduta
Modifica	Effettua un inserimento in Seguito se l'utente può prenotarsi all'evento
Prima	Il cliente non è prenotato all'evento e quindi c'è un posto disponibile
Poi	Il cliente è prenotato all'evento e quindi c'è un posto occupato

Operazione	Disdici_Prenotazione_Posto_Evento
Scopo	Permette ad un cliente di disdire la partecipazione ad un evento
Argomenti	Numero tessera del cliente e Data dell'evento
Risultato	Cancellazione della prenotazione all'evento da Seguito se possibile / Errore altrimenti
Errori	Evento_Concluso se il cliente tenta di disdire la prenotazione ad un evento già concluso, Cliente_Non_Prenotato se il cliente tenta di disdire una prenotazione non effettuata
Usa	Seguito per controllare se il cliente è prenotato all'evento
Modifica	Cancellazione da Seguito
Prima	Il cliente è prenotato all'evento e quindi c'è un posto occupato
Poi	Il cliente non è più prenotato all'evento e quindi si libera un posto

Operazione	Posiziona_Copie
Scopo	Permette ad un bibliotecario di posizionare le copie nella giusta mensola e nel giusto scaffale
Argomenti	Data di acquisto dell'ordine, ID del fornitore
Risultato	Inserimento in Copia se è possibile risalire all'ordine / Errore altrimenti
Errori	Ordine_Non_Trovato se l'ordine a cui si fa riferimento non è stato effettuato dalla biblioteca
Usa	Ordine per controllare che l'ordine a cui si fa riferimento sia stato effettuato dalla biblioteca, Libro per comprendere in quale mensola e in quale scaffale vanno posizionate le copie, Copia per inserire ed aggiornare le informazioni delle copie ordinate
Modifica	Inserimento in Copia
Prima	Le copie ordinate non sono disponibili in quanto non ancora catalogate
Poi	Le copie ordinate sono disponibili e catalogate e possono quindi essere prese in prestito dai clienti

Operazione	Rinnova_Tessera
Scopo	Permette ad un bibliotecario di rinnovare una tessera scaduta se il cliente non possiede tre multe non ancora pagate
Argomenti	Numero tessera del cliente
Risultato	Aggiornamento in Registrazione se il cliente è registrato, la sua tessera scaduta e non possiede tre multe non ancora pagate / Errore altrimenti
Errori	Tessera_Non_Trovata se il cliente non risulta essere registrato alla biblioteca, Troppe_Multe se l'utente possiede tre multe non ancora pagate
Usa	Registrazione per aggiornare la data di scadenza della tessera, Multa per contare il numero di multe non ancora pagate possedute dal cliente
Modifica	Registrazione.data_scadenza_tessera
Prima	La tessera del cliente è scaduta e non può effettuare prestiti o prenotarsi a eventi
Poi	La tessera del cliente è nuovamente attiva e può effettuare prestiti e prenotarsi a eventi

Operazione	Controlla_Inventario
Scopo	Permette al direttore di visualizzare l'inventario della biblioteca per ordinare copie di libri le cui disponibilità sono ridotte
Argomenti	ID del fornitore e Numero di copie da ordinare per ogni libro
Risultato	Inserimento in Ordine di copie le cui disponibilità in biblioteca sono ridotte
Errori	-
Usa	Libro per controllare i libri catalogati in biblioteca, Copia per controllare la disponibilità di copie per ogni libro, Ordine per effettuare un ordine di copie
Modifica	Inserimento in Ordine per ogni libro con quantità di copie disponibili ridotta
Prima	All'interno della biblioteca sono catalogati libri di cui si possiedono poche copie
Poi	I libri catalogati all'interno della biblioteca hanno una disponibilità di copie sufficiente

La tavola delle operazioni seguente ipotizza che in un anno si iscrivano 200 persone alla biblioteca e che tutte le persone che prendono in prestito libri li restituiscono, considerando comunque il caso in cui la restituzione avviene in ritardo.

Operazione	Volume	Periodo
Effettua_Prestito	90	mese
Effettua_Restituzione	90	mese
Prenota_Posto_Evento	80	mese
Disdici_Prenotazione_Posto_Evento	15	mese
Posiziona_Copie	10	mese
Rinnova_Tessera	100	anno
Controllo_Inventario	1	mese

VOLUMI

Con volumi si intende il numero di tuple presenti in un caso reale in ciascuna entità del database con relativo incremento calcolato in base ad un periodo di tempo.
Siccome esiste una politica di aggiornamento dei dati per cui alcune tuple più vecchie sono rimosse dal database, l'incremento di alcune tabelle, come Scaffale, Turno, Evento, Prende, Multa e Seguito è nullo.

Tabella	Tipo	Volume	Incremento	Periodo
Turno	Entità Debole	15	0	Una settimana
Bibliotecario	Entità	15	2	Un anno
Assiste	Tabella di transizione	300	320	Un mese
Cliente	Entità	250	30	Un anno
Registrazione	Entità	200	20	Un anno
Seguito	Tabella di transizione	80	0	Un mese
Evento	Entità	4	0	Un mese
Multa	Entità Debole	10	0	Un mese
Recensione	Entità Debole	50	5	Un anno
Prende	Tabella di transizione	90	0	Un mese
Copia	Entità Debole	675	75	Un anno
Libro	Entità	135	15	Un anno
Scritto	Tabella di transizione	135	15	Un anno
Autore	Entità	115	5	Un anno
Mensola	Entità Debole	702	0	Cinque anni
Scaffale	Entità	27	0	Cinque anni
Ordine	Entità Debole	15	25	Un mese
Fornitore	Entità	30	5	Un anno
Casa Editrice	Entità	15	3	Un anno
Donatore	Entità	15	2	Un anno

VINCOLI DI INTEGRITA'

I vincoli statici riguardano attributi i cui valori non cambiano nel tempo, al contrario dei vincoli dinamici, i quali riguardano anche regole di business.
Di seguito sono elencati i principali vincoli del database.

Vincoli Statici

- La durata di un turno è obbligatoria e deve assumere un valore intero compreso tra quattro e otto;
- La durata di un evento è obbligatoria e deve assumere un valore reale compreso tra uno e due;
- La data e l'ora di un turno e di un evento devono rispettare i giorni e gli orari di apertura e chiusura stabiliti dalla biblioteca (non possono iniziare prima dell'apertura della biblioteca e non possono terminare dopo la sua chiusura);
- Il numero matricola di un bibliotecario deve essere composto esattamente da otto numeri;
- Il nome è obbligatorio e non può contenere caratteri speciali, fatta eccezione per il nome di un evento o di una casa editrice, il quale può contenere degli spazi;
- Il cognome è obbligatorio, fatta eccezione per il donatore, e non può contenere caratteri speciali, fatta eccezione di uno spazio;
- Il numero di telefono deve essere composto esattamente da nove o dieci numeri, inoltre è univoco e obbligatorio (ad esempio 3815962014 oppure 381596624);
- L'email è univoca e obbligatoria, inoltre deve rispettare il seguente formato: "nickname@dominio.estensione" (ad esempio nomecognome@gmail.com);
- Il numero tessera di un cliente deve essere composto esattamente da otto numeri;
- Il codice fiscale deve rispettare il seguente formato: tre lettere per il cognome, tre lettere per il nome, le ultime due cifre dell'anno di nascita, un carattere alfanumerico per il mese, due cifre per il giorno di nascita, un codice alfanumerico di una lettera e tre cifre per il luogo di nascita e un carattere alfanumerico di controllo (ad esempio RSSPSQ08H16F839S);
- La valutazione di una recensione deve assumere un valore intero compreso tra uno e cinque ed è obbligatoria per la recensione di un libro;
- L'ISBN di un libro deve essere composto esattamente da tredici numeri;
- Il nome di una casa editrice è univoco e obbligatorio;
- L'ISNI di un autore deve essere composto da esattamente 16 numeri;
- La condizione di una copia può assumere solamente un valore tra "nuovo" e "usato";
- Il genere di un libro è obbligatorio e ammette le seguenti opzioni: storia, letteratura, geografia, fantasy, matematica, scienze, giallo, romanzo, horror;
- La partita IVA di una casa editrice deve essere composta esattamente da undici numeri;
- L'IUV di una multa deve rispettare il seguente formato: '123-ANNO-MESE-NUMERO";
- Il numero del piano di uno scaffale deve assumere un valore intero compreso tra uno e tre.

Non sono stati descritti i vincoli banali come la necessità che la data di scadenza di una tessera debba essere successiva alla data di registrazione... .

Vincoli Dinamici

- Un bibliotecario può svolgere al più un turno al giorno;
- Nello stesso giorno possono esserci al più tre bibliotecari che svolgono il proprio turno;
- All'interno della biblioteca possono essere registrati al più quindici bibliotecari;
- Per poter essere assunti come bibliotecario è necessario avere un'età compresa tra i diciotto e sessantacinque anni;
- Per potersi registrare alla biblioteca è necessario avere un'età compresa tra i quattordici e i settant'anni;
- I clienti possono registrarsi solo una volta alla biblioteca, salvo disiscrizione;
- Per la disponibilità di posti presenti nella biblioteca, il numero massimo di partecipanti ad un evento è cinquanta;
- Può tenersi al più un evento al giorno e al più un evento dello stesso tipo al mese;
- Un cliente non può prendere in prestito più di dieci copie di un libro senza prima restituirne alcuna;
- Un cliente non può prendere in prestito più copie di uno stesso libro contemporaneamente;
- Non si può posizionare una copia su di una mensola piena o la cui lettera non corrisponde alla lettera inziziale del titolo del libro al quale appartiene la copia, o su di uno scaffale la cui categoria non corrisponde al genere al quale appartiene la copia;
- Non è possibile effettuare ordini di copie dai fornitori se il numero di queste eccede la capienza della mensola sulla quale vanno posizionate.
- La data di acquisto di uno scaffale non può essere precedente alla data corrente di più di cinque anni.

Da notare che non sono stati indicati i vincoli di chiave primaria e di chiave esterna.

VERIFICA DI NORMALITA'

Di seguito si analizza se il database rispetta le forme normali.

PRIMA FORMA NORMALE

Il database rispetta la prima forma normale in quanto non presenta campi di tipo strutturato, nonostante la presenza di campi di tipo data, che seppur apparentemente strutturati, sono, nella maggior parte dei linguaggi, tipi primitivi e quindi considerati atomici.

SECONDA FORMA NORMALE

Il database rispetta la seconda forma normale in quanto per ciascuna relazione ogni attributo non chiave dipende completamente dalla chiave e non parzialmente da uno degli attributi che la compone, nel caso di chiavi multiattributo.

TERZA FORMA NORMALE

Il database rispetta la terza forma normale e la BCFN in quanto non sono presenti dipendenze anomale (gli attributi non chiave dipendono solo dalla chiave e non da altri attributi). Tuttavia, è presente una minima ridondanza in quanto si ha che l'ISBN di ogni libro presente nella biblioteca è ripetuto sia nella tabella Ordine che nella tabella Libro.
Tale scelta è stata effettuata per essere a conoscenza delle copie spedite tramite un ordine alla biblioteca e tenere conto delle giacenze.

IMPLEMENTAZIONE

Conclusa la fase di progettazione, è tempo di convertire il tutto in codice eseguibile.
Il codice utilizzato farà riferimento al DBMS Oracle XE e il linguaggio adottato sarà il PL/SQL.
Le parole riservate saranno evidenziate in **blu**, i commenti in **verde** e i valori fra gli apici in **arancione**.

CREAZIONE UTENTI

Di seguito si riporta la creazione degli utenti.
Siccome lo schema non è stato ancora definito, gli unici privilegi indicati sono quelli detenuti dall'amministratore, che li possiede tutti.

```
CREATE USER REGISTRATO IDENTIFIED BY PASSWORD_REGISTRATO;
CREATE USER ADDETTO IDENTIFIED BY PASSWORD_ADDETTO;
CREATE USER DIRETTORE IDENTIFIED BY PASSWORD_DIRETTORE;
CREATE USER AMMINISTRATORE IDENTIFIED BY PASSWORD_AMMINISTRATORE;
GRANT ALL PRIVILEGIES TO AMMINISTRATORE;
```

DATA DEFINITION LANGUAGE

Di seguito si riporta la traduzione della creazione delle tabelle.

FORNITORE

La tabella Fornitore è popolata tramite inserimento diretto dal direttore, motivo per il quale è necessario che i controlli d'integrità dei dati siano inclusi nel DDL.

```
CREATE TABLE FORNITORE
(
   ID_FORNITORE                    number (3, 0),
   NUMERO_TELEFONO_FORNITORE       varchar (10) UNIQUE NOT NULL,

   CONSTRAINT PRIMARY_KEY_FORNITORE         PRIMARY KEY (ID_FORNITORE),
   CONSTRAINT CHECK_NUMERO_TELEFONO_FORNITORE CHECK
(REGEXP_LIKE(NUMERO_TELEFONO_FORNITORE, '^[0-9]{9,10}+$'))
);
```

BIBLIOTECARIO

La tabella Bibliotecario è popolata ed aggiornata dal direttore tramite inserimento diretto, motivo per il quale è necessario che i controlli d'integrità dei dati siano inclusi nel DDL.

```
CREATE TABLE BIBLIOTECARIO
(
   NUMERO_MATRICOLA                char (8),
   NOME_BIBLIOTECARIO              varchar (20)   NOT NULL,
   COGNOME_BIBLIOTECARIO           varchar (20)   NOT NULL,
   DATA_NASCITA_BIBLIOTECARIO   date           NOT NULL,
   NUMERO_TELEFONO_BIBLIOTECARIO   varchar (10)   UNIQUE NOT NULL,
```

```
  CONSTRAINT PRIMARY_KEY_BIBLIOTECARIO        PRIMARY KEY (NUMERO_MATRICOLA),
  CONSTRAINT CHECK_MATRICOLA                              CHECK
(REGEXP_LIKE (NUMERO_MATRICOLA, '^[0-9]{8}+$')),
  CONSTRAINT CHECK_NOME_BIBLIOTECARIO              CHECK
(REGEXP_LIKE(NOME_BIBLIOTECARIO, '^[A-Za-z ]+$')),
  CONSTRAINT CHECK_COGNOME_BIBLIOTECARIO           CHECK
(REGEXP_LIKE(COGNOME_BIBLIOTECARIO, '^[A-Za-z ]+$')),
  CONSTRAINT CHECK_NUMERO_TELEFONO_BIBLIOTECARIO   CHECK
(REGEXP_LIKE(NUMERO_TELEFONO_BIBLIOTECARIO, '^[0-9]{9,10}+$'))
);
```

CLIENTE

La tabella Cliente è popolata al momento dell'ingresso di un cliente in biblioteca, di conseguenza è necessario che i controlli d'integrità dei dati siano inclusi nel DDL.

```
CREATE TABLE CLIENTE
(
  CODICE_FISCALE_CLIENTE        char (16),
  DATA_NASCITA_CLIENTE          date          NOT NULL,
  NOME_CLIENTE          varchar (20)   NOT NULL,
  COGNOME_CLIENTE               varchar (20)   NOT NULL,
  EMAIL_CLIENTE                 varchar (30)   UNIQUE NOT NULL,

  CONSTRAINT PRIMARY_KEY_CLIENTE       PRIMARY KEY (CODICE_FISCALE_CLIENTE),
  CONSTRAINT CHECK_CODICE_FISCALE_CLIENTE          CHECK
(REGEXP_LIKE(CODICE_FISCALE_CLIENTE, '^[A-Z]{6}\d{2}[A-Z]\d{2}[A-Z]\d{3}[A-Z]$')),
  CONSTRAINT CHECK_NOME_CLIENTE                    CHECK
(REGEXP_LIKE(NOME_CLIENTE, '^[A-Za-z]+$')),
  CONSTRAINT CHECK_COGNOME_CLIENTE                 CHECK
(REGEXP_LIKE(COGNOME_CLIENTE, '^[A-Za-z ]+$')),
  CONSTRAINT CHECK_EMAIL_CLIENTE                   CHECK
(EMAIL_CLIENTE LIKE '%@%.%' AND EMAIL_CLIENTE NOT LIKE '@%' AND EMAIL_CLIENTE NOT
LIKE '%@%@%')
);
```

EVENTO

La tabella Evento è popolata tramite inserimento diretto dal direttore, motivo per il quale è necessario che i controlli d'integrità dei dati siano inclusi nel DDL.
Come si può notare, l'attributo Data_Evento, che nel modello EE/R era indicato come chiave primaria, nel DDL è concatenato all'attributo Ora_Evento.
Questo permetterebbe di avere più eventi in un giorno, contraddicendo quanto dichiarato precedentemente.
Per evitare questo problema è stato definito un vincolo di univocità che garantisca che le date degli eventi siano univoche.

```
CREATE TABLE EVENTO
(
  DATA_E_ORA_EVENTO     date,
```

```
  DURATA_EVENTO          number (5, 2)   NOT NULL,
  NOME_EVENTO            varchar (40)    NOT NULL,

  CONSTRAINT PRIMARY_KEY_EVENTO PRIMARY KEY (DATA_E_ORA_EVENTO),
  CONSTRAINT CHECK_DURATA_EVENTO          CHECK
(DURATA_EVENTO IN (1, 1.30, 2)),
  CONSTRAINT CHECK_NOME_EVENTO            CHECK
(REGEXP_LIKE(NOME_EVENTO, '^[A-Za-z ]+$')),
  CONSTRAINT CHECK_INIZIO_EVENTO          CHECK
(TO_CHAR(DATA_E_ORA_EVENTO, 'HH24') BETWEEN 8 AND 17),
  CONSTRAINT CHECK_FINE_EVENTO            CHECK
(TO_NUMBER(TO_CHAR(DATA_E_ORA_EVENTO, 'HH24')) * 60 +
TO_NUMBER(TO_CHAR(DATA_E_ORA_EVENTO, 'MI')) + DURATA_EVENTO * 60 <= 1140),
  CONSTRAINT CHECK_GIORNO_EVENTO          CHECK
(TO_CHAR(DATA_E_ORA_EVENTO, 'D') BETWEEN 1 AND 5)
);

CREATE UNIQUE INDEX DATA_EVENTO_UNIVOCA ON EVENTO (TRUNC(DATA_E_ORA_EVENTO));
```

SCAFFALE

La tabella Scaffale è popolata tramite inserimento diretto dal direttore e aggiornata ogni cinque anni, motivo per il quale è necessario includere i controlli d'integrità dei dati nel DDL.

```
CREATE TABLE SCAFFALE
(
  NUMERO_PIANO            number (1, 0),
  CATEGORIA_SCAFFALE      varchar (20),
  DATA_ACQUISTO_SCAFFALE  date    NOT NULL,

  CONSTRAINT PRIMARY_KEY_SCAFFALE    PRIMARY KEY (NUMERO_PIANO,
CATEGORIA_SCAFFALE),
  CONSTRAINT CHECK_PIANO_SCAFFALE         CHECK     (NUMERO_PIANO IN (1, 2, 3)),
  CONSTRAINT CHECK_CATEGORIA_SCAFFALE     CHECK     (LOWER(CATEGORIA_SCAFFALE) IN
('storia', 'letteratura', 'geografia', 'fantasy', 'scienze', 'giallo', 'romanzo', 'matematica', 'horror'))
);
```

AUTORE

La tabella Autore è popolata dal direttore tramite inserimento diretto, motivo per il quale è necessario includere i controlli d'integrità dei dati nel DDL.

```
CREATE TABLE AUTORE
(
  ISNI                    char (16),
  NOME_AUTORE             varchar (20)   NOT NULL,
  COGNOME_AUTORE          varchar (20)   NOT NULL,

  CONSTRAINT PRIMARY_KEY_AUTORE   PRIMARY KEY (ISNI),
  CONSTRAINT CHECK_ISNI                          CHECK
```

```
(REGEXP_LIKE(ISNI, '^[0-9]{16}+$')),
   CONSTRAINT CHECK_NOME_AUTORE              CHECK
(REGEXP_LIKE(NOME_AUTORE, '^[A-Za-z]+$')),
   CONSTRAINT CHECK_COGNOME_AUTORE           CHECK
(REGEXP_LIKE(COGNOME_AUTORE, '^[A-Za-z ]+$'))
);
```

LIBRO

La tabella Libro è popolata tramite inserimento diretto dal direttore, motivo per il quale è necessario includere i controlli d'integrità dei dati nel DDL.

```
CREATE TABLE LIBRO
(
   ISBN                    char (13),
   TITOLO                  varchar (30)    NOT NULL,
   ANNO_PUBBLICAZIONE      number (4, 0)   NOT NULL,
   GENERE                  varchar (20)    NOT NULL,

   CONSTRAINT PRIMARY_KEY_LIBRO          PRIMARY KEY (ISBN),
   CONSTRAINT CHECK_ISBN_LIBRO           CHECK
(REGEXP_LIKE(ISBN, '^[0-9]{13}+$')),
   CONSTRAINT CHECK_GENERE_LIBRO      CHECK
(LOWER(GENERE) IN ('storia', 'letteratura', 'geografia', 'fantasy', 'scienze', 'giallo', 'romanzo',
'matematica', 'horror'))
);
```

CASA_EDITRICE

La tabella Casa_Editrice è popolata tramite inserimento diretto dal direttore, motivo per il quale è necessario includere i controlli d'integrità dei dati nel DDL.

```
CREATE TABLE CASA_EDITRICE
(
   PARTITA_IVA             char (11),
   NOME_CASA_EDITRICE      varchar (30)      UNIQUE NOT NULL,
   ID_FORNITORE            number (3, 0)     UNIQUE NOT NULL,

   CONSTRAINT PRIMARY_KEY_CASA_EDITRICE PRIMARY KEY (PARTITA_IVA),
   CONSTRAINT FOREIGN_KEY_CASA_EDITRICE FOREIGN KEY (ID_FORNITORE) REFERENCES
FORNITORE (ID_FORNITORE),
   CONSTRAINT CHECK_PARTITA_IVA       CHECK      (REGEXP_LIKE(PARTITA_IVA, '^[0-9]{11}+$'))
);
```

DONATORE

La tabella Donatore è popolata tramite inserimento diretto dal direttore, motivo per il quale è necessario includere i controlli d'integrità dei dati nel DDL.

```
CREATE TABLE DONATORE
(
```

```
   CODICE_FISCALE_DONATORE      char (16),
   NOME_DONATORE                varchar (20)    NOT NULL,
   COGNOME_DONATORE             varchar (20),
   ID_FORNITORE                 number (3, 0)  UNIQUE NOT NULL,

   CONSTRAINT PRIMARY_KEY_DONATORE       PRIMARY KEY (CODICE_FISCALE_DONATORE),
   CONSTRAINT FOREIGN_KEY_DONATORE       FOREIGN KEY (ID_FORNITORE) REFERENCES
FORNITORE (ID_FORNITORE),
   CONSTRAINT CHECK_CODICE_FISCALE_DONATORE       CHECK
(REGEXP_LIKE(CODICE_FISCALE_DONATORE, '^[A-Z]{6}\d{2}[A-Z]\d{2}[A-Z]\d{3}[A-Z]$')),
   CONSTRAINT CHECK_NOME_DONATORE                 CHECK
(REGEXP_LIKE(NOME_DONATORE, '^[A-Za-z]+$')),
   CONSTRAINT CHECK_COGNOME_DONATORE              CHECK
(REGEXP_LIKE(COGNOME_DONATORE, '^[A-Za-z ]+$'))
);
```

TURNO

La tabella Turno è popolata dal direttore tramite inserimento diretto, motivo per il quale è necessario includere i controlli d'integrità dei dati nel DDL.
Come descritto per la tabella Evento, anche in questo caso il concatenamento della data e dell'ora del turno permetterebbe di violare il vincolo stabilito che prevede che un bibliotecario svolga al più un turno al giorno.
Per evitare questo problema è stato definito un vincolo di univocità che garantisca che le combinazioni tra le date dei turni e i numeri di matricola dei bibliotecari che li svolgono siano univoche.

```
CREATE TABLE TURNO
(
   NUMERO_MATRICOLA      char (8),
   DATA_E_ORA_TURNO      date,
   DURATA_TURNO          number (1, 0) NOT NULL,

   CONSTRAINT PRIMARY_KEY_TURNO  PRIMARY KEY (NUMERO_MATRICOLA,
DATA_E_ORA_TURNO),
   CONSTRAINT FOREIGN_KEY_TURNO  FOREIGN KEY (NUMERO_MATRICOLA) REFERENCES
BIBLIOTECARIO (NUMERO_MATRICOLA),
   CONSTRAINT CHECK_DURATA_TURNO     CHECK
(DURATA_TURNO BETWEEN 4 AND 8),
   CONSTRAINT CHECK_INIZIO_TURNO     CHECK
(TO_CHAR(DATA_E_ORA_TURNO, 'HH24') BETWEEN 8 AND 16),
   CONSTRAINT CHECK_FINE_TURNO       CHECK
(TO_NUMBER(TO_CHAR(DATA_E_ORA_TURNO, 'HH24')) * 60 +
TO_NUMBER(TO_CHAR(DATA_E_ORA_TURNO, 'MI')) + DURATA_TURNO * 60 <= 1200),
   CONSTRAINT CHECK_GIORNO_TURNO     CHECK
(TO_CHAR (DATA_E_ORA_TURNO, 'D') BETWEEN 1 AND 5)
);
```

```
CREATE UNIQUE INDEX DATA_TURNO_UNIVOCA ON TURNO (NUMERO_MATRICOLA,
TRUNC(DATA_E_ORA_TURNO));
```

REGISTRAZIONE

La tabella Registrazione è popolata tramite inserimento diretto al momento della registrazione dei clienti ed è aggiornata tramite la procedura Rinnova_Tessera.
Entrambe le azioni sono effettuate da un bibliotecario.
Al fine di evitare errori di inserimento da parte di un bibliotecario al momento dell'inserimento diretto, nonostante i controlli d'integrità dei dati siano inclusi nella procedura, occorre specificarli anche nel DDL.
Si noti l'utilizzo della clausola **UNIQUE** sulla chiave esterna CODICE_FISCALE_CLIENTE per permettere che ad ogni numero di tessera sia associato un solo codice fiscale ed evitare che un cliente possa registrarsi più volte alla biblioteca.

```
CREATE TABLE REGISTRAZIONE
(
   NUMERO_TESSERA              char (8),
   DATA_REGISTRAZIONE          date        NOT NULL,
   DATA_SCADENZA_TESSERA       date        NOT NULL,
   CODICE_FISCALE_CLIENTE      char (16)   UNIQUE NOT NULL,

   CONSTRAINT PRIMARY_KEY_REGISTRAZIONE      PRIMARY KEY (NUMERO_TESSERA),
   CONSTRAINT FOREIGN_KEY_REGISTRAZIONE      FOREIGN KEY (CODICE_FISCALE_CLIENTE)
REFERENCES CLIENTE (CODICE_FISCALE_CLIENTE),
   CONSTRAINT CHECK_NUMERO_TESSERA_REGISTRAZIONE      CHECK
(REGEXP_LIKE(NUMERO_TESSERA, '^[0-9]{8}+$')),
   CONSTRAINT CHECK_SCADENZA_REGISTRAZIONE            CHECK
(DATA_SCADENZA_TESSERA > DATA_REGISTRAZIONE)
);
```

MULTA

La tabella Multa è popolata tramite l'esecuzione della procedura Effettua_Restituzione al momento di una restituzione avvenuta in ritardo.
Per questo motivo, non occorre controllare il formato dell'IUV nel DDL in quanto il controllo sarà effettuato all'interno della procedura.

```
CREATE TABLE MULTA
(
   IUV   char(20),
   NUMERO_TESSERA          char (8),
   DATA_PAGAMENTO          date,

   CONSTRAINT PRIMARY_KEY_MULTA      PRIMARY KEY (IUV, NUMERO_TESSERA),
   CONSTRAINT FOREIGN_KEY_MULTA      FOREIGN KEY (NUMERO_TESSERA) REFERENCES
REGISTRAZIONE (NUMERO_TESSERA)
);
```

MENSOLA

La tabella Mensola è popolata mediante inserimento diretto dal direttore, motivo per il quale è necessario includere i controlli d'integrità dei dati nel DDL.

```
CREATE TABLE MENSOLA
(
   LETTERA_MENSOLA          char (1),
   NUMERO_PIANO             number (1, 0),
   CATEGORIA_SCAFFALE       varchar (20),
   CAPIENZA                 number (3, 0)   NOT NULL,

   CONSTRAINT PRIMARY_KEY_MENSOLA    PRIMARY KEY (LETTERA_MENSOLA, NUMERO_PIANO,
CATEGORIA_SCAFFALE),
   CONSTRAINT FOREIGN_KEY_MENSOLA    FOREIGN KEY (NUMERO_PIANO,
CATEGORIA_SCAFFALE) REFERENCES SCAFFALE (NUMERO_PIANO, CATEGORIA_SCAFFALE),
   CONSTRAINT CHECK_LETTERA_MENSOLA          CHECK
(LOWER(LETTERA_MENSOLA) BETWEEN 'a' AND 'z'),
   CONSTRAINT CHECK_CAPIENZA_MENSOLA         CHECK      (CAPIENZA > 0)
);
```

ORDINE

La tabella Ordine è popolata mediante inserimento diretto o esecuzione della procedura Controlla_Inventario.
Entrambe le azioni sono effettuate dal direttore.
Al fine di evitare errori di inserimento da parte del direttore al momento dell'inserimento diretto, nonostante i controlli d'integrità dei dati siano inclusi nella procedura, occorre specificarli anche nel DDL.

```
CREATE TABLE ORDINE
(
   DATA_ACQUISTO_ORDINE          date,
   ISBN_COPIE_ACQUISTATE         char  (13)    NOT NULL,
   NUMERO_COPIE_ACQUISTATE       number(3, 0)  NOT NULL,
   ID_FORNITORE                  number(3, 0),

   CONSTRAINT PRIMARY_KEY_ORDINE         PRIMARY KEY (DATA_ACQUISTO_ORDINE,
ID_FORNITORE),
   CONSTRAINT FOREIGN_KEY_ORDINE         FOREIGN KEY (ID_FORNITORE) REFERENCES
FORNITORE (ID_FORNITORE),
   CONSTRAINT CHECK_ISBN_ORDINE          CHECK
(REGEXP_LIKE(ISBN_COPIE_ACQUISTATE, '^[0-9]{13}+$')),
   CONSTRAINT CHECK_NUMERO_COPIE_ACQUISTATE CHECK
 (NUMERO_COPIE_ACQUISTATE > 0)
);
```

COPIA

La tabella Copia è popolata dal bibliotecario mediante la procedura Posiziona_Copie.
Siccome i controlli d'integrità dei dati sono effettuati nella procedura, non occorre includerli nel DDL.

```
CREATE TABLE COPIA
(
   ISBN                    char (13),
   NUMERO_COPIA            number (4, 0),
   CONDIZIONE              char (5)        NOT NULL,
   LETTERA_MENSOLA         char (1)        NOT NULL,
   NUMERO_PIANO            number (1, 0)   NOT NULL,
   CATEGORIA_SCAFFALE      varchar (20)    NOT NULL,
   DATA_ACQUISTO_ORDINE    date            NOT NULL,
   ID_FORNITORE            number (3, 0)   NOT NULL,

   CONSTRAINT PRIMARY_KEY_COPIA      PRIMARY KEY (ISBN, NUMERO_COPIA),
   CONSTRAINT FOREIGN_KEY1_COPIA     FOREIGN KEY (ISBN)
REFERENCES LIBRO   (ISBN),
   CONSTRAINT FOREIGN_KEY2_COPIA     FOREIGN KEY (LETTERA_MENSOLA, NUMERO_PIANO,
CATEGORIA_SCAFFALE) REFERENCES MENSOLA (LETTERA_MENSOLA, NUMERO_PIANO,
CATEGORIA_SCAFFALE),
   CONSTRAINT FOREIGN_KEY3_COPIA     FOREIGN KEY (DATA_ACQUISTO_ORDINE,
ID_FORNITORE)          REFERENCES ORDINE  (DATA_ACQUISTO_ORDINE, ID_FORNITORE),
   CONSTRAINT CHECK_ISBN_COPIA                CHECK
(REGEXP_LIKE(ISBN, '^[0-9]{13}+$')),
   CONSTRAINT CHECK_NUMERO_COPIA              CHECK
(NUMERO_COPIA > 0),
   CONSTRAINT CHECK_CONDIZIONE_COPIA          CHECK
(LOWER(CONDIZIONE) IN ('nuovo', 'usato'))
);
```

RECENSIONE

La tabella Recensione è popolata ed aggiornata tramite inserimenti diretti da parte del cliente, motivo per il quale è necessario includere i controlli d'integrità dei dati nel DDL.

```
CREATE TABLE RECENSIONE
(
   DATA_RECENSIONE         date,
   VALUTAZIONE             number (1, 0) NOT NULL,
   ISBN                    char (13)       NOT NULL,
   NUMERO_TESSERA          char (8),

   CONSTRAINT PRIMARY_KEY_RECENSIONE  PRIMARY KEY (ISBN, NUMERO_TESSERA),
   CONSTRAINT FOREIGN_KEY1_RECENSIONE FOREIGN KEY (ISBN) REFERENCES LIBRO (ISBN),
   CONSTRAINT FOREIGN_KEY2_RECENSIONE FOREIGN KEY (NUMERO_TESSERA) REFERENCES
REGISTRAZIONE (NUMERO_TESSERA),
   CONSTRAINT CHECK_VALUTAZIONE      CHECK      (VALUTAZIONE IN (1, 2, 3, 4, 5))
```

);

PRENDE

La tabella Prende è popolata tramite la procedura Effettua_Prestito e aggiornata tramite la procedura Effettua_Restituzione, eseguite da un cliente registrato.
Siccome la maggior parte dei vincoli è gestita all'interno delle procedure, essi sono omessi dal DDL.

```
CREATE TABLE PRENDE
(
   NUMERO_TESSERA              char (8),
   ISBN                        char (13),
   NUMERO_COPIA                number (4, 0),
   DATA_INIZIO_PRESTITO              date      NOT NULL,
   DATA_SCADENZA_PRESTITO      date      NOT NULL,
   DATA_RESTITUZIONE                 date,

   CONSTRAINT PRIMARY_KEY_PRENDE   PRIMARY KEY (NUMERO_TESSERA, ISBN,
NUMERO_COPIA),
   CONSTRAINT FOREIGN_KEY1_PRENDE  FOREIGN KEY (NUMERO_TESSERA)    REFERENCES
REGISTRAZIONE (NUMERO_TESSERA),
   CONSTRAINT FOREIGN_KEY2_PRENDE  FOREIGN KEY (ISBN, NUMERO_COPIA) REFERENCES
COPIA       (ISBN, NUMERO_COPIA)
);
```

SEGUITO

La tabella Seguito è popolata tramite la procedura Prenota_Posto_Evento e aggiornata dalla procedura Disdici_Prenotazione_Posto_Evento eseguite da un cliente registrato.
Siccome la maggior parte dei vincoli è gestita all'interno delle procedure, essi sono omessi dal DDL.

```
CREATE TABLE SEGUITO
(
   NUMERO_TESSERA          char(8),
   DATA_E_ORA_EVENTO       date,

   CONSTRAINT PRIMARY_KEY_SEGUITO  PRIMARY KEY (NUMERO_TESSERA,
DATA_E_ORA_EVENTO),
   CONSTRAINT FOREIGN_KEY1_SEGUITO FOREIGN KEY (NUMERO_TESSERA)     REFERENCES
REGISTRAZIONE (NUMERO_TESSERA),
   CONSTRAINT FOREIGN_KEY2_SEGUITO FOREIGN KEY (DATA_E_ORA_EVENTO) REFERENCES
EVENTO      (DATA_E_ORA_EVENTO)
);
```

SCRITTO

La tabella Scritto è popolata manualmente al momento dell'inserimento di un nuovo libro all'interno della biblioteca.
Non è necessario l'inserimento dei controlli d'integrità dei dati nel DDL, in quanto possiede solamente chiavi esterne.

Per il vincolo d'integrità referenziale, infatti, esse dovranno far riferimento a chiavi primarie esistenti, che quindi per poter essere state inserite avranno rispettato i controlli d'integrità dei dati.

```
CREATE TABLE SCRITTO
(
   ISNI      char(16)    NOT NULL,
   ISBN      char(13)    NOT NULL,

   CONSTRAINT PRIMARY_KEY_SCRITTO  PRIMARY KEY (ISNI, ISBN),
   CONSTRAINT FOREIGN_KEY1_SCRITTO FOREIGN KEY (ISNI) REFERENCES AUTORE (ISNI),
   CONSTRAINT FOREIGN_KEY2_SCRITTO FOREIGN KEY (ISBN) REFERENCES LIBRO  (ISBN)
);
```

ASSISTE

La tabella Assiste è popolata tramite inserimento diretto dal bibliotecario ogni volta che fornisce assistenza ad un cliente.
Come nel caso della tabella Scritto, non è necessario l'inserimento dei controlli d'integrità dei dati nel DDL, in quanto gli unici controlli da effettuare sono sulle chiavi esterne.

```
CREATE TABLE ASSISTE
(
   CODICE_FISCALE_CLIENTE  char(16),
   NUMERO_MATRICOLA        char(8),
   DATA_ASSISTENZA         date       NOT NULL,

   CONSTRAINT PRIMARY_KEY_ASSISTE  PRIMARY KEY (CODICE_FISCALE_CLIENTE,
NUMERO_MATRICOLA),
   CONSTRAINT FOREIGN_KEY1_ASSISTE FOREIGN KEY (CODICE_FISCALE_CLIENTE) REFERENCES
CLIENTE      (CODICE_FISCALE_CLIENTE),
   CONSTRAINT FOREIGN_KEY2_ASSISTE FOREIGN KEY (NUMERO_MATRICOLA)      REFERENCES
BIBLIOTECARIO (NUMERO_MATRICOLA)
);
```

Per gestire gli ID delle tabelle Fornitore, Casa_Editrice e Donatore, e il numero delle multe, sono state utilizzate le seguenti sequenze di auto-increment.

SEQUENZA PER LA TABELLA FORNITORE

```
CREATE SEQUENCE AUTO_INCREMENT_ID_FORNITORE
START WITH 1
INCREMENT BY 1;
```

SEQUENZA PER LE TABELLE CASA_EDITRICE E DONATORE

```
CREATE SEQUENCE AUTO_INCREMENT_ID_CASA_DONATORE
START WITH 1
INCREMENT BY 1;
```

SEQUENZA PER LA TABELLA MULTA

```
CREATE SEQUENCE AUTO_INCREMENT_NUMERO_MULTA
```

```
START WITH 1
INCREMENT BY 1;
```

Per quanto riguarda la gestione dell'attributo Numero Copia della tabella Copia, è stato ritenuto opportuno non utilizzare una sequenza, in quanto sarebbe stato necessario dichiararne una per ogni genere catalogato nella biblioteca, soluzione impensabile per il numero dei generi disponibili e il loro possibile incremento.
L'utilizzo della parola riservata **PRIMARY KEY** permette l'applicazione implicita dei vincoli di univocità, esprimibili agli attributi non scelti come chiave primaria mediante la parola riservata **UNIQUE**, ed obbligatorietà per gli attributi sulla quale è specificato.
Come si può notare è possibile indicare i vincoli statici al momento della creazione delle tabelle, mediante l'utilizzo della clausola **CONSTRAINT**.
Questo perché gli attributi sottoposti a tali vincoli non vedranno variati i loro valori nel tempo.
Al contrario, i vincoli dinamici non sono esprimibili e per farlo occorrerà l'utilizzo dei trigger.

DATA MANIPULATION LANGUAGE

Le seguenti tabelle sono popolate mediante inserimenti diretti

FORNITORE

```
INSERT INTO FORNITORE (ID_FORNITORE, NUMERO_TELEFONO_FORNITORE) VALUES
(AUTO_INCREMENT_ID_FORNITORE.NEXTVAL, '3477274025');
INSERT INTO FORNITORE (ID_FORNITORE, NUMERO_TELEFONO_FORNITORE) VALUES
(AUTO_INCREMENT_ID_FORNITORE.NEXTVAL, '3928347592');
INSERT INTO FORNITORE (ID_FORNITORE, NUMERO_TELEFONO_FORNITORE) VALUES
(AUTO_INCREMENT_ID_FORNITORE.NEXTVAL, '3815962486');
INSERT INTO FORNITORE (ID_FORNITORE, NUMERO_TELEFONO_FORNITORE) VALUES
(AUTO_INCREMENT_ID_FORNITORE.NEXTVAL, '3815962487');
INSERT INTO FORNITORE (ID_FORNITORE, NUMERO_TELEFONO_FORNITORE) VALUES
(AUTO_INCREMENT_ID_FORNITORE.NEXTVAL, '3815962488');
INSERT INTO FORNITORE (ID_FORNITORE, NUMERO_TELEFONO_FORNITORE) VALUES
(AUTO_INCREMENT_ID_FORNITORE.NEXTVAL, '3815962489');
INSERT INTO FORNITORE (ID_FORNITORE, NUMERO_TELEFONO_FORNITORE) VALUES
(AUTO_INCREMENT_ID_FORNITORE.NEXTVAL, '3815962490');
INSERT INTO FORNITORE (ID_FORNITORE, NUMERO_TELEFONO_FORNITORE) VALUES
(AUTO_INCREMENT_ID_FORNITORE.NEXTVAL, '3815962483');
INSERT INTO FORNITORE (ID_FORNITORE, NUMERO_TELEFONO_FORNITORE) VALUES
(AUTO_INCREMENT_ID_FORNITORE.NEXTVAL, '3815962481');
INSERT INTO FORNITORE (ID_FORNITORE, NUMERO_TELEFONO_FORNITORE) VALUES
(AUTO_INCREMENT_ID_FORNITORE.NEXTVAL, '3815962472');
```

BIBLIOTECARIO

```
INSERT INTO BIBLIOTECARIO (NUMERO_MATRICOLA, NOME_BIBLIOTECARIO,
COGNOME_BIBLIOTECARIO, DATA_NASCITA_BIBLIOTECARIO,
NUMERO_TELEFONO_BIBLIOTECARIO) VALUES ('00011234', 'Ciro', 'Esposito', '01-JUN-2003',
'3317575738');
INSERT INTO BIBLIOTECARIO (NUMERO_MATRICOLA, NOME_BIBLIOTECARIO,
COGNOME_BIBLIOTECARIO, DATA_NASCITA_BIBLIOTECARIO,
```

```
NUMERO_TELEFONO_BIBLIOTECARIO) VALUES ('00011235', 'Luisa', 'Pastrugni', '27-FEB-2000',
'3928347392');
INSERT INTO BIBLIOTECARIO (NUMERO_MATRICOLA, NOME_BIBLIOTECARIO,
COGNOME_BIBLIOTECARIO, DATA_NASCITA_BIBLIOTECARIO,
NUMERO_TELEFONO_BIBLIOTECARIO) VALUES ('00011236', 'Mario', 'Rossi', '30-MAR-1980',
'3958347392');
INSERT INTO BIBLIOTECARIO (NUMERO_MATRICOLA, NOME_BIBLIOTECARIO,
COGNOME_BIBLIOTECARIO, DATA_NASCITA_BIBLIOTECARIO,
NUMERO_TELEFONO_BIBLIOTECARIO) VALUES ('00011237', 'Franco', 'Calogero', '25-DEC-2001',
'3925347392');
INSERT INTO BIBLIOTECARIO (NUMERO_MATRICOLA, NOME_BIBLIOTECARIO,
COGNOME_BIBLIOTECARIO, DATA_NASCITA_BIBLIOTECARIO,
NUMERO_TELEFONO_BIBLIOTECARIO) VALUES ('00011238', 'Maria', 'Bianchi', '12-MAR-2001',
'3928347396');
```

CLIENTE

```
INSERT INTO CLIENTE (CODICE_FISCALE_CLIENTE, DATA_NASCITA_CLIENTE, NOME_CLIENTE,
COGNOME_CLIENTE, EMAIL_CLIENTE) VALUES ('RMOGTN03A22F839U', '22-JAN-2003', 'Gaetano',
'Romeo', 'gaetanoromeo03@gmail.com');
INSERT INTO CLIENTE (CODICE_FISCALE_CLIENTE, DATA_NASCITA_CLIENTE, NOME_CLIENTE,
COGNOME_CLIENTE, EMAIL_CLIENTE) VALUES ('PCRGNN02D09H892T', '09-APR-2002', 'Giovanni',
'Picardi', 'giovannipicardi02@gmail.com');
INSERT INTO CLIENTE (CODICE_FISCALE_CLIENTE, DATA_NASCITA_CLIENTE, NOME_CLIENTE,
COGNOME_CLIENTE, EMAIL_CLIENTE) VALUES ('RCPLNZ01C11F839V', '11-MAR-2001', 'Lorenzo',
'Arcopinto', 'lorenzoarcopinto01@gmail.com');
INSERT INTO CLIENTE (CODICE_FISCALE_CLIENTE, DATA_NASCITA_CLIENTE, NOME_CLIENTE,
COGNOME_CLIENTE, EMAIL_CLIENTE) VALUES ('RSSPSQ80H16F839S', '16-JUN-2004', 'Pasquale',
'Rossi', 'pasqualerossi04@gmail.com');
INSERT INTO CLIENTE (CODICE_FISCALE_CLIENTE, DATA_NASCITA_CLIENTE, NOME_CLIENTE,
COGNOME_CLIENTE, EMAIL_CLIENTE) VALUES ('BCHNTN04R09F839Y', '09-OCT-1999', 'Antonio',
'Bianchi', 'antoniobianchi99@gmail.com');
```

EVENTO

```
INSERT INTO EVENTO (DATA_E_ORA_EVENTO, DURATA_EVENTO, NOME_EVENTO) VALUES
(TO_DATE('31/DEC/2022 12:30','dd/mm/yyyy HH24:MI'), 1, 'LABORATORIO LETTURA');
INSERT INTO EVENTO (DATA_E_ORA_EVENTO, DURATA_EVENTO, NOME_EVENTO) VALUES
(TO_DATE('22/MAR/2018 10:00','dd/mm/yyyy HH24:MI'), 1.30, 'LABORATORIO SCRITTURA');
INSERT INTO EVENTO (DATA_E_ORA_EVENTO, DURATA_EVENTO, NOME_EVENTO) VALUES
(TO_DATE('11/JUL/2023 15:00','dd/mm/yyyy HH24:MI'), 1, 'PRESENTAZIONE LIBRO');
INSERT INTO EVENTO (DATA_E_ORA_EVENTO, DURATA_EVENTO, NOME_EVENTO) VALUES
(TO_DATE('19/APR/2019 16:00','dd/mm/yyyy HH24:MI'), 2, 'LETTURA LIBRI STRANIERI');
INSERT INTO EVENTO (DATA_E_ORA_EVENTO, DURATA_EVENTO, NOME_EVENTO) VALUES
(TO_DATE('23/FEB/2018 11:00','dd/mm/yyyy HH24:MI'), 1.30, 'PROIEZIONE DI FILM');
```

SCAFFALE

```
INSERT INTO SCAFFALE (NUMERO_PIANO, CATEGORIA_SCAFFALE, DATA_ACQUISTO_SCAFFALE)
VALUES (1, 'STORIA', '19-APR-2020');
INSERT INTO SCAFFALE (NUMERO_PIANO, CATEGORIA_SCAFFALE, DATA_ACQUISTO_SCAFFALE)
VALUES (2, 'STORIA', '12-OCT-2021');
INSERT INTO SCAFFALE (NUMERO_PIANO, CATEGORIA_SCAFFALE, DATA_ACQUISTO_SCAFFALE)
VALUES (3, 'STORIA', '23-JAN-2021');
INSERT INTO SCAFFALE (NUMERO_PIANO, CATEGORIA_SCAFFALE, DATA_ACQUISTO_SCAFFALE)
VALUES (1, 'LETTERATURA', '07-MAR-2022');
INSERT INTO SCAFFALE (NUMERO_PIANO, CATEGORIA_SCAFFALE, DATA_ACQUISTO_SCAFFALE)
VALUES (2, 'LETTERATURA', '18-JUN-2022');
INSERT INTO SCAFFALE (NUMERO_PIANO, CATEGORIA_SCAFFALE, DATA_ACQUISTO_SCAFFALE)
VALUES (3, 'LETTERATURA', '14-NOV-2019');
INSERT INTO SCAFFALE (NUMERO_PIANO, CATEGORIA_SCAFFALE, DATA_ACQUISTO_SCAFFALE)
VALUES (1, 'GEOGRAFIA', '01-AUG-2021');
INSERT INTO SCAFFALE (NUMERO_PIANO, CATEGORIA_SCAFFALE, DATA_ACQUISTO_SCAFFALE)
VALUES (2, 'GEOGRAFIA', '15-SEP-2020');
INSERT INTO SCAFFALE (NUMERO_PIANO, CATEGORIA_SCAFFALE, DATA_ACQUISTO_SCAFFALE)
VALUES (3, 'GEOGRAFIA', '23-NOV-2021');
INSERT INTO SCAFFALE (NUMERO_PIANO, CATEGORIA_SCAFFALE, DATA_ACQUISTO_SCAFFALE)
VALUES (1, 'SCIENZE', '17-APR-2019');
INSERT INTO SCAFFALE (NUMERO_PIANO, CATEGORIA_SCAFFALE, DATA_ACQUISTO_SCAFFALE)
VALUES (2, 'SCIENZE', '27-FEB-2022');
INSERT INTO SCAFFALE (NUMERO_PIANO, CATEGORIA_SCAFFALE, DATA_ACQUISTO_SCAFFALE)
VALUES (3, 'SCIENZE', '19-OCT-2021');
INSERT INTO SCAFFALE (NUMERO_PIANO, CATEGORIA_SCAFFALE, DATA_ACQUISTO_SCAFFALE)
VALUES (1, 'MATEMATICA', '21-JUN-2020');
INSERT INTO SCAFFALE (NUMERO_PIANO, CATEGORIA_SCAFFALE, DATA_ACQUISTO_SCAFFALE)
VALUES (2, 'MATEMATICA', '04-DEC-2021');
INSERT INTO SCAFFALE (NUMERO_PIANO, CATEGORIA_SCAFFALE, DATA_ACQUISTO_SCAFFALE)
VALUES (3, 'MATEMATICA', '02-FEB-2023');
INSERT INTO SCAFFALE (NUMERO_PIANO, CATEGORIA_SCAFFALE, DATA_ACQUISTO_SCAFFALE)
VALUES (1, 'HORROR', '18-SEP-2019');
INSERT INTO SCAFFALE (NUMERO_PIANO, CATEGORIA_SCAFFALE, DATA_ACQUISTO_SCAFFALE)
VALUES (2, 'HORROR', '06-AUG-2021');
INSERT INTO SCAFFALE (NUMERO_PIANO, CATEGORIA_SCAFFALE, DATA_ACQUISTO_SCAFFALE)
VALUES (3, 'HORROR', '26-DEC-2022');
INSERT INTO SCAFFALE (NUMERO_PIANO, CATEGORIA_SCAFFALE, DATA_ACQUISTO_SCAFFALE)
VALUES (1, 'FANTASY', '28-MAR-2023');
INSERT INTO SCAFFALE (NUMERO_PIANO, CATEGORIA_SCAFFALE, DATA_ACQUISTO_SCAFFALE)
VALUES (2, 'FANTASY', '15-APR-2019');
INSERT INTO SCAFFALE (NUMERO_PIANO, CATEGORIA_SCAFFALE, DATA_ACQUISTO_SCAFFALE)
VALUES (3, 'FANTASY', '19-JUN-2019');
INSERT INTO SCAFFALE (NUMERO_PIANO, CATEGORIA_SCAFFALE, DATA_ACQUISTO_SCAFFALE)
VALUES (1, 'GIALLO', '07-JUL-2021');
INSERT INTO SCAFFALE (NUMERO_PIANO, CATEGORIA_SCAFFALE, DATA_ACQUISTO_SCAFFALE)
VALUES (2, 'GIALLO', '09-MAR-2020');
```

```
INSERT INTO SCAFFALE (NUMERO_PIANO, CATEGORIA_SCAFFALE, DATA_ACQUISTO_SCAFFALE)
VALUES (3, 'GIALLO', '28-JAN-2023');
INSERT INTO SCAFFALE (NUMERO_PIANO, CATEGORIA_SCAFFALE, DATA_ACQUISTO_SCAFFALE)
VALUES (1, 'ROMANZO', '21-FEB-2017');
INSERT INTO SCAFFALE (NUMERO_PIANO, CATEGORIA_SCAFFALE, DATA_ACQUISTO_SCAFFALE)
VALUES (2, 'ROMANZO', '18-APR-2018');
INSERT INTO SCAFFALE (NUMERO_PIANO, CATEGORIA_SCAFFALE, DATA_ACQUISTO_SCAFFALE)
VALUES (3, 'ROMANZO', '19-DEC-2017');
```

AUTORE

```
INSERT INTO AUTORE (ISNI, NOME_AUTORE, COGNOME_AUTORE) VALUES ('0000000123456789',
'Virginia', 'Woolf');
INSERT INTO AUTORE (ISNI, NOME_AUTORE, COGNOME_AUTORE) VALUES ('0000001123456789',
'Ugo', 'Foscolo');
INSERT INTO AUTORE (ISNI, NOME_AUTORE, COGNOME_AUTORE) VALUES ('0000011123456789',
'Dante', 'Alighieri');
INSERT INTO AUTORE (ISNI, NOME_AUTORE, COGNOME_AUTORE) VALUES ('0000111123456789',
'Mark', 'Twain');
INSERT INTO AUTORE (ISNI, NOME_AUTORE, COGNOME_AUTORE) VALUES ('0001111123456789',
'Oscar', 'Wilde');
```

LIBRO

```
INSERT INTO LIBRO (ISBN, TITOLO, ANNO_PUBBLICAZIONE, GENERE) VALUES ('1234567890123',
'BREVE STORIA DEL TEMPO', 2004, 'SCIENZE');
INSERT INTO LIBRO (ISBN, TITOLO, ANNO_PUBBLICAZIONE, GENERE) VALUES ('1234567890134',
'CALCOLO', 2011, 'MATEMATICA');
INSERT INTO LIBRO (ISBN, TITOLO, ANNO_PUBBLICAZIONE, GENERE) VALUES ('1234567890147',
'TUFFO NEL PASSATO', 2017, 'STORIA');
INSERT INTO LIBRO (ISBN, TITOLO, ANNO_PUBBLICAZIONE, GENERE) VALUES ('1234567890179',
'STORIA ITALIANA', 2017, 'STORIA');
INSERT INTO LIBRO (ISBN, TITOLO, ANNO_PUBBLICAZIONE, GENERE) VALUES ('1234567890159',
'GEOGRAFANDO', 2014, 'GEOGRAFIA');
```

CASA EDITRICE

```
INSERT INTO CASA_EDITRICE (PARTITA_IVA, NOME_CASA_EDITRICE, ID_FORNITORE) VALUES
('12345678901', 'MONDADORI', AUTO_INCREMENT_ID_CASA_DONATORE.NEXTVAL);
INSERT INTO CASA_EDITRICE (PARTITA_IVA, NOME_CASA_EDITRICE, ID_FORNITORE) VALUES
('12345678902', 'FELTRINELLI', AUTO_INCREMENT_ID_CASA_DONATORE.NEXTVAL);
INSERT INTO CASA_EDITRICE (PARTITA_IVA, NOME_CASA_EDITRICE, ID_FORNITORE) VALUES
('12345678903', 'DISNEY', AUTO_INCREMENT_ID_CASA_DONATORE.NEXTVAL);
INSERT INTO CASA_EDITRICE (PARTITA_IVA, NOME_CASA_EDITRICE, ID_FORNITORE) VALUES
('12345678904', 'NERBINI', AUTO_INCREMENT_ID_CASA_DONATORE.NEXTVAL);
INSERT INTO CASA_EDITRICE (PARTITA_IVA, NOME_CASA_EDITRICE, ID_FORNITORE) VALUES
('12345678905', 'DE AGOSTINI', AUTO_INCREMENT_ID_CASA_DONATORE.NEXTVAL);
```

DONATORE

```
INSERT INTO DONATORE (CODICE_FISCALE_DONATORE, NOME_DONATORE,
COGNOME_DONATORE, ID_FORNITORE) VALUES ('EGGMRT22T33F839U', 'EGGIDIO', 'MURATORI',
AUTO_INCREMENT_ID_CASA_DONATORE.NEXTVAL);
INSERT INTO DONATORE (CODICE_FISCALE_DONATORE, NOME_DONATORE,
COGNOME_DONATORE, ID_FORNITORE) VALUES ('FRCCLN47J21L847F', 'FRANCO', 'CALONE',
AUTO_INCREMENT_ID_CASA_DONATORE.NEXTVAL);
INSERT INTO DONATORE (CODICE_FISCALE_DONATORE, NOME_DONATORE,
COGNOME_DONATORE, ID_FORNITORE) VALUES ('NTNSPS07N44Q039L', 'ANTONIO', 'ESPOSITO',
AUTO_INCREMENT_ID_CASA_DONATORE.NEXTVAL);
INSERT INTO DONATORE (CODICE_FISCALE_DONATORE, NOME_DONATORE,
COGNOME_DONATORE, ID_FORNITORE) VALUES ('GLAPLI12J55A817O', 'GIULIA', 'PAOLI',
AUTO_INCREMENT_ID_CASA_DONATORE.NEXTVAL);
INSERT INTO DONATORE (CODICE_FISCALE_DONATORE, NOME_DONATORE,
COGNOME_DONATORE, ID_FORNITORE) VALUES ('ASNTCC01N14Y031I', 'ASIA', 'ANTONUCCI',
AUTO_INCREMENT_ID_CASA_DONATORE.NEXTVAL);
```

TURNO

```
INSERT INTO TURNO (NUMERO_MATRICOLA, DATA_E_ORA_TURNO, DURATA_TURNO) VALUES
('00011234', TO_DATE('12/DEC/2021 08:00','dd/mm/yyyy HH24:MI'), 8);
INSERT INTO TURNO (NUMERO_MATRICOLA, DATA_E_ORA_TURNO, DURATA_TURNO) VALUES
('00011235', TO_DATE('28/MAR/2022 14:00','dd/mm/yyyy HH24:MI'), 6);
INSERT INTO TURNO (NUMERO_MATRICOLA, DATA_E_ORA_TURNO, DURATA_TURNO) VALUES
('00011236', TO_DATE('14/APR/2023 10:00','dd/mm/yyyy HH24:MI'), 6);
INSERT INTO TURNO (NUMERO_MATRICOLA, DATA_E_ORA_TURNO, DURATA_TURNO) VALUES
('00011237', TO_DATE('07/OCT/2020 11:00','dd/mm/yyyy HH24:MI'), 4);
INSERT INTO TURNO (NUMERO_MATRICOLA, DATA_E_ORA_TURNO, DURATA_TURNO) VALUES
('00011238', TO_DATE('21/APR/2022 12:00','dd/mm/yyyy HH24:MI'), 5);
```

REGISTRAZIONE

```
INSERT INTO REGISTRAZIONE (NUMERO_TESSERA, DATA_REGISTRAZIONE,
DATA_SCADENZA_TESSERA, CODICE_FISCALE_CLIENTE) VALUES ('12345678', '15-DEC-2023', '15-
DEC-2024', 'RMOGTN03A22F839U');
INSERT INTO REGISTRAZIONE (NUMERO_TESSERA, DATA_REGISTRAZIONE,
DATA_SCADENZA_TESSERA, CODICE_FISCALE_CLIENTE) VALUES ('12345679', '01-FEB-2023', '01-
FEB-2024', 'PCRGNN02D09H892T');
INSERT INTO REGISTRAZIONE (NUMERO_TESSERA, DATA_REGISTRAZIONE,
DATA_SCADENZA_TESSERA, CODICE_FISCALE_CLIENTE) VALUES ('12345670', '19-APR-2023', '19-
APR-2024', 'RCPLNZ01C11F839V');
INSERT INTO REGISTRAZIONE (NUMERO_TESSERA, DATA_REGISTRAZIONE,
DATA_SCADENZA_TESSERA, CODICE_FISCALE_CLIENTE) VALUES ('12345680', '15-JAN-2023', '15-
JAN-2024', 'RSSPSQ80H16F839S');
INSERT INTO REGISTRAZIONE (NUMERO_TESSERA, DATA_REGISTRAZIONE,
DATA_SCADENZA_TESSERA, CODICE_FISCALE_CLIENTE) VALUES ('12345681', '01-MAY-2022', '01-
MAY-2023', 'BCHNTN04R09F839Y');
```

MENSOLA

```
INSERT INTO MENSOLA (LETTERA_MENSOLA, NUMERO_PIANO, CATEGORIA_SCAFFALE,
CAPIENZA) VALUES ('A', 1, 'STORIA', 29);
INSERT INTO MENSOLA (LETTERA_MENSOLA, NUMERO_PIANO, CATEGORIA_SCAFFALE,
CAPIENZA) VALUES ('B', 1, 'STORIA', 37);
INSERT INTO MENSOLA (LETTERA_MENSOLA, NUMERO_PIANO, CATEGORIA_SCAFFALE,
CAPIENZA) VALUES ('C', 1, 'STORIA', 29);
INSERT INTO MENSOLA (LETTERA_MENSOLA, NUMERO_PIANO, CATEGORIA_SCAFFALE,
CAPIENZA) VALUES ('D', 1, 'STORIA', 35);
INSERT INTO MENSOLA (LETTERA_MENSOLA, NUMERO_PIANO, CATEGORIA_SCAFFALE,
CAPIENZA) VALUES ('E', 1, 'STORIA', 37);
INSERT INTO MENSOLA (LETTERA_MENSOLA, NUMERO_PIANO, CATEGORIA_SCAFFALE,
CAPIENZA) VALUES ('F', 1, 'STORIA', 29);
INSERT INTO MENSOLA (LETTERA_MENSOLA, NUMERO_PIANO, CATEGORIA_SCAFFALE,
CAPIENZA) VALUES ('G', 1, 'STORIA', 35);
INSERT INTO MENSOLA (LETTERA_MENSOLA, NUMERO_PIANO, CATEGORIA_SCAFFALE,
CAPIENZA) VALUES ('H', 1, 'STORIA', 37);
INSERT INTO MENSOLA (LETTERA_MENSOLA, NUMERO_PIANO, CATEGORIA_SCAFFALE,
CAPIENZA) VALUES ('I', 2, 'STORIA', 35);
INSERT INTO MENSOLA (LETTERA_MENSOLA, NUMERO_PIANO, CATEGORIA_SCAFFALE,
CAPIENZA) VALUES ('J', 2, 'STORIA', 37);
INSERT INTO MENSOLA (LETTERA_MENSOLA, NUMERO_PIANO, CATEGORIA_SCAFFALE,
CAPIENZA) VALUES ('K', 2, 'STORIA', 29);
INSERT INTO MENSOLA (LETTERA_MENSOLA, NUMERO_PIANO, CATEGORIA_SCAFFALE,
CAPIENZA) VALUES ('L', 2, 'STORIA', 35);
INSERT INTO MENSOLA (LETTERA_MENSOLA, NUMERO_PIANO, CATEGORIA_SCAFFALE,
CAPIENZA) VALUES ('M', 2, 'STORIA', 37);
INSERT INTO MENSOLA (LETTERA_MENSOLA, NUMERO_PIANO, CATEGORIA_SCAFFALE,
CAPIENZA) VALUES ('N', 2, 'STORIA', 29);
INSERT INTO MENSOLA (LETTERA_MENSOLA, NUMERO_PIANO, CATEGORIA_SCAFFALE,
CAPIENZA) VALUES ('O', 2, 'STORIA', 35);
INSERT INTO MENSOLA (LETTERA_MENSOLA, NUMERO_PIANO, CATEGORIA_SCAFFALE,
CAPIENZA) VALUES ('P', 2, 'STORIA', 37);
INSERT INTO MENSOLA (LETTERA_MENSOLA, NUMERO_PIANO, CATEGORIA_SCAFFALE,
CAPIENZA) VALUES ('Q', 2, 'STORIA', 37);
INSERT INTO MENSOLA (LETTERA_MENSOLA, NUMERO_PIANO, CATEGORIA_SCAFFALE,
CAPIENZA) VALUES ('R', 3, 'STORIA', 35);
INSERT INTO MENSOLA (LETTERA_MENSOLA, NUMERO_PIANO, CATEGORIA_SCAFFALE,
CAPIENZA) VALUES ('S', 3, 'STORIA', 37);
INSERT INTO MENSOLA (LETTERA_MENSOLA, NUMERO_PIANO, CATEGORIA_SCAFFALE,
CAPIENZA) VALUES ('T', 3, 'STORIA', 33);
INSERT INTO MENSOLA (LETTERA_MENSOLA, NUMERO_PIANO, CATEGORIA_SCAFFALE,
CAPIENZA) VALUES ('U', 3, 'STORIA', 35);
INSERT INTO MENSOLA (LETTERA_MENSOLA, NUMERO_PIANO, CATEGORIA_SCAFFALE,
CAPIENZA) VALUES ('V', 3, 'STORIA', 37);
INSERT INTO MENSOLA (LETTERA_MENSOLA, NUMERO_PIANO, CATEGORIA_SCAFFALE,
CAPIENZA) VALUES ('W', 3, 'STORIA', 29);
```

```
INSERT INTO MENSOLA (LETTERA_MENSOLA, NUMERO_PIANO, CATEGORIA_SCAFFALE,
CAPIENZA) VALUES ('X', 3, 'STORIA', 35);
INSERT INTO MENSOLA (LETTERA_MENSOLA, NUMERO_PIANO, CATEGORIA_SCAFFALE,
CAPIENZA) VALUES ('Y', 3, 'STORIA', 37);
INSERT INTO MENSOLA (LETTERA_MENSOLA, NUMERO_PIANO, CATEGORIA_SCAFFALE,
CAPIENZA) VALUES ('Z', 3, 'STORIA', 37);
INSERT INTO MENSOLA (LETTERA_MENSOLA, NUMERO_PIANO, CATEGORIA_SCAFFALE,
CAPIENZA) VALUES ('A', 1, 'LETTERATURA', 35);
INSERT INTO MENSOLA (LETTERA_MENSOLA, NUMERO_PIANO, CATEGORIA_SCAFFALE,
CAPIENZA) VALUES ('B', 1, 'LETTERATURA', 37);
INSERT INTO MENSOLA (LETTERA_MENSOLA, NUMERO_PIANO, CATEGORIA_SCAFFALE,
CAPIENZA) VALUES ('C', 1, 'LETTERATURA', 29);
INSERT INTO MENSOLA (LETTERA_MENSOLA, NUMERO_PIANO, CATEGORIA_SCAFFALE,
CAPIENZA) VALUES ('D', 1, 'LETTERATURA', 35);
INSERT INTO MENSOLA (LETTERA_MENSOLA, NUMERO_PIANO, CATEGORIA_SCAFFALE,
CAPIENZA) VALUES ('E', 1, 'LETTERATURA', 37);
INSERT INTO MENSOLA (LETTERA_MENSOLA, NUMERO_PIANO, CATEGORIA_SCAFFALE,
CAPIENZA) VALUES ('F', 1, 'LETTERATURA', 29);
INSERT INTO MENSOLA (LETTERA_MENSOLA, NUMERO_PIANO, CATEGORIA_SCAFFALE,
CAPIENZA) VALUES ('G', 1, 'LETTERATURA', 35);
INSERT INTO MENSOLA (LETTERA_MENSOLA, NUMERO_PIANO, CATEGORIA_SCAFFALE,
CAPIENZA) VALUES ('H', 1, 'LETTERATURA', 37);
INSERT INTO MENSOLA (LETTERA_MENSOLA, NUMERO_PIANO, CATEGORIA_SCAFFALE,
CAPIENZA) VALUES ('I', 2, 'LETTERATURA', 35);
INSERT INTO MENSOLA (LETTERA_MENSOLA, NUMERO_PIANO, CATEGORIA_SCAFFALE,
CAPIENZA) VALUES ('J', 2, 'LETTERATURA', 37);
INSERT INTO MENSOLA (LETTERA_MENSOLA, NUMERO_PIANO, CATEGORIA_SCAFFALE,
CAPIENZA) VALUES ('K', 2, 'LETTERATURA', 29);
INSERT INTO MENSOLA (LETTERA_MENSOLA, NUMERO_PIANO, CATEGORIA_SCAFFALE,
CAPIENZA) VALUES ('L', 2, 'LETTERATURA', 35);
INSERT INTO MENSOLA (LETTERA_MENSOLA, NUMERO_PIANO, CATEGORIA_SCAFFALE,
CAPIENZA) VALUES ('M', 2, 'LETTERATURA', 37);
INSERT INTO MENSOLA (LETTERA_MENSOLA, NUMERO_PIANO, CATEGORIA_SCAFFALE,
CAPIENZA) VALUES ('N', 2, 'LETTERATURA', 29);
INSERT INTO MENSOLA (LETTERA_MENSOLA, NUMERO_PIANO, CATEGORIA_SCAFFALE,
CAPIENZA) VALUES ('O', 2, 'LETTERATURA', 35);
INSERT INTO MENSOLA (LETTERA_MENSOLA, NUMERO_PIANO, CATEGORIA_SCAFFALE,
CAPIENZA) VALUES ('P', 2, 'LETTERATURA', 37);
INSERT INTO MENSOLA (LETTERA_MENSOLA, NUMERO_PIANO, CATEGORIA_SCAFFALE,
CAPIENZA) VALUES ('Q', 2, 'LETTERATURA', 37);
INSERT INTO MENSOLA (LETTERA_MENSOLA, NUMERO_PIANO, CATEGORIA_SCAFFALE,
CAPIENZA) VALUES ('R', 3, 'LETTERATURA', 35);
INSERT INTO MENSOLA (LETTERA_MENSOLA, NUMERO_PIANO, CATEGORIA_SCAFFALE,
CAPIENZA) VALUES ('S', 3, 'LETTERATURA', 37);
INSERT INTO MENSOLA (LETTERA_MENSOLA, NUMERO_PIANO, CATEGORIA_SCAFFALE,
CAPIENZA) VALUES ('T', 3, 'LETTERATURA', 29);
INSERT INTO MENSOLA (LETTERA_MENSOLA, NUMERO_PIANO, CATEGORIA_SCAFFALE,
CAPIENZA) VALUES ('U', 3, 'LETTERATURA', 35);
```

```
INSERT INTO MENSOLA (LETTERA_MENSOLA, NUMERO_PIANO, CATEGORIA_SCAFFALE,
CAPIENZA) VALUES ('V', 3, 'LETTERATURA', 37);
INSERT INTO MENSOLA (LETTERA_MENSOLA, NUMERO_PIANO, CATEGORIA_SCAFFALE,
CAPIENZA) VALUES ('W', 3, 'LETTERATURA', 29);
INSERT INTO MENSOLA (LETTERA_MENSOLA, NUMERO_PIANO, CATEGORIA_SCAFFALE,
CAPIENZA) VALUES ('X', 3, 'LETTERATURA', 35);
INSERT INTO MENSOLA (LETTERA_MENSOLA, NUMERO_PIANO, CATEGORIA_SCAFFALE,
CAPIENZA) VALUES ('Y', 3, 'LETTERATURA', 37);
INSERT INTO MENSOLA (LETTERA_MENSOLA, NUMERO_PIANO, CATEGORIA_SCAFFALE,
CAPIENZA) VALUES ('Z', 3, 'LETTERATURA', 37);
INSERT INTO MENSOLA (LETTERA_MENSOLA, NUMERO_PIANO, CATEGORIA_SCAFFALE,
CAPIENZA) VALUES ('A', 1, 'GEOGRAFIA', 35);
INSERT INTO MENSOLA (LETTERA_MENSOLA, NUMERO_PIANO, CATEGORIA_SCAFFALE,
CAPIENZA) VALUES ('B', 1, 'GEOGRAFIA', 37);
INSERT INTO MENSOLA (LETTERA_MENSOLA, NUMERO_PIANO, CATEGORIA_SCAFFALE,
CAPIENZA) VALUES ('C', 1, 'GEOGRAFIA', 29);
INSERT INTO MENSOLA (LETTERA_MENSOLA, NUMERO_PIANO, CATEGORIA_SCAFFALE,
CAPIENZA) VALUES ('D', 1, 'GEOGRAFIA', 35);
INSERT INTO MENSOLA (LETTERA_MENSOLA, NUMERO_PIANO, CATEGORIA_SCAFFALE,
CAPIENZA) VALUES ('E', 1, 'GEOGRAFIA', 37);
INSERT INTO MENSOLA (LETTERA_MENSOLA, NUMERO_PIANO, CATEGORIA_SCAFFALE,
CAPIENZA) VALUES ('F', 1, 'GEOGRAFIA', 29);
INSERT INTO MENSOLA (LETTERA_MENSOLA, NUMERO_PIANO, CATEGORIA_SCAFFALE,
CAPIENZA) VALUES ('G', 1, 'GEOGRAFIA', 35);
INSERT INTO MENSOLA (LETTERA_MENSOLA, NUMERO_PIANO, CATEGORIA_SCAFFALE,
CAPIENZA) VALUES ('H', 1, 'GEOGRAFIA', 37);
INSERT INTO MENSOLA (LETTERA_MENSOLA, NUMERO_PIANO, CATEGORIA_SCAFFALE,
CAPIENZA) VALUES ('I', 2, 'GEOGRAFIA', 35);
INSERT INTO MENSOLA (LETTERA_MENSOLA, NUMERO_PIANO, CATEGORIA_SCAFFALE,
CAPIENZA) VALUES ('J', 2, 'GEOGRAFIA', 37);
INSERT INTO MENSOLA (LETTERA_MENSOLA, NUMERO_PIANO, CATEGORIA_SCAFFALE,
CAPIENZA) VALUES ('K', 2, 'GEOGRAFIA', 29);
INSERT INTO MENSOLA (LETTERA_MENSOLA, NUMERO_PIANO, CATEGORIA_SCAFFALE,
CAPIENZA) VALUES ('L', 2, 'GEOGRAFIA', 35);
INSERT INTO MENSOLA (LETTERA_MENSOLA, NUMERO_PIANO, CATEGORIA_SCAFFALE,
CAPIENZA) VALUES ('M', 2, 'GEOGRAFIA', 37);
INSERT INTO MENSOLA (LETTERA_MENSOLA, NUMERO_PIANO, CATEGORIA_SCAFFALE,
CAPIENZA) VALUES ('N', 2, 'GEOGRAFIA', 29);
INSERT INTO MENSOLA (LETTERA_MENSOLA, NUMERO_PIANO, CATEGORIA_SCAFFALE,
CAPIENZA) VALUES ('O', 2, 'GEOGRAFIA', 35);
INSERT INTO MENSOLA (LETTERA_MENSOLA, NUMERO_PIANO, CATEGORIA_SCAFFALE,
CAPIENZA) VALUES ('P', 2, 'GEOGRAFIA', 37);
INSERT INTO MENSOLA (LETTERA_MENSOLA, NUMERO_PIANO, CATEGORIA_SCAFFALE,
CAPIENZA) VALUES ('Q', 2, 'GEOGRAFIA', 37);
INSERT INTO MENSOLA (LETTERA_MENSOLA, NUMERO_PIANO, CATEGORIA_SCAFFALE,
CAPIENZA) VALUES ('R', 3, 'GEOGRAFIA', 35);
INSERT INTO MENSOLA (LETTERA_MENSOLA, NUMERO_PIANO, CATEGORIA_SCAFFALE,
CAPIENZA) VALUES ('S', 3, 'GEOGRAFIA', 37);
```

```
INSERT INTO MENSOLA (LETTERA_MENSOLA, NUMERO_PIANO, CATEGORIA_SCAFFALE,
CAPIENZA) VALUES ('T', 3, 'GEOGRAFIA', 29);
INSERT INTO MENSOLA (LETTERA_MENSOLA, NUMERO_PIANO, CATEGORIA_SCAFFALE,
CAPIENZA) VALUES ('U', 3, 'GEOGRAFIA', 35);
INSERT INTO MENSOLA (LETTERA_MENSOLA, NUMERO_PIANO, CATEGORIA_SCAFFALE,
CAPIENZA) VALUES ('V', 3, 'GEOGRAFIA', 37);
INSERT INTO MENSOLA (LETTERA_MENSOLA, NUMERO_PIANO, CATEGORIA_SCAFFALE,
CAPIENZA) VALUES ('W', 3, 'GEOGRAFIA', 29);
INSERT INTO MENSOLA (LETTERA_MENSOLA, NUMERO_PIANO, CATEGORIA_SCAFFALE,
CAPIENZA) VALUES ('X', 3, 'GEOGRAFIA', 35);
INSERT INTO MENSOLA (LETTERA_MENSOLA, NUMERO_PIANO, CATEGORIA_SCAFFALE,
CAPIENZA) VALUES ('Y', 3, 'GEOGRAFIA', 37);
INSERT INTO MENSOLA (LETTERA_MENSOLA, NUMERO_PIANO, CATEGORIA_SCAFFALE,
CAPIENZA) VALUES ('Z', 3, 'GEOGRAFIA', 37);
INSERT INTO MENSOLA (LETTERA_MENSOLA, NUMERO_PIANO, CATEGORIA_SCAFFALE,
CAPIENZA) VALUES ('A', 1, 'SCIENZE', 35);
INSERT INTO MENSOLA (LETTERA_MENSOLA, NUMERO_PIANO, CATEGORIA_SCAFFALE,
CAPIENZA) VALUES ('B', 1, 'SCIENZE', 37);
INSERT INTO MENSOLA (LETTERA_MENSOLA, NUMERO_PIANO, CATEGORIA_SCAFFALE,
CAPIENZA) VALUES ('C', 1, 'SCIENZE', 29);
INSERT INTO MENSOLA (LETTERA_MENSOLA, NUMERO_PIANO, CATEGORIA_SCAFFALE,
CAPIENZA) VALUES ('D', 1, 'SCIENZE', 35);
INSERT INTO MENSOLA (LETTERA_MENSOLA, NUMERO_PIANO, CATEGORIA_SCAFFALE,
CAPIENZA) VALUES ('E', 1, 'SCIENZE', 37);
INSERT INTO MENSOLA (LETTERA_MENSOLA, NUMERO_PIANO, CATEGORIA_SCAFFALE,
CAPIENZA) VALUES ('F', 1, 'SCIENZE', 29);
INSERT INTO MENSOLA (LETTERA_MENSOLA, NUMERO_PIANO, CATEGORIA_SCAFFALE,
CAPIENZA) VALUES ('G', 1, 'SCIENZE', 35);
INSERT INTO MENSOLA (LETTERA_MENSOLA, NUMERO_PIANO, CATEGORIA_SCAFFALE,
CAPIENZA) VALUES ('H', 1, 'SCIENZE', 37);
INSERT INTO MENSOLA (LETTERA_MENSOLA, NUMERO_PIANO, CATEGORIA_SCAFFALE,
CAPIENZA) VALUES ('I', 2, 'SCIENZE', 35);
INSERT INTO MENSOLA (LETTERA_MENSOLA, NUMERO_PIANO, CATEGORIA_SCAFFALE,
CAPIENZA) VALUES ('J', 2, 'SCIENZE', 37);
INSERT INTO MENSOLA (LETTERA_MENSOLA, NUMERO_PIANO, CATEGORIA_SCAFFALE,
CAPIENZA) VALUES ('K', 2, 'SCIENZE', 29);
INSERT INTO MENSOLA (LETTERA_MENSOLA, NUMERO_PIANO, CATEGORIA_SCAFFALE,
CAPIENZA) VALUES ('L', 2, 'SCIENZE', 35);
INSERT INTO MENSOLA (LETTERA_MENSOLA, NUMERO_PIANO, CATEGORIA_SCAFFALE,
CAPIENZA) VALUES ('M', 2, 'SCIENZE', 37);
INSERT INTO MENSOLA (LETTERA_MENSOLA, NUMERO_PIANO, CATEGORIA_SCAFFALE,
CAPIENZA) VALUES ('N', 2, 'SCIENZE', 29);
INSERT INTO MENSOLA (LETTERA_MENSOLA, NUMERO_PIANO, CATEGORIA_SCAFFALE,
CAPIENZA) VALUES ('O', 2, 'SCIENZE', 35);
INSERT INTO MENSOLA (LETTERA_MENSOLA, NUMERO_PIANO, CATEGORIA_SCAFFALE,
CAPIENZA) VALUES ('P', 2, 'SCIENZE', 37);
INSERT INTO MENSOLA (LETTERA_MENSOLA, NUMERO_PIANO, CATEGORIA_SCAFFALE,
CAPIENZA) VALUES ('Q', 2, 'SCIENZE', 37);
```

```
INSERT INTO MENSOLA (LETTERA_MENSOLA, NUMERO_PIANO, CATEGORIA_SCAFFALE,
CAPIENZA) VALUES ('R', 3, 'SCIENZE', 35);
INSERT INTO MENSOLA (LETTERA_MENSOLA, NUMERO_PIANO, CATEGORIA_SCAFFALE,
CAPIENZA) VALUES ('S', 3, 'SCIENZE', 37);
INSERT INTO MENSOLA (LETTERA_MENSOLA, NUMERO_PIANO, CATEGORIA_SCAFFALE,
CAPIENZA) VALUES ('T', 3, 'SCIENZE', 29);
INSERT INTO MENSOLA (LETTERA_MENSOLA, NUMERO_PIANO, CATEGORIA_SCAFFALE,
CAPIENZA) VALUES ('U', 3, 'SCIENZE', 35);
INSERT INTO MENSOLA (LETTERA_MENSOLA, NUMERO_PIANO, CATEGORIA_SCAFFALE,
CAPIENZA) VALUES ('V', 3, 'SCIENZE', 37);
INSERT INTO MENSOLA (LETTERA_MENSOLA, NUMERO_PIANO, CATEGORIA_SCAFFALE,
CAPIENZA) VALUES ('W', 3, 'SCIENZE', 29);
INSERT INTO MENSOLA (LETTERA_MENSOLA, NUMERO_PIANO, CATEGORIA_SCAFFALE,
CAPIENZA) VALUES ('X', 3, 'SCIENZE', 35);
INSERT INTO MENSOLA (LETTERA_MENSOLA, NUMERO_PIANO, CATEGORIA_SCAFFALE,
CAPIENZA) VALUES ('Y', 3, 'SCIENZE', 37);
INSERT INTO MENSOLA (LETTERA_MENSOLA, NUMERO_PIANO, CATEGORIA_SCAFFALE,
CAPIENZA) VALUES ('Z', 3, 'SCIENZE', 37);
INSERT INTO MENSOLA (LETTERA_MENSOLA, NUMERO_PIANO, CATEGORIA_SCAFFALE,
CAPIENZA) VALUES ('A', 1, 'MATEMATICA', 35);
INSERT INTO MENSOLA (LETTERA_MENSOLA, NUMERO_PIANO, CATEGORIA_SCAFFALE,
CAPIENZA) VALUES ('B', 1, 'MATEMATICA', 37);
INSERT INTO MENSOLA (LETTERA_MENSOLA, NUMERO_PIANO, CATEGORIA_SCAFFALE,
CAPIENZA) VALUES ('C', 1, 'MATEMATICA', 29);
INSERT INTO MENSOLA (LETTERA_MENSOLA, NUMERO_PIANO, CATEGORIA_SCAFFALE,
CAPIENZA) VALUES ('D', 1, 'MATEMATICA', 35);
INSERT INTO MENSOLA (LETTERA_MENSOLA, NUMERO_PIANO, CATEGORIA_SCAFFALE,
CAPIENZA) VALUES ('E', 1, 'MATEMATICA', 37);
INSERT INTO MENSOLA (LETTERA_MENSOLA, NUMERO_PIANO, CATEGORIA_SCAFFALE,
CAPIENZA) VALUES ('F', 1, 'MATEMATICA', 29);
INSERT INTO MENSOLA (LETTERA_MENSOLA, NUMERO_PIANO, CATEGORIA_SCAFFALE,
CAPIENZA) VALUES ('G', 1, 'MATEMATICA', 35);
INSERT INTO MENSOLA (LETTERA_MENSOLA, NUMERO_PIANO, CATEGORIA_SCAFFALE,
CAPIENZA) VALUES ('H', 1, 'MATEMATICA', 37);
INSERT INTO MENSOLA (LETTERA_MENSOLA, NUMERO_PIANO, CATEGORIA_SCAFFALE,
CAPIENZA) VALUES ('I', 2, 'MATEMATICA', 35);
INSERT INTO MENSOLA (LETTERA_MENSOLA, NUMERO_PIANO, CATEGORIA_SCAFFALE,
CAPIENZA) VALUES ('J', 2, 'MATEMATICA', 37);
INSERT INTO MENSOLA (LETTERA_MENSOLA, NUMERO_PIANO, CATEGORIA_SCAFFALE,
CAPIENZA) VALUES ('K', 2, 'MATEMATICA', 29);
INSERT INTO MENSOLA (LETTERA_MENSOLA, NUMERO_PIANO, CATEGORIA_SCAFFALE,
CAPIENZA) VALUES ('L', 2, 'MATEMATICA', 35);
INSERT INTO MENSOLA (LETTERA_MENSOLA, NUMERO_PIANO, CATEGORIA_SCAFFALE,
CAPIENZA) VALUES ('M', 2, 'MATEMATICA', 37);
INSERT INTO MENSOLA (LETTERA_MENSOLA, NUMERO_PIANO, CATEGORIA_SCAFFALE,
CAPIENZA) VALUES ('N', 2, 'MATEMATICA', 29);
INSERT INTO MENSOLA (LETTERA_MENSOLA, NUMERO_PIANO, CATEGORIA_SCAFFALE,
CAPIENZA) VALUES ('O', 2, 'MATEMATICA', 35);
```

```
INSERT INTO MENSOLA (LETTERA_MENSOLA, NUMERO_PIANO, CATEGORIA_SCAFFALE,
CAPIENZA) VALUES ('P', 2, 'MATEMATICA', 37);
INSERT INTO MENSOLA (LETTERA_MENSOLA, NUMERO_PIANO, CATEGORIA_SCAFFALE,
CAPIENZA) VALUES ('Q', 2, 'MATEMATICA', 37);
INSERT INTO MENSOLA (LETTERA_MENSOLA, NUMERO_PIANO, CATEGORIA_SCAFFALE,
CAPIENZA) VALUES ('R', 3, 'MATEMATICA', 35);
INSERT INTO MENSOLA (LETTERA_MENSOLA, NUMERO_PIANO, CATEGORIA_SCAFFALE,
CAPIENZA) VALUES ('S', 3, 'MATEMATICA', 37);
INSERT INTO MENSOLA (LETTERA_MENSOLA, NUMERO_PIANO, CATEGORIA_SCAFFALE,
CAPIENZA) VALUES ('T', 3, 'MATEMATICA', 29);
INSERT INTO MENSOLA (LETTERA_MENSOLA, NUMERO_PIANO, CATEGORIA_SCAFFALE,
CAPIENZA) VALUES ('U', 3, 'MATEMATICA', 35);
INSERT INTO MENSOLA (LETTERA_MENSOLA, NUMERO_PIANO, CATEGORIA_SCAFFALE,
CAPIENZA) VALUES ('V', 3, 'MATEMATICA', 37);
INSERT INTO MENSOLA (LETTERA_MENSOLA, NUMERO_PIANO, CATEGORIA_SCAFFALE,
CAPIENZA) VALUES ('W', 3, 'MATEMATICA', 29);
INSERT INTO MENSOLA (LETTERA_MENSOLA, NUMERO_PIANO, CATEGORIA_SCAFFALE,
CAPIENZA) VALUES ('X', 3, 'MATEMATICA', 35);
INSERT INTO MENSOLA (LETTERA_MENSOLA, NUMERO_PIANO, CATEGORIA_SCAFFALE,
CAPIENZA) VALUES ('Y', 3, 'MATEMATICA', 37);
INSERT INTO MENSOLA (LETTERA_MENSOLA, NUMERO_PIANO, CATEGORIA_SCAFFALE,
CAPIENZA) VALUES ('Z', 3, 'MATEMATICA', 37);
INSERT INTO MENSOLA (LETTERA_MENSOLA, NUMERO_PIANO, CATEGORIA_SCAFFALE,
CAPIENZA) VALUES ('A', 1, 'HORROR', 35);
INSERT INTO MENSOLA (LETTERA_MENSOLA, NUMERO_PIANO, CATEGORIA_SCAFFALE,
CAPIENZA) VALUES ('B', 1, 'HORROR', 37);
INSERT INTO MENSOLA (LETTERA_MENSOLA, NUMERO_PIANO, CATEGORIA_SCAFFALE,
CAPIENZA) VALUES ('C', 1, 'HORROR', 29);
INSERT INTO MENSOLA (LETTERA_MENSOLA, NUMERO_PIANO, CATEGORIA_SCAFFALE,
CAPIENZA) VALUES ('D', 1, 'HORROR', 35);
INSERT INTO MENSOLA (LETTERA_MENSOLA, NUMERO_PIANO, CATEGORIA_SCAFFALE,
CAPIENZA) VALUES ('E', 1, 'HORROR', 37);
INSERT INTO MENSOLA (LETTERA_MENSOLA, NUMERO_PIANO, CATEGORIA_SCAFFALE,
CAPIENZA) VALUES ('F', 1, 'HORROR', 29);
INSERT INTO MENSOLA (LETTERA_MENSOLA, NUMERO_PIANO, CATEGORIA_SCAFFALE,
CAPIENZA) VALUES ('G', 1, 'HORROR', 35);
INSERT INTO MENSOLA (LETTERA_MENSOLA, NUMERO_PIANO, CATEGORIA_SCAFFALE,
CAPIENZA) VALUES ('H', 1, 'HORROR', 37);
INSERT INTO MENSOLA (LETTERA_MENSOLA, NUMERO_PIANO, CATEGORIA_SCAFFALE,
CAPIENZA) VALUES ('I', 2, 'HORROR', 35);
INSERT INTO MENSOLA (LETTERA_MENSOLA, NUMERO_PIANO, CATEGORIA_SCAFFALE,
CAPIENZA) VALUES ('J', 2, 'HORROR', 37);
INSERT INTO MENSOLA (LETTERA_MENSOLA, NUMERO_PIANO, CATEGORIA_SCAFFALE,
CAPIENZA) VALUES ('K', 2, 'HORROR', 29);
INSERT INTO MENSOLA (LETTERA_MENSOLA, NUMERO_PIANO, CATEGORIA_SCAFFALE,
CAPIENZA) VALUES ('L', 2, 'HORROR', 35);
INSERT INTO MENSOLA (LETTERA_MENSOLA, NUMERO_PIANO, CATEGORIA_SCAFFALE,
CAPIENZA) VALUES ('M', 2, 'HORROR', 37);
```

```
INSERT INTO MENSOLA (LETTERA_MENSOLA, NUMERO_PIANO, CATEGORIA_SCAFFALE,
CAPIENZA) VALUES ('N', 2, 'HORROR', 29);
INSERT INTO MENSOLA (LETTERA_MENSOLA, NUMERO_PIANO, CATEGORIA_SCAFFALE,
CAPIENZA) VALUES ('O', 2, 'HORROR', 35);
INSERT INTO MENSOLA (LETTERA_MENSOLA, NUMERO_PIANO, CATEGORIA_SCAFFALE,
CAPIENZA) VALUES ('P', 2, 'HORROR', 37);
INSERT INTO MENSOLA (LETTERA_MENSOLA, NUMERO_PIANO, CATEGORIA_SCAFFALE,
CAPIENZA) VALUES ('Q', 2, 'HORROR', 37);
INSERT INTO MENSOLA (LETTERA_MENSOLA, NUMERO_PIANO, CATEGORIA_SCAFFALE,
CAPIENZA) VALUES ('R', 3, 'HORROR', 35);
INSERT INTO MENSOLA (LETTERA_MENSOLA, NUMERO_PIANO, CATEGORIA_SCAFFALE,
CAPIENZA) VALUES ('S', 3, 'HORROR', 37);
INSERT INTO MENSOLA (LETTERA_MENSOLA, NUMERO_PIANO, CATEGORIA_SCAFFALE,
CAPIENZA) VALUES ('T', 3, 'HORROR', 29);
INSERT INTO MENSOLA (LETTERA_MENSOLA, NUMERO_PIANO, CATEGORIA_SCAFFALE,
CAPIENZA) VALUES ('U', 3, 'HORROR', 35);
INSERT INTO MENSOLA (LETTERA_MENSOLA, NUMERO_PIANO, CATEGORIA_SCAFFALE,
CAPIENZA) VALUES ('V', 3, 'HORROR', 37);
INSERT INTO MENSOLA (LETTERA_MENSOLA, NUMERO_PIANO, CATEGORIA_SCAFFALE,
CAPIENZA) VALUES ('W', 3, 'HORROR', 29);
INSERT INTO MENSOLA (LETTERA_MENSOLA, NUMERO_PIANO, CATEGORIA_SCAFFALE,
CAPIENZA) VALUES ('X', 3, 'HORROR', 35);
INSERT INTO MENSOLA (LETTERA_MENSOLA, NUMERO_PIANO, CATEGORIA_SCAFFALE,
CAPIENZA) VALUES ('Y', 3, 'HORROR', 37);
INSERT INTO MENSOLA (LETTERA_MENSOLA, NUMERO_PIANO, CATEGORIA_SCAFFALE,
CAPIENZA) VALUES ('Z', 3, 'HORROR', 37);
INSERT INTO MENSOLA (LETTERA_MENSOLA, NUMERO_PIANO, CATEGORIA_SCAFFALE,
CAPIENZA) VALUES ('A', 1, 'FANTASY', 35);
INSERT INTO MENSOLA (LETTERA_MENSOLA, NUMERO_PIANO, CATEGORIA_SCAFFALE,
CAPIENZA) VALUES ('B', 1, 'FANTASY', 37);
INSERT INTO MENSOLA (LETTERA_MENSOLA, NUMERO_PIANO, CATEGORIA_SCAFFALE,
CAPIENZA) VALUES ('C', 1, 'FANTASY', 29);
INSERT INTO MENSOLA (LETTERA_MENSOLA, NUMERO_PIANO, CATEGORIA_SCAFFALE,
CAPIENZA) VALUES ('D', 1, 'FANTASY', 35);
INSERT INTO MENSOLA (LETTERA_MENSOLA, NUMERO_PIANO, CATEGORIA_SCAFFALE,
CAPIENZA) VALUES ('E', 1, 'FANTASY', 37);
INSERT INTO MENSOLA (LETTERA_MENSOLA, NUMERO_PIANO, CATEGORIA_SCAFFALE,
CAPIENZA) VALUES ('F', 1, 'FANTASY', 29);
INSERT INTO MENSOLA (LETTERA_MENSOLA, NUMERO_PIANO, CATEGORIA_SCAFFALE,
CAPIENZA) VALUES ('G', 1, 'FANTASY', 35);
INSERT INTO MENSOLA (LETTERA_MENSOLA, NUMERO_PIANO, CATEGORIA_SCAFFALE,
CAPIENZA) VALUES ('H', 1, 'FANTASY', 37);
INSERT INTO MENSOLA (LETTERA_MENSOLA, NUMERO_PIANO, CATEGORIA_SCAFFALE,
CAPIENZA) VALUES ('I', 2, 'FANTASY', 35);
INSERT INTO MENSOLA (LETTERA_MENSOLA, NUMERO_PIANO, CATEGORIA_SCAFFALE,
CAPIENZA) VALUES ('J', 2, 'FANTASY', 37);
INSERT INTO MENSOLA (LETTERA_MENSOLA, NUMERO_PIANO, CATEGORIA_SCAFFALE,
CAPIENZA) VALUES ('K', 2, 'FANTASY', 29);
```

```
INSERT INTO MENSOLA (LETTERA_MENSOLA, NUMERO_PIANO, CATEGORIA_SCAFFALE,
CAPIENZA) VALUES ('L', 2, 'FANTASY', 35);
INSERT INTO MENSOLA (LETTERA_MENSOLA, NUMERO_PIANO, CATEGORIA_SCAFFALE,
CAPIENZA) VALUES ('M', 2, 'FANTASY', 37);
INSERT INTO MENSOLA (LETTERA_MENSOLA, NUMERO_PIANO, CATEGORIA_SCAFFALE,
CAPIENZA) VALUES ('N', 2, 'FANTASY', 29);
INSERT INTO MENSOLA (LETTERA_MENSOLA, NUMERO_PIANO, CATEGORIA_SCAFFALE,
CAPIENZA) VALUES ('O', 2, 'FANTASY', 35);
INSERT INTO MENSOLA (LETTERA_MENSOLA, NUMERO_PIANO, CATEGORIA_SCAFFALE,
CAPIENZA) VALUES ('P', 2, 'FANTASY', 37);
INSERT INTO MENSOLA (LETTERA_MENSOLA, NUMERO_PIANO, CATEGORIA_SCAFFALE,
CAPIENZA) VALUES ('Q', 2, 'FANTASY', 37);
INSERT INTO MENSOLA (LETTERA_MENSOLA, NUMERO_PIANO, CATEGORIA_SCAFFALE,
CAPIENZA) VALUES ('R', 3, 'FANTASY', 35);
INSERT INTO MENSOLA (LETTERA_MENSOLA, NUMERO_PIANO, CATEGORIA_SCAFFALE,
CAPIENZA) VALUES ('S', 3, 'FANTASY', 37);
INSERT INTO MENSOLA (LETTERA_MENSOLA, NUMERO_PIANO, CATEGORIA_SCAFFALE,
CAPIENZA) VALUES ('T', 3, 'FANTASY', 29);
INSERT INTO MENSOLA (LETTERA_MENSOLA, NUMERO_PIANO, CATEGORIA_SCAFFALE,
CAPIENZA) VALUES ('U', 3, 'FANTASY', 35);
INSERT INTO MENSOLA (LETTERA_MENSOLA, NUMERO_PIANO, CATEGORIA_SCAFFALE,
CAPIENZA) VALUES ('V', 3, 'FANTASY', 37);
INSERT INTO MENSOLA (LETTERA_MENSOLA, NUMERO_PIANO, CATEGORIA_SCAFFALE,
CAPIENZA) VALUES ('W', 3, 'FANTASY', 29);
INSERT INTO MENSOLA (LETTERA_MENSOLA, NUMERO_PIANO, CATEGORIA_SCAFFALE,
CAPIENZA) VALUES ('X', 3, 'FANTASY', 35);
INSERT INTO MENSOLA (LETTERA_MENSOLA, NUMERO_PIANO, CATEGORIA_SCAFFALE,
CAPIENZA) VALUES ('Y', 3, 'FANTASY', 37);
INSERT INTO MENSOLA (LETTERA_MENSOLA, NUMERO_PIANO, CATEGORIA_SCAFFALE,
CAPIENZA) VALUES ('Z', 3, 'FANTASY', 37);
INSERT INTO MENSOLA (LETTERA_MENSOLA, NUMERO_PIANO, CATEGORIA_SCAFFALE,
CAPIENZA) VALUES ('A', 1, 'GIALLO', 35);
INSERT INTO MENSOLA (LETTERA_MENSOLA, NUMERO_PIANO, CATEGORIA_SCAFFALE,
CAPIENZA) VALUES ('B', 1, 'GIALLO', 37);
INSERT INTO MENSOLA (LETTERA_MENSOLA, NUMERO_PIANO, CATEGORIA_SCAFFALE,
CAPIENZA) VALUES ('C', 1, 'GIALLO', 29);
INSERT INTO MENSOLA (LETTERA_MENSOLA, NUMERO_PIANO, CATEGORIA_SCAFFALE,
CAPIENZA) VALUES ('D', 1, 'GIALLO', 35);
INSERT INTO MENSOLA (LETTERA_MENSOLA, NUMERO_PIANO, CATEGORIA_SCAFFALE,
CAPIENZA) VALUES ('E', 1, 'GIALLO', 37);
INSERT INTO MENSOLA (LETTERA_MENSOLA, NUMERO_PIANO, CATEGORIA_SCAFFALE,
CAPIENZA) VALUES ('F', 1, 'GIALLO', 29);
INSERT INTO MENSOLA (LETTERA_MENSOLA, NUMERO_PIANO, CATEGORIA_SCAFFALE,
CAPIENZA) VALUES ('G', 1, 'GIALLO', 35);
INSERT INTO MENSOLA (LETTERA_MENSOLA, NUMERO_PIANO, CATEGORIA_SCAFFALE,
CAPIENZA) VALUES ('H', 1, 'GIALLO', 37);
INSERT INTO MENSOLA (LETTERA_MENSOLA, NUMERO_PIANO, CATEGORIA_SCAFFALE,
CAPIENZA) VALUES ('I', 2, 'GIALLO', 35);
```

```
INSERT INTO MENSOLA (LETTERA_MENSOLA, NUMERO_PIANO, CATEGORIA_SCAFFALE,
CAPIENZA) VALUES ('J', 2, 'GIALLO', 37);
INSERT INTO MENSOLA (LETTERA_MENSOLA, NUMERO_PIANO, CATEGORIA_SCAFFALE,
CAPIENZA) VALUES ('K', 2, 'GIALLO', 29);
INSERT INTO MENSOLA (LETTERA_MENSOLA, NUMERO_PIANO, CATEGORIA_SCAFFALE,
CAPIENZA) VALUES ('L', 2, 'GIALLO', 35);
INSERT INTO MENSOLA (LETTERA_MENSOLA, NUMERO_PIANO, CATEGORIA_SCAFFALE,
CAPIENZA) VALUES ('M', 2, 'GIALLO', 37);
INSERT INTO MENSOLA (LETTERA_MENSOLA, NUMERO_PIANO, CATEGORIA_SCAFFALE,
CAPIENZA) VALUES ('N', 2, 'GIALLO', 29);
INSERT INTO MENSOLA (LETTERA_MENSOLA, NUMERO_PIANO, CATEGORIA_SCAFFALE,
CAPIENZA) VALUES ('O', 2, 'GIALLO', 35);
INSERT INTO MENSOLA (LETTERA_MENSOLA, NUMERO_PIANO, CATEGORIA_SCAFFALE,
CAPIENZA) VALUES ('P', 2, 'GIALLO', 37);
INSERT INTO MENSOLA (LETTERA_MENSOLA, NUMERO_PIANO, CATEGORIA_SCAFFALE,
CAPIENZA) VALUES ('Q', 2, 'GIALLO', 37);
INSERT INTO MENSOLA (LETTERA_MENSOLA, NUMERO_PIANO, CATEGORIA_SCAFFALE,
CAPIENZA) VALUES ('R', 3, 'GIALLO', 35);
INSERT INTO MENSOLA (LETTERA_MENSOLA, NUMERO_PIANO, CATEGORIA_SCAFFALE,
CAPIENZA) VALUES ('S', 3, 'GIALLO', 37);
INSERT INTO MENSOLA (LETTERA_MENSOLA, NUMERO_PIANO, CATEGORIA_SCAFFALE,
CAPIENZA) VALUES ('T', 3, 'GIALLO', 29);
INSERT INTO MENSOLA (LETTERA_MENSOLA, NUMERO_PIANO, CATEGORIA_SCAFFALE,
CAPIENZA) VALUES ('U', 3, 'GIALLO', 35);
INSERT INTO MENSOLA (LETTERA_MENSOLA, NUMERO_PIANO, CATEGORIA_SCAFFALE,
CAPIENZA) VALUES ('V', 3, 'GIALLO', 37);
INSERT INTO MENSOLA (LETTERA_MENSOLA, NUMERO_PIANO, CATEGORIA_SCAFFALE,
CAPIENZA) VALUES ('W', 3, 'GIALLO', 29);
INSERT INTO MENSOLA (LETTERA_MENSOLA, NUMERO_PIANO, CATEGORIA_SCAFFALE,
CAPIENZA) VALUES ('X', 3, 'GIALLO', 35);
INSERT INTO MENSOLA (LETTERA_MENSOLA, NUMERO_PIANO, CATEGORIA_SCAFFALE,
CAPIENZA) VALUES ('Y', 3, 'GIALLO', 37);
INSERT INTO MENSOLA (LETTERA_MENSOLA, NUMERO_PIANO, CATEGORIA_SCAFFALE,
CAPIENZA) VALUES ('Z', 3, 'GIALLO', 37);
INSERT INTO MENSOLA (LETTERA_MENSOLA, NUMERO_PIANO, CATEGORIA_SCAFFALE,
CAPIENZA) VALUES ('A', 1, 'ROMANZO', 35);
INSERT INTO MENSOLA (LETTERA_MENSOLA, NUMERO_PIANO, CATEGORIA_SCAFFALE,
CAPIENZA) VALUES ('B', 1, 'ROMANZO', 37);
INSERT INTO MENSOLA (LETTERA_MENSOLA, NUMERO_PIANO, CATEGORIA_SCAFFALE,
CAPIENZA) VALUES ('C', 1, 'ROMANZO', 29);
INSERT INTO MENSOLA (LETTERA_MENSOLA, NUMERO_PIANO, CATEGORIA_SCAFFALE,
CAPIENZA) VALUES ('D', 1, 'ROMANZO', 35);
INSERT INTO MENSOLA (LETTERA_MENSOLA, NUMERO_PIANO, CATEGORIA_SCAFFALE,
CAPIENZA) VALUES ('E', 1, 'ROMANZO', 37);
INSERT INTO MENSOLA (LETTERA_MENSOLA, NUMERO_PIANO, CATEGORIA_SCAFFALE,
CAPIENZA) VALUES ('F', 1, 'ROMANZO', 29);
INSERT INTO MENSOLA (LETTERA_MENSOLA, NUMERO_PIANO, CATEGORIA_SCAFFALE,
CAPIENZA) VALUES ('G', 1, 'ROMANZO', 35);
```

```
INSERT INTO MENSOLA (LETTERA_MENSOLA, NUMERO_PIANO, CATEGORIA_SCAFFALE,
CAPIENZA) VALUES ('H', 1, 'ROMANZO', 37);
INSERT INTO MENSOLA (LETTERA_MENSOLA, NUMERO_PIANO, CATEGORIA_SCAFFALE,
CAPIENZA) VALUES ('I', 2, 'ROMANZO', 35);
INSERT INTO MENSOLA (LETTERA_MENSOLA, NUMERO_PIANO, CATEGORIA_SCAFFALE,
CAPIENZA) VALUES ('J', 2, 'ROMANZO', 37);
INSERT INTO MENSOLA (LETTERA_MENSOLA, NUMERO_PIANO, CATEGORIA_SCAFFALE,
CAPIENZA) VALUES ('K', 2, 'ROMANZO', 29);
INSERT INTO MENSOLA (LETTERA_MENSOLA, NUMERO_PIANO, CATEGORIA_SCAFFALE,
CAPIENZA) VALUES ('L', 2, 'ROMANZO', 35);
INSERT INTO MENSOLA (LETTERA_MENSOLA, NUMERO_PIANO, CATEGORIA_SCAFFALE,
CAPIENZA) VALUES ('M', 2, 'ROMANZO', 37);
INSERT INTO MENSOLA (LETTERA_MENSOLA, NUMERO_PIANO, CATEGORIA_SCAFFALE,
CAPIENZA) VALUES ('N', 2, 'ROMANZO', 29);
INSERT INTO MENSOLA (LETTERA_MENSOLA, NUMERO_PIANO, CATEGORIA_SCAFFALE,
CAPIENZA) VALUES ('O', 2, 'ROMANZO', 35);
INSERT INTO MENSOLA (LETTERA_MENSOLA, NUMERO_PIANO, CATEGORIA_SCAFFALE,
CAPIENZA) VALUES ('P', 2, 'ROMANZO', 37);
INSERT INTO MENSOLA (LETTERA_MENSOLA, NUMERO_PIANO, CATEGORIA_SCAFFALE,
CAPIENZA) VALUES ('Q', 2, 'ROMANZO', 37);
INSERT INTO MENSOLA (LETTERA_MENSOLA, NUMERO_PIANO, CATEGORIA_SCAFFALE,
CAPIENZA) VALUES ('R', 3, 'ROMANZO', 35);
INSERT INTO MENSOLA (LETTERA_MENSOLA, NUMERO_PIANO, CATEGORIA_SCAFFALE,
CAPIENZA) VALUES ('S', 3, 'ROMANZO', 37);
INSERT INTO MENSOLA (LETTERA_MENSOLA, NUMERO_PIANO, CATEGORIA_SCAFFALE,
CAPIENZA) VALUES ('T', 3, 'ROMANZO', 29);
INSERT INTO MENSOLA (LETTERA_MENSOLA, NUMERO_PIANO, CATEGORIA_SCAFFALE,
CAPIENZA) VALUES ('U', 3, 'ROMANZO', 35);
INSERT INTO MENSOLA (LETTERA_MENSOLA, NUMERO_PIANO, CATEGORIA_SCAFFALE,
CAPIENZA) VALUES ('V', 3, 'ROMANZO', 37);
INSERT INTO MENSOLA (LETTERA_MENSOLA, NUMERO_PIANO, CATEGORIA_SCAFFALE,
CAPIENZA) VALUES ('W', 3, 'ROMANZO', 29);
INSERT INTO MENSOLA (LETTERA_MENSOLA, NUMERO_PIANO, CATEGORIA_SCAFFALE,
CAPIENZA) VALUES ('X', 3, 'ROMANZO', 35);
INSERT INTO MENSOLA (LETTERA_MENSOLA, NUMERO_PIANO, CATEGORIA_SCAFFALE,
CAPIENZA) VALUES ('Y', 3, 'ROMANZO', 37);
INSERT INTO MENSOLA (LETTERA_MENSOLA, NUMERO_PIANO, CATEGORIA_SCAFFALE,
CAPIENZA) VALUES ('Z', 3, 'ROMANZO', 37);
```

RECENSIONE

```
INSERT INTO RECENSIONE (DATA_RECENSIONE, VALUTAZIONE, ISBN, NUMERO_TESSERA) VALUES
('24-FEB-2020', 3, '1234567890123', '12345678');
INSERT INTO RECENSIONE (DATA_RECENSIONE, VALUTAZIONE, ISBN, NUMERO_TESSERA) VALUES
('12-MAR-2019', 1, '1234567890134', '12345679');
INSERT INTO RECENSIONE (DATA_RECENSIONE, VALUTAZIONE, ISBN, NUMERO_TESSERA) VALUES
('04-APR-2022', 5, '1234567890169', '12345670');
INSERT INTO RECENSIONE (DATA_RECENSIONE, VALUTAZIONE, ISBN, NUMERO_TESSERA) VALUES
('16-MAY-2021', 5, '1234567890179', '12345680');
```

```
INSERT INTO RECENSIONE (DATA_RECENSIONE, VALUTAZIONE, ISBN, NUMERO_TESSERA) VALUES
('09-APR-2020', 4, '1234567890159', '12345681');
```

SCRITTO

```
INSERT INTO SCRITTO (ISNI, ISBN) VALUES ('0000000123456789', '1234567890123');
INSERT INTO SCRITTO (ISNI, ISBN) VALUES ('0000001123456789', '1234567890134');
INSERT INTO SCRITTO (ISNI, ISBN) VALUES ('0000011123456789', '1234567890147');
INSERT INTO SCRITTO (ISNI, ISBN) VALUES ('0000111123456789', '1234567890179');
INSERT INTO SCRITTO (ISNI, ISBN) VALUES ('0001111123456789', '1234567890159');
```

ASSISTE

```
INSERT INTO ASSISTE (CODICE_FISCALE_CLIENTE, NUMERO_MATRICOLA, DATA_ASSISTENZA)
VALUES ('RMOGTN03A22F839U', '00011234', '12-DEC-2021');
INSERT INTO ASSISTE (CODICE_FISCALE_CLIENTE, NUMERO_MATRICOLA, DATA_ASSISTENZA)
VALUES ('PCRGNN02D09H892T', '00011235', '28-MAR-2022');
INSERT INTO ASSISTE (CODICE_FISCALE_CLIENTE, NUMERO_MATRICOLA, DATA_ASSISTENZA)
VALUES ('RCPLNZ01C11F839V', '00011236', '14-APR-2023');
INSERT INTO ASSISTE (CODICE_FISCALE_CLIENTE, NUMERO_MATRICOLA, DATA_ASSISTENZA)
VALUES ('RSSPSQ80H16F839S', '00011237', '07-OCT-2020');
INSERT INTO ASSISTE (CODICE_FISCALE_CLIENTE, NUMERO_MATRICOLA, DATA_ASSISTENZA)
VALUES ('BCHNTN04R09F839Y', '00011238', '21-APR-2022');
```

ORDINE

```
INSERT INTO ORDINE (DATA_ACQUISTO_ORDINE, ISBN_COPIE_ACQUISTATE,
NUMERO_COPIE_ACQUISTATE, ID_FORNITORE) VALUES ('31-DEC-2022', '1234567890123', 13, 1);
INSERT INTO ORDINE (DATA_ACQUISTO_ORDINE, ISBN_COPIE_ACQUISTATE,
NUMERO_COPIE_ACQUISTATE, ID_FORNITORE) VALUES ('22-APR-2021', '1234567890134', 14, 2);
INSERT INTO ORDINE (DATA_ACQUISTO_ORDINE, ISBN_COPIE_ACQUISTATE,
NUMERO_COPIE_ACQUISTATE, ID_FORNITORE) VALUES ('14-MAR-2023', '1234567890147', 11, 3);
INSERT INTO ORDINE (DATA_ACQUISTO_ORDINE, ISBN_COPIE_ACQUISTATE,
NUMERO_COPIE_ACQUISTATE, ID_FORNITORE) VALUES ('12-JUN-2019', '1234567890179', 7, 4);
INSERT INTO ORDINE (DATA_ACQUISTO_ORDINE, ISBN_COPIE_ACQUISTATE,
NUMERO_COPIE_ACQUISTATE, ID_FORNITORE) VALUES ('07-SEP-2018', '1234567890159', 19, 5);
```

COPIA

```
INSERT INTO COPIA (ISBN, NUMERO_COPIA, CONDIZIONE, LETTERA_MENSOLA, NUMERO_PIANO,
CATEGORIA_SCAFFALE, DATA_ACQUISTO_ORDINE, ID_FORNITORE) VALUES ('1234567890123',
1000, 'USATO', 'B', 1, 'SCIENZE', '31-DEC-2022', 1);
INSERT INTO COPIA (ISBN, NUMERO_COPIA, CONDIZIONE, LETTERA_MENSOLA, NUMERO_PIANO,
CATEGORIA_SCAFFALE, DATA_ACQUISTO_ORDINE, ID_FORNITORE) VALUES ('1234567890123',
1001, 'NUOVO', 'B', 1, 'SCIENZE', '31-DEC-2022', 1);
INSERT INTO COPIA (ISBN, NUMERO_COPIA, CONDIZIONE, LETTERA_MENSOLA, NUMERO_PIANO,
CATEGORIA_SCAFFALE, DATA_ACQUISTO_ORDINE, ID_FORNITORE) VALUES ('1234567890166',
1000, 'NUOVO', 'I', 2, 'SCIENZE', '19-MAR-2016', 13);
INSERT INTO COPIA (ISBN, NUMERO_COPIA, CONDIZIONE, LETTERA_MENSOLA, NUMERO_PIANO,
CATEGORIA_SCAFFALE, DATA_ACQUISTO_ORDINE, ID_FORNITORE) VALUES ('1234567890166',
1001, 'NUOVO', 'I', 2, 'SCIENZE', '19-MAR-2016', 13);
```

```
INSERT INTO COPIA (ISBN, NUMERO_COPIA, CONDIZIONE, LETTERA_MENSOLA, NUMERO_PIANO,
CATEGORIA_SCAFFALE, DATA_ACQUISTO_ORDINE, ID_FORNITORE) VALUES ('1234567890123',
1002, 'NUOVO', 'B', 1, 'SCIENZE', '31-DEC-2022', 1);
INSERT INTO COPIA (ISBN, NUMERO_COPIA, CONDIZIONE, LETTERA_MENSOLA, NUMERO_PIANO,
CATEGORIA_SCAFFALE, DATA_ACQUISTO_ORDINE, ID_FORNITORE) VALUES ('1234567890166',
1003, 'USATO', 'I', 2, 'SCIENZE', '19-MAR-2016', 13);
INSERT INTO COPIA (ISBN, NUMERO_COPIA, CONDIZIONE, LETTERA_MENSOLA, NUMERO_PIANO,
CATEGORIA_SCAFFALE, DATA_ACQUISTO_ORDINE, ID_FORNITORE) VALUES ('1234567890134',
1000, 'NUOVO', 'C', 1, 'MATEMATICA', '22-APR-2021', 2);
INSERT INTO COPIA (ISBN, NUMERO_COPIA, CONDIZIONE, LETTERA_MENSOLA, NUMERO_PIANO,
CATEGORIA_SCAFFALE, DATA_ACQUISTO_ORDINE, ID_FORNITORE) VALUES ('1234567890134',
1001, 'USATO', 'C', 1, 'MATEMATICA', '22-APR-2021', 2);
INSERT INTO COPIA (ISBN, NUMERO_COPIA, CONDIZIONE, LETTERA_MENSOLA, NUMERO_PIANO,
CATEGORIA_SCAFFALE, DATA_ACQUISTO_ORDINE, ID_FORNITORE) VALUES ('1234567890134',
1002, 'USATO', 'C', 1, 'MATEMATICA', '22-APR-2021', 2);
INSERT INTO COPIA (ISBN, NUMERO_COPIA, CONDIZIONE, LETTERA_MENSOLA, NUMERO_PIANO,
CATEGORIA_SCAFFALE, DATA_ACQUISTO_ORDINE, ID_FORNITORE) VALUES ('1234567890179',
1000, 'NUOVO', 'S', 3, 'STORIA', '12-JUN-2019', 4);
INSERT INTO COPIA (ISBN, NUMERO_COPIA, CONDIZIONE, LETTERA_MENSOLA, NUMERO_PIANO,
CATEGORIA_SCAFFALE, DATA_ACQUISTO_ORDINE, ID_FORNITORE) VALUES ('1234567890179',
1001, 'NUOVO', 'S', 3, 'STORIA', '12-JUN-2019', 4);
INSERT INTO COPIA (ISBN, NUMERO_COPIA, CONDIZIONE, LETTERA_MENSOLA, NUMERO_PIANO,
CATEGORIA_SCAFFALE, DATA_ACQUISTO_ORDINE, ID_FORNITORE) VALUES ('1234567890147',
1001, 'NUOVO', 'T', 3, 'STORIA', '07-SEP-2018', 5);
INSERT INTO COPIA (ISBN, NUMERO_COPIA, CONDIZIONE, LETTERA_MENSOLA, NUMERO_PIANO,
CATEGORIA_SCAFFALE, DATA_ACQUISTO_ORDINE, ID_FORNITORE) VALUES ('1234567890179',
1002, 'USATO', 'S', 3, 'STORIA', '12-JUN-2019', 4);
INSERT INTO COPIA (ISBN, NUMERO_COPIA, CONDIZIONE, LETTERA_MENSOLA, NUMERO_PIANO,
CATEGORIA_SCAFFALE, DATA_ACQUISTO_ORDINE, ID_FORNITORE) VALUES ('1234567890147',
1002, 'NUOVO', 'T', 3, 'STORIA', '07-SEP-2018', 5);
INSERT INTO COPIA (ISBN, NUMERO_COPIA, CONDIZIONE, LETTERA_MENSOLA, NUMERO_PIANO,
CATEGORIA_SCAFFALE, DATA_ACQUISTO_ORDINE, ID_FORNITORE) VALUES ('1234567890159',
1000, 'USATO', 'G', 1, 'GEOGRAFIA', '07-OCT-2019', 6);
INSERT INTO COPIA (ISBN, NUMERO_COPIA, CONDIZIONE, LETTERA_MENSOLA, NUMERO_PIANO,
CATEGORIA_SCAFFALE, DATA_ACQUISTO_ORDINE, ID_FORNITORE) VALUES ('1234567890159',
1001, 'USATO', 'G', 1, 'GEOGRAFIA', '07-OCT-2019', 6);
INSERT INTO COPIA (ISBN, NUMERO_COPIA, CONDIZIONE, LETTERA_MENSOLA, NUMERO_PIANO,
CATEGORIA_SCAFFALE, DATA_ACQUISTO_ORDINE, ID_FORNITORE) VALUES ('1234567890159',
1002, 'NUOVO', 'G', 1, 'GEOGRAFIA', '07-OCT-2019', 6);
INSERT INTO COPIA (ISBN, NUMERO_COPIA, CONDIZIONE, LETTERA_MENSOLA, NUMERO_PIANO,
CATEGORIA_SCAFFALE, DATA_ACQUISTO_ORDINE, ID_FORNITORE) VALUES ('1234567890160',
1000, 'USATO', 'I', 2, 'LETTERATURA', '08-APR-2020', 7);
INSERT INTO COPIA (ISBN, NUMERO_COPIA, CONDIZIONE, LETTERA_MENSOLA, NUMERO_PIANO,
CATEGORIA_SCAFFALE, DATA_ACQUISTO_ORDINE, ID_FORNITORE) VALUES ('1234567890165',
1000, 'USATO', 'I', 2, 'LETTERATURA', '15-AUG-2019', 12);
INSERT INTO COPIA (ISBN, NUMERO_COPIA, CONDIZIONE, LETTERA_MENSOLA, NUMERO_PIANO,
CATEGORIA_SCAFFALE, DATA_ACQUISTO_ORDINE, ID_FORNITORE) VALUES ('1234567890160',
1001, 'NUOVO', 'I', 2, 'LETTERATURA', '08-APR-2020', 7);
```

```
INSERT INTO COPIA (ISBN, NUMERO_COPIA, CONDIZIONE, LETTERA_MENSOLA, NUMERO_PIANO,
CATEGORIA_SCAFFALE, DATA_ACQUISTO_ORDINE, ID_FORNITORE) VALUES ('1234567890160',
1002, 'NUOVO', 'I', 2, 'LETTERATURA', '08-APR-2020', 7);
INSERT INTO COPIA (ISBN, NUMERO_COPIA, CONDIZIONE, LETTERA_MENSOLA, NUMERO_PIANO,
CATEGORIA_SCAFFALE, DATA_ACQUISTO_ORDINE, ID_FORNITORE) VALUES ('1234567890165',
1001, 'USATO', 'I', 2, 'LETTERATURA', '15-AUG-2019', 12);
INSERT INTO COPIA (ISBN, NUMERO_COPIA, CONDIZIONE, LETTERA_MENSOLA, NUMERO_PIANO,
CATEGORIA_SCAFFALE, DATA_ACQUISTO_ORDINE, ID_FORNITORE) VALUES ('1234567890165',
1002, 'NUOVO', 'I', 2, 'LETTERATURA', '15-AUG-2019', 12);
INSERT INTO COPIA (ISBN, NUMERO_COPIA, CONDIZIONE, LETTERA_MENSOLA, NUMERO_PIANO,
CATEGORIA_SCAFFALE, DATA_ACQUISTO_ORDINE, ID_FORNITORE) VALUES ('1234567890161',
1000, 'NUOVO', 'I', 2, 'HORROR', '17-JUN-2021', 8);
INSERT INTO COPIA (ISBN, NUMERO_COPIA, CONDIZIONE, LETTERA_MENSOLA, NUMERO_PIANO,
CATEGORIA_SCAFFALE, DATA_ACQUISTO_ORDINE, ID_FORNITORE) VALUES ('1234567890161',
1001, 'NUOVO', 'I', 2, 'HORROR', '17-JUN-2021', 8);
INSERT INTO COPIA (ISBN, NUMERO_COPIA, CONDIZIONE, LETTERA_MENSOLA, NUMERO_PIANO,
CATEGORIA_SCAFFALE, DATA_ACQUISTO_ORDINE, ID_FORNITORE) VALUES ('1234567890161',
1002, 'USATO', 'I', 2, 'HORROR', '17-JUN-2021', 8);
INSERT INTO COPIA (ISBN, NUMERO_COPIA, CONDIZIONE, LETTERA_MENSOLA, NUMERO_PIANO,
CATEGORIA_SCAFFALE, DATA_ACQUISTO_ORDINE, ID_FORNITORE) VALUES ('1234567890162',
1000, 'USATO', 'I', 2, 'FANTASY', '27-SEP-2019', 9);
INSERT INTO COPIA (ISBN, NUMERO_COPIA, CONDIZIONE, LETTERA_MENSOLA, NUMERO_PIANO,
CATEGORIA_SCAFFALE, DATA_ACQUISTO_ORDINE, ID_FORNITORE) VALUES ('1234567890168',
1000, 'USATO', 'L', 2, 'FANTASY', '03-SEP-2019', 15);
INSERT INTO COPIA (ISBN, NUMERO_COPIA, CONDIZIONE, LETTERA_MENSOLA, NUMERO_PIANO,
CATEGORIA_SCAFFALE, DATA_ACQUISTO_ORDINE, ID_FORNITORE) VALUES ('1234567890162',
1001, 'USATO', 'I', 2, 'FANTASY', '27-SEP-2019', 9);
INSERT INTO COPIA (ISBN, NUMERO_COPIA, CONDIZIONE, LETTERA_MENSOLA, NUMERO_PIANO,
CATEGORIA_SCAFFALE, DATA_ACQUISTO_ORDINE, ID_FORNITORE) VALUES ('1234567890168',
1001, 'NUOVO', 'L', 2, 'FANTASY', '03-SEP-2019', 15);
INSERT INTO COPIA (ISBN, NUMERO_COPIA, CONDIZIONE, LETTERA_MENSOLA, NUMERO_PIANO,
CATEGORIA_SCAFFALE, DATA_ACQUISTO_ORDINE, ID_FORNITORE) VALUES ('1234567890162',
1002, 'USATO', 'I', 2, 'FANTASY', '27-SEP-2019', 9);
INSERT INTO COPIA (ISBN, NUMERO_COPIA, CONDIZIONE, LETTERA_MENSOLA, NUMERO_PIANO,
CATEGORIA_SCAFFALE, DATA_ACQUISTO_ORDINE, ID_FORNITORE) VALUES ('1234567890168',
1002, 'USATO', 'L', 2, 'FANTASY', '03-SEP-2019', 15);
INSERT INTO COPIA (ISBN, NUMERO_COPIA, CONDIZIONE, LETTERA_MENSOLA, NUMERO_PIANO,
CATEGORIA_SCAFFALE, DATA_ACQUISTO_ORDINE, ID_FORNITORE) VALUES ('1234567890163',
1000, 'USATO', 'S', 3, 'GIALLO', '30-NOV-2018', 10);
INSERT INTO COPIA (ISBN, NUMERO_COPIA, CONDIZIONE, LETTERA_MENSOLA, NUMERO_PIANO,
CATEGORIA_SCAFFALE, DATA_ACQUISTO_ORDINE, ID_FORNITORE) VALUES ('1234567890167',
1000, 'NUOVO', 'I', 2, 'GIALLO', '21-AUG-2018', 14);
INSERT INTO COPIA (ISBN, NUMERO_COPIA, CONDIZIONE, LETTERA_MENSOLA, NUMERO_PIANO,
CATEGORIA_SCAFFALE, DATA_ACQUISTO_ORDINE, ID_FORNITORE) VALUES ('1234567890163',
1001, 'USATO', 'S', 3, 'GIALLO', '30-NOV-2018', 10);
INSERT INTO COPIA (ISBN, NUMERO_COPIA, CONDIZIONE, LETTERA_MENSOLA, NUMERO_PIANO,
CATEGORIA_SCAFFALE, DATA_ACQUISTO_ORDINE, ID_FORNITORE) VALUES ('1234567890167',
1001, 'USATO', 'I', 2, 'GIALLO', '21-AUG-2018', 14);
```

```
INSERT INTO COPIA (ISBN, NUMERO_COPIA, CONDIZIONE, LETTERA_MENSOLA, NUMERO_PIANO,
CATEGORIA_SCAFFALE, DATA_ACQUISTO_ORDINE, ID_FORNITORE) VALUES ('1234567890163',
1002, 'NUOVO', 'S', 3, 'GIALLO', '30-NOV-2018', 10);
INSERT INTO COPIA (ISBN, NUMERO_COPIA, CONDIZIONE, LETTERA_MENSOLA, NUMERO_PIANO,
CATEGORIA_SCAFFALE, DATA_ACQUISTO_ORDINE, ID_FORNITORE) VALUES ('1234567890167',
1002, 'USATO', 'I', 2, 'GIALLO', '21-AUG-2018', 14);
INSERT INTO COPIA (ISBN, NUMERO_COPIA, CONDIZIONE, LETTERA_MENSOLA, NUMERO_PIANO,
CATEGORIA_SCAFFALE, DATA_ACQUISTO_ORDINE, ID_FORNITORE) VALUES ('1234567890164',
1000, 'NUOVO', 'I', 2, 'ROMANZO', '12-DEC-2017', 11);
INSERT INTO COPIA (ISBN, NUMERO_COPIA, CONDIZIONE, LETTERA_MENSOLA, NUMERO_PIANO,
CATEGORIA_SCAFFALE, DATA_ACQUISTO_ORDINE, ID_FORNITORE) VALUES ('1234567890169',
1000, 'USATO', 'L', 2, 'ROMANZO', '14-MAR-2023', 3);
INSERT INTO COPIA (ISBN, NUMERO_COPIA, CONDIZIONE, LETTERA_MENSOLA, NUMERO_PIANO,
CATEGORIA_SCAFFALE, DATA_ACQUISTO_ORDINE, ID_FORNITORE) VALUES ('1234567890164',
1001, 'USATO', 'I', 2, 'ROMANZO', '12-DEC-2017', 11);
INSERT INTO COPIA (ISBN, NUMERO_COPIA, CONDIZIONE, LETTERA_MENSOLA, NUMERO_PIANO,
CATEGORIA_SCAFFALE, DATA_ACQUISTO_ORDINE, ID_FORNITORE) VALUES ('1234567890169',
1001, 'NUOVO', 'L', 2, 'ROMANZO', '14-MAR-2023', 3);
INSERT INTO COPIA (ISBN, NUMERO_COPIA, CONDIZIONE, LETTERA_MENSOLA, NUMERO_PIANO,
CATEGORIA_SCAFFALE, DATA_ACQUISTO_ORDINE, ID_FORNITORE) VALUES ('1234567890164',
1002, 'NUOVO', 'I', 2, 'ROMANZO', '12-DEC-2017', 11);
INSERT INTO COPIA (ISBN, NUMERO_COPIA, CONDIZIONE, LETTERA_MENSOLA, NUMERO_PIANO,
CATEGORIA_SCAFFALE, DATA_ACQUISTO_ORDINE, ID_FORNITORE) VALUES ('1234567890169',
1002, 'NUOVO', 'L', 2, 'ROMANZO', '14-MAR-2023', 3);
```

Sono stati effettuati cinque inserimenti per ogni tabella per mostrare il corretto funzionamento dei CONSTRAINT e dei trigger.
Si noti che per le tabelle popolate tramite procedure non sono stati effettuati inserimenti diretti, eccetto per la tabella Copia e che per le tabelle Scaffale, Mensola e Copia è stato necessario effettuare tutti gli inserimenti per garantire il corretto funzionamento del database.
Inoltre, per la tabella Fornitore sono stati effettuati dieci inserimenti, a differenza di tutte le altre, per permettere la presenza di cinque case editrici e cinque donatori.

TRIGGER

I seguenti trigger sono stati definiti per il controllo dei vincoli dinamici.

CONTROLLO_TURNO

Controlla che nello stesso giorno non ci siano più di tre bibliotecari che svolgono il proprio turno, che un bibliotecario non svolga più di due turni a settimana e che non gli venga assegnato un turno il giorno dopo aver lavorato per otto ore.

```
CREATE OR REPLACE TRIGGER CONTROLLO_TURNO
BEFORE INSERT OR UPDATE ON TURNO
FOR EACH ROW

DECLARE
        NUM_BIBLIO   NUMBER(1, 0);        --Numero di bibliotecari che svolgono il proprio turno
nella
```

```
                                        data indicata
        NUM_TURNI           NUMBER(1, 0);  --Numero di turni svolti dal bibliotecario in questa
settimana
        DUR_TUR_PREC        NUMBER(1, 0);  --Durata del turno del bibliotecario nel giorno
precedente

        TROPPI_BIBLIOTECARI EXCEPTION;    --Si verifica quando nella data indicata ci sono già 3
                                          bibliotecari che svolgono il proprio turno
TROPPI_TURNI        EXCEPTION;    --Si verifica quando il bibliotecario svolge più di 2 turni a
settimana

        GIORNO_FERIALE      EXCEPTION;    --Si verifica quando si cerca di inserire un turno un
giorno in
                                          cui la biblioteca è chiusa (sabato o domenica)
        TURNO_VECCHIO       EXCEPTION;    --Si verifica quando si cerca di inserire un turno in una
data
                                          precedente a quella corrente
BEGIN
   --Se si tenta di inserire un turno in una data precedente a quella corrente, il trigger genera
     l'eccezione
   IF :NEW.DATA_E_ORA_TURNO < SYSDATE
        THEN RAISE TURNO_VECCHIO;
   END IF;

--Conta il numero di bibliotecari che svolgono il proprio turno nella data in cui si vuole inserire il
  turno
   SELECT COUNT(*) INTO NUM_BIBLIO
   FROM TURNO
   WHERE TRUNC(DATA_E_ORA_TURNO) = TRUNC(:NEW.DATA_E_ORA_TURNO);

   IF NUM_BIBLIO > 2
      THEN RAISE TROPPI_BIBLIOTECARI;
   END IF;
   --Restituisce la durata del turno assegnato al bibliotecario nel giorno precedente (se gli è stato
assegnato)
   SELECT DURATA_TURNO INTO DUR_TUR_PREC
   FROM TURNO
   WHERE NUMERO_MATRICOLA = :NEW.NUMERO_MATRICOLA AND DATA_E_ORA_TURNO =
:NEW.DATA_E_ORA_TURNO - 1;

   --Se il giorno precedente il bibliotecario ha svolto un turno di 8 ore, il trigger genera l'eccezione
   IF DUR_TUR_PREC = 8
      THEN RAISE BIBLIOTECARIO_STANCO;
   END IF;

EXCEPTION
        WHEN NO_DATA_FOUND      THEN NULL;
```

```
        WHEN TURNO_VECCHIO      THEN RAISE_APPLICATION_ERROR(-20003, 'Impossibile
        inserire un turno in una data precedente a quella corrente');
        WHEN TROPPI_BIBLIOTECARI THEN RAISE_APPLICATION_ERROR(-20004, 'Nella data
        indicata ci sono già 3 bibliotecari che svolgeranno il proprio turno');
        WHEN TROPPI_TURNI        THEN RAISE_APPLICATION_ERROR(-20003, 'Il bibliotecario
        svolge  già 2 turni nella settimana indicata');
        WHEN BIBLIOTECARIO_STANCO   THEN RAISE_APPLICATION_ERROR(-20004, 'Il
        bibliotecario ha svolto un turno di 8 ore e deve riposare');

END;
```

CONTROLLO_BIBLIOTECARIO

Controlla che non siano registrati più di quindici bibliotecari contemporaneamente e che ognuno abbia un'età compresa tra i diciotto e i sessantacinque anni.

```
CREATE OR REPLACE TRIGGER CONTROLLO_BIBLIOTECARIO
BEFORE INSERT OR UPDATE ON BIBLIOTECARIO
FOR EACH ROW

DECLARE
   NUM_BIBLIO           NUMBER (2, 0);    --Numero di bibliotecari registrati nella biblioteca

   TROPPI_BIBLIOTECARI        EXCEPTION;      --Si verifica quando si supera il limite massimo di
                                               bibliotecari
   BIBLIOTECARIO_MINORENNE EXCEPTION;   --Si verifica quando il bibliotecario che si vuole
                                          inserire è minorenne
   BIBLIOTECARIO_ANZIANO   EXCEPTION;      --Si verifica quando il bibliotecario che si vuole
                                          inserire è troppo anziano

BEGIN
   --Conta il numero di bibliotecari registrati alla biblioteca
   SELECT COUNT (*) INTO NUM_BIBLIO
   FROM BIBLIOTECARIO;

   --Se il numero di bibliotecari registrati alla biblioteca supera il limite massimo, il trigger genera
     l'eccezione
   IF NUM_BIBLIO > 14
      THEN RAISE TROPPI_BIBLIOTECARI;
   END IF;

   --Se il bibliotecario che si sta tentando di inserire è minorenne, il trigger genera l'eccezione
   IF TRUNC (MONTHS_BETWEEN(SYSDATE, :NEW.DATA_NASCITA_BIBLIOTECARIO) / 12) < 18
         THEN RAISE BIBLIOTECARIO_MINORENNE;
   END IF;

   --Se il bibliotecario che si sta tentando di inserire è troppo anziano, il trigger genera l'eccezione
   IF TRUNC (MONTHS_BETWEEN(SYSDATE, :NEW.DATA_NASCITA_BIBLIOTECARIO) / 12) > 65
         THEN RAISE BIBLIOTECARIO_ANZIANO;
```

```
   END IF;

EXCEPTION
   WHEN TROPPI_BIBLIOTECARI     THEN RAISE_APPLICATION_ERROR(-20001, 'Numero massimo
di bibliotecari raggiunto');
   WHEN BIBLIOTECARIO_MINORENNE THEN RAISE_APPLICATION_ERROR(-20002, 'Il bibliotecario
che si sta tentando di inserire è minorenne');
   WHEN BIBLIOTECARIO_ANZIANO THEN RAISE_APPLICATION_ERROR(-20003, 'Il bibliotecario
che si sta tentando di inserire è troppo anziano');

END;
```

CONTROLLO_ASSISTENZA

Controlla che la data in cui un bibliotecario assiste un cliente corrisponda ad una data in cui ha svolto il proprio turno.

```
CREATE OR REPLACE TRIGGER CONTROLLO_ASSISTENZA
BEFORE INSERT OR UPDATE ON ASSISTE
FOR EACH ROW

DECLARE
   CHECK_TURNO              NUMBER(1, 0);    --Vale 0 se il bibliotecario non ha svolto il turno nella
data indicata e 1 altrimenti

   DATA_ASSISTENZA_SBAGLIATA EXCEPTION;      --Si verifica quando si tenta di inserire una data
                                               di assistenza in cui il bibliotecario non ha
                                               svolto il proprio turno
   DATA_INCOERENTE          EXCEPTION;      --Si verifica quando la data di assistenza è successiva a
                                               quella corrente

BEGIN
   --Restituisce 0 se il bibliotecario non ha svolto il proprio turno nella data dell'assistenza o un
     valore maggiore altrimenti
   SELECT COUNT (*) INTO CHECK_TURNO
   FROM TURNO T JOIN BIBLIOTECARIO B ON T.NUMERO_MATRICOLA = B.NUMERO_MATRICOLA
   WHERE T.NUMERO_MATRICOLA = :NEW.NUMERO_MATRICOLA AND
TRUNC(DATA_E_ORA_TURNO) = :NEW.DATA_ASSISTENZA;

   --Se il bibliotecario non ha svolto il proprio turno nella data dell'assistenza, il trigger genera
     l'eccezione
   IF CHECK_TURNO = 0
      THEN RAISE DATA_ASSISTENZA_SBAGLIATA;
   END IF;

   --Se la data di assistenza è successiva a quella corrente, il trigger genera l'eccezione
   IF :NEW.DATA_ASSISTENZA > SYSDATE
      THEN RAISE DATA_INCOERENTE;
   END IF;
```

```
EXCEPTION
   WHEN DATA_ASSISTENZA_SBAGLIATA THEN RAISE_APPLICATION_ERROR(-20001, 'Il
bibliotecario non ha svolto il proprio turno nella data indicata');
   WHEN DATA_INCOERENTE THEN RAISE_APPLICATION_ERROR(-20002, 'La data di assistenza è
incoerente');
END;
```

CONTROLLO_CLIENTE

Controlla che un cliente che si vuole registrare abbia un'età compresa tra i quattordici e i settant'anni.

```
CREATE OR REPLACE TRIGGER CONTROLLO_CLIENTE
BEFORE INSERT OR UPDATE ON CLIENTE
FOR EACH ROW

DECLARE
   CLIENTE_PICCOLO EXCEPTION;        --Si verifica quando si tenta di inserire un cliente con età
                                       minore ai 14 anni
   CLIENTE_ANZIANO EXCEPTION;        --Si verifica quando si tenta di inserire un cliente troppo
                                       anziano

BEGIN
   --Il trigger genera un'eccezione se il cliente che si vuole inserire è minorenne
   IF TRUNC (MONTHS_BETWEEN(SYSDATE, :NEW.DATA_NASCITA_CLIENTE) / 12) < 14
      THEN RAISE CLIENTE_PICCOLO;
   END IF;

   IF TRUNC (MONTHS_BETWEEN(SYSDATE, :NEW.DATA_NASCITA_CLIENTE) / 12) > 70
         THEN RAISE CLIENTE_ANZIANO;
   END IF;

EXCEPTION
   WHEN CLIENTE_PICCOLO THEN RAISE_APPLICATION_ERROR(-20001, 'Il cliente che si sta
tentando di registrare ha meno di 14 anni');
   WHEN CLIENTE_ANZIANO THEN RAISE_APPLICATION_ERROR(-20002, 'Il cliente che si sta
tentando di registrare è troppo anziano');

END;
```

CONTROLLO_EVENTO

Controlla che si tenga un solo evento dello stesso tipo al mese.

```
CREATE OR REPLACE TRIGGER CONTROLLO_EVENTO
BEFORE INSERT OR UPDATE ON EVENTO
FOR EACH ROW

DECLARE
```

```
   CHECK_EVENTO              NUMBER(1, 0);        --Vale 1 se nel mese indicato è già
programmato un evento dello stesso tipo e 0 altrimenti

   TROPPI_EVENTI_MESE      EXCEPTION;           --Si verifica quando si tiene più di un evento
                                                  dello stesso tipo lo stesso mese
   EVENTO_VECCHIO          EXCEPTION;           --Si verifica quando si tenta di inserire un
                                                  evento in una data precedente a quella
                                                  corrente

BEGIN
   --Se si tenta di inserire un evento in una data precedente a quella corrente, il trigger genera
     l'eccezione
   IF :NEW.DATA_E_ORA_EVENTO < SYSDATE
      THEN RAISE EVENTO_VECCHIO;
   END IF;

   --Conta il numero di eventi, dello stesso tipo di quello che si vuole inserire, che si sono tenuti
   nello stesso mese
   SELECT COUNT(*) INTO CONTATORE
   FROM EVENTO
   WHERE NOME_EVENTO = :NEW.NOME_EVENTO AND TO_CHAR(DATA_E_ORA_EVENTO, 'MM-
YYYY') = TO_CHAR(:NEW.DATA_E_ORA_EVENTO, 'MM-YYYY');

   --Se si terrà già un evento dello stesso tipo nel mese indicato, il trigger genera l'eccezione
   IF CHECK_EVENTO > 0
      THEN RAISE TROPPI_EVENTI_MESE;
   END IF;

EXCEPTION
   WHEN NO_DATA_FOUND THEN NULL;
   WHEN TROPPI_EVENTI_MESE   THEN RAISE_APPLICATION_ERROR(-20001, 'Nel mese indicato si
terrà già un evento dello stesso tipo');
   WHEN EVENTO_VECCHIO      THEN RAISE_APPLICATION_ERROR(-20002, 'La data del nuovo
evento non può essere precedente a quella corrente');

END;
```

CONTROLLO_RECENSIONE

Controlla che la data di recensione di un libro sia successiva al suo anno di pubblicazione e alla data di registrazione del cliente.

```
CREATE OR REPLACE TRIGGER CONTROLLO_RECENSIONE
BEFORE INSERT ON RECENSIONE
FOR EACH ROW

DECLARE
   ANNO_LIBRO          NUMBER(4, 0);       --Anno di pubblicazione del libro che si vuole recensire
   DATA_REG            DATE;               --Data di registrazione del cliente alla biblioteca
```

```
SCADENZA                DATE;                --Data di scadenza della tessera del cliente

   DATA_INCOERENTE      EXCEPTION;       --Si verifica quando la data di recensione di un libro è
                                           minore del suo anno di pubblicazione
   CLIENTE_NON_REGISTRATO EXCEPTION; --Si verifica quando un cliente tenta di recensire
                                           un libro senza essersi registrato alla biblioteca

BEGIN
   --Restituisce la data di registrazione del cliente alla biblioteca
   SELECT DATA_REGISTRAZIONE, DATA_SCADENZA_TESSERA INTO DATA_REG, SCADENZA
   FROM REGISTRAZIONE
   WHERE NUMERO_TESSERA = :NEW.NUMERO_TESSERA;

   --Se l'utente tenta di recensire un libro senza essere registrato, il trigger genera l'eccezione
   IF DATA_REG > :NEW.DATA_RECENSIONE OR :NEW.DATA_RECENSIONE > SCADENZA
      THEN RAISE CLIENTE_NON_REGISTRATO;
   END IF;

   --Restituisce l'anno di pubblicazione del libro che si vuole recensire
   SELECT ANNO_PUBBLICAZIONE INTO ANNO_LIBRO
   FROM   LIBRO
   WHERE  ISBN = :NEW.ISBN;

   --Se la data della recensione è minore dell'anno di pubblicazione del libro o maggiore della data
attuale, il trigger genera l'eccezione
   IF EXTRACT(YEAR FROM :NEW.DATA_RECENSIONE) < ANNO_LIBRO OR
:NEW.DATA_RECENSIONE > SYSDATE
      THEN RAISE DATA_INCOERENTE;
   END IF;

EXCEPTION
   WHEN NO_DATA_FOUND       THEN RAISE_APPLICATION_ERROR(-20001, 'Cliente non
registrato o libro non presente in biblioteca');
   WHEN DATA_INCOERENTE     THEN RAISE_APPLICATION_ERROR(-20002, 'La data della
recensione è incoerente');
   WHEN CLIENTE_NON_REGISTRATO THEN RAISE_APPLICATION_ERROR(-20003, 'Il cliente non è
registrato alla biblioteca');

END;
```

CONTROLLO_LIBRO

Controlla che l'anno di pubblicazione del libro non sia successivo all'anno corrente.

```
CREATE OR REPLACE TRIGGER CONTROLLO_LIBRO
BEFORE INSERT OR UPDATE ON LIBRO
FOR EACH ROW

DECLARE
```

```
   ANNO_PUBBLICAZIONE_INCOERENTE EXCEPTION;   --Si verifica quando si tenta di inserire un
                                                libro non ancora pubblicato

BEGIN
   --Se si tenta di inserire un libro non ancora pubblicato, il trigger genera l'eccezione
   IF :NEW.ANNO_PUBBLICAZIONE > EXTRACT(YEAR FROM SYSDATE)
      THEN RAISE ANNO_PUBBLICAZIONE_INCOERENTE;
   END IF;

EXCEPTION
   WHEN ANNO_PUBBLICAZIONE_INCOERENTE THEN RAISE_APPLICATION_ERROR(-20001, 'Il libro
non è stato ancora pubblicato');

END;
```

CONTROLLO_ORDINE

Controlla che una copia non sia ordinata prima della pubblicazione del libro al quale appartiene e che non vengano ordinate copie se il loro numero eccederebbe la capienza della mensola.

```
CREATE OR REPLACE TRIGGER CONTROLLO_ORDINE
BEFORE INSERT OR UPDATE ON ORDINE
FOR EACH ROW

DECLARE
   PUBBLICAZIONE          NUMBER (4, 0);   --Anno di pubblicazione del libro spedito con
                                             l'ordine
   COPIE_POSSEDUTE        NUMBER (4, 0);   --Numero di copie dello stesso tipo già possedute
in
                                              biblioteca
   INIZIALE_LIBRO         CHAR   (1);       --Lettera iniziale del libro acquistato
   GENERE_LIBRO           VARCHAR (20);     --Genere del libro acquistato
   NUM_PIA                NUMBER (1, 0);   --Numero del piano dello scaffale sul quale le copie
                                             vanno posizionate
   CAPIENZA_MENSOLA       NUMBER (4, 0);   --Capienza massima della mensola

   DATA_INCOERENTE EXCEPTION;      --Si verifica quando si tenta di ordinare un libro non
                                     ancora pubblicato
   TROPPE_COPIE           EXCEPTION;      --Si verifica quando si tenta di ordinare delle copie
                                            che non si riescono a posizionare nella mensola
                                            perchè già piena

BEGIN
--Restituisce l'anno di pubblicazione del libro ordinato
   SELECT ANNO_PUBBLICAZIONE INTO PUBBLICAZIONE
   FROM LIBRO
   WHERE ISBN = :NEW.ISBN_COPIE_ACQUISTATE;
```

```
   --Se la data di acquisto dell'ordine è successiva a quella corrente o precedente all'anno di pubblicazione del libro ordinato, il trigger genera l'eccezione
   IF :NEW.DATA_ACQUISTO_ORDINE > SYSDATE OR EXTRACT(YEAR FROM :NEW.DATA_ACQUISTO_ORDINE) < PUBBLICAZIONE
      THEN RAISE ORDINE_FUTURO;
   END IF;

   --Restituisce la lettera iniziale del libro e il genere al quale appartiene
   SELECT DISTINCT SUBSTR(TITOLO, 1, 1), GENERE INTO INIZIALE_LIBRO, GENERE_LIBRO
   FROM ORDINE O JOIN COPIA C ON O.DATA_ACQUISTO_ORDINE = C.DATA_ACQUISTO_ORDINE AND O.ID_FORNITORE = C.ID_FORNITORE JOIN LIBRO L ON C.ISBN = L.ISBN
   WHERE O.DATA_ACQUISTO_ORDINE = :NEW.DATA_ACQUISTO_ORDINE;

   --Restituisce il numero del piano dello scaffale sul quale vanno posizionate le copie
   SELECT S.NUMERO_PIANO INTO NUM_PIA
   FROM MENSOLA M JOIN SCAFFALE S ON M.NUMERO_PIANO = S.NUMERO_PIANO AND M.CATEGORIA_SCAFFALE = S.CATEGORIA_SCAFFALE
   WHERE S.CATEGORIA_SCAFFALE = GENERE_LIBRO AND M.LETTERA_MENSOLA = INIZIALE_LIBRO;

   --Restituisce il numero di copie posizionate sulla mensola sulla quale andrebbero posizionate le copie acquistate e la relativa capienza massima
   SELECT COUNT(*), CAPIENZA INTO COPIE_POSSEDUTE, CAPIENZA_MENSOLA
   FROM COPIA C JOIN MENSOLA M ON C.LETTERA_MENSOLA = M.LETTERA_MENSOLA AND C.CATEGORIA_SCAFFALE = M.CATEGORIA_SCAFFALE
   WHERE M.LETTERA_MENSOLA = INIZIALE_LIBRO AND M.CATEGORIA_SCAFFALE = GENERE_LIBRO AND M.NUMERO_PIANO = NUM_PIA
   GROUP BY (CAPIENZA);

   --Se non c'è più spazio sulla mensola, il trigger genera l'eccezione
   IF COPIE_POSSEDUTE + :NEW.NUMERO_COPIE_ACQUISTATE > CAPIENZA_MENSOLA
      THEN RAISE TROPPE_COPIE;
   END IF;

EXCEPTION
   WHEN NO_DATA_FOUND       THEN RAISE_APPLICATION_ERROR(-20001, 'Le copie che si sta tentando di ordinare appartengono a un libro non presente in biblioteca');
   WHEN LIBRO_NON_PUBBLICATO THEN RAISE_APPLICATION_ERROR(-20002, 'Le copie che si sta tentando di ordinare appartengono a un libro non ancora pubblicato');
   WHEN TROPPE_COPIE        THEN RAISE_APPLICATION_ERROR(-20003, 'Non è possibile acquistare nuove copie di questo libro in quanto la mensola è piena');
   WHEN ORDINE_FUTURO       THEN RAISE_APPLICATION_ERROR(-20004, 'La data di acquisto dell'ordine è incoerente');

END;
```

CONTROLLO_ORDINE

Controlla che il numero del piano dello scaffale al quale appartiene una mensola corrisponda alla lettera della mensola. In base alla gestione delle copie descritta nella fase introduttiva, le mensole la cui lettera sarà compresa tra le lettere 'A' e 'H' apparterranno agli scaffali al primo piano, quelle la cui lettera sarà compresa tra le lettere 'I' e 'Q' apparterranno agli scaffali al secondo piano e le restanti a quelli al terzo piano.

```
CREATE OR REPLACE TRIGGER CONTROLLO_MENSOLA
BEFORE INSERT OR UPDATE ON MENSOLA
FOR EACH ROW

DECLARE
  NUMERO_PIANO_SBAGLIATO EXCEPTION;      --Si verifica quando il numero del piano dello scaffale al quale appartiene la mensola non corrisponde alla lettera della mensola

BEGIN
  --Se il numero del piano dello scaffale al quale appartiene la mensola non corrisponde alla lettera della mensola, il trigger genera l'eccezione
  IF (LOWER(:NEW.LETTERA_MENSOLA) BETWEEN 'a' AND 'h' AND :NEW.NUMERO_PIANO <> 1)
OR (LOWER(:NEW.LETTERA_MENSOLA) BETWEEN 'i' AND 'q' AND :NEW.NUMERO_PIANO <> 2)
  OR (LOWER(:NEW.LETTERA_MENSOLA) BETWEEN 'r' AND 'z' AND :NEW.NUMERO_PIANO <> 3)
    THEN RAISE NUMERO_PIANO_SBAGLIATO;
  END IF;

EXCEPTION
  WHEN NUMERO_PIANO_SBAGLIATO THEN RAISE_APPLICATION_ERROR(-20001, 'Il piano dello scaffale non corrisponde a quello a cui la mensola dovrebbe appartenere');

END;
```

CONTROLLO_SCAFFALE

Controlla che la data di acquisto di uno scaffale non sia successiva a quella attuale e precedente a quest'ultima di più di cinque anni.

```
CREATE OR REPLACE TRIGGER CONTROLLO_SCAFFALE
BEFORE INSERT OR UPDATE ON SCAFFALE
FOR EACH ROW

DECLARE
  SCAFFALE_VECCHIO EXCEPTION;
  DATA_INCOERENTE  EXCEPTION;

BEGIN
  --Se lo scaffale è troppo vecchio, il trigger genera l'eccezione
  IF ADD_MONTHS(:NEW.DATA_ACQUISTO_SCAFFALE, 60) < SYSDATE
    THEN RAISE SCAFFALE_VECCHIO;
  END IF;
```

```
   --Se la data di acquisto è successiva alla data corrente, il trigger genera l'eccezione
   IF :NEW.DATA_ACQUISTO_ORDINE > SYSDATE
      THEN RAISE DATA_INCOERENTE;
   END IF;
EXCEPTION
   WHEN SCAFFALE_VECCHIO THEN RAISE_APPLICATION_ERROR(-20001, 'Lo scaffale è troppo
vecchio per essere montato nella biblioteca');
   WHEN DATA_INCOERENTE  THEN RAISE_APPLICATION_ERROR(-20002, 'La data di acquisto
dello scaffale è incoerente');

END;
```

PROCEDURE

Le seguenti procedure permettono di automatizzare le operazioni e sono legate alle regole di business stabilite.

EFFETTUA_PRESTITO

Un cliente ottiene in prestito la copia di un libro lanciando la procedura Effettua_Prestito e passandogli come argomenti il suo numero di tessera e l'ISBN del libro che vuole ottenere in prestito.
La procedura controlla che la tessera del cliente non sia scaduta, che non abbia tre multe non ancora pagate, che non abbia già preso in prestito dieci copie nel mese corrente e che non abbia già in prestito una copia dello stesso libro.
Se i controlli sono superati, seleziona, se disponibile, la prima copia disponibile del libro e la presta al cliente, impostando la data di scadenza entro la quale sarà tenuto a restituirla.
Supponendo per esempio che la procedura sia lanciata allo stesso istante da più clienti (molto verosimile), si presenterebbe un problema nel caso due o più di essi tentino di prendere in prestito una copia dello stesso libro e questa fosse l'ultima disponibile.
Un'ulteriore problema potrebbe verificarsi se mentre un cliente lancia la procedura per prendere in prestito una copia, un altro lanci la procedura e prenda in prestito la copia richiesta dal primo cliente, che quindi potrà visualizzarla ancora anche se non più disponibile.
L'utilizzo della clausola **COMMIT** è necessario per gestire la concorrenza sull'utilizzo della procedura, in quanto permette di confermare le modifiche applicate al database e renderle visibili a tutte le altre procedure.

```
CREATE OR REPLACE PROCEDURE EFFETTUA_PRESTITO (NUM_TESS
REGISTRAZIONE.NUMERO_TESSERA%TYPE, ISBN_C LIBRO.ISBN%TYPE) IS
   NUMERO_COPIA_PRESTATA  NUMBER(4, 0);   --Numero della copia appartenente al libro
                                            prestata al cliente
   COPIE_LIBRO_PRESTATE    NUMBER(4, 0);      --Numero di copie del libro che il cliente ha già
                                          attualmente in prestito
   COPIE_TOTALI_PRESTATE    NUMBER(4, 0);     --Numero di copie che il cliente ha già
                                          attualmente in prestito

   NUM_MULTE        NUMBER(1, 0);       --Numero di multe non pagate dal cliente
   SCADENZA         DATE;               --Scadenza della tessera del cliente

   TROPPI_PRESTITI         EXCEPTION;     --Si verifica quando il cliente ha preso in prestito più di
```

```
                                        10 copie senza averne restituito alcuno
  LIBRO_GIA_PRESTATO    EXCEPTION;      --Si verifica quando il cliente tenta di prendere in
                                        prestito la copia di un libro di cui ha già una copia
                                        attualmente in prestito
  MULTE_NON_PAGATE   EXCEPTION;         --Si verifica quando il cliente possiede almeno 3 multe
                                        da pagare
  TESSERA_SCADUTA       EXCEPTION;      --Si verifica quando il cliente ha la tessera scaduta
BEGIN
  --Restituisce il numero di multe non pagate dal cliente
  SELECT COUNT(*) INTO NUM_MULTE
  FROM MULTA
  WHERE NUMERO_TESSERA = NUM_TESS AND DATA_PAGAMENTO IS NULL;

  --Se il cliente ha 3 multe non pagate, la procedura genera l'eccezione
  IF NUM_MULTE > 2
    THEN RAISE MULTE_NON_PAGATE;
  END IF;

  --Restituisce la data di scadenza della tessera del cliente
  SELECT DATA_SCADENZA_TESSERA INTO SCADENZA
  FROM REGISTRAZIONE
  WHERE NUMERO_TESSERA = NUM_TESS;

  --Se la tessera del cliente è scaduta, la procedura genera l'eccezione
  IF SCADENZA < SYSDATE
    THEN RAISE TESSERA_SCADUTA;
  END IF;

  --Conta il numero di copie prese in prestito dal cliente
  SELECT COUNT (*) INTO COPIE_TOTALI_PRESTATE
  FROM   PRENDE
  WHERE  NUMERO_TESSERA = :NEW.NUMERO_TESSERA;

  --Se il cliente ha già preso in prestito 10 copie nel mese indicato, il trigger genera l'eccezione
  IF COPIE_TOTALI_PRESTATE > 9
    THEN RAISE TROPPI_PRESTITI;
  END IF;

  --Controlla se il cliente ha già attualmente in prestito una copia del libro
  SELECT COUNT(*) INTO COPIE_LIBRO_PRESTATE
  FROM PRENDE
  WHERE NUMERO_TESSERA = NUM_TESS AND ISBN = ISBN_C;

  --Se il cliente ha già una copia del libro attualmente in prestito, la procedura genera l'eccezione
  IF COPIE_LIBRO_PRESTATE > 0
    THEN RAISE LIBRO_GIA_PRESTATO;
  END IF;
```

```
   --Seleziona la prima copia disponibile (non attualmente prestata) del libro
   SELECT NUMERO_COPIA INTO NUMERO_COPIA_PRESTATA
   FROM COPIA
   WHERE ISBN = ISBN_C AND NUMERO_COPIA NOT IN (SELECT NUMERO_COPIA
                                FROM PRENDE
                                WHERE ISBN = ISBN_C)
   FETCH FIRST 1 ROW ONLY;
   --Presta la copia al cliente, stabilendo la data di scadenza del prestito
   INSERT INTO PRENDE (NUMERO_TESSERA, ISBN, NUMERO_COPIA, DATA_INIZIO_PRESTITO,
DATA_SCADENZA_PRESTITO)
   VALUES (NUM_TESS, ISBN_C, NUMERO_COPIA_PRESTATA, SYSDATE, ADD_MONTHS(SYSDATE,
12));

   DBMS_OUTPUT.PUT_LINE ('Il prestito della copia è avvenuto con successo');

   COMMIT;

EXCEPTION
   WHEN NO_DATA_FOUND      THEN RAISE_APPLICATION_ERROR(-20001, 'Non ci sono copie del
libro disponibili da prendere in prestito');
   WHEN TROPPI_PRESTITI    THEN RAISE_APPLICATION_ERROR(-20002, 'Il cliente ha già preso in
prestito 10 libri senza restituirne alcuno');
   WHEN LIBRO_GIA_PRESTATO THEN RAISE_APPLICATION_ERROR(-20003, 'Il cliente ha già
attualmente in prestito una copia del libro');
   WHEN MULTE_NON_PAGATE   THEN RAISE_APPLICATION_ERROR(-20004, 'Il cliente non può
prendere in prestito copie in quanto ha 3 multe da pagare');
   WHEN TESSERA_SCADUTA    THEN RAISE_APPLICATION_ERROR(-20005, 'Il cliente non può
prendere in prestito copie in quanto la sua tessera è scaduta');

END;
```

EFFETTUA_RESTITUZIONE

Un cliente restituisce una copia di un libro presa in prestito lanciando la procedura Effettua_Restituzione e passando come parametri il numero della tessera e l'ISBN e il numero della copia restituita.
La procedura controlla che il cliente abbia effettivamente preso in prestito la copia che sta cercando di restituire e, se la data in cui avviene la restituzione è successiva alla data di scadenza stabilita al momento del prestito, assegna una multa al cliente.
In ogni caso aggiorna la data di restituzione.
L'utilizzo della clausola **COMMIT** non è necessario in quanto la procedura effettua una lettura e un inserimento destinato a modificare esclusivamente i dati dell'utente che la esegue.

```
CREATE OR REPLACE PROCEDURE EFFETTUA_RESTITUZIONE (NUM_TESS
REGISTRAZIONE.NUMERO_TESSERA%TYPE, ISBN_C LIBRO.ISBN%TYPE, NUM_COPIA NUMBER) IS
   SCADENZA  DATE;       --Scadenza prefissata al momento del prestito

BEGIN
   --Restituisce la data di scadenza prefissata al momento del prestito
```

```
   SELECT DATA_SCADENZA_PRESTITO INTO SCADENZA
   FROM PRENDE
   WHERE NUMERO_TESSERA = NUM_TESS AND ISBN = ISBN_C AND NUMERO_COPIA =
NUM_COPIA;

   --Se la copia è restituita oltre la data di scadenza, la procedura genera una multa e la attesta al
     cliente
   IF SCADENZA < SYSDATE
   THEN
         IUV_MULTA := '123-' || TO_CHAR(SYSDATE, 'YYYY-MM') || '-' ||
LPAD(AUTO_INCREMENT_NUMERO_MULTA.NEXTVAL, 4, 0);

      INSERT INTO MULTA (IUV, NUMERO_TESSERA) VALUES (IUV_MULTA, NUM_TESS);
      DBMS_OUTPUT.PUT_LINE  ('Il cliente è stato multato perchè ha restituito la copia in ritardo');
   ELSE
      DBMS_OUTPUT.PUT_LINE  ('La copia è stata restituita entro la scadenza prefissata');

   END IF;

   --Aggiorna la data di restituzione della copia alla data corrente
   UPDATE PRENDE SET DATA_RESTITUZIONE = SYSDATE WHERE NUMERO_TESSERA = NUM_TESS
AND ISBN = ISBN_C AND NUMERO_COPIA = NUM_COPIA;

EXCEPTION
   WHEN NO_DATA_FOUND THEN RAISE_APPLICATION_ERROR(-20001, 'La copia che si sta
tentando di restituire non è stata presa in prestito dal cliente o non è presente nel database');
END;
```

PRENOTA_POSTO_EVENTO

Un cliente si prenota ad un evento lanciando la procedura Prenota_Posto_Evento e passandogli come parametri il numero della tessera e la data dell'evento a cui vuole prenotarsi.
La procedura controlla che la tessera del cliente non sia scaduta, che il cliente non possieda tre multe non pagate, che nella data indicata sia presente un evento e che ci sia un posto disponibile. Se i controlli vengono superati, il cliente è prenotato all'evento.
Come descritto per la procedura Effettua_Prestito, anche in questo caso bisogna utilizzare la clausola **COMMIT** per gestire il caso in cui più utenti lanciano in contemporanea tale procedura per prenotarsi ad un evento ed evitare problemi nel caso più di loro cerchino di prenotare l'ultimo posto disponibile.

```
CREATE OR REPLACE PROCEDURE PRENOTA_POSTO_EVENTO (NUM_TESS
REGISTRAZIONE.NUMERO_TESSERA%TYPE, DATA_EVENTO DATE) IS
   PARTECIPANTI       NUMBER(3, 0);     --Numero di partecipanti all'evento
   NUM_MULTE          NUMBER(1, 0);     --Numero di multe non pagate dal cliente
   SCADENZA           DATE;             --Data di scadenza della tessera del cliente

   EVENTO_PIENO    EXCEPTION;      --Si verifica quando ci si tenta di prenotarsi a un evento pieno
   MULTE_NON_PAGATE EXCEPTION;     --Si verifica quando il cliente ha 3 multe non pagate
   TESSERA_SCADUTA  EXCEPTION;     --Si verifica quando il cliente ha la tessera scaduta
```

```
BEGIN
   --Restituisce il numero di multe non pagate dal cliente
   SELECT COUNT(*) INTO NUM_MULTE
   FROM MULTA
   WHERE NUMERO_TESSERA = NUM_TESS AND DATA_PAGAMENTO IS NULL;

   --Se il cliente ha 3 multe non pagate, la procedura genera l'eccezione
   IF NUM_MULTE > 2
      THEN RAISE MULTE_NON_PAGATE;
   END IF;

   --Restituisce la data di scadenza della tessera del cliente
   SELECT DATA_SCADENZA_TESSERA INTO SCADENZA
   FROM REGISTRAZIONE
   WHERE NUMERO_TESSERA = NUM_TESS;

   --Se la tessera del cliente è scaduta, la procedura genera l'eccezione
   IF SCADENZA < SYSDATE
      THEN RAISE TESSERA_SCADUTA;
   END IF;

   --Restituisce il numero di partecipanti all'evento
   SELECT COUNT(*) INTO PARTECIPANTI
   FROM SEGUITO
   WHERE TRUNC(DATA_E_ORA_EVENTO) = TRUNC(DATA_EVENTO);

   --Se non ci sono più posti disponibili, la procedura genera l'eccezione
   IF PARTECIPANTI >= 50
      THEN RAISE EVENTO_PIENO;
   END IF;

   --Altrimenti, il cliente viene prenotato per l'evento
   INSERT INTO SEGUITO (DATA_E_ORA_EVENTO, NUMERO_TESSERA)
   VALUES (DATA_EVENTO, NUM_TESS);
   DBMS_OUTPUT.PUT_LINE ('Prenotazione effettuata con successo');

   COMMIT;

EXCEPTION
   WHEN NO_DATA_FOUND     THEN RAISE_APPLICATION_ERROR(-20001, 'Il giorno indicato non
si terrà nessun evento');
   WHEN EVENTO_PIENO      THEN RAISE_APPLICATION_ERROR(-20002, 'L evento a cui si vuole
partecipare non ha più posti disponibili');
   WHEN MULTE_NON_PAGATE  THEN RAISE_APPLICATION_ERROR(-20003, 'Il cliente non può
prendere prenotarsi in quanto ha 3 multe da pagare');
   WHEN TESSERA_SCADUTA   THEN RAISE_APPLICATION_ERROR(-20004, 'Il cliente non può
prendere prenotarsi in quanto la sua tessera è scaduta');
```

```
END;
```

DISDICI_PRENOTAZIONE_POSTO_EVENTO

Un cliente può disdire la prenotazione ad un evento lanciando la procedura Disdici_Prenotazione_Posto_Evento e passandogli come argomenti il numero della tessera e la data in cui si terrà l'evento.
La procedura controlla che il cliente sia effettivamente prenotato all'evento e che l'evento non si sia già tenuto.
Se i controlli sono superati, la prenotazione del cliente è disdetta e il posto occupato precedentemente è nuovamente disponibile.
L'utilizzo della clausola **COMMIT** non è necessario in quanto la procedura esegue una cancellazione su una tupla collegata esclusivamente all'utente che la esegue.

```
CREATE OR REPLACE PROCEDURE DISDICI_PRENOTAZIONE_POSTO_EVENTO (NUM_TESS
REGISTRAZIONE.NUMERO_TESSERA%TYPE, DATA_EVENTO DATE) IS
   CLIENTE_NON_PRENOTATO    EXCEPTION;     --Si verifica quando il cliente tenta di disdire la
                                             prenotazione ad un evento per la quale non è
                                             prenotato
   EVENTO_CONCLUSO          EXCEPTION;     --Si verifica quando si cerca di disdire la
                                             prenotazione ad un evento concluso

BEGIN
   --Se il cliente tenta di disdire la prenotazione ad un evento già concluso, la procedura genera
     l'eccezione
   IF DATA_EVENTO < SYSDATE
      THEN RAISE EVENTO_CONCLUSO;
   END IF;

   --Disdice la prenotazione del cliente all'evento
   DELETE FROM SEGUITO WHERE NUMERO_TESSERA = NUM_TESS AND
TRUNC(DATA_E_ORA_EVENTO) = TRUNC(DATA_EVENTO);

   --Se il cliente non è prenotato all'evento, la procedura genera l'eccezione
   IF SQL%ROWCOUNT = 0
      THEN RAISE CLIENTE_NON_PRENOTATO;
   END IF;

EXCEPTION
   WHEN EVENTO_CONCLUSO      THEN RAISE_APPLICATION_ERROR (-20001, 'Impossibile disdire
la prenotazione per un evento concluso');
   WHEN CLIENTE_NON_PRENOTATO THEN RAISE_APPLICATION_ERROR (-20002, 'Il cliente non è
prenotato per l'evento indicato');

END;
```

POSIZIONA_COPIE

Un bibliotecario posiziona una copia nel giusto scaffale e sulla giusta mensola lanciando la procedura Posiziona_Copie e passandogli come argomenti la data d'acquisto dell'ordine con la quale le copie sono state fornite alla biblioteca e l'ID del fornitore da cui è stato ricevuto l'ordine.
La procedura seleziona l'iniziale del nome del libro al quale appartengono le copie acquistate e il suo genere.
In base a questi valori, calcola il numero del piano dello scaffale e la mensola sulla quale andranno posizionate le copie.
Infine, effettua tanti inserimenti quante sono le copie contenute nell'ordine.
L'utilizzo della clausola **COMMIT** è necessario, in quanto la procedura potrebbe essere eseguita allo stesso istante da più bibliotecari, per permettere alla procedura di risalire correttamente al numero delle copie che deve inserire.

```
CREATE OR REPLACE PROCEDURE POSIZIONA_COPIE (DATA_ACQUISTO DATE, ID_F NUMBER) IS
   ISBN_C          LIBRO.ISBN%TYPE;          --ISBN delle copie arrivate
   NUM_COP_ACQ     NUMBER (4, 0);            --Numero totale di copie acquistate con l'ordine
   INIT            CHAR (1);                 --Lettera iniziale del nome del libro al quale
                                               appartengono le copie acquistate (per il
                                               posizionamento sulla giusta mensola)
   CAT_C           LIBRO.GENERE%TYPE;        --Genere delle copie (per il posizionamento sul giusto
                                               scaffale)
   NUM_P           COPIA.NUMERO_PIANO%TYPE;  --Numero di piano dello scaffale su cui vanno
                                                 posizionate le copie (in base alla lettera
                                                iniziale)
   CONT            NUMBER (4, 0);            --Ultimo numero di copia inserito (l'inserimento parte
                                               da questo valore)

BEGIN
   --Restituisce l'ISBN e il numero di copie acquistate
   SELECT ISBN_COPIE_ACQUISTATE, NUMERO_COPIE_ACQUISTATE INTO ISBN_C, NUM_COP_ACQ
   FROM ORDINE
   WHERE DATA_ACQUISTO_ORDINE = DATA_ACQUISTO AND ID_FORNITORE = ID_F;

   --Restituisce il genere al quale appartengono le copie
   SELECT DISTINCT SUBSTR(TITOLO, 1, 1), GENERE INTO INIT, CAT_C
   FROM LIBRO
   WHERE ISBN = ISBN_C;

   --Aggiorna il numero del piano dello scaffale sul quale vanno inserite le copie in base alla lettera
     iniziale del libro a cui appartengono
   IF UPPER(INIT) BETWEEN 'A' AND 'H'
      THEN NUM_P := 1;
   ELSIF UPPER(INIT) BETWEEN 'I' AND 'Q'
      THEN NUM_P := 2;
   ELSE
      NUM_P := 3;
   END IF;
```

```
  --Restituisce il massimo numero di copia inserito da cui iniziare l'inserimento
  SELECT MAX(NUMERO_COPIA) INTO CONT
  FROM COPIA
  WHERE ISBN = ISBN_C;

  --Se non ci sono copie di quel determinato libro in biblioteca, il conteggio parte da 0
  IF CONT IS NULL
    THEN CONT := 0;
  END IF;

  --Effettua tanti inserimenti nella tabella copia quante ne sono le copie incluse nell'ordine
  FOR I IN 1..NUM_COP_ACQ LOOP
    CONT := CONT + 1;
    INSERT INTO COPIA (ISBN, NUMERO_COPIA, CONDIZIONE, LETTERA_MENSOLA,
NUMERO_PIANO, CATEGORIA_SCAFFALE, DATA_ACQUISTO_ORDINE, ID_FORNITORE)
    VALUES (ISBN_C, CONT, 'NUOVO', INIT, NUM_P, CAT_C, DATA_ACQUISTO, ID_F);
  END LOOP;

  DBMS_OUTPUT.PUT_LINE ('Ordini inseriti con successo');

  COMMIT;

EXCEPTION
  WHEN NO_DATA_FOUND THEN RAISE_APPLICATION_ERROR (-20001, 'Ordine non trovato');

END;
```

RINNOVA_TESSERA

Un bibliotecario rinnova la tessera di un cliente lanciando la procedura Rinnova_Tessera e passandogli come parametro il numero della tessera da rinnovare.
La procedura controlla che il cliente non possieda tre multe non ancora pagate.
Se il controllo è superato, la tessera del cliente viene rinnovata di un anno.
L'utilizzo della clausola **COMMIT** è necessario in quanto potrebbe accadere che mentre il bibliotecario rinnova la tessera del cliente, a quest'ultimo venga assegnata una multa, che sfuggirebbe quindi al controllo effettuato.

```
CREATE OR REPLACE PROCEDURE RINNOVA_TESSERA (NUM_TESS
REGISTRAZIONE.NUMERO_TESSERA%TYPE) IS
  NUM_MULTE      NUMBER (1, 0);  --Numero di multe da pagare del cliente

  TROPPE_MULTE   EXCEPTION;      --Si verifica quando il cliente possiede 3 multe non ancora
                                   pagate

BEGIN
  --Restituisce il numero di multe non pagate dal cliente
  SELECT COUNT(*) INTO NUM_MULTE
  FROM MULTA
```

```
   WHERE NUMERO_TESSERA = NUM_TESS AND DATA_PAGAMENTO IS NULL;

   --Se il cliente ha tre multe ancora da pagare, la procedura genera l'eccezione
   IF NUM_MULTE > 2
      THEN RAISE TROPPE_MULTE;
   END IF;

   --Se la tessera è scaduta la procedura rinnova la tessera, impostando come nuova data di
     scadenza 1 anno a partire dalla data corrente
   THEN UPDATE REGISTRAZIONE SET DATA_SCADENZA_TESSERA = ADD_MONTHS(SYSDATE, 12)
WHERE NUMERO_TESSERA = NUM_TESS;
   DBMS_OUTPUT.PUT_LINE ('Rinnovo tessera effettuato con successo');

   COMMIT;

EXCEPTION
   WHEN NO_DATA_FOUND THEN RAISE_APPLICATION_ERROR(-20001, 'Tessera non trovata');
   WHEN TROPPE_MULTE  THEN RAISE_APPLICATION_ERROR(-20002, 'Impossibile rinnovare la
tessera in quanto il cliente possiede 3 multe da pagare');

END;
```

CONTROLLA_INVENTARIO

Il direttore può visualizzare l'inventario della biblioteca lanciando la procedura Controlla_Inventario e passandogli come parametri l'ID del fornitore dal quale eventualmente ordinare le copie e il numero di copie da ordinare.
La procedura controlla il numero di copie disponibili per ogni libro catalogato in biblioteca ed effettua tanti ordini quanti sono i libri con poche copie disponibili.
L'utilizzo della clausola **COMMIT** non è necessario in quanto la procedura non può essere eseguita in contemporanea da più utenti, dal momento che solo il direttore è autorizzato a farlo. Risulterebbe necessaria nel caso la procedura sia eseguibile anche da altri utenti, come ad esempio il bibliotecario, o la biblioteca abbia più direttori.

```
CREATE OR REPLACE PROCEDURE CONTROLLA_INVENTARIO (ID_F NUMBER, NUM_COPIE) IS
BEGIN
   --Restituisce i libri presenti in biblioteca e il relativo numero di copie disponibili
   FOR BOOK IN (SELECT L.ISBN, COUNT(NUMERO_COPIA) AS COPIE_DISPONIBILI
            FROM LIBRO L JOIN COPIA C ON L.ISBN = C.ISBN
            GROUP BY (L.ISBN))
   LOOP
      --Se il numero di copie disponibili di un libro è insufficiente, si effettua un ordine di copie di
        quel libro da un fornitore casuale
      IF BOOK.COPIE_DISPONIBILI < 5
         THEN INSERT INTO ORDINE (DATA_ACQUISTO_ORDINE, ISBN_COPIE_ACQUISTATE,
NUMERO_COPIE_ACQUISTATE, ID_FORNITORE)
         VALUES (SYSDATE, BOOK.ISBN, NUM_COPIE, ID_F);
      END IF;
   END LOOP;
```

```
END;
```

FUNZIONI

VISUALIZZA_DISPONIBILITA_COPIE

Viene utilizzata dalla vista Libri_Catalogati per visualizzare la disponibilità attuale di copie prestabili di un libro.
Il parametro della funzione è l'ISBN del libro.

```
CREATE OR REPLACE FUNCTION VISUALIZZA_DISPONIBILITA_COPIE (ISBN_C LIBRO.ISBN%TYPE)
RETURN NUMBER
IS
   COPIE_DISPONIBILI NUMBER(4, 0);     --Numero di copie del libro non attualmente in prestito a
                                         qualche cliente

BEGIN
   --Conta il numero di copie del libro disponibili
   SELECT COUNT(NUMERO_COPIA) INTO COPIE_DISPONIBILI
   FROM COPIA
   WHERE ISBN = ISBN_C AND NUMERO_COPIA NOT IN (SELECT NUMERO_COPIA
                                 FROM PRENDE
                                 WHERE ISBN = ISBN_C);
   --E lo restituisce
   RETURN COPIE_DISPONIBILI;

EXCEPTION
   WHEN NO_DATA_FOUND THEN DBMS_OUTPUT.PUT_LINE  ('Il libro non è presente in
biblioteca, si prega di sceglierne un altro');

END;
```

VISUALIZZA_POSTI_DISPONIBILI_EVENTO

Viene utilizzata dalla vista Eventi_Disponibili per visualizzare gli eventi che si terranno in biblioteca e i relativi posti disponibili.
Il parametro della funzione è la data in cui si terrà l'evento.

```
CREATE OR REPLACE FUNCTION VISUALIZZA_POSTI_DISPONIBILI_EVENTO (DATA_EVENTO)
RETURN NUMBER
IS
   POSTI_OCCUPATI NUMBER(2, 0);     --Numero di posti occupati per l'evento

BEGIN
```

```
   --Conta il numero di posti disponibili per l'evento
   SELECT COUNT(*) INTO POSTI_OCCUPATI
   FROM SEGUITO
   WHERE TRUNC(DATA_E_ORA_EVENTO) = TRUNC(DATA_EVENTO) AND
TRUNC(DATA_E_ORA_EVENTO) > SYSDATE;

   --Restituisce il numero di posti disponibili per l'evento
   RETURN (50 - POSTI_OCCUPATI);

EXCEPTION
   WHEN NO_DATA_FOUND    THEN DBMS_OUTPUT.PUT_LINE  ('Evento non trovato');

END;
```

VISUALIZZA_VALUTAZIONE_MEDIA_LIBRO

Viene utilizzata dalla vista Libri_Catalogati per visualizzare la valutazione media dei libri catalogati in biblioteca.
Il parametro della funzione è l'ISBN del libro.

```
CREATE OR REPLACE FUNCTION VISUALIZZA_VALUTAZIONE_MEDIA_LIBRO (ISBN_C
LIBRO.ISBN%TYPE)
RETURN NUMBER
IS
   VALUTAZIONE_MEDIA NUMBER(2, 0);    --Valutazione media delle recensioni del libro

BEGIN
   --Calcola la valutazione media delle recensioni libro
   SELECT AVG(VALUTAZIONE) INTO VALUTAZIONE_MEDIA
   FROM RECENSIONE
   WHERE ISBN = ISBN_C;

   --Restituisce la valutazione media delle recensioni del libro
   RETURN VALUTAZIONE_MEDIA;

EXCEPTION
   WHEN NO_DATA_FOUND    THEN DBMS_OUTPUT.PUT_LINE  ('Libro non trovato');

END;
```

VISUALIZZA_NUMERO_COPIE

Viene utilizzata dalla vista Copie_Disponibili per visualizzare il numero totale di copie di un libro presenti in biblioteca.
Il parametro della funzione è l'ISBN del libro.

```
CREATE OR REPLACE FUNCTION VISUALIZZA_NUMERO_COPIE (ISBN_C LIBRO.ISBN%TYPE)
RETURN NUMBER
IS
   COPIE_TOTALI NUMBER(3, 0);  --Numero totali di copie del libro presenti in biblioteca
```

```
BEGIN
   --Conta il numero di copie del libro presenti in biblioteca
   SELECT COUNT(NUMERO_COPIA) INTO COPIE_TOTALI
   FROM COPIA
   WHERE ISBN = ISBN_C;

   --E lo restituisce
   RETURN COPIE_TOTALI;

EXCEPTION
   WHEN NO_DATA_FOUND THEN DBMS_OUTPUT.PUT_LINE  ('Il libro non è presente in
biblioteca, si prega di sceglierne un altro');

END;
```

VISUALIZZA_ORDINE

Viene utilizzata dalla vista Copie_Ordinate per visualizzare il numero totale di copie ordinate dalla biblioteca per ogni libro.
Il parametro d'ingresso della funzione è l'ISBN del libro.

```
CREATE OR REPLACE FUNCTION VISUALIZZA_ORDINE (ISBN_C LIBRO.ISBN%TYPE)
RETURN NUMBER
IS
   NUM_COP_ACQ NUMBER(3, 0);      --Numero totale di copie del libro ordinate

BEGIN
   --Calcola il numero totale di copie del libro ordinate
   SELECT SUM(NUMERO_COPIE_ACQUISTATE) INTO NUM_COP_ACQ
   FROM ORDINE
   WHERE ISBN_COPIE_ACQUISTATE = ISBN_C;

   --E lo restituisce
   RETURN NUM_COP_ACQ;

EXCEPTION
   WHEN NO_DATA_FOUND    THEN DBMS_OUTPUT.PUT_LINE  ('Ordine non trovato');

END;
```

VISTE

Per regolamentare l'accesso ai dati del database, sono state definite le seguenti viste.

EVENTI_DISPONIBILI

Permette a un cliente di visualizzare i prossimi eventi che si terranno in biblioteca e i relativi posti disponibili, utilizzando la function Visualizza_Posti_Disponibili_Evento.

```
CREATE OR REPLACE VIEW EVENTI_DISPONIBILI AS
SELECT DISTINCT DATA_E_ORA_EVENTO, VISUALIZZA_POSTI_DISPONIBILI_EVENTO
(DATA_E_ORA_EVENTO) AS POSTI_DISPONIBILI
FROM EVENTO
WHERE TRUNC(DATA_E_ORA_EVENTO) > SYSDATE;
```

LIBRI_CATALOGATI

Permette a un cliente di visualizzare i libri catalogati all'interno della biblioteca ed eventuali informazioni come l'ISBN, il titolo e il genere di appartenenza, oltre al numero di copie disponibili al momento, e quindi quelle prestabili, e alla valutazione media di ognuno di essi.
Utilizza le function Visualizza_Disponibilita_Copie e Visualizza_Valutazione_Media_Libro.

```
CREATE OR REPLACE VIEW LIBRI_CATALOGATI AS
SELECT DISTINCT ISBN, TITOLO, GENERE, VISUALIZZA_DISPONIBILITA_COPIE (ISBN) AS
COPIE_DISPONIBILI, VISUALIZZA_VALUTAZIONE_MEDIA_LIBRO (ISBN) AS VALUTAZIONE_MEDIA
FROM LIBRO;
```

RECENSIONI_EFFETTUATE

Permette a un cliente di visualizzare i libri che ha recensito e le valutazioni che ha assegnato.

```
CREATE OR REPLACE VIEW RECENSIONI_EFFETTUATE AS
SELECT DISTINCT TITOLO AS LIBRO_RECENSITO, VALUTAZIONE, DATA_RECENSIONE AS DATA,
NOME_AUTORE, COGNOME_AUTORE, NUMERO_TESSERA
FROM RECENSIONE R JOIN LIBRO L ON R.ISBN = L.ISBN JOIN SCRITTO S ON L.ISBN = S.ISBN JOIN
AUTORE A ON S.ISNI = A.ISNI;

--Utilizzo della vista
SELECT LIBRO_RECENSITO, NOME_AUTORE, COGNOME_AUTORE, VALUTAZIONE, DATA
FROM RECENSIONI_EFFETTUATE
WHERE NUMERO_TESSERA = '12345678';
```

PRESTITI_EFFETTUATI

Permette ad un cliente di visualizzare i libri che ha attualmente in prestito e la data di scadenza entro la quale deve restituirli.

```
CREATE OR REPLACE VIEW PRESTITI_EFFETTUATI AS
SELECT DISTINCT TITOLO AS LIBRO_PRESO_IN_PRESTITO, DATA_INIZIO_PRESTITO AS
INIZIO_PRESTITO, DATA_SCADENZA_PRESTITO AS SCADENZA, R.NUMERO_TESSERA
FROM REGISTRAZIONE R JOIN PRENDE P ON R.NUMERO_TESSERA = P.NUMERO_TESSERA JOIN

CREATE OR REPLACE VIEW COPIE_DISPONIBILI AS
SELECT DISTINCT L.ISBN, TITOLO, GENERE, VISUALIZZA_NUMERO_COPIE (L.ISBN) AS
COPIE_DISPONIBILI, LETTERA_MENSOLA, NUMERO_PIANO
FROM LIBRO L JOIN COPIA C ON L.ISBN = C.ISBN;
```

COPIE_ORDINATE

Permette al direttore di visualizzare il numero di copie di ogni libro ordinate e il numero di copie effettivamente presente in biblioteca per poter controllare eventuali incongruenze.

Utilizza le function Visualizza_Numero_Copie e Visualizza_Disponibilita_Copie.

```
CREATE OR REPLACE VIEW COPIE_ORDINATE AS
SELECT DISTINCT ISBN, VISUALIZZA_NUMERO_COPIE(ISBN) AS COPIE_DISPONIBILI,
VISUALIZZA_ORDINE(ISBN) AS NUMERO_COPIE_ORDINATE
FROM LIBRO;
```

DATA CONTROL LANGUAGE

Permette di impostare i privilegi agli utenti, identificati da una password, per decidere a quali dati possono e non possono accedere.
I seguenti privilegi sono assegnati agli utenti seguendo le indicazioni della tabella **Utenti e Categoria** precedentemente descritta.

REGISTRATO

```
GRANT CONNECT, CREATE SESSION TO REGISTRATO;
GRANT SELECT  ON MULTA TO REGISTRATO;
GRANT SELECT, INSERT, UPDATE, DELETE ON RECENSIONE TO REGISTRATO;
GRANT EXECUTE ON EFFETTUA_PRESTITO TO REGISTRATO;
GRANT EXECUTE ON EFFETTUA_RESTITUZIONE TO REGISTRATO;
GRANT EXECUTE ON PRENOTA_POSTO_EVENTO TO REGISTRATO;
GRANT EXECUTE ON DISDICI_PRENOTAZIONE_POSTO_EVENTO TO REGISTRATO;
GRANT SELECT  ON EVENTI_DISPONIBILI TO REGISTRATO;
GRANT SELECT  ON LIBRI_CATALOGATI TO REGISTRATO;
GRANT SELECT  ON PRESTITI_EFFETTUATI TO REGISTRATO;
GRANT SELECT  ON RECENSIONI_EFFETTUATE TO REGISTRATO;
```

ADDETTO

```
GRANT CONNECT, CREATE SESSION TO ADDETTO;
GRANT SELECT ON TURNO TO ADDETTO;
GRANT SELECT, UPDATE, INSERT ON ASSISTE TO ADDETTO;
GRANT SELECT, UPDATE ON REGISTRAZIONE TO ADDETTO;
GRANT SELECT, UPDATE ON PRENDE TO ADDETTO;
GRANT EXECUTE ON POSIZIONA_COPIE TO ADDETTO;
GRANT SELECT ON SCAFFALE TO ADDETTO;
GRANT SELECT ON MENSOLA TO ADDETTO;
GRANT SELECT ON ORDINE TO ADDETTO;
GRANT EXECUTE ON RINNOVA_TESSERA TO ADDETTO;
GRANT SELECT ON COPIE_DISPONIBILI TO ADDETTO;
```

DIRETTORE

```
GRANT CONNECT, CREATE SESSION TO DIRETTORE;
GRANT SELECT, INSERT, DELETE ON BIBLIOTECARIO TO DIRETTORE;
GRANT SELECT, INSERT, UPDATE, DELETE ON TURNO TO DIRETTORE;
GRANT SELECT, INSERT, UPDATE, DELETE ON EVENTO TO DIRETTORE;
GRANT INSERT, DELETE ON LIBRO TO DIRETTORE;
GRANT INSERT, DELETE ON COPIA TO DIRETTORE;
GRANT INSERT, DELETE ON AUTORE TO DIRETTORE;
```

```
GRANT SELECT, INSERT, UPDATE, DELETE ON SCAFFALE TO DIRETTORE;
GRANT SELECT, INSERT, UPDATE, DELETE ON MENSOLA TO DIRETTORE;
GRANT SELECT, INSERT, UPDATE ON ORDINE TO DIRETTORE;
GRANT SELECT, INSERT, UPDATE, DELETE ON CASA_EDITRICE TO DIRETTORE;
GRANT EXECUTE ON CONTROLLA_INVENTARIO TO DIRETTORE;
GRANT SELECT ON COPIE_ORDINATE TO DIRETTORE;
```

AMMINISTRATORE

```
GRANT CONNECT, CREATE SESSION TO AMMINISTRATORE;
GRANT ALL PRIVILEGIES TO AMMINISTRATORE;
```

SCHEDULER

L'utilizzo di uno scheduler permette la programmazione di operazioni periodiche, con periodi di tempo stabiliti, come manutenzione, aggiornamenti ed eliminazione dei dati.
In questo caso, lo scheduler è utilizzato per la rimozione di dati obsoleti dal database, in quanto non necessari al suo futuro funzionamento.

```
BEGIN
        DBMS_SCHEDULER.CREATE_JOB
        (
                job_name   => 'Pulizia_Database',
                job_type    => 'PLSQL_BLOCK',
                job_action  =>
                'BEGIN
                        --Elimina gli eventi precedenti di un mese e le informazioni relative ai
                          partecipanti
                        DELETE FROM SEGUITO
                        WHERE DATA_E_ORA_EVENTO < SYSDATE - 30;
                        DELETE FROM EVENTO
                        WHERE DATA_E_ORA_EVENTO < SYSDATE - 30;

                        --Elimina i turni precedenti di un mese
                        DELETE FROM TURNO
                        WHERE DATA_E_ORA_TURNO < SYSDATE - 30;

                        --Elimina i prestiti precedenti di un mese
                        DELETE FROM PRENDE
                        WHERE DATA_SCADENZA_PRESTITO < SYSDATE - 30;

                        --Elimina le multe precedenti di un mese
                        DELETE FROM MULTA
                        WHERE DATA_PAGAMENTO < SYSDATE - 30;
                  END;',
                start_date      => TO_DATE('01-SEP-2023', 'DD-MM-YYYY'),
                repeat_interval => 'FREQ = MONTHLY',
                end_date        => NULL,
                auto_drop       => FALSE,
                enabled         => TRUE,
```

```
                comments        => 'Effettua la pulizia del database eliminando dati obsoleti.'
        );
END;

--Eliminazione dello scheduler
BEGIN
   DBMS_SCHEDULER.DROP_JOB('Pulizia_Database');
END;
```