



UNIVERSITÀ
DEGLI STUDI
FIRENZE

**Scuola di Scienze
Matematiche
Fisiche e Naturali**

RELAZIONE BASI DI DATI E SISTEMI INFORMATIVI

DATABASE BIBLIOTECA UNIVERSITARIA

Autore:

Matteo Pascuzzo - matteo.pascuzzo@edu.unifi.it - 7072913

Informazioni progetto:

Università degli Studi di Firenze

Anno: 2024

Ambiente di sviluppo: MySQL Workbench

Versione: MySQL 8.4.0

BREVE DESCRIZIONE

Il progetto volge alla creazione e allo sviluppo di una base di dati elaborata e progettata per gestire efficacemente le biblioteche dislocate in diversi dipartimenti universitari facenti parte di una stessa università. Questo sistema non solo centralizza le informazioni, ma consente anche operazioni dinamiche e collaborative tra i vari dipartimenti, migliorando l'accessibilità e l'efficienza del servizio bibliotecario per studenti, professori e utenti esterni.

Il principale ruolo della base di dati è quello di permettere alle biblioteche situate all'interno dei vari dipartimenti universitari di effettuare una corretta gestione dei noleggi e dei resi effettuati dai vari utenti.

Ciò sarà possibile attraverso diverse funzionalità, quali:

1. **Gestione degli Utenti:** Gli utenti sono classificati in tre categorie principali: studenti, professori e utenti esterni. Ogni utente dispone di un profilo personale, che include un indirizzo email e una password necessari per accedere al sito della biblioteca. Questo profilo permette agli utenti di effettuare prenotazioni, visualizzare lo stato dei noleggi e consultare lo storico delle loro attività.
2. **Gestione del Personale Bibliotecario:** Il personale bibliotecario ha accesso a un'interfaccia gestionale che consente di eseguire varie operazioni essenziali come noleggi, acquisti e resi.
3. **Gestione di Prenotazioni e Prestiti:** Per effettuare una prenotazione (e quindi successivamente usufruire di un noleggio), gli utenti devono accedere al sito della biblioteca utilizzando le proprie credenziali (email e password). Una volta autenticati, possono visualizzare un catalogo completo degli articoli disponibili, con dettagli specifici su ciascun articolo.

PROGETTAZIONE CONCETTUALE

Si voglia analizzare ora la gestione di biblioteche locate in vari dipartimenti universitari facenti parte di una stessa università. Tali biblioteche devono gestire tante prenotazioni e richieste di noleggio effettuate da persone interne ed esterne al dipartimento.

L'utente che si approccia al sistema verrà identificato tramite nome, cognome, indirizzo e-mail ed un attributo che può essere studente, docente o esterno.

Il dipartimento è rappresentato da proprio codice identificativo, il nome, la città in cui si trova e l'orario di chiusura in modo tale da sapere se è possibile accettare o meno prenotazioni effettuate dagli utenti.

La biblioteca è identificata dal proprio nome, email, numero di telefono e dal nome del dipartimento nel quale è collocata.

Sarà possibile visualizzare un elenco degli articoli che la biblioteca offre in modo tale che l'utente possa scegliere quale vuole e possa così inoltrare la richiesta di noleggio. L'utente potrà fare una prenotazione nella quale specifica l'articolo richiesto, la quantità, e la data di inizio del noleggio la quale coinciderà con la data di effettuazione della richiesta, dato che quell'articolo non sarà più prenotabile da nessun altro utente. Inoltre l'utente potrà richiedere un solo articolo per prenotazione, ergo se uno stesso utente vuole ricevere in prestito più articoli dovrà effettuare più prenotazioni.

Al termine della scelta dell'articolo, l'utente dovrà confermare la propria prenotazione tramite l'inserimento della propria password, effettuando quindi un login nel sito della biblioteca che quindi segnerà quella prenotazione come effettuata e restituirà la data di effettuazione della prenotazione, il nome dell'oggetto richiesto e la tipologia.

DIZIONARI

Definite le caratteristiche del progetto e terminata la fase di progettazione concettuale si procede a catalogare tutte le entità coinvolte nel sistema. Verranno definiti, per ciascuna entità, gli attributi, le chiavi primarie e i vincoli di integrità tramite la seguente tabella:

Dizionario delle entità

Nome	Descrizione	Chiave	Attributo
Utente	Fruitore del servizio. Può essere studente, docente o esterno	Email (primary key)	Nome Cognome
Dipartimento	Struttura che ha al proprio interno la Biblioteca e i suoi Utenti	CodDipartimentale (primary key)	Nome Città OrarioChiusura
Biblioteca	Entità centrale del servizio	LibraryID (primary key)	Nome Email Telefono
Prenotazione	Prenotazione di un articolo della Biblioteca	Id (primary key)	ArticoloRichiesto Quantità Data
Conferma	Conferma data dall'utente al termine della Prenotazione	Id (primary key)	ArticoloRichiesto Tipo Data
Articolo	Articolo disponibile per la Prenotazione	Id (primary key)	Nome Tipo

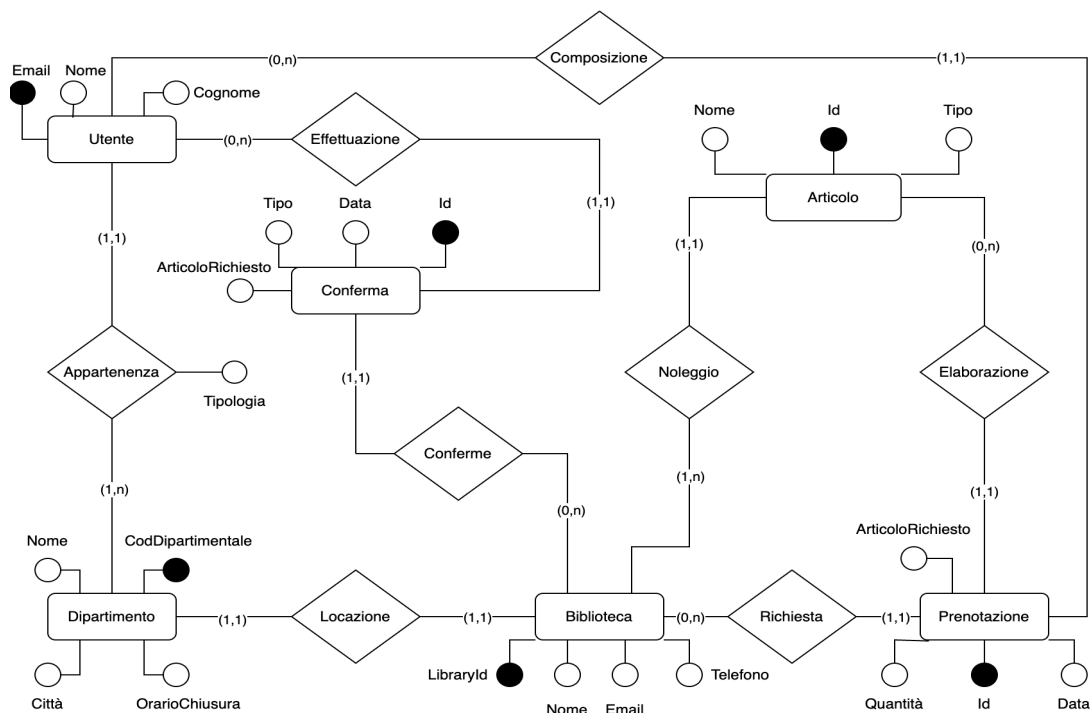
Dizionario delle relazioni

Nome	Descrizione	Entità	Attributo
Composizione	Effettuazione di richiesta di noleggio di un Articolo da parte di un Utente	Utente (0,n) Prenotazione (1,1)	X
Effettuazione	Conferme di noleggio effettuate da un Utente	Utente (0,n) Conferma (1,1)	X
Appartenenza	Decreta a quale Dipartimento appartiene un Utente	Utente (1,1) Dipartimento (1,n)	Tipologia
Conferme	Richieste completate ed effettuate nella Biblioteca	Conferma (1,1) Biblioteca (0,n)	X
Locazione	Locazione della Biblioteca di un Dipartimento	Dipartimento (1,1) Biblioteca (1,1)	X
Noleggio	Lista degli Articoli che un Utente può noleggiare nella Biblioteca	Articolo (1,1) Biblioteca (1,n)	X
Richiesta	Richieste di noleggio effettuate ad una Biblioteca	Biblioteca (0,n) Prenotazione (1,1)	X
Elaborazione	Ultimazione di una Richiesta di noleggio	Articolo (0,n) Prenotazione (1,1)	X

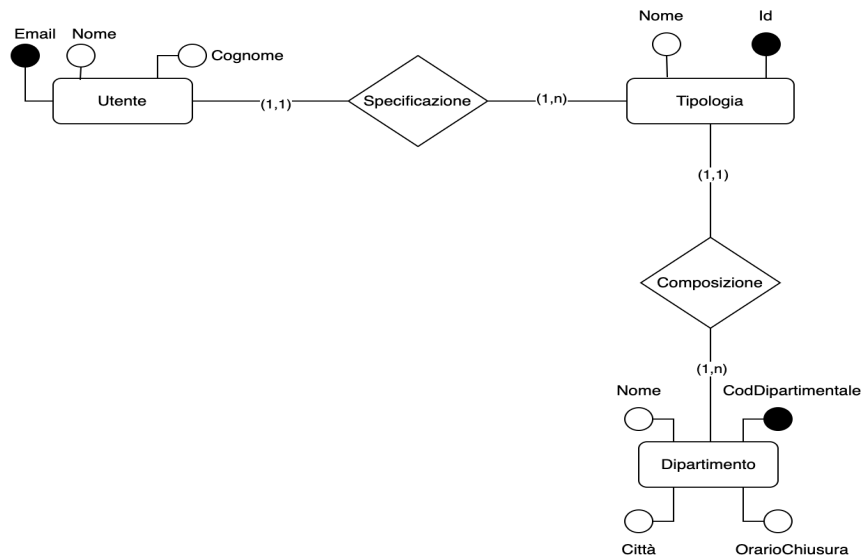
SCHEMA CONCETTUALE

Si procede ora alla costruzione di un primo schema concettuale che sarà successivamente rimodellato. Verranno elencate in seguito tutte le entità con i relativi attributi e le altre entità con le quali si collega tramite relazione.

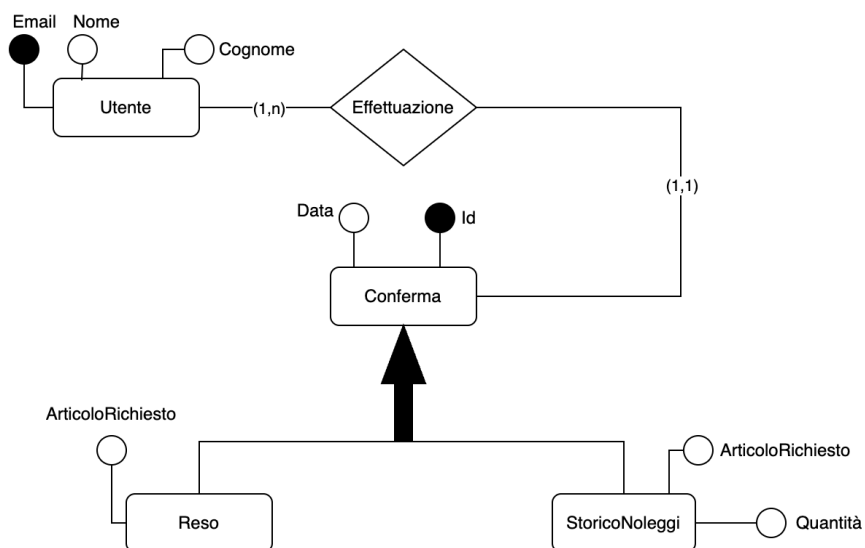
- Biblioteca ha come attributi il Codice Unico della Biblioteca (LibraryID), Nome, Email e Telefono e si collega con le entità Dipartimento, Conferma, Articolo e Prenotazione.
- Utente ha come attributi il proprio indirizzo email, il Nome ed il Cognome. Si collega alle entità Dipartimento, Conferma e Prenotazione.
- Dipartimento ha come attributi il proprio codice dipartimentale univoco, Nome, Città e l'orario di chiusura del dipartimento stesso, è collegata tramite relazione alle entità Utente e Biblioteca.
- Prenotazione è identificata univocamente da un proprio Id, l'articolo in oggetto, la quantità e la data in cui si è effettuata la prenotazione. Si collega alle entità Utente, Articolo e Biblioteca.
- Conferma ha come attributi il proprio Id univoco, nome, tipo e la data in cui si è confermata la prenotazione. Si collega alle entità Utente e Biblioteca.
- Articolo ha come attributi un proprio Id che lo determina univocamente, il Nome e il Tipo.

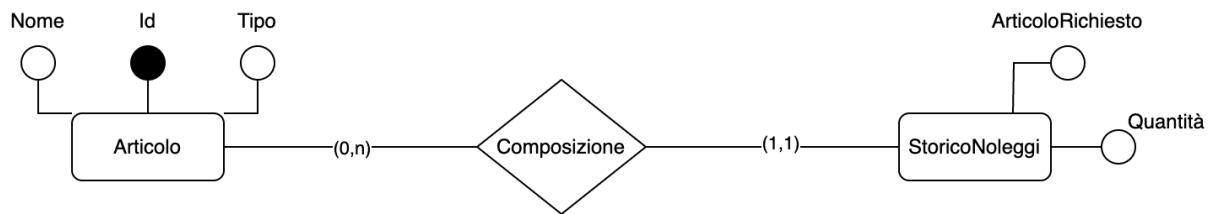


In seguito alla prima definizione dello schema, emerge l'esigenza di rendere esplicita la relazione binaria Utente – Appartenenza– Dipartimento, per poter così formalizzare la Tipologia di Utente che si interfaccia con la Biblioteca.



L'entità Conferma viene inoltre divisa in altre due entità tramite generalizzazione, ottenendo le entità Reso e StoricoNoleggi. In questo modo sarà possibile distinguere tra i due diversi tipi di conferma: di reso o di noleggio. Sarà inoltre possibile confermare la propria volontà di rendere un articolo alla biblioteca, non solo nel momento effettivo dello scadere del tempo di noleggio ma anche prima. L'entità StoricoNoleggi tiene conto di tutti i noleggi chiesti dall'utente e quindi bisogna creare una nuova relazione con l'entità Articolo.





PROGETTAZIONE LOGICA

Tabella dei Volumi

La base di dati è stata concepita per la gestione di biblioteche all'interno di dipartimenti universitari di uno stesso ateneo. Si considera che un ateneo sia composto da circa 20 dipartimenti. Ogni dipartimento ha al proprio interno una biblioteca alla quale uno studente può fare richiesta di noleggio o rendere un articolo che aveva già preso in prestito. Ogni biblioteca riceve circa 20 richieste di noleggio al giorno. Il numero medio di studenti appartenenti ad uno stesso dipartimento è circa 2mila e, considerando anche insegnanti e utenti esterni possiamo arrivare a 2050 utenti totali per dipartimento, quindi il numero totale di utenti del database sarà circa 41mila. Ogni biblioteca offre circa 15 tipi di articoli diversi. Si consideri che ogni utente faccia 2 resi al mese. Tipologia può assumere 3 differenti valori: un utente può essere uno studente, un docente oppure una persona esterna all'ateneo. Le parentesi tonde all'interno della colonna volume specificano che le relazioni sono di tipo (1,1) con l'entità specificata tra parentesi, di conseguenza i loro volumi coincidono.

CONCETTO	TIPO	VOLUME
Biblioteca	E	20
Conferma	E	10000
Noleggi	E	10000
Reso	E	8000
Dipartimento	E	20
Tipologia	E	1000
Utente	E	41000

Prenotazione	E	10000
Articolo	E	700
Composizione Prenotazione	R	10000
Effettuazione	R	10000
Locazione	R	20
Conferma	R	10000
Richiesta	R	10000
Noleggio	R	700
Elaborazione	R	10000
Composizione Dipartimento	R	1000
Specificazione	R	41000
Composizione Noleggi	R	10000

Tabella degli Accessi

Si analizzano ora gli accessi al database effettuati in un giorno. Si assume che il numero di accessi alla tabella Biblioteca corrisponda alle volte in cui è necessario ottenere qualsiasi tipo di informazione relativa alla Biblioteca. Nella tabella Conferme, invece, si prendono in considerazione, ad esempio, le consultazioni dello storico dei Noleggi o tutte le volte in cui si effettuano Noleggi o Resi con operazioni di scrittura.

CONCETTO	COSTRUTTO	ACCESSI	TIPO
Biblioteca	E	10000	Lettura
Conferma	E	5000	Lettura/Scrittura
Noleggi	E	10000	Lettura/Scrittura
Reso	E	4000	Lettura/Scrittura
Dipartimento	E	20	Lettura/Scrittura
Tipologia	E	1000	Lettura

Utente	E	200	Lettura
Prenotazione	E	4000	Lettura/Scrittura
Articolo	E	40000	Lettura
Composizione Prenotazione	R	4000	Lettura/Scrittura
Effettuazione	R	1000	Lettura/Scrittura
Locazione	R	500	Lettura/Scrittura
Conferma	R	3000	Lettura/Scrittura
Richiesta	R	4000	Lettura/Scrittura
Noleggio	R	10000	Lettura/Scrittura
Elaborazione	R	40000	Lettura/Scrittura
Composizione Dipartimento	R	600	Lettura
Specificazione	R	200	Lettura
Composizione Noleggi	R	8000	Lettura/Scrittura

RISTRUTTURAZIONE SCHEMA ER

Dopo aver terminato il modello concettuale, è indispensabile adattarlo per la conversione in un progetto logico e, successivamente, per la sua realizzazione pratica. Questo processo comporta la verifica di quattro fasi che verranno illustrate in dettaglio: analisi delle ridondanze, rimozione delle generalizzazioni, partizionamento e fusione dei concetti e la selezione degli identificatori primari.

Analisi delle ridondanze

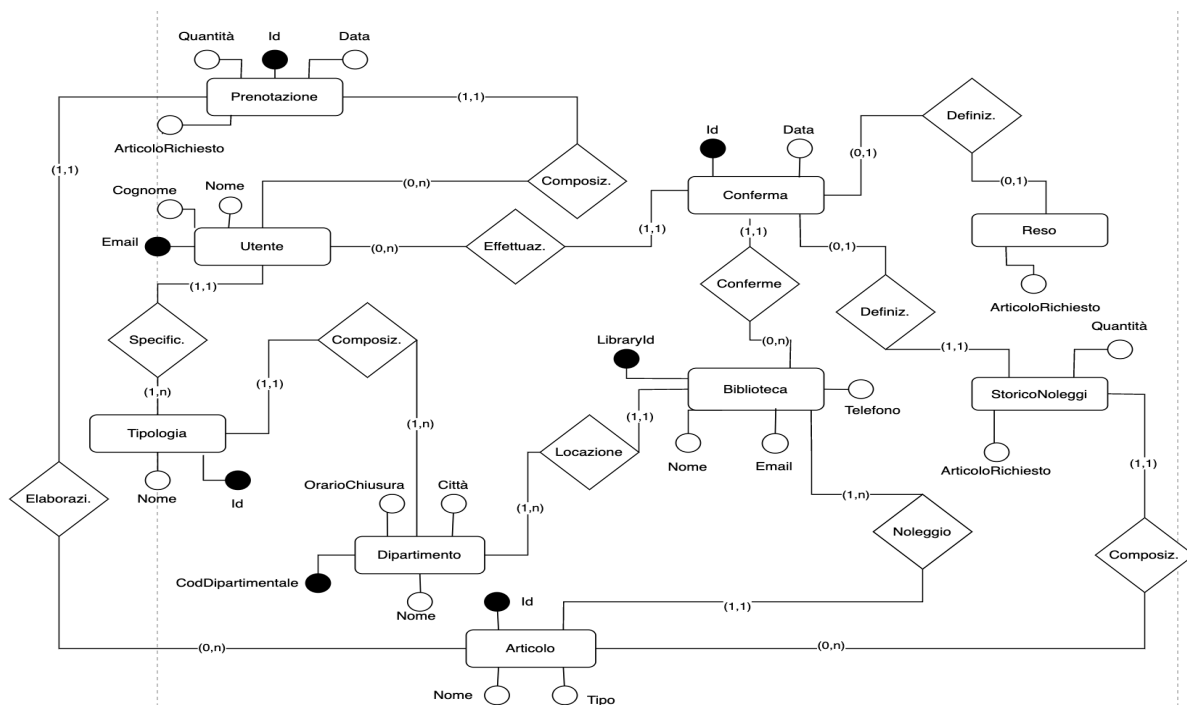
L'analisi delle ridondanze comporta l'eliminazione di attributi superflui. Esaminando il modello ER, viene individuata una ridondanza tra l'entità 'Biblioteca' e l'entità 'Prenotazione' attraverso la relazione Richiesta. Si può infatti determinare la Biblioteca tramite l'entità 'Articolo'. La risoluzione di questa ridondanza è essenziale per evitare possibili anomalie, motivo per cui è stato deciso di eliminare l'associazione tra le entità 'Biblioteca' e 'Prenotazione', mantenendo invece quella tra 'Prenotazione' e 'Articolo'.

La rimozione delle generalizzazioni si è resa necessaria dopo aver formalizzato l'Entità principale Conferma e le due entità derivate Reso e StoricoNoleggi. Questa generalizzazione è totalmente esclusiva ed è stato necessario sostituirla con due associazioni, imponendo la chiave di transazione come chiave per le entità derivate.

Viene verificato che il flusso dei dati sia consono. Non ci sono attributi multi valore né altre entità che vadano modificate in questo senso.

Non c'è bisogno di modifiche poiché ogni entità ha già una chiave primaria definita durante la progettazione.

Schema finale ristrutturato



TRADUZIONE AL MODELLO LOGICO

Di particolare utilità prima di avviare la fase di traduzione è la possibilità di rappresentare tutte le associazioni e le entità in una forma pseudo-tabellare. Questo metodo facilita la visualizzazione delle informazioni e semplifica il processo di trasformazione. Attraverso la rappresentazione tabellare, si possono organizzare i dettagli delle associazioni e delle entità in modo strutturato, rendendo l'analisi e la comprensione del contesto più agevoli. In particolare, il processo di traduzione adottato ha permesso di trasformare ogni relazione del tipo $(1,1) - (1,n)$ nell'entità dal lato $(1,1)$. Ad esempio, la relazione tra Prenotazione e Articolo viene collassata, definendo Articolo come chiave esterna di Prenotazione.

Dipartimento (CodDipartimentale, Nome, Città, OrarioChiusura)

Tipologia (Id, Dipartimento, Nome)

Primary Key: Id

Tipologia.Dipartimento → Dipartimento.CodDipartimentale

Utente (Email, Tipologia, Nome, Cognome)

Primary Key: Email

Utente.Tipologia → Tipologia.Id

Biblioteca (LibraryId, Dipartimento, Nome, Email, Telefono)

Primary Key: LibraryId

Biblioteca.Dipartimento → Dipartimento.CodDipartimentale

Articolo (Id, Biblioteca, Nome, Tipo)

Primary Key: Id

Articolo.Biblioteca → Biblioteca.LibraryId

Prenotazione (Id, Utente, Articolo, ArticoloRichiesto, Quantità, Data)

Primary Key: Id

Prenotazione.Utente → Utente.Email

Prenotazione.Articolo → Articolo.Id

Conferma (Id, Utente, Biblioteca, Data)

Primary Key: Id

Conferma.Utente → Utente.Email

Conferma.Biblioteca → Biblioteca.LibraryId

StoricoNoleggi (Conferma, Articolo, Quantità, ArticoloRichiesto)

Primary Key: Conferma

StoricoNoleggi.Conferma → Conferma.Id

Storiconoleggi.Articolo → Articolo.Id

Reso (Conferma, ArticoloRichiesto)

PrimaryKey: Conferma

Reso.Conferma → Conferma.Id

DIPENDENZE FUNZIONALI

Dipartimento:

CodDipartimentale → Nome, Città, OrarioChiusura

Tipologia:

Id → Dipartimento, Nome

Utente:

Email → Tipologia, Nome, Cognome

Biblioteca:

LibraryId → Dipartimento, Nome, Email, Telefono

Articolo:

Id → Biblioteca, Nome, Tipo

Prenotazione:

Id → Utente, Articolo, Quantità, ArticoloRichiesto, Data

Articolo, Quantità → ArticoloRichiesto

Conferma:

Id → Utente, Biblioteca, Data

StoricoNoleggi:

Conferma → Articolo, Quantità, ArticoloRichiesto

Articolo, Quantità → ArticoloRichiesto

Reso:

Conferma → ArticoloRichiesto

NORMALIZZAZIONE

Prima Forma Normale

Poiché non si hanno attributi multi valore nè attributi con cardinalità, allora siamo in 1FN

Seconda Forma Normale

Siccome tutte le superchiavi sono definiti da un singolo attributo allora non ci sono dipendenze parziali quindi siamo in 2FN

Terza Forma Normale

Si ha la terza forma normale quando, per ogni dipendenza funzionale non banale $x \rightarrow y$, allora x è la superchiave oppure y è un attributo della chiave.

Sorge, quindi, un problema nelle tabelle Prenotazione e StoricoNoleggi. Questo si risolve eliminando l'attributo ArticoloRichiesto dalle due entità. Infatti, ArticoloRichiesto può essere ricavato da Conferma usando l'Id e la data della prenotazione.

Quindi siamo in 3FN.

Di seguito le modifiche alle tabelle Prenotazione e StoricoNoleggi:

Prenotazione (Id, Utente, Articolo, Quantità, Data)

Primary Key: Id

Prenotazione.Utente \rightarrow Utente.Email

Prenotazione.Articolo \rightarrow Articolo.Id

StoricoNoleggi (Conferma, Articolo, Quantità)

Primary Key: Conferma

StoricoNoleggi.Conferma \rightarrow Conferma.Id

StoricoNoleggi.Articolo \rightarrow Articolo.Id

e le rispettive dipendenze funzionali:

Prenotazione:

Id \rightarrow Utente, Articolo, Quantità, Data

StoricoNoleggi:

Conferma \rightarrow Articolo, Quantità

BCNF

Si afferma anche che siamo in BCNF dato che per ogni dipendenza funzionale non banale $x \rightarrow y$, x è la superchiave. Ciò è reso manifesto dalla lista delle dipendenze funzionali esposta sopra.

IMPLEMENTAZIONE SQL

Lo sviluppo SQL inizia dallo schema finale e procede attraverso una serie di fasi sequenziali che garantiscono una corretta implementazione della base di dati. Queste fasi possono essere riassunte come segue:

CREAZIONE DEL DATABASE

Per assicurare che non esista un database preesistente o obsoleto, utilizziamo un'opzione che permette di eliminare il database attuale e ricostruirlo da zero.

INIZIALIZZAZIONE DEL DATABASE

Un'alternativa consiste nel mantenere il database esistente, eliminando però le tabelle al suo interno per poi ricrearle. Questo metodo assicura una visualizzazione corretta del database durante l'esecuzione e previene modifiche strutturali indesiderate alle tabelle.

È fondamentale ricordare che le tabelle sono collegate tramite chiavi esterne (FK), il che richiede particolare attenzione nell'ordine di eliminazione: prima devono essere rimosse le tabelle figlie che dipendono da tabelle padre, e solo successivamente le tabelle padre.

CREAZIONE DELLE TABELLE

In questa fase cruciale, vengono create le tabelle. Tuttavia, l'ordine di creazione è inverso rispetto all'eliminazione, poiché non è possibile definire una dipendenza senza avere già la tabella padre.

CREAZIONE DEI TRIGGER E DELLE VISTE

A questo punto, aggiungiamo i Trigger per imporre condizioni specifiche durante operazioni come inserimenti o cancellazioni. Creiamo anche le viste che permettono un accesso semplificato ai dati.

FASE DI POPOLAMENTO

Durante questa fase, vengono inseriti nel database dati generati casualmente. E' essenziale considerare le dipendenze per evitare errori dovuti alla mancanza di dati. L'inserimento avviene sia tramite file che con codice SQL.

PROCEDURE

Procedura: **EseguiPrenotazione**

Questa procedura riceve come input un utente, una biblioteca, un articolo, la quantità e la data. Inserisce i dati e la data nella tabella 'Conferma', e aggiunge l'id della conferma, l'articolo e la quantità noleggiata nella tabella 'StoricoNoleggi'.

Procedura: **ConfermaPrenotazione**

Questa procedura prende come input l'id della prenotazione. Dopo aver verificato l'utente, chiama la procedura 'EseguiPrenotazione' per confermare la prenotazione, inserendola nella tabella 'StoricoNoleggi'. Il record originale viene rimosso dalla tabella 'Prenotazioni'.

FUNZIONE

Funzione: **BibliotecaAccessibileDaUtente**

Questa funzione accetta come input l'email dell'utente e una biblioteca, restituendo un valore booleano True se l'utente ha il permesso di effettuare Prenotazioni in quella biblioteca. Questo è possibile solo se la biblioteca si trova nello stesso dipartimento dell'utente; in caso contrario, restituisce False.

VISTE

Vista: **UtentiConTipologie**

Questa vista mostra le informazioni di contatto e i nomi degli utenti insieme alla loro tipologia associata. Include l'email, il nome e il cognome dell'utente, e il nome della tipologia.

Vista: **DettagliPrenotazioni**

Questa vista fornisce i dettagli completi delle prenotazioni effettuate dagli utenti. Include l'ID della prenotazione, l'email dell'utente, il nome dell'articolo, l'ID dell'articolo richiesto, la quantità e la data della prenotazione.

Vista: **InfoDipartimentoConBiblioteca**

Questa vista elenca le biblioteche con informazioni complete sui loro dipartimenti associati. Include l'ID della biblioteca, il nome, l'email, il telefono, il nome del dipartimento, la città e l'orario di chiusura del dipartimento.

Vista: **DettagliResi**

Questa vista fornisce i dettagli completi dei resi di articoli. Include l'ID di conferma del reso, l'ID dell'articolo richiesto, l'utente che ha effettuato il reso, la biblioteca di conferma, la data della conferma e il nome dell'articolo.

Vista: **DettagliConfermeUtenti**

Questa vista mostra i dettagli delle conferme delle operazioni con informazioni sugli utenti. Include l'ID della conferma, l'utente, il nome e il cognome dell'utente, il nome della biblioteca e la data della conferma.

Vista: **PrenotazioniBiblioteca**

Questa vista elenca le prenotazioni effettuate presso una biblioteca specifica identificata da LibraryId = 'LIB01'. Include l'ID della prenotazione, l'email, il nome e il cognome dell'utente, il nome dell'articolo, l'ID dell'articolo richiesto, la quantità e la data della prenotazione.

TRIGGER

Trigger: **CheckBibliotecaInPrenotazione**

Questo trigger, chiamato CheckBibliotecaInPrenotazione, viene eseguito prima di ogni inserimento nella tabella Prenotazione. La sua funzione è verificare se l'utente ha i permessi per prenotare un articolo in una specifica biblioteca. Il trigger esegue i seguenti passaggi:

1. Recupera l'ID della biblioteca associata all'articolo che si sta cercando di prenotare.
2. Utilizza la funzione BibliotecaAccessibileDaUtente(...) per determinare se l'utente ha i permessi per prenotare presso quella biblioteca.
3. Se l'utente non ha i permessi (ovvero, se canBook è FALSE), il trigger genera un errore e blocca l'inserimento, visualizzando un messaggio che informa che l'utente non ha i permessi necessari per prenotare presso quella biblioteca.