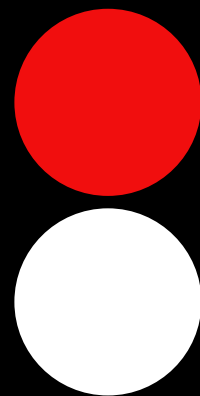


COMIC VERSE

APPLICAZIONE
GESTIONALE
FUMETTERIA



DESCRIZIONE DEL PROGETTO



Gestionale fumetteria

ComicVerse è un gestionale per una fumetteria progettato per gestire l'inventario dei fumetti, le vendite, i clienti e le attività di gestione delle transazioni. Questo sistema sarà utilizzato dai dipendenti della fumetteria per gestire efficacemente tutte le operazioni quotidiane.

Richieste di gestione

Nel contesto della gestione di un negozio di fumetti, è fondamentale disporre di efficaci procedure e operazioni per garantire un corretto funzionamento delle attività quotidiane. Le seguenti sezioni dettagliano le principali operazioni coinvolte nella gestione dell'inventario, delle vendite, dei clienti e delle transazioni finanziarie. Ogni operazione è descritta in modo approfondito, evidenziando i dettagli e le azioni necessarie per eseguirle con successo. Questa guida fornisce un quadro completo delle attività di gestione essenziali per un negozio di fumetti.

Gestione database

La gestione del database in ComicVerse ha un obiettivo chiaro: organizzare, memorizzare e gestire i dati della fumetteria in modo efficiente. Grazie a una struttura ben progettata e relazioni ben definite, il sistema assicura che i dati siano facilmente accessibili e manipolabili. Questo significa che i dipendenti della fumetteria possono svolgere le loro attività senza problemi, avendo sempre a disposizione le informazioni di cui hanno bisogno.

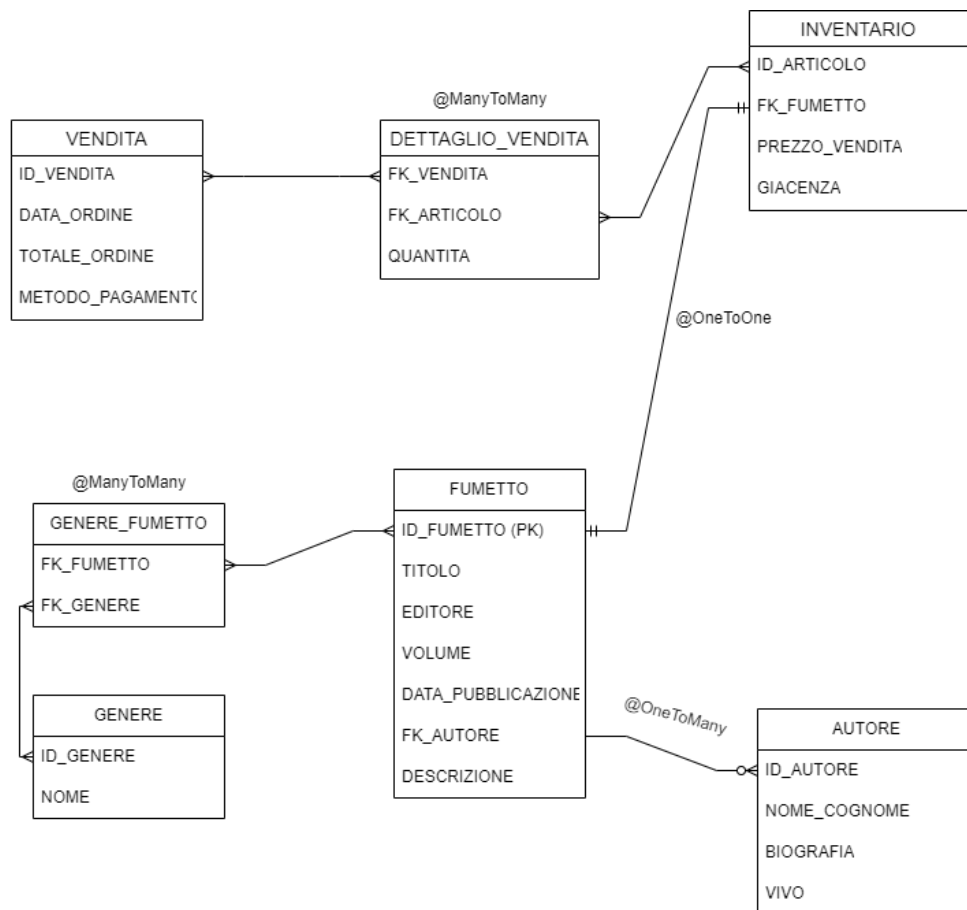


Tabella AUTORE:

Campo	Descrizione	Tipo di Dato
ID_AUTORE	Identificatore univoco	Integer
NOME_COGNOME	Nome e cognome dell'autore	String
BIOGRAFIA	Descrizione della biografia dell'autore	Text
VIVO	Indica se l'autore è deceduto	Boolean

Tabella GENERE:

Campo	Descrizione	Tipo di Dato
ID_GENERE	Identificatore univoco	Integer
NOME	Nome del genere	String

Tabella FUMETTO:

Campo	Descrizione	Tipo di Dato
ID_FUMETTO	Identificatore univoco	Integer
TITOLO	Titolo del fumetto	String
EDITORE	Editore del fumetto	String
VOLUME	Numero del volume	Integer
DATA_PUBBLICAZIONE	Data di pubblicazione del fumetto	Date
FK_AUTORE	Chiave esterna riferente all'autore del fumetto	Integer
DESCRIZIONE	Descrizione del fumetto	String

Tabella GENERI_FUMETTO:

Campo	Descrizione	Tipo di Dato
FK_FUMETTO	Chiave esterna riferente al fumetto nell'inventario	Integer
FK_GENERE	Chiave esterna riferente al genere	Integer

Tabella VENDITA:

Campo	Descrizione	Tipo di Dato
ID_VENDITA	Identificatore univoco	Integer
DATA_ORDINE	Data dell'ordine di vendita	Date
TOTALE_ORDINE	Totale dell'ordine di vendita	Decimal
METODO_PAGAMENTO	Metodo di pagamento utilizzato per l'ordine	ENUM (Contanti, Carta di credito, Bancomat)

Tabella INVENTARIO:

Campo	Descrizione	Tipo di Dato
ID_ARTICOLO	Identificatore univoco	Integer
FK_FUMETTO	Chiave esterna riferente al fumetto nell'inventario	Integer
PREZZO_VENDITA	Prezzo di vendita dell'oggetto	Decimal
GIACENZA	Quantità disponibile in magazzino	Integer

Tabella DETTAGLIO_VENDITA:

ID_DETTAGLIO	Identificatore univoco	Integer
FK_VENDITA	Chiave esterna riferente alla transazione di vendita	Integer
FK_ARTICOLO	Chiave esterna riferente all'articolo venduto	Integer
QUANTITA	Quantità dell'oggetto venduto nella transazione	Integer



Connessioni Essenziali: Esplorando le Relazioni Tra Tabelle del Database

Le relazioni nei database sono connessioni logiche tra tabelle che definiscono come i dati sono correlati tra loro. Queste relazioni consentono di organizzare e strutturare i dati in modo efficiente, consentendo alle applicazioni di recuperare e manipolare le informazioni in modo coerente e accurato.

Le tipologie comuni di relazioni includono "**one-to-one**", "**one-to-many**" e "**many-to-many**", ciascuna con un ruolo specifico nel modellare le interazioni tra i dati.

Autore e fumetto

La relazione "one-to-many" tra le tabelle AUTORE e FUMETTO indica che un autore può essere associato a più fumetti, ma un singolo fumetto può essere associato a un solo autore. Quindi, se un autore ha scritto o disegnato più fumetti, ci sarà una riga corrispondente nella tabella FUMETTO per ciascun fumetto, ciascuna con lo stesso ID_AUTORE che fa riferimento all'ID dell'autore nella tabella AUTORE.

Dettaglio e vendita

La relazione "many-to-many" tra le tabelle DETTAGLIO_VENDITA e VENDITA indica che una vendita può includere molti dettagli di vendita, e allo stesso tempo, un dettaglio di vendita può essere incluso in più vendite. Quindi, per ogni vendita, ci saranno uno o più record nella tabella DETTAGLIO_VENDITA che corrispondono ai dettagli specifici dell'ordine. Allo stesso tempo, ogni dettaglio di vendita nella tabella DETTAGLIO_VENDITA può essere associato a una o più vendite nella tabella VENDITA.

- Un singolo ordine (VENDITA) può includere più fumetti (DETTAGLIO_VENDITA).
- Allo stesso tempo, ogni fumetto può essere incluso in più ordini. Ad esempio, "Spiderman Vol.52" potrebbe essere incluso in un'altra transazione di vendita per un altro cliente.

Fumetto e inventario

La relazione "one-to-one" tra le tabelle FUMETTO e INVENTARIO indica che ogni fumetto presente nell'inventario è associato ad un unico record nella tabella INVENTARIO e viceversa. Quindi, ogni record nella tabella INVENTARIO è associato esclusivamente a un fumetto nella tabella FUMETTO.

Genere e fumetto

La relazione "many-to-many" tra le tabelle GENERE e FUMETTO indica che ogni genere può essere associato a molti fumetti e, allo stesso tempo, ogni fumetto può essere associato a più generi contemporaneamente.

Gestione inserimento dati

Definizione delle classi

Prima di poter gestire i dati della nostra fumetteria, è essenziale definire le strutture dati fondamentali. In questo primo passaggio, creeremo le classi necessarie per rappresentare fumetti, autori e generi.

Crea le seguenti classi Java per rappresentare le entità principali:

- Fumetto: attributi idFumetto(PK), titolo, editore, volume, dataPubblicazione, descrizione, listaGeneri(FK), autoreFumetto(FK).
- Autore: attributi idAutore(PK), nomeCognome, biografia, vivo, listaFumetti(FK).
- Genere: attributi idGenere(PK), nome.
- Inventario: attributi idArticolo(PK), listaFumetti(FK), prezzoVendita, giacenza.
- Vendita: attributi idVendita(PK), dataOrdine, totaleOrdine, metodoPagamento, listaDettaglio(FK).
- Dettaglio: attributi idDettaglio(PK), vendita(FK), oggetto(FK), quantità.
- FumettoDTO: rappresenta un DTO per un fumetto, con gli stessi attributi di Fumetto.
- AutoreDTO: rappresenta un DTO per un autore, con gli stessi attributi di Autore.
- GenereDTO: rappresenta un DTO per un genere, con gli stessi attributi di Genere.
- InventarioDTO: rappresenta un DTO per inventario, con gli stessi attributi di Inventario.
- VenditaDTO: rappresenta un DTO per vendita, con gli stessi attributi di Vendita.

Implementazione delle CRUD e gestione dell'inserimento

In questo passaggio, realizzeremo la logica per gestire l'inserimento dei dati dei fumetti, degli autori e dei generi nel nostro sistema. Inoltre, implementeremo le operazioni CRUD per consentire la creazione, la lettura, l'aggiornamento e l'eliminazione dei dati.

Creare una classe GestioneInserimentoDati che contenga metodi per gestire l'inserimento e le operazioni CRUD dei dati nel sistema:

Create

- Implementa un metodo inserisciNuovoFumetto che accetta un oggetto FumettoDTO come parametro e inserisce il nuovo fumetto nel database della fumetteria.
- Implementa un metodo inserisciNuovoAutore che accetta un oggetto AutoreDTO come parametro e inserisce il nuovo autore nel database.
- Implementa un metodo inserisciNuovoGenere che accetta un oggetto GenereDTO come parametro e inserisce il nuovo genere nel database.
- Implementa un metodo inserisciNuovoArticolo che accetta un oggetto ArticoloDTO come parametro e inserisce il nuovo articolo nel database.
- Implementa un metodo inserisciNuovaVendita che accetta un oggetto VenditaDTO come parametro e inserisce la nuova vendita nel database.

- Implementa un metodo `inserisciNuovoDettaglio` che accetta un oggetto `DettaglioDTO` come parametro e inserisce il nuovo dettaglio nel database.

Read

- Implementa un metodo `recuperaFumettoPerId` che accetta l'ID di un fumetto e restituisce l'oggetto `FumettoDTO` corrispondente.
- Implementa un metodo `recuperaAutorePerId` che accetta l'ID di un autore restituisce l'oggetto `AutoreDTO` corrispondente
- Implementa un metodo `recuperaGenerePerId` che accetta l'ID di un genere e restituisce l'oggetto `GenereDTO` corrispondente
- Implementa un metodo `recuperaArticoloPerId` che accetta l'ID di un articolo e restituisce l'oggetto `ArticoloDTO` corrispondente.
- Implementa un metodo `recuperaVenditaPerId` che accetta l'ID di una vendita e restituisce l'oggetto `VenditaDTO` corrispondente.
- Implementa un metodo `recuperaDettaglioPerId` che accetta l'ID di un dettaglio e restituisce l'oggetto `DettaglioDTO` corrispondente.

Update

- Implementa un metodo `aggiornaFumetto` che accetta un oggetto `FumettoDTO` aggiornato e lo utilizza per aggiornare le informazioni del fumetto nel database.
- Implementa un metodo `aggiornaAutore` che accetta un oggetto `AutoreDTO` aggiornato e lo utilizza per aggiornare le informazioni dell'autore nel database.
- Implementa un metodo `aggiornaGenere` che accetta un oggetto `GenereDTO` aggiornato e lo utilizza per aggiornare le informazioni del genere nel database.
- Implementa un metodo `aggiornaArticolo` che accetta un oggetto `ArticoloDTO` aggiornato e lo utilizza per aggiornare le informazioni dell'articolo nel database.
- Implementa un metodo `aggiornaVendita` che accetta un oggetto `VenditaDTO` aggiornato e lo utilizza per aggiornare le informazioni della vendita nel database.
- Implementa un metodo `aggiornaDettaglio` che accetta un oggetto `DettaglioDTO` aggiornato e lo utilizza per aggiornare le informazioni del dettaglio nel database.

Delete

- Implementa un metodo `eliminaFumettoPerId` che accetta l'ID di un fumetto e lo elimina dal database della fumetteria.
- Implementa un metodo `eliminaAutorePerId` che accetta l'ID di un autore e lo elimina dal database degli autori.
- Implementa un metodo `eliminaGenerePerId` che accetta l'ID di un fumetto e lo elimina dal database dei generi.
- Implementa un metodo `eliminaArticoloPerId` che accetta l'ID di un articolo e lo elimina dal database dell'inventario.
- Implementa un metodo `eliminaVenditaPerId` che accetta l'ID di una vendita e la elimina dal database della vendita.
- Implementa un metodo `eliminaDettaglioPerId` che accetta l'ID di un dettaglio e lo elimina dal database `dettaglio_vendita`.

Inventario fumetteria

Nel contesto della gestione dell'inventario di una fumetteria, avere accesso tempestivo e diretto alle informazioni sui prodotti disponibili è fondamentale, l'accessibilità immediata consente ai dipendenti di monitorare costantemente lo stato delle scorte e di rispondere alle richieste dei clienti.

Ecco alcune funzioni che possono risultare utili per la gestione dell'inventario:

- **Recupero del numero totale di oggetti presenti nell'inventario**

Questa query fornisce il conteggio totale degli oggetti attualmente disponibili nell'inventario della fumetteria.

Consiglio: Utilizza la funzione COUNT(*) per contare il numero totale di righe nella tabella dell'inventario.

- **Recupero del prezzo medio di vendita degli oggetti nell'inventario**

Questa query calcola il prezzo medio di vendita degli oggetti presenti nell'inventario della fumetteria.

Consiglio: Utilizza la funzione AVG() per calcolare il prezzo medio di vendita dei prodotti.

- **Recupero del fumetto con il titolo che inizia con una determinata lettera**

Questa query restituisce tutti i fumetti il cui titolo inizia con una lettera specifica, facilitando la ricerca di fumetti in base ai loro titoli.

Consiglio: Utilizza l'operatore LIKE per filtrare i titoli dei fumetti che iniziano con una lettera specifica.

- **Recupero dei fumetti pubblicati dopo una determinata data**

Questa query restituisce tutti i fumetti pubblicati dopo una data specifica.

- **Recupero degli autori ancora vivi**

Questa query restituirà tutti gli autori che sono ancora vivi.

- **Recupero dell'oggetto con la quantità disponibile più bassa**

Questa query restituirà l'oggetto nell'inventario con la quantità disponibile più bassa.

Consiglio: Utilizza la funzione MIN() per trovare la quantità minima disponibile e quindi recuperare l'oggetto corrispondente, è una subquery.

- **Recupero del fumetto con il prezzo di vendita più alto**

Questa query restituirà il fumetto con il prezzo di vendita più alto.

Consiglio: Utilizza la funzione MAX() per trovare il prezzo di vendita più alto e quindi recuperare il fumetto corrispondente, è una subquery.

- **Recupero dei fumetti disponibili in un certo intervallo di prezzi**

Questa query restituirà tutti i fumetti disponibili in un intervallo di prezzi specificato.

Consiglio: Utilizza un operatore di confronto per filtrare i fumetti in base al prezzo di vendita.

Gestione delle vendite

La gestione delle vendite di una fumetteria, è essenziale accedere rapidamente alle informazioni sui prodotti disponibili. L'accessibilità immediata consente ai dipendenti di monitorare le vendite e soddisfare le esigenze dei clienti. Ecco alcune funzioni che possono risultare utili per la gestione delle vendite.

- **Recupero totale delle vendite**

Questa query restituirà il totale delle vendite.

Consiglio: Utilizza la funzione SUM(TOTALE_ORDINE) per gestire la somma.

- **Recupero del numero totale di ordini**

Questa query restituirà il numero totale di ordini effettuati.

Consiglio: Utilizza la funzione COUNT(*) per contare il numero totale di righe nella tabella degl'ordini.

- **Recupero dell'importo medio per ordine**

Questa query calcolerà l'importo medio per ordine effettuato.

Consiglio: Utilizza la funzione AVG() per calcolare il prezzo medio del totale ordine.

- **Recupero del numero totale di articoli venduti**

Questa query restituirà il numero totale di articoli venduti in tutti gli ordini.

Consiglio: Utilizza la funzione SUM(QUANTITA) per gestire la somma.

API REST

Le API, (Application Programming Interface) sono un insieme di definizioni e protocolli con i quali vengono realizzati e integrati software applicativi. Possono essere considerate come un contratto tra un fornitore di informazioni e l'utente destinatario di tali dati: l'API stabilisce il contenuto richiesto dal consumatore (la chiamata) e il contenuto richiesto dal produttore (la risposta).

Un'API REST utilizzerà una richiesta GET per recuperare un record. Una richiesta POST crea un nuovo record. Una richiesta PUT aggiorna un record e una richiesta DELETE ne elimina uno. Tutti i metodi HTTP possono essere utilizzati nelle chiamate API. Queste informazioni possono essere inviate a un client praticamente in qualsiasi formato, tra cui JavaScript Object Notation (JSON), HTML, XLT, Python, PHP o testo semplice. Noi utilizzeremo JSON in quanto più leggibile sia dagli esseri umani che dalle macchine ed è indipendente dal linguaggio di programmazione.

ControllerFumetti

Il controller per i fumetti gestisce le operazioni relative alla gestione dei fumetti all'interno del sistema ComicVerse. Questo controller fornisce un'interfaccia per l'accesso alle informazioni sui fumetti, inclusa la creazione, la lettura, l'aggiornamento e l'eliminazione di fumetti nel database. Utilizzando le convenzioni REST, il controller per i fumetti definisce diversi endpoint per gestire le operazioni CRUD sui fumetti.

CRUD:

Tipo	Endpoint	Input	Output
Create	POST /fumetti	FumettoDTO	FumettoDTO
	POST /autori	AutoreDTO	AutoreDTO
	POST /generi	GenereDTO	GenereDTO
Read	GET /fumetti/{id}	-	FumettoDTO
	GET /autori/{id}	-	AutoreDTO
	GET /generi/{id}	-	GenereDTO
Update	PUT /fumetti/{id}	FumettoDTO	FumettoDTO
	PUT /autori/{id}	AutoreDTO	AutoreDTO
	PUT /generi/{id}	GenereDTO	GenereDTO
Delete	DELETE /fumetti/{id}	-	Nessun output
	DELETE /autori/{id}	-	Nessun output
	DELETE /generi/{id}	-	Nessun output

ControllerInventario

Il controller per l'inventario gestisce le operazioni relative alla gestione dell'inventario all'interno del sistema ComicVerse. Questo controller fornisce un'interfaccia per l'accesso alle informazioni sull'inventario, inclusa la visualizzazione della disponibilità degli oggetti, l'aggiornamento delle quantità e altre operazioni correlate all'inventario. Utilizzando le convenzioni REST, il controller per l'inventario definisce diversi endpoint per gestire le operazioni relative all'inventario.

CRUD:

Tipo	Endpoint	Input	Output
Create	POST /articoli	ArticoloDTO	ArticoloDTO
	POST /vendite	VenditaDTO	VenditaDTO
	POST /dettagli	DettaglioDTO	DettaglioDTO
Read	GET /articoli/{id}	-	ArticoloDTO
	GET /vendite/{id}	-	VenditaDTO
	GET /dettagli/{id}	-	DettaglioDTO
Update	PUT /articoli/{id}	ArticoloDTO	ArticoloDTO
	PUT /vendite/{id}	VenditaDTO	VenditaDTO
	PUT /dettagli/{id}	DettaglioDTO	DettaglioDTO
Delete	DELETE /articoli/{id}	-	Nessun output
	DELETE /vendite/{id}	-	Nessun output
	DELETE /dettagli/{id}	-	Nessun output

APIREST:

Metodo o HTTP	URL	Input	Output
GET	/inventario/numero-oggetti	-	Integer: Numero totale di oggetti presenti nell'inventario
GET	/inventario/prezzo-medio-vendita	-	Double: Prezzo medio di vendita degli oggetti nell'inventario
GET	/inventario/fumetti-lettera-specifica	@RequestParam ("inizia-con") String lettera	List<Fumetto>: Lista di fumetti il cui titolo inizia con la lettera specificata
GET	/inventario/fumetti-dopo-data	@RequestParam ("pubblicati-dopo") LocalDate data	List<Fumetto>: Lista di fumetti pubblicati dopo la data specificata
GET	/autori/vivi	-	List<Autore>: Lista di autori ancora vivi
GET	/inventario/oggetto-min-quantita	-	Articolo: Oggetto nell'inventario con la quantità disponibile più bassa
GET	/inventario/fumetto-prezzo-massimo	-	Fumetto: Fumetto con il prezzo di vendita più alto
GET	/inventario/fumetti-prezzo-specifico	@RequestParam("prezzo-min") double prezzoMin, @RequestParam("prezzo-max") double prezzoMax	List<Fumetto>: Lista di fumetti disponibili in un intervallo di prezzi specificato

ControllerVendite

Il controller per le vendite gestisce le operazioni relative alla gestione delle vendite all'interno del sistema ComicVerse, fornisce un'interfaccia per l'accesso alle informazioni sulle vendite, inclusa la visualizzazione del totale delle vendite, il numero di ordini effettuati, l'importo medio per ordine e altre operazioni correlate alle vendite. Utilizzando le convenzioni REST, il controller per le vendite definisce diversi endpoint per gestire le operazioni relative alle vendite.

API REST:

Metodo HTTP	URL	Input	Output
GET	/vendite/totale	-	Double: Totale delle vendite
GET	/vendite/numero-ordini	-	Integer: Numero totale di ordini
GET	/vendite/importo-medio-ordine	-	Double: Importo medio per ordine
GET	/vendite/numero-articoli-venduti	-	Integer: Numero totale di articoli venduti

