

SPECIFICHE DB:

Si vuole realizzare una base di dati che permetta di gestire un e-commerce di libri. Gli impiegati, si autenticano al sistema tramite nome utente e una password, inseriscono nuovi prodotti e aggiornano la quantità di questi prodotti. Gli impiegati possono spedire gli ordini fatti dai clienti registrati e possono visualizzare statistiche sulle vendite dei prodotti.

I Libri sono caratterizzati da un codice identificativo, titolo, autore, editore, anno di pubblicazione, descrizione, immagine di copertina, formato, prezzo, quantità, numero di pagine, lingua e genere letterario.

Ogni libro che ha i seguenti parametri uguali

(titolo, autore, editore, anno di pubblicazione, formato, lingua) può essere presente una sola volta all'interno del sistema.

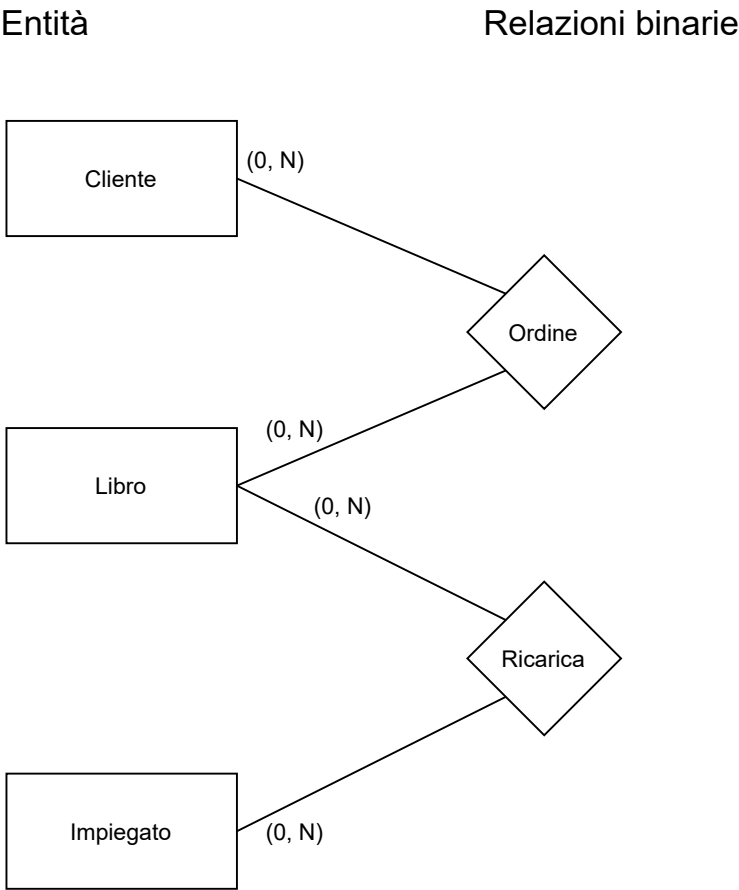
Si vogliono mantenere le informazioni solo sui clienti registrati, caratterizzati da: codice identificativo, nome, cognome, e-mail, numero di telefono indirizzo e città. Tali clienti acquistano i libri e visualizzano i propri ordini.

Si vuole tenere traccia degli ordini dei clienti, ricariche e spedizioni degli impiegati.

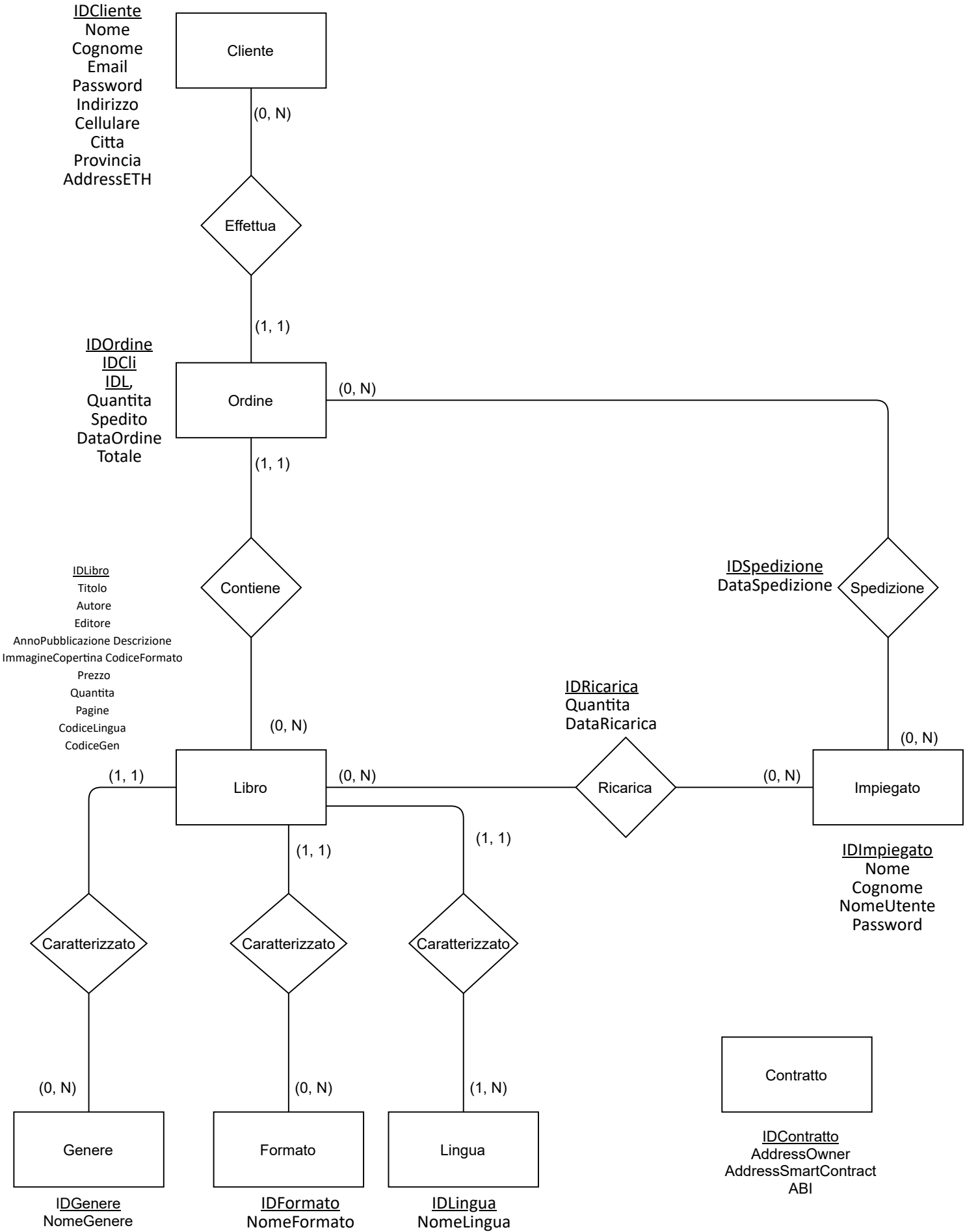
Si vuole inoltre avere una tabella che contenga le informazioni relative a:

- Indirizzo del possessore dello Smart Contract.
- Indirizzo dello Smart Contract.
- ABI.

Schema scheletro



Schema concettuale



SCHEMA LOGICO:

Cliente (IDCliente, Nome, Cognome, Email, Password, Indirizzo, Cellulare, Citta, Provincia, AddressETH)

➔ Vincoli:

- IDCliente: PRIMARY KEY
- Email: UNIQUE
- Nome, Cognome, Email, Password, Indirizzo, Citta: NOT NULL

Impiegato (IDImpiegato, Nome, Cognome, NomeUtente, Password)

➔ Vincoli:

- IDImpiegato: PRIMARY KEY
- NomeUtente: UNIQUE
- Nome, Cognome, NomeUtente, Password: NOT NULL

Genere (IDGenere, NomeGenere)

➔ Vincoli:

- IDGenere: PRIMARY KEY
- NomeGenere: NOT NULL

Formato (IDFormato, NomeFormato)

➔ Vincoli:

- IDFormato: PRIMARY KEY
- NomeFormato: NOT NULL

Lingua (IDLingua, NomeLingua)

➔ Vincoli:

- IDLingua: PRIMARY KEY
- NomeLingua: NOT NULL

Libro (IDLibro, Titolo, Autore, Editore, AnnoPubblicazione, Descrizione, ImmagineCopertina, CodiceFormato, Prezzo, Quantita, Pagine, CodiceLingua, CodiceGen)

➔ Vincoli:

- IDLibro: PRIMARY KEY.
- CodiceFormato → Formato. IDFormato
- CodiceLingua → Lingua. IDLingua
- CodiceGen → Genere. IDGenere
- Titolo, Autore, Editore, AnnoPubblicazione, Descrizione, ImmagineCopertina, CodiceFormato, Prezzo, Quantita, Pagine, CodiceLingua, CodiceGen: NOT NULL
- Titolo, Autore, Editore, AnnoPubblicazione, CodiceFormato, CodiceLingua: UNIQUE

Ordine (IDOrdine, IDCli, IDL, Quantita, Spedito, DataOrdine, Totale)

➔ Vincoli:

- IDOrdine, IDCli, IDL: PRIMARY KEY
- IDCli → Cliente.IDCliente
- IDL → Libro.IDLibro
- Quantita, Spedito, DataOrdine, Totale: NOT NULL

Ricarica (IDRicarica, CodLibro, CodImpiegato, Quantita, DataRicarica)

→ Vincoli:

- IDRicarica, CodLibro, CodImpiegato: PRIMARY KEY
- CodLibro → Libro.IdLibro
- CodImpiegato → Impiegato.IdImpiegato
- Quantita, DataRicarica: NOT NULL

Spedizione(IDSpedizione, CodOrdine, CodImpiegato, DataSpedizione)

→ Vincoli:

- IDSpedizione, CodOrdine, CodImpiegato: PRIMARY KEY
- CodOrdine → Ordine.IdOrdine
- CodImpiegato → Impiegato.IdImpiegato
- DataSpedizione: NOT NULL

Contratto (IDContratto, ABI, AddressSmartContract, AddressOwner);

Ipotesi sui vincoli:

1. L'e-mail del cliente è stata resa UNIQUE perché non possono esistere due clienti registrati con la stessa e-mail. Tuttavia, non si è voluto usare la mail come chiave primaria per una più facile gestione dei collegamenti con gli acquisti tramite i codici utente (chiave primaria).
2. Il nome utente dell'impiegato è stato reso UNIQUE perché non possono esistere due impiegati con lo stesso nome utente.
3. Si sono voluti introdurre le tabelle genere, formato e lingua per evitare che durante l'introduzione di tali parametri si possano commettere errori di battitura.
4. Ogni libro ha una chiave primaria (codice). Tuttavia, si sono voluti introdurre i parametri Titolo, Autore, Editore, AnnoPubblicazione, CodiceFormato, CodicelLingua come UNIQUE per far sì che lo stesso libro non possa essere inserito più volte ma si possa solo aggiornare la quantità. I libri possono differire per uno o più dei parametri precedenti. Se tutti i parametri precedenti sono identici a un libro già esistente verrà generato un errore.
5. La tabella ricarica è utilizzata per mantenere traccia delle operazioni di inserimento (da parte degli impiegati) di un nuovo libro all'interno del database di sistema e delle operazioni di aggiornamento del numero di copie di un prodotto.
6. La tabella spedizione è utilizzata per mantenere traccia delle operazioni di spedizione (da parte degli impiegati) di uno o più prodotti.