

Noleggio video

Scopo dei documento

Questo documento ha lo scopo di analizzare i requisiti dell'esercizio di creazione di un'applicazione di gestione di un noleggio video.

In questo documento verranno analizzati i requisiti indicati nelle slide descrittive dell'esercizio, verrà documentato il diagramma E-R e le scelte che hanno portato alla ristrutturazione verso lo schema logico.

Il documento contiene inoltre lo schema fisico del DB.

Requisiti

Nel requisito vengono identificate tutte le entità (in rosso) e le relazioni (in verde), insieme agli attributi (in blu)

Una società di noleggio video vuole automatizzare i **punti vendita**. La società gestisce 8 **punti vendita** in varie **località**; in ogni **punto vendita** lavorano da 1 a 6 **addetti** più il **responsabile del negozio**. La società ha 6 **fornitori** che dopo 90 giorni di affitto **ritirano** la **merce**.

Ogni settimana **avviene** uno scarico dei nuovi **video** e la **restituzione** dei **video scaduti**.

Ogni **video** è contraddistinto da **più attori**, **un titolo**, **un regista**, **la casa produttrice**, **la durata in minuti** ed **un genere**.

I **punti vendita** **noleggiano** i **video** a giorni interi e le **tariffe** sono **decescenti** **all'aumentare** **del periodo** **di noleggio**. Il **responsabile** del **punto vendita** **può applicare** degli **sconti** ad alcuni **clienti fidelizzati**, ma la politica degli sconti deve essere codificata.

I **clienti** dovranno essere rintracciabili e **deve essere tenuta traccia** del **cliente** e dei **video** che ha in noleggio. Per ogni **cliente** si dovrà conoscere **nome**, **cognome**, **indirizzo**, **telefono** **abitazione**, **telefono cellulare**, **indirizzo email** e **data di nascita**.

Ogni **cliente** dovrà **firmare** una **liberatoria per la legge sulla privacy** e si dovrà tenere traccia delle **coordinate del documento cartaceo corrispondente**. Per ogni **cliente** si dovrà mantenere uno **storico dei noleggi effettuati**.

Gli **addetti** saranno caratterizzati da un **nome**, **cognome**, **matricola**. Ogni addetto riceverà incentivi se si supererà certi volumi di vendita

Operazioni:

- OP1: Carico e scarico dei video a magazzino con frequenza 1 operazione a settimana per punto vendita.
- OP2: Stampa della ricevuta di attivazione noleggio al cliente e contestuale diminuzione della disponibilità (7g000/gg)
- OP3: Stampa della ricevuta di termine noleggio al cliente e contestuale aumento della disponibilità (7.000/gg)
- OP4: Ricerca sul catalogo per genere (10.000/gg)
- OP5: Ricerca sul catalogo per titolo (10.000/gg)
- OP6: Calcolo incasso giornaliero per punto vendita e per addetto.
- OP7: Prenotazione di titoli in uscita da parte dei clienti (500/gg)
- OP8: Stampa ricevuta di addebito al cliente dei **video restituiti in cattivo** stato e quindi da rimborsare al fornitore

Progettazione concettuale

Analisi dei requisiti

Dal testo dei requisiti si evincono le seguenti entità:

Entità	Attributi	In relazione con
Punto vendita	località	Addetto (impiego) Punto vendita (impiego) Noleggio (noleggiano)
Fornitore	nome	Merce (proprietà) Merce (affitto) Merce (ritiro)
Merce		Video (rappresenta) Fornitore (affitto) Fornitore (ritiro)
Addetto	nome, cognome, matricola	Punto vendita (impiego)
Responsabile	nome, cognome, matricola	Punto vendita (impiego) Cliente (applica sconti)
Video	attori (1,n), un titolo, un regista, la casa produttrice, durata in minuti, genere	Merce (rappresenta) Cliente (prenotazione)
Noleggio	Giorni di noleggio	Punto vendita (noleggiano) Tariffa (condizione di noleggio) Cliente (video in noleggio)
Tariffa	Le tariffe sono decrescenti all'aumentare del periodo di noleggio	Noleggio (condizione di noleggio)
Clienti	nome, cognome, indirizzo, telefono abitazione, telefono cellulare, indirizzo email e data di nascita.	Noleggio (storico) Responsabile (applica sconti) Noleggio (video in noleggio) Liberatoria (firma) Video (prenotazione)
Liberatoria	Coordinate del documento cartaceo	Cliente (firma)

Impegnato e Responsabile

E' stato introdotto un attributo composito che descrive le credenziali di un impiegato, per permettere il login al sistema.

Mentre è implicitamente richiesto un sistema di autenticazione, non si accenna alla necessità di un controllo di autorizzazione, pertanto le autorizzazioni verranno gestite direttamente nell'applicativo a seconda del ruolo dell'impiegato (dirigente, addetto)

Il direttore viene considerato come una specializzazione di un addetto, in quanto entrambi sono logicamente impiegati di un'azienda, hanno ruoli diversi ma attributi comuni. Il direttore può infatti fare operazioni aggiuntive come applicare degli sconti.

Entità Video

Dal fatto che un utente può prenotare un titolo in uscita, si deduce che l'entità *Video* può essere sia *Video in uscita* che *Video disponibile*

Entità Merce (supporto fisico)

Per tracciare le istanze dei supporti provenienti dai fornitori, verrà creata una nuova entità *Supporto Fisico* che rappresenta un supporto fornito da un fornitore ad uno specifico negozio e che rappresenta un video.

Si assume che il supporto fisico abbia un codice seriale (per permettere al fornitore di identificare il supporto).

Un Supporto fisico potrebbe essere affidato da un fornitore ad un negozio più volte (es: il negozio di Roma ha il supporto dal 1.01.2020 al 31.03.2020, il negozio di Milano ha il supporto dal 1.04.2020 al 30.06.2020)

Ulteriori considerazioni su questa entità verranno fatte successivamente, in quanto viola la forma normale BCNF

Dal requisito di pagamento del costo di danneggiamento del supporto Il supporto fisico (OP8) si deduce che un supporto fisico è noleggiabile solo se non danneggiato da un precedente noleggio, e che il supporto fisico ha anche un costo, che verrà pagato dal cliente in caso di danneggiamento

Batch

Il requisito esprime una relazione tra un fornitore e la merce attraverso un'operazione di carico/scarico della merce.

Per rappresentare le operazioni di carico e scarico, si prevede la necessità di registrare le operazioni in batch per identificare tutti i supporti caricati/scaricati e l'operatore che ha avviato l'operazione. Pertanto, la relazione tra Fornitore e

Catalogo

Un punto vendita ha un catalogo, che traccia la disponibilità al punto vendita di un film (è quindi una relazione tra punto vendita e video). L'attributo quantità disponibile sulla relazione catalogo sarà oggetto di studio per capire se convenga conservare l'informazione aggregata, oppure convenga svolgere una ricerca su *Punto vendita* -> *Stoccaggio* -> *Supporto Fisico* per conoscerne la disponibilità

Entità noleggio e tariffa

Si assume che il noleggio venga pagato alla riconsegna del titolo, e che il numero di giorni di noleggio venga scelto all'atto del noleggio.

Può succedere che un cliente ritardi la consegna, per cui il calcolo della tariffa di noleggio è costituita da una parte fissa – che include tutti i giorni del noleggio stabiliti, e da una parte opzionale che include un costo per ogni giorno di ritardo nella consegna.

Il costo iniziale ed il costo per ogni giorno di ritardo variano in funzione del numero di giorni iniziali di noleggio.

Con questo requisito, l'entità *Termine di noleggio* rappresenta la tariffa applicata al

noleggio, che dipende dal numero di giorni di noleggio scelti dal cliente, che viene accettata da un cliente, e che lo impegna in un *Contratto di noleggio* a pagare i corrispettivi indicati nel termine di noleggio.

La necessità di tracciare i costi di noleggio e di distruzione del supporto da parte del cliente (OP8), identifica una ulteriore entità non descritta che rappresenta le ricevute di fine noleggio. Si assume pertanto che al termine di ogni noleggio venga creata una ricevuta contentente i dettagli dei costi, inclusi i costi di distruzione del supporto

Entità Cliente

Ogni cliente deve sottoscrivere un documento di assunzione di responsabilità. Si assume che tale documento possa essere sottoscritto anche in un momento successivo alla creazione del cliente, ma fino a quando il cliente non consegna il documento, non sarà possibile noleggiare alcun supporto fisico.

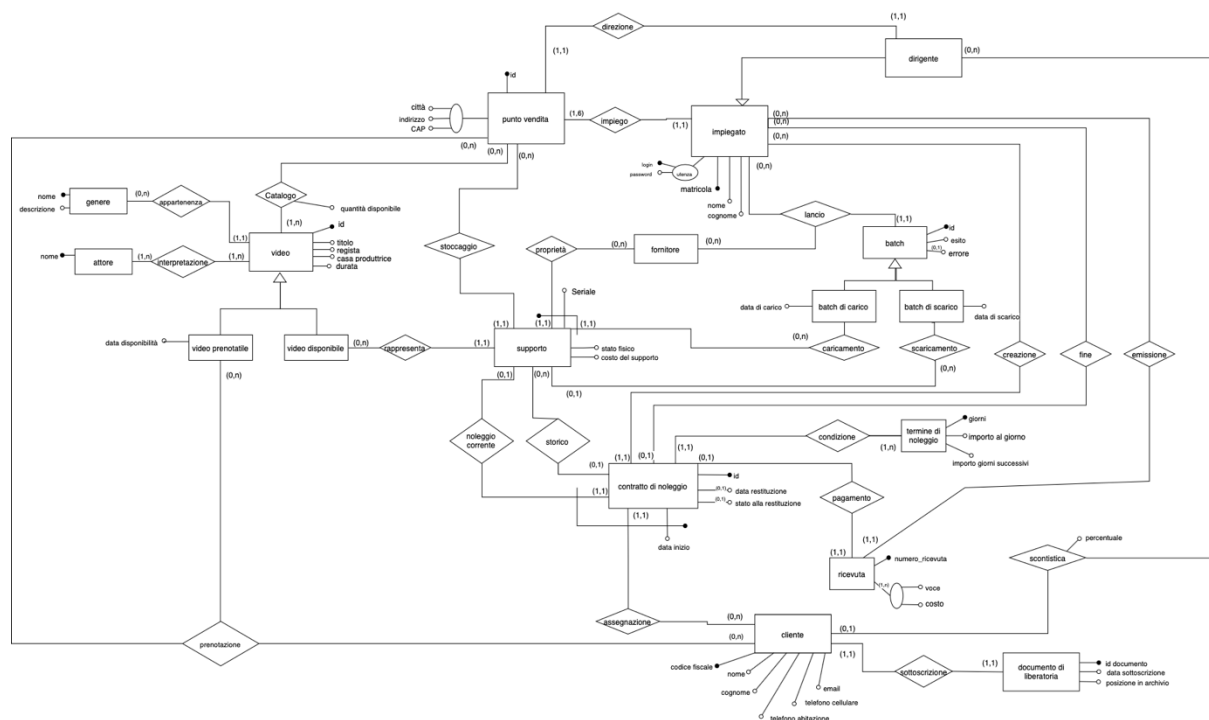
Incentivi

Non è definita alcuna logica per tracciare gli incentivi che deve ricevere un addetto, in particolare non è definito se gli incentivi si ricevono per volume di noleggi effettuati, terminati o effettivamente pagati. Pertanto si procede a modellare una serie di relazioni tra le entità Noleggio e Addetto: creazione, termine e tra l'entità Ricevuta e addetto: emissione

Scontistica

La scontistica non è ancora determinata nei requisiti. Si prevede quindi di modellare un modello semplice, che però verrà eliminato nella progettazione logica in quanto nessuna operazione è attualmente richiesta sullo sconto.

Diagramma E-R



Progettazione logica

Tabella dei volumi

ID	Concetto	Tipo	Volume	Criterio	Descrizione
E1	Punto Vendita	E	8		Da requisiti
E2	Impiegato	E	56	$= E1 * 7$	Da requisiti
E3	Dirigente	E	8	$= E1$	Da requisiti
R1	Direzione	R	8	$= E3$	Da requisiti
R2	Impiego	R	56	$= E2$	Da requisiti
E4	Fornitore	E	6		Da requisiti
E5	Video	E	1000		Assunzione
R4	Catalogo	R	8000	$E5 * E1$	
E6	Genere	E	20		Assunzione
R5	Appartenenza	R	1000	E5	
E7	Attore	E	100		Assunzione
R6	Interpretazione	R	5000	$5 * E5$	Assunzione: 5 attori / film
E8	Video prenotabile	E	20		Assunzione: 2% dei film sono in uscita
E9	Video disponibile	E	980	$= E5 - E8$	
E10	Supporto fisico	E	28365		Assunzione: 544 ingressi/uscite a settimana, storico di 1 anno
R7	Contenuto	R	28365	$= E10$	
R8	Proprietà	R	28365	$= E10$	
R9	Stoccaggio	R	28365	$= E10$	
E11	Cliente	E	16000	$= 2000 * E1$	Assunzione: 2000 / p.v.
E12	Contratto di noleggio	E	2555000	$= 365 * 7000$	Assunzione: storico di 1 anno, 7000 noleggi al giorno
R10	Assegnazione contratto	R	2555000	$= E12$	
R11	Noleggio corrente	R	28000	$= 4 * 7000$	Assunzione: durata media noleggio: 4gg
R12	Storico noleggio	R	2555000	$= E12$	
R13	Creazione noleggio	R	2555000	$= E12$	
E13	Termine di noleggio	E	5		Assunzione: 5 termini di noleggio (a seconda della durata prevista)
R14	Condizione di noleggio	R	2555000	$= E12$	
E14	Sconto	E	1600	$= E11 * 0.1$	Assunzione: 10% clienti

E15	Documento liberatoria	E	16000	= E11	
R15	Sottoscrizione	R	16000	= E15	
R16	Prenotazione	R	182500	= 500 * 365	Assunzione: storico di 1 anno, 500/g
E16	Batch	E	5184	= E17 + E18	1 carico / scarico a settimana per punto vendita e fornitore
E17	Batch di carico	E	2592	= E1 * E4 * 54	
E18	Batch di scarico	E	2592	= E1 * E4 * 54	
R18	Caricamento	R	2592	= E17	
R19	Scaricamento	R	2592	= E18	
R20	Lancio	R	5184	= E16	
E17	Ricevuta	E	2555000	<= E12	
R21	Fine	R	2555000	<= E12	
R22	Emissione	R	2555000	= E17	

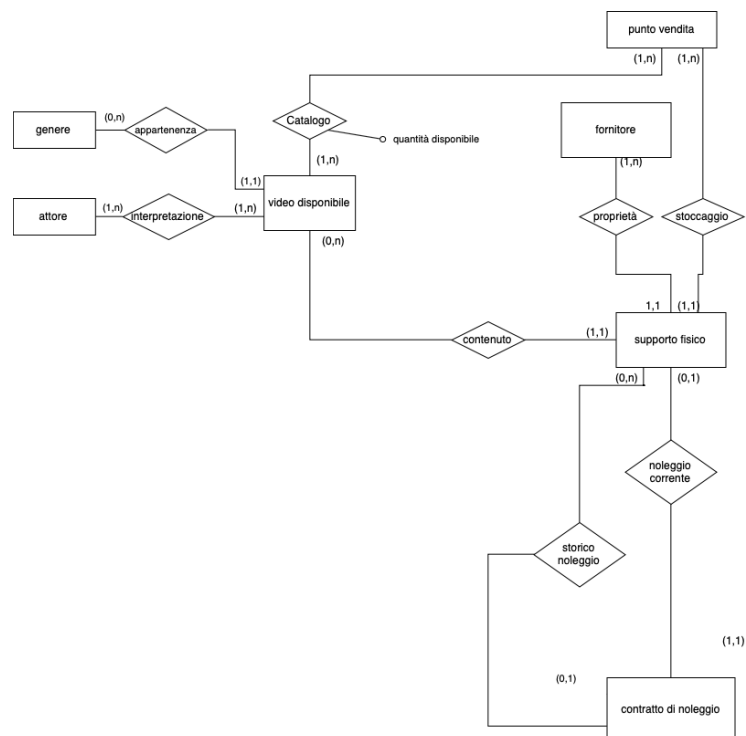
Tabella delle operazioni

ID	Descrizione	Volume
OP1	Carico magazzino e scarico magazzino	1 / 7gg
OP2	Attivazione noleggio	7000 / g
OP3	Termine noleggio	7000 / g
OP4	Ricerca per genere	10000 / g
OP5	Ricerca per titolo	10000 / g
OP6	Calcolo incasso giornaliero per punto vendita e per addetto	8 / g
OP7	Prenotazione titoli in uscita	500 / g
OP8	Stampa ricevuta addebito titoli in cattivo stato	7 / g

Analisi ridondanza Quantità disponibile

In questa sezione si analizza la ridondanza Quantità disponibile della relazione Catalogo. Tale informazione è ridondante in quanto desumibile dal conteggio delle istanze dell'entità *Supporto* che non sono noleggiate (non presenti nella relazione Noleggio corrente) e non sono scaricate (non presenti nella relazione Scaricamento).

Lo schema semplificato del modello E-R mostra tutte le entità e relazioni coinvolte nell'analisi della ridondanza:



Le operazioni che coinvolgono l'informazione quantità disponibile sono: OP2, OP3, OP4, OP5. Viene esclusa OP1 in quanto operazione batch.

OP2: Attivazione noleggio

Accesso	Con ridondanza
Contratto di noleggio	1W
Noleggio corrente	1W
Supporto fisico	1W
Stoccaggio	1R
Catalogo	1R 1W

Accesso	Senza ridondanza
Contratto di noleggio	1W
Noleggio corrente	1W
Supporto fisico	1W

OP3: Termine noleggio

Accesso	Con ridondanza
Contratto di noleggio	1W
Noleggio corrente	1W
Storico noleggio	1W
Supporto fisico	1W
Stoccaggio	1R
Catalogo	1R 1W

Accesso	Senza ridondanza
Contratto di noleggio	1W
Noleggio corrente	1W
Storico noleggio	1W
Supporto fisico	1W
Stoccaggio	-
Catalogo	-

OP4: Ricerca per genere

Accesso	Con ridondanza	Note
Genere	1R	
Appartenenza	49R	1-1 con video disponibile
Video disponibile	49R	980 video / 20 generi
Catalogo	49R	1-1 con video disponibile

Accesso	Senza ridondanza	Note
Genere	1R	
Appartenenza	49R	1-1 con video disponibile
Video disponibile	49R	980 video / 20 generi
Contenuto	1.421R	1-1 con supporto
Supporto	1.421R	(28.365 supporti / 980 video) * 49

OP5: Ricerca per titolo

Non si calcola in questo caso il costo della ricerca per titolo, ma il path di accesso - per ogni titolo trovato – al conteggio dei titoli disponibili. Si assume che ogni ricerca per titolo ritorni un solo elemento.

Accesso	Con ridondanza
Video disponibile	1R
Catalogo	1R

Accesso	Senza ridondanza	Note
Video disponibile	1R	
Contenuto	29R	1-1 con supporto
Supporto fisico	29R	(28.365 supporti / 980 video)

Totali

Operazione	Cardinalità	Con ridondanza	Senza ridondanza
OP2	7000x	14.000R + 28.000W	21.000W
OP3	7000x	14.000R + 35.000W	28.000W

OP4	10.000x	1.480.000R	29.410.000R
OP5	10.000x	20.000R	590.000R
Totali		1.528.000R + 63.000W	30.000.000R + 49000W

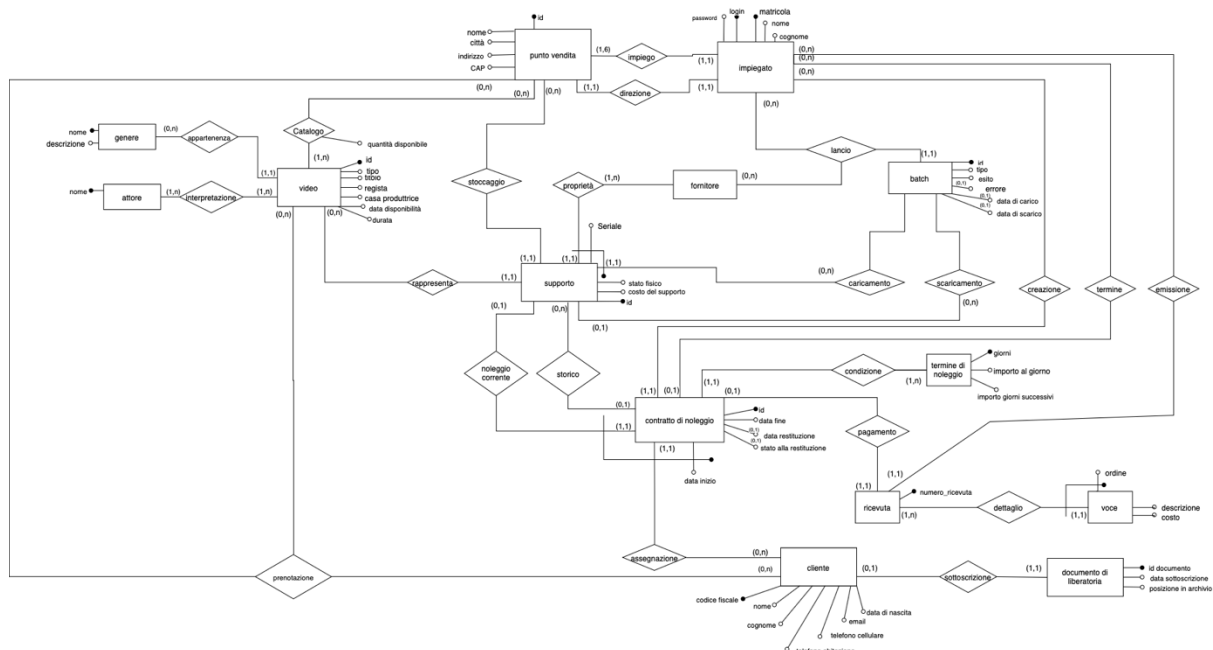
dal calcolo del numero di operazioni si evince chiaramente che la ridondanza deve essere tenuta, in quanto i volumi di carico per l'OP4 ed OP5 sono di un ordine di grandezza superiore senza la ridondanza!

Ristrutturazione dello schema E-R

Per la ristrutturazione dello schema E-R, si provvede prima a semplificare lo schema:

- La gerarchia Impiegato viene collassata in un'unica entità, con l'aggiunta di un attributo tipo per determinare se un impiegato è un dirigente o un addetto. Bisogna tenere in considerazione che può esistere un solo direttore per ogni filiale. Non è specificato se un direttore è anche un addetto, ma si assume un direttore possa anche svolgere le mansioni di un addetto.
- La gerarchia Video viene collassata aggiungendo gli attributi *Data disponibilità* in Video ed inserendo un attributo *tipo*, per determinare se il video è disponibile o meno. L'attributo tipo è in realtà ridondante, in quanto l'informazione che un video è prenotabile è anche espressa dal fatto che esiste una data disponibilità.
- L'attributo composito in *Punto vendita* viene collassato a livello di entità
- L'attributo composito in Impiegato riportante le credenziali di accesso viene collassato a livello di entità
- La relazione *Scontistica* viene eliminata in quanto ancora da definire
- La gerarchia Batch viene collassata a livello di entità inserendo un attributo tipo che identifica se il batch è un carico o uno scarico. I due attributi data di carico e data di scarico provenienti dalle due entità figlio vengono rimappate nell'attributo data: l'attributo tipo permette di capire semanticamente se la data è di carico o di scarico.
- L'attributo composito in Ricevuta riportante le voci della ricevuta viene modellato come una nuova entità (voce) ed una nuova relazione (dettaglio)

Il diagramma ER ristrutturato viene riportato in seguito:



Scelta delle chiavi

Entità	Chiave	Note
Punto vendita	ID	
Impiegato	Matricola	
Fornitore	ID	
Video	ID	
Batch	ID	
Supporto	ID	Chiave naturale è (fornitore, batch di carico, seriale). Si sceglie di creare una chiave univoca come UUID, che può essere utilizzato con un barcode scanner. Si sceglie di utilizzare ID come chiave primaria per evitare di propagare a tutte le relazioni a valle (contratto noleggio, ricevuta) le informazioni su fornitore, batch di carico e seriale.
Genere	Nome	
Attore	ID	l'unico altro attributo della tabella – nome – non è univoco
Contratto di noleggio	ID	La chiave naturale è (supporto, data inizio, cliente), tuttavia si sceglie di utilizzare un UUID come chiave primaria per avere un codice univoco che possa essere comunicato al cliente e che nasconda le informazioni sulla chiave naturale
Termine di noleggio	Giorni	
Cliente	Codice fiscale	
Documento di liberatoria	ID documento	

Relazioni algebriche

Per la traduzione dello schema ER verso lo schema relazionale, si procede a ridurre le relazioni come segue:

- Direzione: La relazione direzione viene rimossa in quanto l'entità dirigente è già in relazione con il punto vendita tramite la relazione Impiego. Verrà aggiunto un vincolo che non può esistere più di un dirigente per ogni negozio.
- Impiego: viene espressa tramite una chiave esterna in impiegato
- Creazione: viene espressa tramite una chiave esterna in contratto_noleggio
- Condizione: viene espressa tramite una chiave esterna in contratto_noleggio
- Assegnazione: viene espressa tramite una chiave esterna in contratto_noleggio
- Storico: viene espressa tramite una chiave esterna in contratto_noleggio
- Sottoscrizione: viene espressa tramite una chiave esterna in cliente
- Stoccaggio: viene espressa tramite una chiave esterna in supporto
- Proprietà: viene espressa tramite una chiave esterna in supporto
- Rappresenta: viene espressa tramite una chiave esterna in supporto
- Catalogo: molti-a-molti – viene modellata come una nuova relazione
- Appartenenza: viene espressa tramite una chiave esterna in Video
- Attore: molti-a-molti - richiede una relazione
- Prenotazione: molti-a-molti - richiede una relazione
- Caricamento: viene espressa tramite una chiave esterna in Supporto
- Scaricamento: viene espressa tramite una chiave esterna in Supporto

```
punto_vendita (id, nome, citta, indirizzo, cap)
impiegato (matricola, punto_vendita, login, password, tipo, nome, cognome)
fornitore (id, nome)
batch(id, tipo, esito, errore, impiegato, fornitore, data)
genere (nome, descrizione)
attore (nome)
video (id, genere, tipo, titolo, regista, casa_produttrice, durata,
data_disponibilita)
supporto (id, video, fornitore, punto_vendita, seriale, batch_carico,
batch_scarico, stato_fisico, costo_supporto, noleggio_corrente)
contratto_noleggio (id, supporto, cliente, impiegato, data_inizio,
data_restituzione, stato_restituzione, termine_noleggio)
ricevuta (numero_ricevuta, impiegato, contratto_noleggio)
voce_ricevuta(numero_ricevuta, ordine, descrizione, costo)
cliente (cod_fiscale, nome,cognome,telefono_abitazione, telefono_cellulare, email,
documento_liberatoria)
documento_liberatoria (id, data_sottoscrizione)
termine_noleggio (giorni, importo_iniziale, importo_gg_successivi)
interpretazione(attore,video, ruolo)
catalogo (punto_vendita, video, quantita_disponibile)
prenotazione (cliente, video, punto_vendita)
```

Considerazioni sulle forme normali

La relazione supporto è in 3NF in quanto seriale -> (video, fornitore) ma non da batch_carico.

Schema fisico

Considerazioni sulla ricerca dei video per nome

Per rendere il sistema utilizzabile, si è scelto di scaricare il database di tutti i titoli disponibili da IMDB (<https://www.imdb.com/interfaces/>), un servizio pubblico di indicizzazione di materiale video contenente oltre 3 milioni di video. Da IMDB sono stati estratti

- Tutti i titoli
- Tutti gli alias (per conoscere i titoli in italiano)
- Tutti i nomi
- Tutti i registi legati ai film

Il data set è stato filtrato per ottenere tutti i film successivi al 1970, che abbiano un genere. Il risultato ha permesso il caricamento nella tabella video di oltre 1.000.000 di entries. Una select distinct sui generi dei video ha permesso di estrarre tutti i generi (23) in cui sono suddivisi i titoli di IMDB.

Per la ricerca sul titolo, si sarebbe potuto utilizzare uno statement di like '%parola%'. Questo avrebbe permesso di ricercare una parola (o meglio, una sequenza di caratteri) all'interno della colonna, tuttavia ci sono due limitazioni da tenere in considerazione: la maggior parte dei DB relazionali – incluso MariaDB e MySQL - non sono in grado di utilizzare indici se il like utilizza un '%' sul lato sinistro.

La tabella dei video è estremamente onerosa in termini di blocchi, pertanto va evitato un full table scan per la ricerca.

Un modo pratico per evitare un full table scan utilizzando un indice sarebbe di eseguire una ricerca di una parte del titolo a partire dall'inizio (like 'parola%'). L'indice ridurrebbe i tempi di ricerca da $O(n)$ a $O(\log n)$.

Nella vita pratica, la ricerca di un titolo testuale richiede di cercare una parola anche in mezzo al titolo (Es: nessuno ricercerebbe il film 'The Matrix' con 'The Ma', ma con 'Matrix'), e quindi il '%' sul lato sinistro della ricerca sarebbe comunque necessario.

Dato il volume dei dati e le necessità di ricerca, si è optato per la creazione di un indice di full text search sul campo *titolo* in modo da poter effettuare una ricerca in linguaggio naturale.

Questo indice è molto costoso in termini di spazi e di update, ma la lista dei video disponibili non verrà aggiornata di frequente, ed inoltre i costi in termini di spazio verranno più che colmati dalle minori richieste in termini di CPU, e la maggiore rapidità degli addetti ad ottenere risposte in tempi rapidi e quindi a servire più clienti.

Un ulteriore vantaggio degli indici fulltext è la capacità di tali indici di ritornare risultati simili, tollerando errori di scrittura quali accenti nelle posizioni sbagliate o ümlaut mancanti per i titoli che hanno – ad esempio - parole in tedesco.

Considerazioni sulla relazione Supporto

La relazione supporto è in 3NF in quanto seriale -> (supporto, fornitore) ma non da batch_carico.

La tabella supporto ha un vincolo difficilmente esprimibile nel diagramma E-R: la tupla (supporto, fornitore) può essere presente più volte se il supporto viene caricato, scaricato, poi ricaricato.

Es:

Id	supporto	fornitore	Batch_carico	Batch_scarico
1	SUP-01	CAT	12345	67890
2	SUP-01	CAT	23456	NULL

Da questo punto di vista, la tupla (supporto, fornitore, batch_scarico) è una chiave, tuttavia MySQL non considera i valori NULL all'interno delle chiavi, rendendo possibile una situazione simile:

Id	supporto	fornitore	Batch_carico	Batch_scarico
1	SUP-01	CAT	12345	67890
2	SUP-01	CAT	23456	NULL
3	SUP-01	CAT	34567	NULL

Tale situazione violerebbe il vincolo che vogliamo imporre: uno stesso supporto di un fornitore non può essere caricato due volte simultaneamente. Per ovviare a questo problema, si decide di creare un campo generato "disponibile", con valore:

- 'true' se il campo batch_scarico è null
- null se il batch_scarico è settato.

Tale campo viene usato per costruire la chiave (supporto, fornitore, disponibile)
In questo modo, si sfrutta la *feature* di MySQL di ignorare le chiavi con valori null, per rimuovere dall'indice della chiave tutti i supporti che sono stati scaricati, mentre i supporti attualmente caricati verranno inseriti nell'indice della chiave.

Il campo viene generato la seguente configurazione:

```
disponibile boolean generated always as (case when batch_scarico is null then true ELSE null end) persistent
```

Con una chiave simile, l'inserimento del record 2 avverrà senza problemi, mentre l'inserimento del record 3 non sarà ammesso in quanto viola la chiave.

Id	supporto	fornitore	Batch_carico	Batch_scarico	Disponibile
1	SUP-01	CAT	12345	67890	NULL
2	SUP-01	CAT	23456	NULL	1
3	SUP-01	CAT	34567	NULL	1

Schema SQL

```
create database noleggio;  
use noleggio;
```

```
create or replace table punto_vendita (  
  id int not null,  
  nome varchar(255) not null,  
  citta varchar(255) not null,
```

```

indirizzo varchar(255) not null,
cap varchar(5) not null,
CONSTRAINT pk_punto_vendita primary key (id)
);

create or replace table impiegato (
    matricola varchar(36) not null,
    punto_vendita int not null,
    login varchar(64) not null,
    password varchar(64) not null,
    tipo enum('ADDETTO', 'DIRIGENTE') not null,
    nome varchar(64) not null,
    cognome varchar(64) not null,
    constraint pk_impiegato primary key (matricola),
    constraint fk_impiegato_punto_vendita FOREIGN key (punto_vendita) REFERENCES
punto_vendita(id)
);

create or replace table fornitore (
    id varchar(36) not null,
    nome varchar(64) not null,
    CONSTRAINT pk_fornitore PRIMARY key (id)
);

create or replace table batch (
    id varchar(36) not null,
    tipo enum('CARICO', 'SCARICO') not null check (tipo in ('CARICO', 'SCARICO')),
    esito enum('SUCCESSO', 'FALLITO', 'IN-ESECUZIONE') not null default 'IN-ESECUZIONE'
check (tipo in ('SUCCESSO', 'FALLITO', 'IN-ESECUZIONE')),
    errore varchar(256),
    punto_vendita int not null,
    impiegato varchar(64) not null,
    data date not null,
    CONSTRAINT pk_batch primary key (id),
    CONSTRAINT fk_batch_punto_vendita foreign key (punto_vendita) REFERENCES
punto_vendita (id),
    CONSTRAINT fk_batch_impiegato foreign key (impiegato) REFERENCES impiegato
(matricola)
);

create or replace table genere (
    nome varchar(36) not null,
    descrizione varchar(255) not null,
    CONSTRAINT pk_genere PRIMARY key (nome)
);

create or replace table attore (
    id varchar(36) not null,
    nome varchar(255) not null,
    CONSTRAINT pk_attore PRIMARY key (id)
);

create or replace table video (
    id varchar(36) not null,
    genere varchar(36) not null,
    tipo enum('DISPONIBILE', 'NON-DISPONIBILE') not null,
    titolo varchar(255) not null,
    regista varchar(255) not null,
    casa_produttrice varchar(255) not null,
    data_disponibilita date,
    CONSTRAINT pk_video PRIMARY key (id),
    CONSTRAINT fk_video_genere FOREIGN key (genere) REFERENCES genere(nome),
    FULLTEXT (titolo)
);

```

```

create or replace table supporto (
    id varchar(36) not null,
    seriale varchar(36) not null,
    video varchar(36) not null,
    fornitore varchar(36) not null,
    punto_vendita int not null,
    batch_carico varchar(36) not null,
    batch_scarico varchar(36),
    stato_fisico enum ('BUONO', 'DANNEGGIATO') not null default 'BUONO' check
    (stato_fisico in ('BUONO', 'DANNEGGIATO')),
    costo_supporto NUMERIC(10,2),
    noleggio_corrente varchar(36),
    disponibile boolean generated always as (case when batch_scarico is null
then true ELSE null end) persistent,
    CONSTRAINT pk_supporto PRIMARY key (id),
    CONSTRAINT key_supporto_gia_presente UNIQUE key (fornitore, seriale,
disponibile),
    CONSTRAINT fk_supporto_video FOREIGN key (video) REFERENCES video(id),
    CONSTRAINT fk_supporto_fornitore FOREIGN key (fornitore) REFERENCES
fornitore(id),
    CONSTRAINT fk_supporto_punto_vendita FOREIGN key (punto_vendita) REFERENCES
punto_vendita(id),
    CONSTRAINT fk_supporto_batch_carico FOREIGN key (batch_carico) REFERENCES
batch(id),
    CONSTRAINT fk_supporto_batch_scarico FOREIGN key (batch_scarico) REFERENCES
batch(id),
    CONSTRAINT ck_supporto_batch_carico CHECK ((select tipo from batch where
id='batch_carico') = 'CARICO'),
    CONSTRAINT ck_supporto_batch_scarico CHECK ((select tipo from batch where
id='batch_carico') = 'SCARICO')
);

```

```

create or replace table documento_liberatoria (
    id varchar(36) not null,
    data_sottoscrizione date not null,
    posizione_archivio varchar(255),
    CONSTRAINT pk_documento_liberatoria PRIMARY key (id)
);

```

```

create or replace table cliente (
    cod_fiscale varchar(16) not null,
    nome varchar(255) not null,
    cognome varchar(255) not null,
    indirizzo varchar(255) not null,
    data_nascita date not null,
    citta varchar(255) not null,
    cap varchar(5) not null,
    telefono_abitazione varchar(10) not null,
    telefono_cellulare varchar(10) not null,
    email varchar(255) not null,
    documento_liberatoria varchar(64),
    CONSTRAINT pk_cliente PRIMARY key (cod_fiscale),
    CONSTRAINT fk_cliente_documento_liberatoria FOREIGN key
(documento_liberatoria) REFERENCES documento_liberatoria(id)
);

```

```

create or replace table termine_noleggio (
    giorni int not null,
    importo_iniziale NUMERIC(10,2) not null,
    importo_gg_successivi numeric(10,2) not null,
    CONSTRAINT pk_termine_noleggio PRIMARY key (giorni)
);

```

```

create or replace table contratto_noleggio (

```

```

        id varchar(36) not null,
        supporto varchar(36) not null,
        cliente varchar(16) not null,
        impiegato_creazione varchar(36) not null,
        impiegato_restituzione varchar(36) check (data_restituzione is not null),
        data_inizio date not null,
        data_restituzione date,
        stato_restituzione enum ('BUONO','DANNEGGIATO') check (stato_restituzione in
('BUONO','DANNEGGIATO')),
        termine_noleggio int not null,
        CONSTRAINT pk_contratto_noleggio PRIMARY key (id),
        CONSTRAINT fk_contratto_noleggio_supporto FOREIGN key (supporto) REFERENCES
supporto(id),
        CONSTRAINT fk_contratto_noleggio_cliente FOREIGN key (cliente) REFERENCES
cliente(cod_fiscale),
        CONSTRAINT fk_contratto_noleggio_impiegato_creazione FOREIGN key
(impiegato_creazione) REFERENCES impiegato(matricola),
        CONSTRAINT fk_contratto_noleggio_impiegato_restituzione FOREIGN key
(impiegato_restituzione) REFERENCES impiegato(matricola),
        CONSTRAINT fk_contratto_termine_noleggio FOREIGN key (termine_noleggio)
REFERENCES termine_noleggio(giorni)
);

create or replace table ricevuta (
    numero_ricevuta varchar(36) not null,
    impiegato varchar(36) not null,
    data date not null,
    contratto_noleggio varchar(36) not null,
    CONSTRAINT pk_ricevuta PRIMARY key (numero_ricevuta),
    CONSTRAINT uk_contratto_noleggio UNIQUE key (contratto_noleggio),
    CONSTRAINT fk_ricevuta_impiegato FOREIGN key (impiegato) REFERENCES
impiegato(matricola),
    CONSTRAINT fk_ricevuta_contratto_noleggio FOREIGN key (contratto_noleggio)
REFERENCES contratto_noleggio(id)
);

create or replace table voce_ricevuta(
    ricevuta varchar(36),
    ordine int,
    descrizione varchar(255),
    costo numeric(10,2),
    CONSTRAINT pk_voce_ricevuta PRIMARY key (ricevuta, ordine),
    CONSTRAINT fk_voce_ricevuta_ricevuta FOREIGN key (ricevuta) REFERENCES
ricevuta(numero_ricevuta)
);

create or replace table interpretazione(
    attore varchar(36) not null,
    video varchar(36) not null,
    ruolo varchar(255),
    constraint pk_interpretazione primary key (attore,video),
    CONSTRAINT fk_interpretazione_attore foreign key (attore) references
attore(id),
    CONSTRAINT fk_interpretazione_video foreign key (video) references video(id)
);

create or replace table catalogo(
    punto_vendita int not null,
    video varchar(36) not null,
    quantita_disponibile int not null default 0,
    CONSTRAINT pk_catalogo primary key (punto_vendita, video),
    CONSTRAINT fk_catalogo_video foreign key (video) REFERENCES video(id),
    CONSTRAINT fk_catalogo_punto_vendita foreign key (punto_vendita) references
punto_vendita(id)
);

```



```

create or replace table prenotazione(
    cliente varchar(16) not null,
    video varchar(36) not null,
    punto_vendita int not null,
    CONSTRAINT pk_prenotazione primary key (cliente, video, punto_vendita),
    CONSTRAINT fk_prenotazione_video foreign key (video) REFERENCES video(id),
    CONSTRAINT fk_prenotazione_punto_vendita foreign key (punto_vendita)
REFERENCES punto_vendita(id),
    CONSTRAINT fk_prenotazione_cliente foreign key (cliente) references
cliente(cod_fiscale)
);

```

```

create or replace view v_contratto_noleggio_attivo as
    select contratto_noleggio.id, supporto, punto_vendita,
    cliente, impiegato_creazione, data_inizio, termine_noleggio, video.id as video,
    titolo
    from cliente
    join contratto_noleggio on cliente.cod_fiscale = contratto_noleggio.cliente
    join supporto on supporto.id = contratto_noleggio.supporto
    join video on video.id = supporto.video
    where contratto_noleggio.data_restituzione is NULL
;

```

```

create or replace view v_contratto_noleggio_terminato as
    select contratto_noleggio.id, supporto, punto_vendita,
    cliente, impiegato_creazione, impiegato_restituzione, data_inizio,
    data_restituzione, termine_noleggio, video.id as video, titolo,
    ricevuta.numero_ricevuta as ricevuta
    from cliente
    join contratto_noleggio on cliente.cod_fiscale = contratto_noleggio.cliente
    join supporto on supporto.id = contratto_noleggio.supporto
    join video on video.id = supporto.video
    left join ricevuta on ricevuta.contratto_noleggio = contratto_noleggio.id
    where contratto_noleggio.data_restituzione is not NULL
;

```

```

create or replace view v_ricevuta as
    select numero_ricevuta, data,
    supporto,
    titolo,
    cliente, cliente.nome, cliente.cognome, indirizzo, citta, cap,
    impiegato.nome as impiegato_nome, impiegato.cognome as impiegato_cognome,
    matricola, impiegato.punto_vendita as punto_vendita,
    data_inizio, data_restituzione,
    sum(costo) as totale
    from ricevuta
    join contratto_noleggio on contratto_noleggio.id =
    ricevuta.contratto_noleggio
    join supporto on supporto.id = contratto_noleggio.supporto
    join video on video.id = supporto.video
    join impiegato on impiegato.matricola = ricevuta.impiegato
    join cliente on cliente.cod_fiscale = contratto_noleggio.cliente
    join voce_ricevuta on voce_ricevuta.ricevuta = ricevuta.numero_ricevuta
    group by numero_ricevuta
;

```

```

create or replace view v_statistica_per_punto_vendita as
    select punto_vendita.id, punto_vendita.nome, punto_vendita.citta,
    punto_vendita.indirizzo, punto_vendita.cap, data, count(numero_ricevuta) as
    numero_ricevute, ifnull(sum(totale),0) as totale_incasso
    from punto_vendita
    left join v_ricevuta
        on v_ricevuta.punto_vendita = punto_vendita.id
    group by punto_vendita.id, data
    order by punto_vendita.id, totale_incasso, data

```

```

;

create or replace view v_statistica_per_impiegato as
select impiegato.punto_vendita, punto_vendita.nome as punto_vendita_nome,
matricola, impiegato.nome, cognome, data, ifnull(sum(costo),0) as totale_incasso
from impiegato
join punto_vendita on punto_vendita.id = impiegato.punto_vendita
left join ricevuta
on ricevuta.impiegato = impiegato.matricola
left join voce_ricevuta
on voce_ricevuta.ricevuta = ricevuta.numero_ricevuta
group by matricola, data
order by punto_vendita, matricola, data
;

create or replace view v_supporto_disponibile as
select *
from supporto
where stato_fisico='BUONO'
and noleggio_corrente is null
and batch_scarico is null
;

```

Progettazione architetturale

L'applicazione è stata creata come un'applicazione a tre livelli, con

- Database relazionale MariaDB
- Server applicativo in PHP che espone servizi con una RESTful API.
- Client applicativo creato come una Progressive Web Application su stack React

Per questioni di fruibilità, l'applicazione contiene delle funzioni in più rispetto a quanto richiesto, in particolare permette:

- Creazione del cliente
- Ricerca del cliente
- Visualizzazione del dettaglio di un cliente
- Assegnazione del documento di liberatoria ad un cliente
- Lista delle operazioni batch
- Visualizzazione del dettaglio delle operazioni batch

Per caricare i video è necessario fornire un file CSV con i dati dei titoli da caricare. Un esempio è fornito nella pagina di carico dei supporti.

Per simulare lo scarico in data futura, sulla pagina di scarico è possibile settare una data operazione.

Al momento sono disponibili nel DB 1.023.000 video e 50.000 supporti. Per ogni genere sono disponibili dei titoli a noleggio ed un titolo prenotabile

Accessi

Per accedere alle applicazioni si possono utilizzare i seguenti URL

APPLICAZIONE WEB	http://romagnuolo-zh.synology.me:18090/webapp
PHPMYADMIN	http://romagnuolo-zh.synology.me:18090/phpmyadmin
MARIA DB	host: 1 port: 13306 user: admin password: pass
CODICE SORGENTE	https://github.com/marcoromag/bd-noleggio

Credenziali di accesso:

Al momento sono presenti gli account per due punti vendita:

- Roma 1 (ID: 1)
- Milano 1 (ID: 2)

P.V.	Nome utente	Password	Tipo
1	admin_rm_1	password	DIRIGENTE
1	mario.rossi	password	ADDETTO
1	luigi.bianchi	password	ADDETTO
1	andrea.pini	password	ADDETTO
1	italo.gelsi	password	ADDETTO
1	marco.romagnuolo	password	ADDETTO
1	svevo.svevi	password	ADDETTO
2	admin_mi_1	password	DIRIGENTE
2	mario.bianchi	password	ADDETTO
2	luigi.verdi	password	ADDETTO
2	andrea.scarpa	password	ADDETTO
2	italo.mora	password	ADDETTO
2	lucio	password	ADDETTO
2	svevo.alpini	password	ADDETTO