

Università degli studi di Padova
Anno Accademico 2023/24



Progetto Basi di Dati

Videoteche Manager

Realizzato da:

Alberto Canavese - N. Matricola 2076423
Cheng Feng Liao - N. Matricola N. Matricola 2082843

1 Abstract

Questo sistema di gestione per le videoteche consente di organizzare e archiviare informazioni importanti. Queste informazioni riguardano i registi dei film, le videoteche stesse, il personale che lavora nelle videoteche, i film su supporto fisico e/o digitale, i clienti, i dettagli dei comuni e altro ancora.

Le tabelle all'interno del database sono interconnesse attraverso relazioni, permettendo di collegare le informazioni in modo significativo: ad esempio, si possono collegare un regista ai film che ha diretto, associare i film alle videoteche in cui sono disponibili, tenere traccia delle iniziative organizzate dalle videoteche.

Questo sistema consente di gestire una varietà di attività all'interno delle videoteche, come il noleggio dei film, l'organizzazione di eventi e la gestione dei dati relativi ai registi e ai clienti, come, per esempio, i dettagli delle carte di credito utilizzate dai clienti.

2 Analisi dei requisiti

Descrizione testuale

Nel database sono presenti i dati dei **registi** che hanno diretto almeno un film che è presente in una delle videoteche contenuta nella base di dati.

Di ogni regista sono noti:

- Codice fiscale (CF)
- Nome
- Cognome
- Data di nascita
- Nazionalità

Un regista può dirigere film e concederli a una o più videoteche presenti nel database.

Di ogni videoteca si conoscono i seguenti attributi:

- Partita IVA
- Nome
- Orario di apertura
- Orario di chiusura
- Indirizzo

Gli indirizzi di ogni videoteca fanno riferimento a dei **comuni** dei quali si conoscono:

- Nome
- Provincia
- Numero di abitanti

Le varie videoteche sono gestite da **gestori** dei quali si conoscono i seguenti dettagli:

- Codice fiscale (CF)
- Nome
- Cognome
- Data di nascita
- Nazionalità
- Stipendio

Da ogni videoteca un **cliente** può noleggiare sia copie fisiche, sotto forma di **DVD**, oppure, tramite un sito web, è può noleggiare i film fruibili direttamente dal web per un periodo limitato di tempo.

Se un DVD è già in prestito ad un'altra persona, non sarà possibile per un cliente noleggiare lo stesso per il medesimo periodo di tempo.

Di ogni **DVD** si conosce:

- ID
- Titolo
- Genere
- Lingua
- Anno di pubblicazione
- Durata in minuti

Invece, per ogni **copia digitale**, sono noti:

- ID
- Titolo
- Genere
- Lingua
- Anno di pubblicazione
- Durata in minuti
- Dimensioni in GB

Di ogni **cliente** si conoscono le seguenti informazioni:

- Codice fiscale (CF)
- Nome
- Cognome
- Data di nascita

Ogni cliente deve avere una **carta di credito** come garanzia, di questa vengono salvati:

- Numero
- Data di scadenza

Le copie fisiche dei DVD vengono rese disponibili dai **fornitori**, dei quali si deve conoscere:

- Nome
- Sede

Infine, ogni videoteca può proporre delle iniziative culturali come i **cineforum**, dei quali servono i seguenti dettagli:

- Data
- Nome film

3 Progettazione concettuale

3.1 Lista entità

se non specificato l'attributo è *non null*

- **Persone:**

- CF: varchar(100) primary key
- Nome: varchar (100)
- Data di nascita: date
- Nazionalità varchar (100)

L'entità "persone" si divide in 3 sottogerarchie:

- Regista:
 - Nome comune: varchar(100) foreign key
- Clienti:
 - Nome comune: varchar(100) foreign key
- Gestore:
 - Stipendio: double precision
 - Nome videoteca: varchar(100) foreign key
 - Nome comune: varchar(100) foreign key

- **Film:**

- ID: varchar(100) primary key
- Titolo: varchar(100)
- Genere: varchar(100)
- Anno: date
- Minuti: integer
- Lingua: varchar(100)

L'entità "film" si divide in 2 sottogerarchie:

- DVD:
 - CF regista: varchar(100) foreign key
 - Nome fornitore: varchar (100) foreign key
 - Nome videoteca: varchar(100) foreign key
- File:
 - GB: integer
 - CF regista: varchar(100) foreign key

- **Fornitori:**

- Nome: varchar(100) primary key
- Sede centrale: varchar(100)

- **Videoteca:**

- P. IVA: varchar(100) primary key
- Nome: varchar(100)
- Orario di apertura: time (6) without time zone
- Orario di chiusura: time (6) without time zone
- Indirizzo: varchar(100)
- Nome comune: varchar(100) foreign key

- **Cineforum:**

- Data: date primary key
- Nome film: varchar(100)

- **Comune:**
 - Nome: varchar(100) primary key
 - Provincia: varchar(100)
 - Numero abitanti: INT
- **Carta di credito:**
 - Numero: varchar(100) primary key
 - Data di scadenza: date

3.2 Tabella delle relazioni

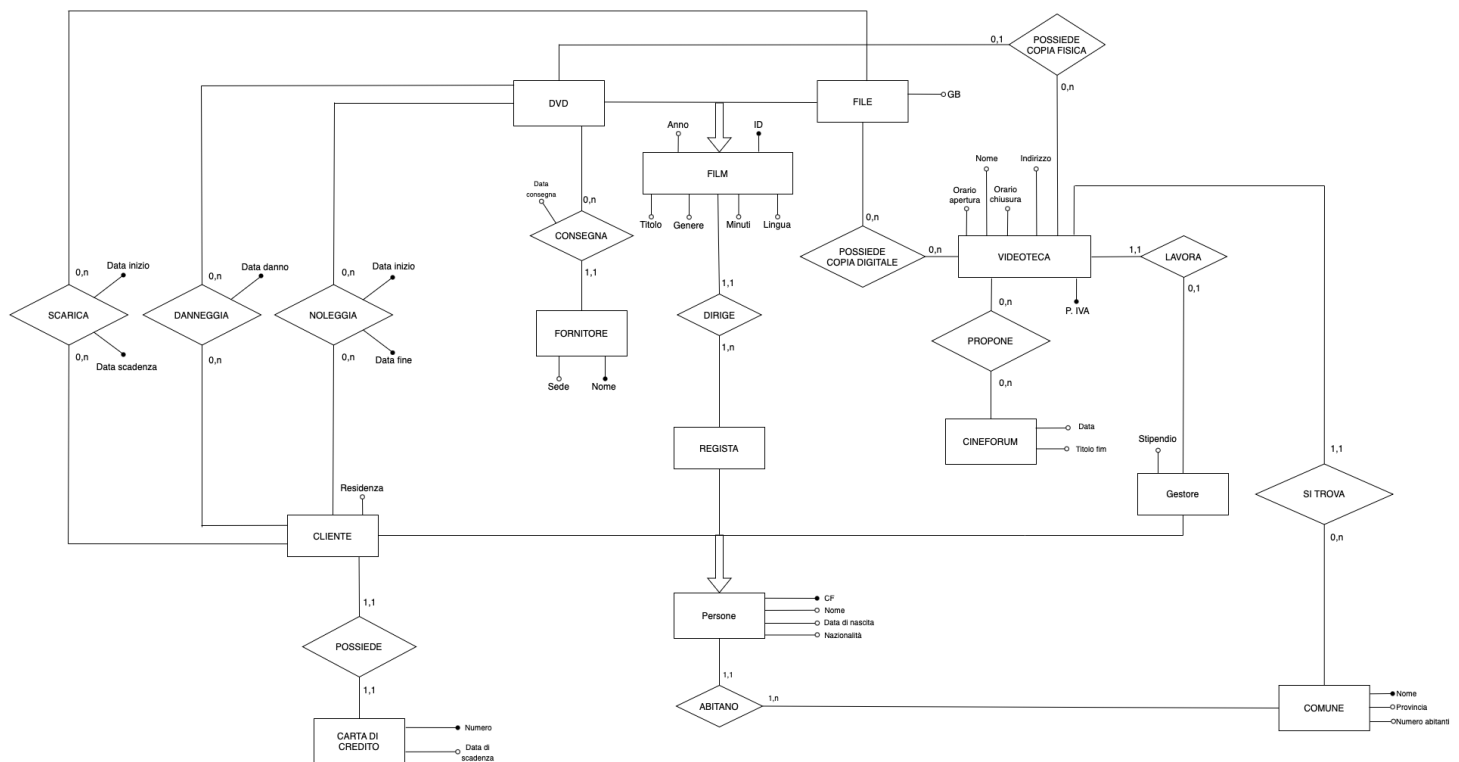
Relazione	Entità coinvolte	Descrizione	Attributi
Scarica	File (0,n) Cliente (0,n)	Un cliente può scaricare più film, ma anche zero. Un file può essere scaricato da zero o più clienti.	Data inizio: date data scadenza: date
Danneggia	DVD (0,n) Cliente (0,n)	Un cliente può danneggiare più DVD, ma anche zero. Un DVD può essere danneggiato da zero o più clienti.	Nessuno
Noleggia	DVD (0,n) Cliente (0,n)	Un cliente può noleggiare più DVD, ma anche zero. Un DVD può essere noleggiato da zero o più clienti.	Data inizio: date Data fine: date
Consegna	Fornitore (1,1) DVD (0,n)	Ogni DVD viene consegnato da un solo fornitore. Un fornitore può consegnare zero o più DVD	Data consegna: date
Possiede	Cliente (1,1) Carta di credito (1,1)	Ogni cliente è associato a una sola carta di credito. Ogni carta di credito è associata ad un solo cliente	Nessuno
Dirige	Regista (1,n) Film (1,1)	Un regista può dirigere uno o più film. Un film può essere diretto da un solo regista.	Nessuno

Propone	Videoteca (0,n) Cineforum (0,n)	Una videoteca può proporre nessun o molti cineforum. Un cineforum può essere proposto da nessuna o molte videoteche.	Nessuno
Possiede copia digitale	Videoteca (0,n) File (0,n)	Una videoteca può possedere zero o molti file. Un file può essere posseduto da zero o molte videoteche	Nessuno
Possiede copia fisica	Videoteca (0,n) DVD (0,1)	Un DVD è posseduto da una sola videoteca. Una videoteca può possedere zero o più DVD.	Nessuno
Si trova	Videoteca (1,1) Comune (0,n)	Una videoteca si trova solo in un comune. In un comune ci possono essere zero videoteche	Nessuno
Lavora	Gestore (0,1) Videoteca (1,1)	Ogni gestore può gestire una o zero videoteche. Ogni videoteca può e deve avere solo un gestore.	Nessuno
Abitano	Persone (1,1) Comune (1,n)	Una persona abita in un solo comune. Ogni comune è abitato da una o più persone	Nessuno

Vincoli non rappresentabili tramite schema ER:

- Ogni videoteca è gestita da una sola persona e ogni gestore gestisce solo una videoteca.
- Lo stesso DVD non può essere noleggiato da più persone durante la stessa finestra di tempo.
- Lo stesso film può essere scaricato da più utenti allo stesso tempo.

Schema ER:



4 Progettazione logica

4.1 Ristrutturazione

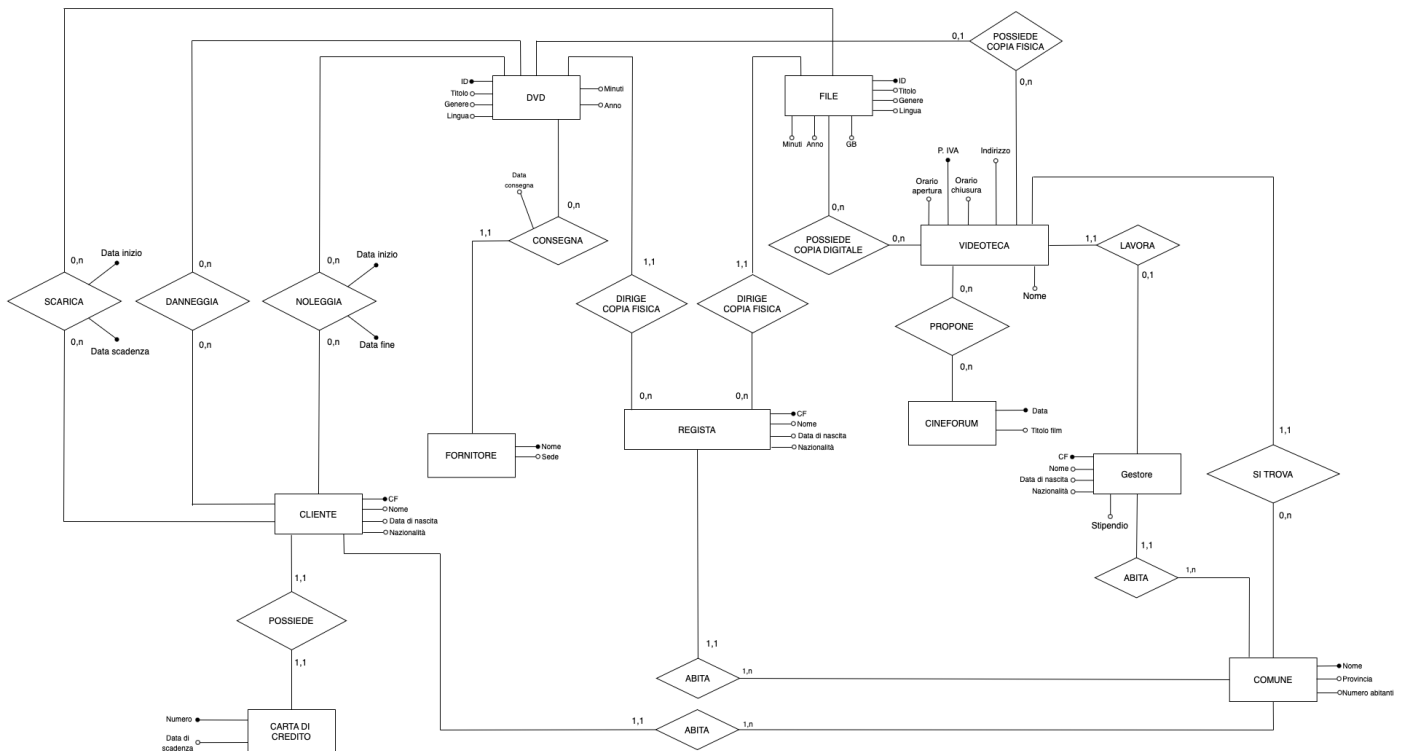
• Eliminazione delle generalizzazioni

Generalizzazione	Risoluzione
Regista, cliente, gestore → Persone	L'entità Persone viene rimossa, i suoi attributi vengono assegnati alle entità figlie. La relazione "ABITA" viene assegnata indipendentemente ad ognuna delle 3 entità figlie.
DVD, file → film	L'entità Libri viene rimossa, i suoi attributi vengono assegnati alle entità DVD e file . La relazione "DIRIGE" viene duplicata

• Eliminazione delle ridondanze

Ridondanza	Risoluzione
<p>Cliente::Residenza</p> <p>Il comune di residenza di un cliente è ridondante in quanto esiste la tabella “Comune”.</p> <p>Si può facilmente risalire alla residenza di un cliente tramite la tabella “Comune”.</p>	<p>Rimozione dell’attributo “Residenza” dalla tabella “Cliente”</p>
<p>Danneggia::data danno</p> <p>La finestra di tempo in cui un DVD può essere stato danneggiato è reperibile grazie alla relazione “Noleggia”.</p> <p>Quindi l’attributo “Data danno” è ridondante</p>	<p>Rimozione dell’attributo “Data danno” dalla relazione “Danneggia”</p>

• Schema ER ristrutturato:



4.2 Creazione delle tabelle

($A \rightarrow B$ indica che B è chiave esterna di A)

Comune (Nome, Provincia, Numero_abitanti)

Regista (CF, nome, Data_di_nascita, Nazionalità, Nome_comune \rightarrow comune.Nome)

Videoteca (P_IVA, Nome, Orario_apertura, Orario_chiusura, Indirizzo, Nome_comune \rightarrow comune.Nome)

Gestore (CF, Nome, data_di_nascita, nazionalità, Nome_comune \rightarrow Comune.nome, P_IVA_videoteca \rightarrow Videoteca.P_IVA)

DVD (ID, Titolo, Genere, Lingua, Minuti, Anno, CF_cliente \rightarrow Cliente.CF, CF_regista \rightarrow Regista.CF, Nome_fornitore \rightarrow Fornitore.nome, Nome_videoteca \rightarrow Videoteca.nome)

File (ID, Titolo, Genere, Lingua, Minuti, Anno, GB, CF_cliente \rightarrow Cliente.CF)

Cliente (CF, Nome, Data_di_nascita, Nazionalità, Nome_comune \rightarrow comune.Nome)

Danneggia (ID_copia \rightarrow DVD.ID, CF_cliente \rightarrow Cliente.CF)

Carta di credito (Numero, Data_di_scadenza, CF_cliente \rightarrow Cliente.CF)

Fornitore (Nome, sede_centrale)

Cineforum (Data, titolo_film)

Propone (P_IVA_videoteca \rightarrow Videoteca.P_IVA, Data \rightarrow cineforum.Data)

Scarica (ID_film \rightarrow File.ID, CF_cliente \rightarrow Cliente.CF, Data_inizio, Data_scadenza)

Noleggia (ID \rightarrow DVD_ID, data_inizio, data_fine, CF_cliente \rightarrow Cliente.CF)

5 Query e indici

5.1 Queries

- **Numero totale di noleggi effettuati in ciascuna videoteca:**

```
SELECT v.Nome AS Videoteca, COUNT(n.ID) AS Totale_Noleggi
FROM Noleggia n
JOIN DVD d ON n.ID = d.ID
JOIN Videoteca v ON d.Nome_videoteca = v.Nome
GROUP BY v.Nome
ORDER BY Totale_Noleggi DESC;
```

	videoteca character varying (100) 	totale_noleggi bigint 
1	Videoteca Parma	20
2	Videoteca Firenze	20
3	Videoteca Bologna	10

- **Numero totale di DVD danneggiati per ciascun cliente:**

```
SELECT c.Nome AS Cliente, COUNT(dg.ID_copia) AS Totale_Danneggiati
FROM Cliente c
JOIN Danneggia dg ON c.CF = dg.CF_cliente
GROUP BY c.Nome
ORDER BY Totale_Danneggiati DESC;
```

	cliente character varying (100) 🔒	totale_danneggiati bigint 🔒
1	Davide Giallo	1
2	Paolo Gialli	1
3	Carla Neri	1
4	Giulia Verdi	1
5	Luigi Bianchi	1
6	Sara Verde	1
7	Elena Arancio	1
8	Mario Rossi	1
9	Roberto Nero	1
10	Anna Blu	1

- **Numero di film scaricati per ciascun cliente e l'importo totale di spazio occupato in GB:**

```
SELECT c.Nome AS Cliente, COUNT(s.ID_film) AS Totale_Scaricati, SUM(f.GB) AS
Totale_GB
FROM Cliente c
JOIN Scarica s ON c.CF = s.CF_cliente
JOIN File f ON s.ID_film = f.ID
GROUP BY c.Nome
ORDER BY Totale_Scaricati DESC;
```

	cliente character varying (100) 🔒	totale_scaricati bigint 🔒	totale_gb numeric 🔒
1	Davide Giallo	1	3.00
2	Paolo Gialli	1	6.00
3	Carla Neri	1	2.50
4	Giulia Verdi	1	3.80
5	Luigi Bianchi	1	3.20
6	Sara Verde	1	4.50
7	Elena Arancio	1	5.00
8	Mario Rossi	1	4.50
9	Roberto Nero	1	3.50
10	Anna Blu	1	4.00

- **Registi che hanno diretto più di un film insieme al numero totale di film diretti**

```
SELECT r.Nome AS Regista, COUNT(d.ID) AS Totale_Film
FROM Regista r
JOIN DVD d ON r.CF = d.CF_regista
GROUP BY r.Nome
HAVING COUNT(d.ID) > 1
ORDER BY Totale_Film DESC;
```

	regista character varying (100) 🔒	totale_film bigint 🔒
1	Federico Fellini	3
2	Sergio Leone	2

- **Numero medio di noleggi per cliente per ciascuna videoteca:**

```
SELECT v.Nome AS Videoteca, AVG(n.Noleggi_Per_Cliente) AS
Media_Noleggi_Per_Cliente
FROM (
    SELECT c.CF, d.Nome_videoteca, COUNT(n.ID) AS Noleggi_Per_Cliente
    FROM Cliente c
    JOIN Noleggia n ON c.CF = n.CF_cliente
    JOIN DVD d ON n.ID = d.ID
    GROUP BY c.CF, d.Nome_videoteca
) AS n
JOIN Videoteca v ON n.Nome_videoteca = v.Nome
GROUP BY v.Nome
ORDER BY Media_Noleggi_Per_Cliente DESC;
```

	videoteca character varying (100) 🔒	media_noleggi_per_cliente numeric 🔒
1	Videoteca Parma	1.6666666666666667
2	Videoteca Firenze	1.4285714285714286
3	Videoteca Bologna	1.2500000000000000

5.2 Indici

Per la definizione di un indice prendiamo in esame la query 5: “*Numero medio di noleggi per cliente per ciascuna videoteca*”.

Motivazione per la scelta dell’indice:

Questa query aggrega i dati dalla tabella **Noleggia** e **DVD** per contare il numero totale di noleggi per ciascuna videoteca. L'indicizzazione delle colonne utilizzate nelle condizioni di join e di aggregazione può migliorare significativamente le prestazioni della query, riducendo i tempi di scansione delle tabelle.

Creazione dell'indice sulla tabella Noleggia per ottimizzare le ricerche di noleggi per DVD

```
CREATE INDEX idx_noleggia_id ON Noleggia (ID);
```

Creazione dell'indice sulla tabella DVD per ottimizzare le ricerche di noleggi per videoteca

```
CREATE INDEX idx_dvd_id_nome_videoteca ON DVD (ID, Nome_videoteca);
```

Le queries originali e gli indici definiti contribuiranno a migliorare le prestazioni complessive delle operazioni di lettura del database, specialmente per queries che richiedono l'aggregazione e il join su tabelle grandi.

Includendo questi indici, si garantisce che le operazioni di ricerca e aggregazione siano eseguite in modo più efficiente, riducendo i tempi di risposta per le queries più complesse.

Tempo impiegato per eseguire la query 5 senza indici:

	videoteca character varying (100) 🔒	media_noleggi_per_cliente numeric 🔒
1	Videoteca Parma	1.6666666666666667
2	Videoteca Firenze	1.4285714285714286
3	Videoteca Bologna	1.2500000000000000

✓ Successfully run. Total query runtime: 44 msec. 3 rows affected. ✗

Tempo impiegato per eseguire la query 5 con gli indici appena descritti:

	videoteca character varying (100) 🔒	media_noleggi_per_cliente numeric 🔒
1	Videoteca Parma	1.6666666666666667
2	Videoteca Firenze	1.4285714285714286
3	Videoteca Bologna	1.2500000000000000

✓ Successfully run. Total query runtime: 37 msec. 3 rows affected. ✗

Come si può notare il tempo d'esecuzione della query è inferiore quando vengono utilizzati degli indici.

6 Codice C

Il codice C per l'esecuzione delle query consiste in un unico file .c che va compilato attraverso il makefile, in modo da linkare facilmente le librerie aggiuntive installate. Quindi i comandi da eseguire per eseguire il codice C sono:

Modificare il makefile in modo da linkare correttamente le librerie aggiuntive: nel nostro caso abbiamo utilizzato il tool "Homebrew" per la gestione di quest'ultime.

```
cd <Percorso della cartella contenete il file .c>
```

```
make
```

```
<Percorso della cartella di output della compilazione>/<nome del file eseguibile>
```

Prima della compilazione è necessario sostituire nelle variabili globali **PG_HOST**, **PG_USER**, **PG_DB**, **PG_PASS**, **PG_PORT** rispettivamente l'indirizzo dell'host, il nome utente di PostgreSQL, il nome del database, la password e la porta utilizzata.

Il programma utilizza la funzione PQconnectdb per connettersi al database, la funzione PQstatus per verificare lo stato della connessione, la funzione PQfinish per terminare la connessione.

Una volta avviato correttamente il programma sullo schermo apparirà il seguente Menù con la lista delle query, identificate da 1 a 6, mentre lo 0 terminerà.

Alcune query richiedono l'inserimento di dati aggiuntivi da parte dell'utente, in particolare:

- La query numero 1 restituisce tutti i gestori che lavorano in videoteche situate in comuni con più di un certo numero di abitanti. L'utente deve inserire il numero di abitanti.
- La query numero 2 restituisce tutti i registi che hanno diretto DVD con più di un certo numero di minuti e i loro rispettivi fornitori. All'utente è richiesto di inserire il minutaggio.
- La query numero 6 restituisce i cineforum organizzati in videoteche con orari di apertura precedenti ad un certo orario. L'utente deve inserire un orario di apertura

AMBIENTE DI SVILUPPO

Alberto Canavese	Cheng Feng Liao
SO: MacOS 14.2	Ubuntu 22.04.4 LTS
Editor C: Visual Studio Code	Editor C: Visual Studio Code
Compilatore: g++ 14.0.3	Compilatore: gcc version 11.4.0 (Ubuntu 11.4.0-1ubuntu1~22.04)
Editor SQL: PostgreSQL 16.3	Editor SQL: PostgreSQL 16.3