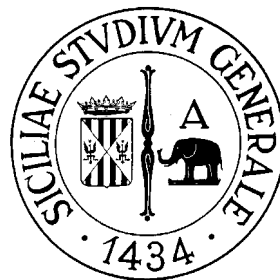


Basi Di Dati

AA 22/23

- Documentazione Progetto Basi Di Dati



Relazione presentata da Messina Michele

Relazione relativa al progetto di Basi Di Dati

Si vuole implementare un database inerente alla gestione di un cinema

Indice

-Descrizione e specifiche sui dati.....	3
-Glossario dei termini.....	5
-Glossario dei simboli.....	7
-Analisi dei requisiti.....	8
-Analisi della strategia	5
-Schema scheletro completo	18
-Schema E-R	26
-Business rules	27
-Analisi dei costi dovuti alle ridondanze	28
-Conseguenze delle analisi dei costi	31
-Ristrutturazione dello schema E-R	32
-Traduzione nel modello logico	37
-Implementazione database in SQL	38
-Varie Interrogazioni in SQL.....	40
-Creazione Trigger SQL.....	42
-Implementazione database in XML.....	43

Descrizione e Specifiche sui dati

Introduzione: La presente documentazione fornisce una descrizione dettagliata del progetto relativo alla creazione di un database per la gestione di un cinema. Saranno presentati tutti i dettagli, le modifiche e le aggiunte pertinenti alla creazione della base di dati, al fine di soddisfare i vari requisiti stabiliti.

1. Analisi dei requisiti: Questa sezione analizza in dettaglio i requisiti del sistema di gestione del cinema e fornisce una panoramica delle funzionalità richieste. Include una descrizione dei dati che devono essere memorizzati nel database, come ad esempio informazioni sugli spettacoli, sugli utenti, sulle sale, sui film e sulle prenotazioni.

2. Schema concettuale: In questa sezione viene presentato lo schema concettuale del database utilizzando il modello Entità-Relazione (E-R). Lo schema fornisce una rappresentazione top-down della progettazione del database, identificando le entità principali, le relazioni tra di esse e gli attributi associati. Vengono fornite spiegazioni dettagliate delle scelte di progettazione effettuate.

3. Operazioni del database: Qui vengono descritte tutte le operazioni che possono essere effettuate nel database, come ad esempio l'inserimento di nuovi spettacoli, la prenotazione di biglietti, la gestione dei pagamenti e la generazione di report. Per ciascuna operazione vengono fornite le specifiche dettagliate, comprese le tabelle coinvolte e le regole di validazione.

4. Progettazione fisica: Questa sezione descrive le fasi successive della progettazione del database, che includono la traduzione dello schema concettuale in uno schema logico, la definizione delle tabelle, degli indici e delle chiavi esterne. Viene fornito uno schema finale che rappresenta la struttura del database completo.

5. Interrogazioni e istruzioni SQL: Qui vengono elencate diverse interrogazioni che possono essere eseguite nel database, insieme alle relative istruzioni SQL. Le interrogazioni includono esempi di ricerca di film per genere, visualizzazione degli spettatori di uno spettacolo specifico e aggiornamento delle informazioni sugli spettacoli. Vengono fornite anche istruzioni per l'inserimento, l'aggiornamento e la cancellazione dei dati.

Conclusione: La documentazione di progettazione per il database di gestione di un cinema fornisce una guida completa per la creazione e l'utilizzo del sistema. Le informazioni dettagliate riguardanti l'analisi dei requisiti, lo schema concettuale, le operazioni del database, la progettazione fisica e le interrogazioni SQL consentono una comprensione completa del funzionamento del database e delle sue funzionalità.

Descrizione e Specifiche sui dati

Si intende realizzare un progetto relativo a un database finalizzato alla corretta funzionalità di un cinema. Il sistema gestirà i dipendenti che operano secondo turni prestabiliti, svolgendo diverse attività come la vendita di biglietti, la vendita di bevande e il controllo dei biglietti prima dell'ingresso nelle sale. Sarà possibile effettuare prenotazioni dei biglietti per i diversi film disponibili al momento dell'acquisto. È necessario disporre di una relazione (tabella) relativa a un catalogo di film, i quali avranno una disponibilità limitata per un periodo di tempo prestabilito a partire dalla data di uscita nelle sale. Ogni impiegato sarà associato a un dirigente come referente. Nel caso in cui il dirigente sia assente per motivi opportuni, sarà possibile designare un vice-dirigente come dirigente per quella giornata. Pertanto, sarà necessario registrare la presenza giornaliera di ogni dipendente/vice-dirigente/dirigente. Sarà prevista la possibilità di modificare la tabella dei film una volta che questi abbiano raggiunto la data di scadenza prefissata per l'uscita. Il database dovrà anche contenere le informazioni relative alle diverse sale destinate alla proiezione dei film. Ad ogni sala sarà associato un orario prestabilito e un film specifico in modo univoco. Le sale saranno accessibili solo al personale autorizzato e potranno essere aperte solo tramite un'identificazione mediante un ID.

Sarà possibile acquistare i biglietti, e una volta che un posto specifico per un determinato film è stato acquistato, il biglietto per quel posto non sarà più disponibile. Saranno previsti sconti per gli studenti durante determinati giorni della settimana. Inoltre, saranno offerte promozioni per i bambini di età inferiore ai 12 anni e per le persone anziane con un'età superiore ai 58 anni. Il giovedì sarà dedicato a uno sconto per le persone di sesso femminile, mentre il venerdì per quelle di sesso maschile.

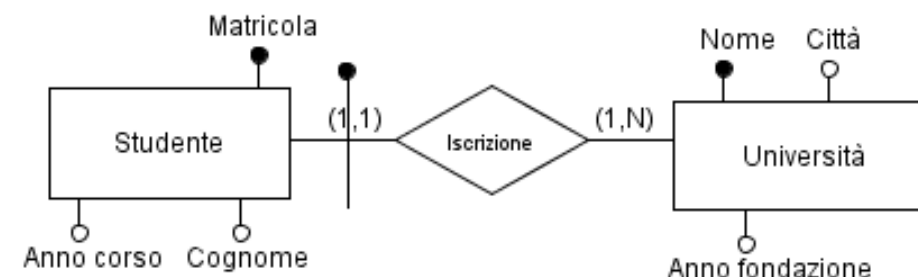
Sarà possibile acquistare bevande e cibi all'interno del cinema, con costi diversi in base alle dimensioni e al tipo di alimento/bevanda. Sarà anche prevista la possibilità per i dipendenti di ricevere un premio in denaro una volta al mese, in base al lavoro svolto.

Glossario Dei Termini : Modello Logico

Termine	Descrizione	Sinonimo
DataBase	uno strumento che permette di organizzare, archiviare, gestire, proteggere e controllare l'accesso ai dati.	Banca Dati
Relazione	legame logico che permette di aggregare informazioni	Tabella
Chiave	vincolo intrarelazionale che interessa una sola tabella, che nel modello relazionale viene detta relazione.	Key
Chiave Primaria	serve a identificare univocamente le righe della tabella ove essa è definita.	Primary Key
Chiave Secondaria	un vincolo di integrità referenziale tra due o più tabelle.	Chiave esterna,Secondary Key

Glossario Dei Termini : Modello Logico

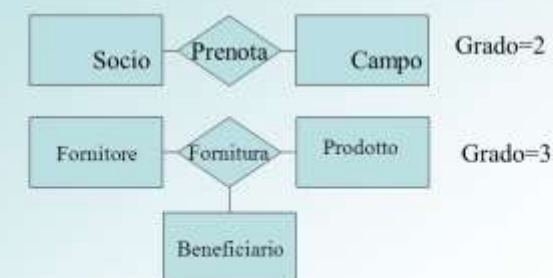
- Le **Associazioni** (o **Relazioni**) rappresentano legami logici fra due o più entità dell'applicazione.
 - Possono esserci più relazioni fra le stesse entità (associazioni ricorsive).
- **Istanza di Associazione:** fatto che descrive un'azione o una situazione e che stabilisce legami tra istanze di entità (associa, mette in relazione)



- **Entità** sono classi di oggetti (cose, persone) che hanno proprietà comuni ai fini dell'applicazione di interesse che si intende modellare.
- **Istanza di Entità:** cosa (oggetto, persona) che esiste di per sé nel dominio applicativo, della quale si vogliono registrare fatti specifici e che può essere chiaramente identificata in modo da poterla distinguere dalle altre

Grado di una relazione

- **Grado di una relazione:** numero di entità partecipanti



Glossario Dei Termini : Cinema

Termine	Descrizione	Sinonimo	Legame
Dipendenti	Chi svolge le proprie mansioni alle dipendenze di altri o di un ente.	Lavoratore	Biglietto, Prodotto,Sala
Clienti	Chi abitualmente si vale delle prestazioni di qualcuno o acquista quanto gli occorre dallo stesso fornitore;	Acquirente	Biglietto, Prodotto,Sala
Biglietto	Piccolo stampato rilasciato come contrassegno del diritto di usufruire di un pubblico servizio.	Scontrino	Sala, Dipendente
Prodotto	Quanto si ottiene da attività tecnicamente ed economicamente definite	Articolo	Dipendente, Cliente, Biglietto
Sala	Luogo indicato con un valore numerico con posti fissi.	Stanza	Biglietto, Prodotto

Creazione Relazioni Per Le Varie Entità : Dipendenti

Personale

Ogni Impiegato sarà fornito di un id con il quale potrà accedere all'interno delle varie sale e sarà riconosciuto da quest'ultimo nella tavola delle presenze .Inoltre avrà anche un saldo ed un orario fisso .

Vice-Dirigente

Il Vice-Dirigente verrà riconosciuto mediante la sua occupazione ,necessiterà anch'esso di un identificativo con un saldo maggiore rispetto ad un normale Impiegato ma minore rispetto ad un Dirigente .Come nel caso dell'Impiegato anch'esso avrà un saldo ed un orario fisso.

Dirigente

Il Dirigente verrà riconosciuto mediante la sua occupazione ,necessiterà anch'esso di un identificativo con un saldo maggiore rispetto ad un normale Impiegato ma minore rispetto ad un Dirigente .Come nel caso dell'Impiegato e del Vice-Dirigente anch'esso avrà un saldo ed un orario fisso.

Si noti bene inoltre che sia nel caso del Vice-Dirigente che del Dirigente l'orario di lavoro inizierà all'apertura del cinema e terminerà alla sua chiusura rispetto ad un normale dipendente.

Creazione Relazioni Per Le Varie Entità : Dipendenti

Personale

Id	Nome	Cognome	Occupazione	Stipendio	Orario Inizio	Orario Fine	Presenza	Id cinema
3892839289	Marco	Giustioliso	Dipendente	1800€	15:30	22:00	Assente	Multicinema
2498174829	Salvatore	Pisa	Dirigente	2400€	15:00	01:00	Assente	THE CITY CINEMA
2189589185	Christian	Latenza	Vice-Dirigente	2100€	15:00	01:00	Assente	Multicinema
5892789835	Franco	Catania	Dipendente	1800€	17:00	00:30	Assente	EPlanet Vasquez
3589683080	Giusi	Regina	Dipendente	1800€	17:00	00:30	Assente	Multicinema

Creazione Relazioni Per Le Varie Entità: Film

Film

Si crei una relazione in cui vengano descritti i film disponibili all'interno del cinema mediante:

<->Un identificatore del film.

<->Il nome del film.

<->La tipologia del film scelto(Horror, Drammatico, Commedia, Fantasy,ecc..).

<->La durata effettiva del film(Durata espressa in minuti)

<->L'orario di inizio del film

<->L'orario finale del film(Orario finale ottenuto con l'orario di inizio e la durata)

<->La sala in cui il film verrà proiettato(A,B,C,D,E)

<->La data di uscita del film

<->La data di ritiro del film(data di ritiro che potrà essere modificabile)

Si assocerà successivamente nella relazione Biglietti il costo relativo ad ogni film.

Inoltre si consideri che ad ogni film viene messa in relazione un'unica sala , che due film potrebbero avere anche la stessa durata e potrebbero essere usciti nello stesso giorno.

Creazione Relazioni Per Le Varie Entità : Film

Id	Nome	Genere	Durata	Orario inizio	Orario fine	Sala	Data di uscita	Data di ritiro
5197353	Guardiani della Galassia	Azione, Fantascienza, Commedia	150 minuti	21:00	23:30	E	03/05/2023	04/06/2023
5390905	Book Club - Il Capitolo Successivo	Commedia	110 minuti	20:30	22:20	A	11/05/2023	28/05/2023
4690966	Love Again	Commedia, Drammatico, Sentimentale	104 minuti	19:00	20:04	D	11/05/2023	25/05/2023
3489786	Super Mario Bros. - Il Film	Animazione, Avventura, Commedia	92 minuti	17:00	18:32	C	05/04/2023	07/05/2023
4768420	La Caccia	Drammatico	100 minuti	16:00	17:40	A	11/05/2023	18/05/2023

Creazione Relazioni Per Le Varie Entità: Sale

Sala

Si nota la creazione di diverse Sale dove ognuna di essa sarà identificata mediante un carattere dell'alfabeto che va dal primo carattere (A) fino al carattere E per un totale di 5 Sale. Ogni sala inoltre potrà essere aperta solamente dai Dipendenti mediante identificazione id ed avrà una capienza di posti fissi dove ,una volta arrivati alla capienza totale non sarà più possibile prenotare per la proiezione del film .

Dunque la relazione Sala avrà queste peculiarità:

- <-> Carattere identificativo (A,B,C,D,E)
- <-> Numero posti(es.100 posti valore fisso)
- <->Aperta e/o chiusa
- <->Prenotabile

Creazione Relazioni Per Le Varie Entità : Sale

Id Sala	Numero Posti	Aperta e/o Chiusa	Prenotabile
A	100	Chiusa	Prenotabile
B	130	Chiusa	Prenotabile
C	127	Chiusa	Prenotabile
D	146	Chiusa	Prenotabile
E	110	Chiusa	Prenotabile

Creazione Relazioni Per Le Varie Entità : Biglietti e Clienti

Biglietto:

Si noti la creazione di una relazione di un biglietto ,dove quest'ultima dovrà essere:

- <-> identificabile mediante un identificativo numero
- <->id del film
- <-> Nome del film
- <->Costo biglietto
- <->La Sala del film per cui è stato richiesto
- <-> Numero posti



Clienti:

Si noti la creazione di una relazione Cliente ,dove quest'ultima dovrà essere:

- <->Identificabile mediante un id
- <->Genere
- <->Se Studente

Creazione Relazioni Per Le Varie Entità : Biglietti

Id biglietto	Id del film	3D	Posti	Costo Biglietto	Nome	Sala
3892839289	5197353	Si	18 J	18	Guardiani della Galassia	E
2498174829	5390905	No	27 T	24	Book Club - Il Capitolo Successivo	A
2189589185	4690966	No	5 D	12	Love Again	D
5892789835	3489786	Si	3Q	8	Super Mario Bros. - Il Film	C
3589683080	4768420	Si	23 L	68	La Caccia	B

Creazione Relazioni Per Le Varie Entità : Clienti

Id	Nome	Cognome	Genere	Età	Studente	Id biglietto
389283928	Franco	Capitale	Maschio	18	No	2189589185
2498174829	Federico	Salvino	Maschio	24	Si	3892839289
2189589185	Christian	Latenza	Femmina	12	Si	3589683080
589278983	Luigi	Magnarolo	Maschio	8	No	5892789835
358968308	Nichol	Regina	Femmina	68	No	2498174829

Creazione Relazioni Per Le Varie Entità : Cinema

Cinema

Alle varie relazione Indicate si aggiunge anche la relazione cinema che sarà definita da attributi come:

<->Nome

<->Città

<->Numero sale

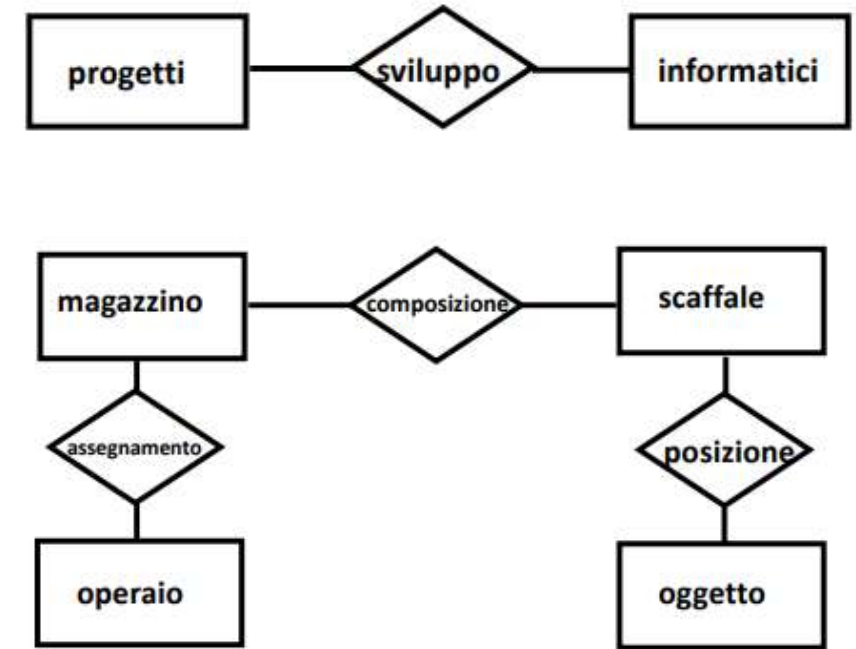
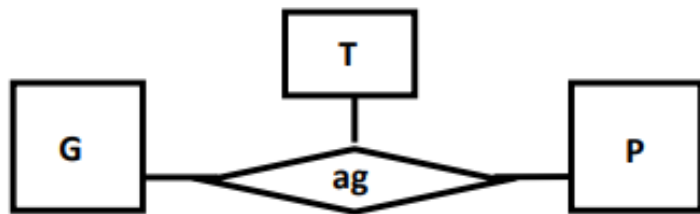


Attributi
Cinema

Nome	Città	Numero Sale
Multicinema	Moncalieri	5
EPlanet Vasquez	Siracusa	5
THE CITY CINEMA	Salerno	5

Schema Scheletro Completo

- Gli schemi scheletro descrivono una prima struttura di massima dello schema senza indicazioni sul TIPO delle entità e delle associazioni
- Descrivono in generale i collegamenti tra le entità di interesse e le associazioni che le legano



Gli schemi scheletro descrivono in generale i collegamenti tra le entità e le associazioni;

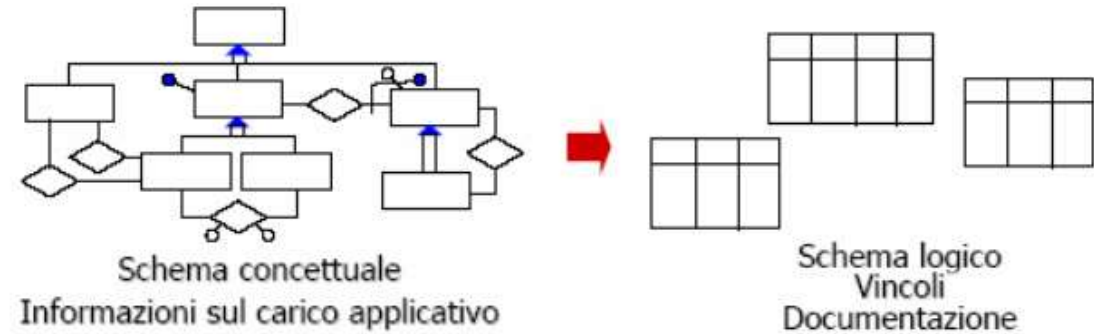
- Le entità e le associazioni devono essere descritte attraverso l'aggregazione di proprietà.

L'obiettivo della fase di progettazione logica è prevenire, a partire dallo schema concettuale, a uno schema logico che lo **rappresenti** in modo fedele e che sia, al tempo stesso **"efficiente"**.

-**Input**: schema concettuale + informazioni sul carico applicativo

-**Output**: schema logico con vincoli e documentazione

Nel nostro caso la seguente progettazione logica ci permetterà di identificare e migliorare qualsiasi problema legato alla prima versione di uno schema logico fatto precedentemente.



● Schema ER iniziale + specifiche sul carico applicativo

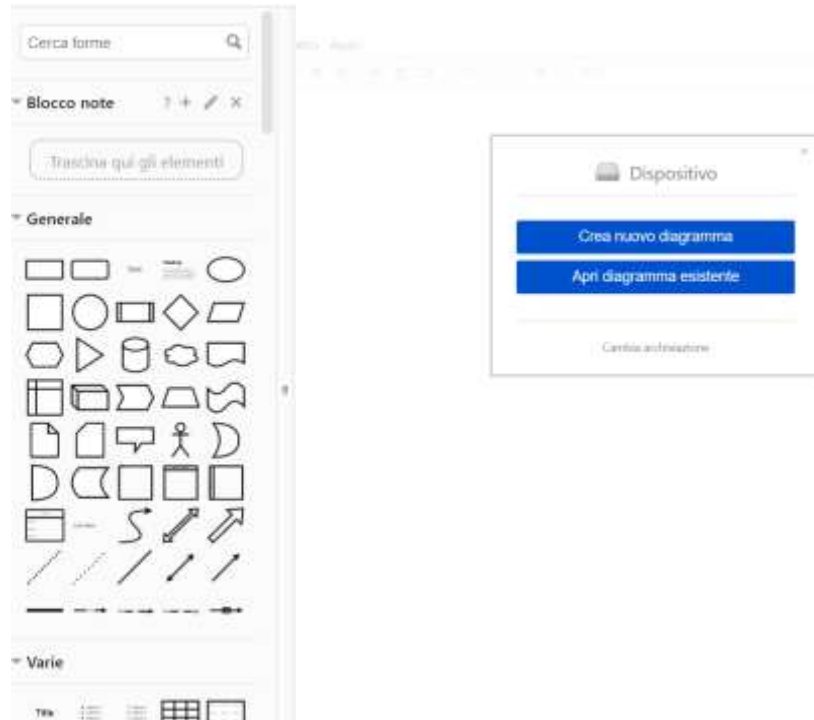
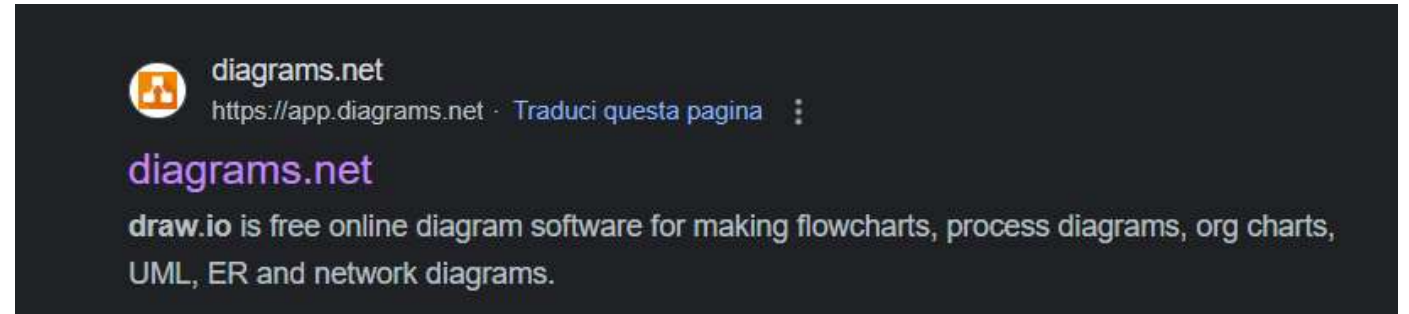
- **Analisi delle ridondanze**
 - » Mantenere/rimuovere le ridondanze dello schema iniziale
- **Eliminazione delle generalizzazioni**
 - » Le generalizzazioni sono analizzate e sostituite da altri costrutti
- **Partizionamento/accorpamento di entità e associazioni**
 - » Suddivisione di un concetto (entità o relazione) in più concetti
 - » Accorpamento di più concetti separati in un solo concetto
- **Eliminazione di attributi composti e/o multivalore**
- **Scelta degli identificatori primari**
 - » Per le entità che hanno più di un identificatore

La progettazione logica può articolarsi in due fasi principali:

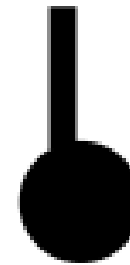
- **Ristrutturazione**: eliminazione dallo schema E-R di tutti i costrutti che non possono essere direttamente rappresentati nel modello logico target (relazionale nel nostro caso):
 - » **Eliminazione** delle gerarchie di generalizzazione
 - » **Partizionamento/accorpamento** di entità e associazioni
 - » **Eliminazione** degli attributi composti e/o multivalore
 - » Scelta degli **identificatori** principali

Creazione Schema Scheletro

Per la progettazione del nostro Schema Scheletro completo usufruiremo di un sito creato per l'apposita creazioni di schemi concettuali di varie strutture e tipologie: <https://app.diagrams.net/>



Partiremo seguendo tutte le informazioni scritte precedentemente, aggiungendo passo per passo durante la costruzione le relative Chiavi primarie (A), Chiavi Secondarie (B) e le relazioni che ci devono essere tra le varie relazioni (C),



(A)



(B)

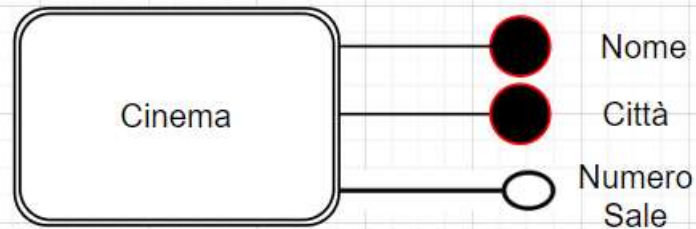


(C)

Creazione Schema Scheletro

Partiamo andando a creare un primo schema delle varie relazioni che abbiamo anticipato prima:

Cinema



Sale



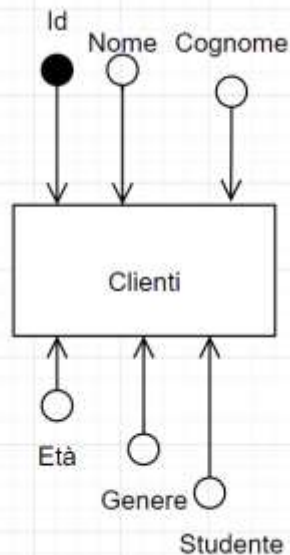
Si notino i diversi punti di unioni tra tabelle

Biglietto



Film

Creazione Schema Scheletro



Clienti

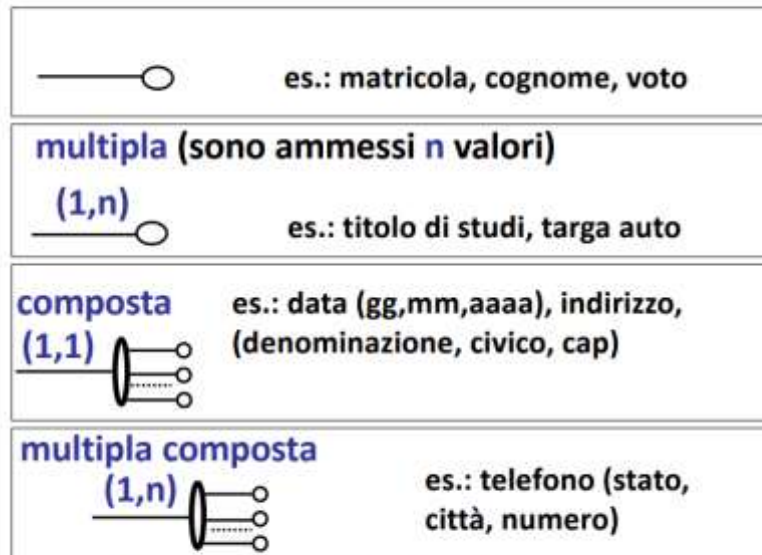
Dipendenti



- Gli schemi scheletro descrivono in generale i collegamenti tra le entità e le associazioni;
- Le entità e le associazioni devono essere descritte attraverso l'aggregazione di proprietà.

Andremo ad utilizzare il concetto di proprietà anche nel nostro schema concettuale

Proprietà



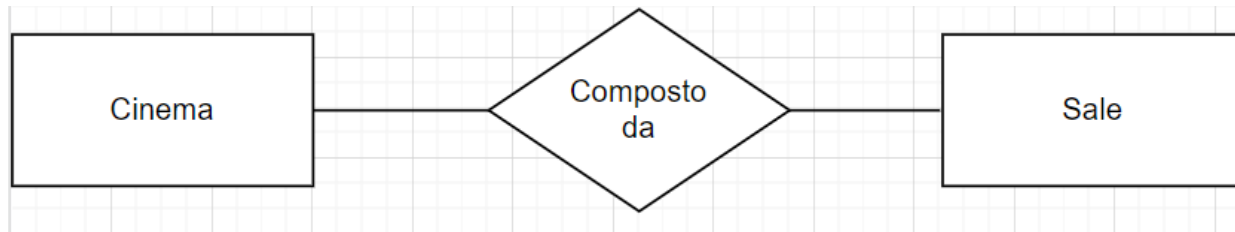
Il simbolo **(n,m)** esprime la **cardinalità** minima e massima della proprietà

Proprietà **opzionale** (è ammessa la "non esistenza del valore")



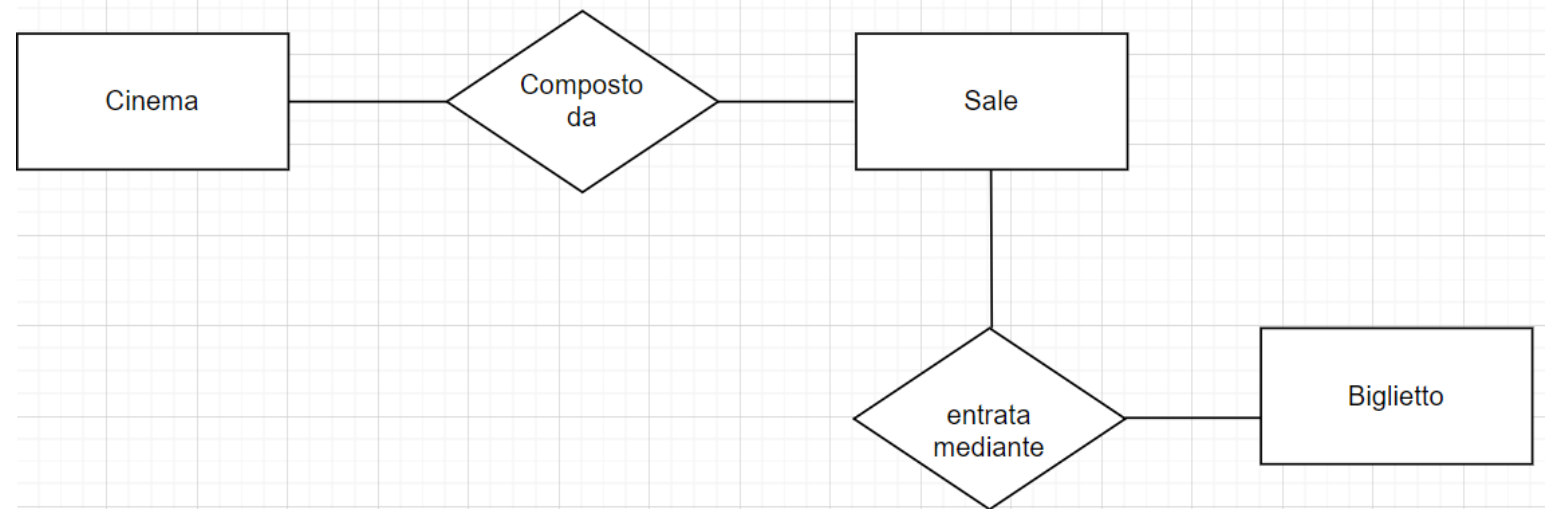
Schema scheletro

Si identifica l'entità **Cinema**. Si vuole quindi modellare che sia suddiviso in diverse **Sale** per la proiezione del film

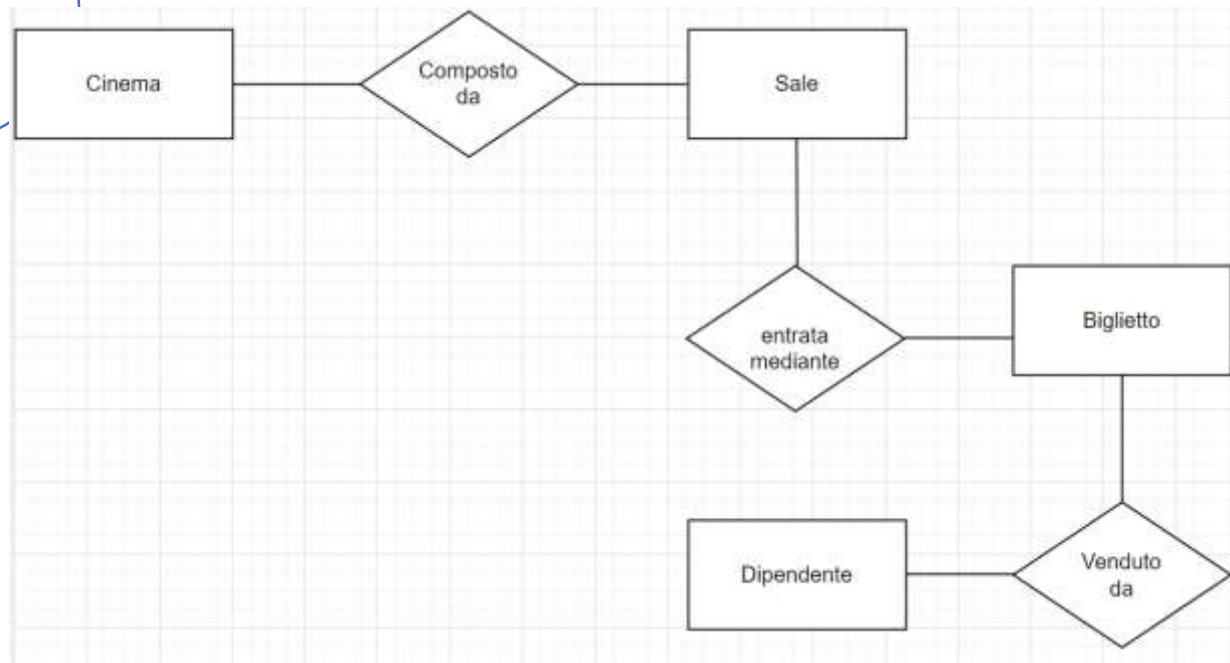


Questi schemi permettono di capire le relazioni ed i collegamenti che ci sono tra le varie relazioni

Ci si rende subito conto che per potere entrare in una **Sala** vi è la necessità di avere comprato l'apposito **Biglietto**

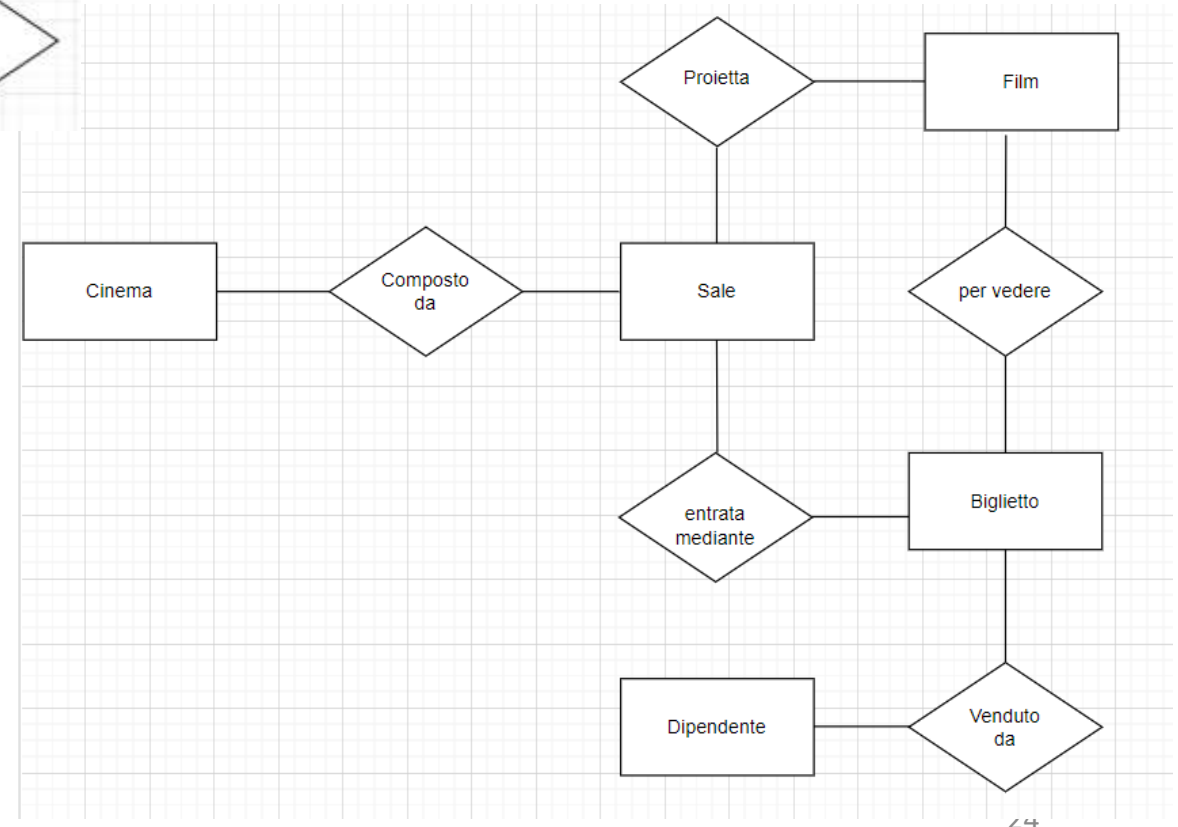


Si necessita a questo punto di un venditore del **Biglietto** per potere entrare in una sala ; esattamente ci serve un **Dipendente**:



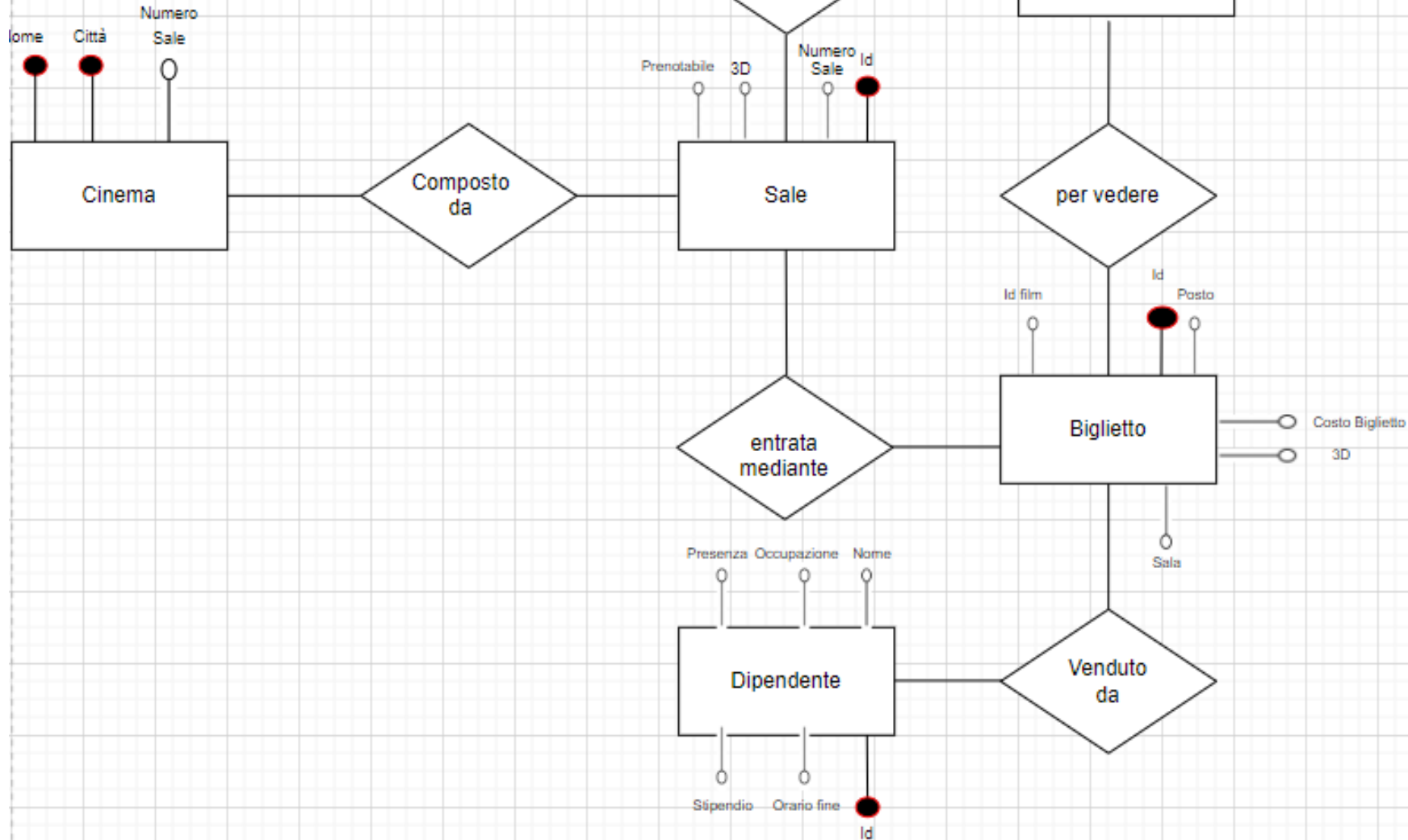
Dunque si necessita ora di aggiungere come consono un collegamento ad un entità di tipo **Film**. Un film viene proiettato in ogni **Sala** di un cinema ed ad un **Biglietto** corrisponde un film dunque:

Otteniamo in questo modo uno schema definitivo privo però delle proprietà che abbiamo inserito separatamente anteriormente ; Rappresentiamo quindi lo schema finale con tutte le proprietà

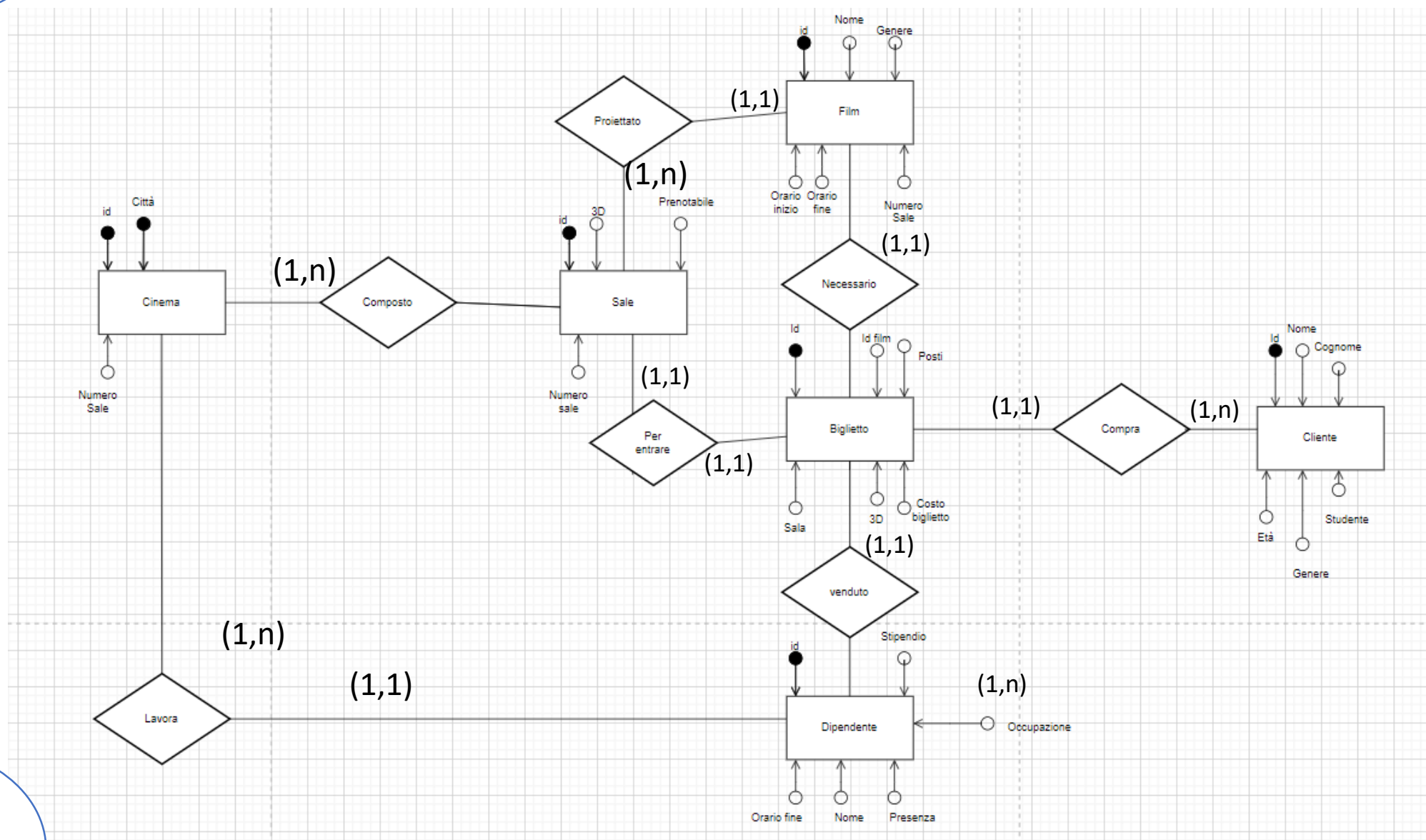


Schema definitivo

Di fianco si nota lo schema proposto precedentemente con l'aggiunta delle varie chiavi tra cui le chiavi **primarie** (descritte con un punto nero) e le chiavi **secondarie**



Schema definitivo



-Affianco si nota lo schema definitivo del modello concettuale, dopo l'aggiunta delle varie proprietà e/o collegamenti per ogni entità

Business Rule

Una **regola aziendale** (business rule) definisce o vincola determinati aspetti dell'attività commerciale. Può essere formulata per specificare un'azione da intraprendere quando determinate condizioni sono soddisfatte, oppure può essere espressa in modo che la sua verità o falsità possa essere determinata.

Vincolo sui Clienti: I clienti possono guardare qualsiasi film desiderino e acquistare bevande e/o cibo senza restrizioni. Tuttavia, i clienti non possono acquistare biglietti per spettacoli quando non sono più disponibili e non devono avere la possibilità di aprire le sale.

Vincolo sui Dipendenti: I dipendenti, indipendentemente dalla categoria di appartenenza (dirigente, vice-dirigente, impiegato), hanno un orario di ingresso e di uscita fisso e possono svolgere sia la vendita dei biglietti che la vendita di cibo e/o bevande. Tuttavia, ogni impiegato deve essere presente all'orario stabilito e può uscire solamente dopo il superamento dell'orario di chiusura.

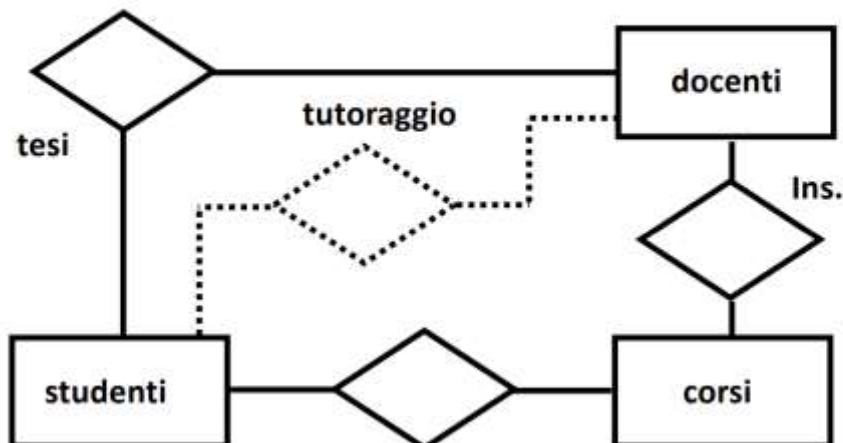
Vincolo sui Biglietti: Un cliente può acquistare quanti biglietti desidera, anche se poi non si presenta per la proiezione del film. Tuttavia, nel caso in cui un cliente desideri modificare il proprio biglietto, dovrà pagare un importo uguale o

Ristrutturazione degli schemi ER

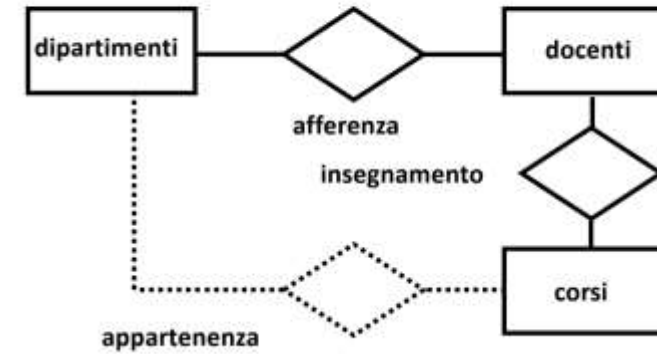
- Schema ER iniziale + specifiche sul carico applicativo

- **Analisi delle ridondanze**
 - » Mantenere/rimuovere le ridondanze dello schema iniziale
- **Eliminazione delle generalizzazioni**
 - » Le generalizzazioni sono analizzate e sostituite da altri costrutti
- **Partizionamento/accorpamento di entità e associazioni**
 - » Suddivisione di un concetto (entità o relazione) in più concetti
 - » Accorpamento di più concetti separati in un solo concetto
- **Eliminazione di attributi composti e/o multivalore**
- **Scelta degli identificatori primari**
 - » Per le entità che hanno più di un identificatore

Associazioni ridondanti?



Associazioni ridondanti?



Analisi delle ridondanze

- Una **ridondanza** in uno schema E-R è una **informazione significativa ma derivabile da altre**
- In questa fase si decide se eliminare le ridondanze eventualmente presenti o mantenerle
è quindi comunque importante averle individuate in fase di progettazione concettuale!
- Se si mantiene una ridondanza
 - 🔹 si **semplificano** alcune interrogazioni, ma
 - 🔹 si **appesantiscono** gli aggiornamenti
 - 🔹 si occupa **maggior spazio**
- La decisione viene presa confrontando **i costi delle operazioni** nel caso di presenza e assenza della ridondanza
- Le possibili ridondanze riguardano
 - **Attributi derivabili** da altri attributi
 - Associazioni derivabili dalla composizione di altre associazioni (**presenza di cicli**)

Le principali operazioni richieste dalla base di dati sono:

1. Potere comprare dei biglietti
2. Potere cambiare film
3. Potere entrare all'interno delle sale
4. Aprire le sale
5. Visualizzare l'orario di inizio dei film
6. Potere decidere di modificare la sala di un film

Per analizzare i volumi e la frequenza dei dati e determinare se è necessaria la ridondanza o se si verifica un'elevata inefficienza, è necessario esaminare le diverse tabelle coinvolte nel database del cinema. Di seguito verranno fornite alcune tabelle di esempio e verrà discussa la potenziale ridondanza dei dati in ciascuna tabella.

1.Tabella "Film": Questa tabella conterrà le informazioni sui film disponibili nel cinema, come titolo, regista, genere, durata, data di uscita, etc. La frequenza di inserimento dei nuovi film dipenderà dalla programmazione del cinema e dalla frequenza con cui vengono rilasciati nuovi film. La ridondanza potrebbe verificarsi se i dati relativi ai film vengono duplicati in altre tabelle, ad esempio nelle prenotazioni o negli spettacoli.

2.Tabella "Clienti": Questa tabella conterrà le informazioni sui clienti, come nome, cognome, indirizzo, numero di telefono, etc. La frequenza di inserimento dei nuovi clienti dipenderà dal numero di persone che frequentano il cinema e creano un account. In generale, non si prevede una ridondanza elevata per questa tabella, poiché i dati dei clienti dovrebbero essere unici e non dovrebbero essere duplicati in altre tabelle.

3.Tabella "Prenotazioni": Questa tabella conterrà le informazioni sulle prenotazioni effettuate dai clienti, come id prenotazione, id cliente, id film, data e ora della prenotazione, etc. La frequenza di inserimento dei nuovi dati dipenderà dal numero di prenotazioni effettuate dai clienti. Non è necessaria una ridondanza significativa in questa tabella, poiché le prenotazioni sono uniche e non dovrebbero essere duplicate in altre tabelle.

4.Tabella "Spettacoli": Questa tabella conterrà le informazioni sugli spettacoli, come id spettacolo, id film, id sala, data e ora dello spettacolo, etc. La frequenza di inserimento dei nuovi spettacoli dipenderà dalla programmazione del cinema e dalla frequenza con cui vengono organizzati gli spettacoli. Potrebbe essere presente una ridondanza se i dati sugli spettacoli vengono duplicati in altre tabelle, come nelle prenotazioni.

È importante analizzare attentamente le relazioni tra le tabelle e le necessità del sistema per valutare se è necessaria una ridondanza dei dati. L'obiettivo è evitare la duplicazione inutilizzata dei dati e garantire un'efficiente gestione delle informazioni. Se i dati sono adeguatamente normalizzati e le relazioni tra le tabelle sono ben definite, potrebbe non essere necessaria una ridondanza significativa nel database del cinema.

Tavola dei volumi

Concetto	Tipo	Volume
Impiegato	E	5
Cliente	E	5
Biglietto	E	10
Film	E	5
Cinema	E	3
Sale	E	5
Composto	R	n
Lavora	R	n
Proiettare	R	n
Entrare	R	n
Necessario	R	n
Venduto	R	n
Comprare	R	n

- Specifica il numero stimato di istanze per ogni entità (E) e relazione (R) dello schema
- I valori sono necessariamente approssimati, ma indicativi

Si noti che i valori relativi al numero di istanze di entità e relazioni nella tabella dei volumi sono influenzati:

- dalle cardinalità nello schema
- dal numero medio di volte che le istanze delle entità partecipano alle relazioni

n è un valore reale che possiamo scegliere

Tavole delle Frequenze

Concetto	Tipo	Volume
Potere comprare dei biglietti		10v/mese
Potere cambiare film		10v/mese
Potere entrare all'interno delle sale		1v/mese
Aprire le sale		1v/mese
Visualizzare l'orario di inizio dei film		20v/mese
Potere decidere di modificare la sala di un film		5v/mese

Tavola degli accessi

- Per ogni entità e associazione interessata dall'operazione, riporta il numero di istanze interessate, e il tipo di accesso (L: lettura; S: scrittura)

Il numero delle istanze si ricava dalla tavola dei volumi mediante semplici operazioni di base(ad esempio ad ogni cinema corrispondono almeno 3 lavoratori quindi a Lavora= $5 \times 3 = 15$)

Concetto	Costrutto	Accessi	Tipo
Dipendente	E	1	L
Biglietti	E	1	L
Cinema	E	1	L
Film	E	1	L
Sale	E	1	L
Clienti	E	1	L
Lavorare	I	3	L
Visionare	I	25	L
Composto	I	15	L
Entrata	I	20	L
Proiettato	I	25	L

Analisi dei costi dovuti alle ridondanze

Nota :

Durante l'analisi si considera il costo degli accessi in scrittura doppio rispetto a quello degli accessi in lettura

Analisi costi dovuti alla prima
operazione 1:

Potere comprare dei biglietti

Con ridondanza:

<-> 1 scrittura in "Cliente"

<-> 1 lettura in compra

<->10 scritture in "Biglietto"

$23 \cdot 10v/\text{mese} = 230 \text{ accessi/mese}$

Senza Ridondanza:

<-> 1 scrittura in "Cliente"

<-> 1 lettura in compra

<->10 scritture in "Biglietto"

$23 \cdot 10v/\text{mese} = 230 \text{ accessi/mese}$

Analisi costi dovuti alla prima operazione 2

Potere cambiare film

Con ridondanza:

<-> 1 scrittura in "Cliente"

<-> 1 lettura in compra

<->10 scritture in " Biglietto'

$23 \cdot 10v/mese = 230$ accessi/mese

Senza Ridondanza:

<-> 1 scrittura in "Cliente"

<-> 1 lettura in compra

<->10 scritture in "Dipendente"

$23 \cdot 10v/mese = 230$ accessi/mese

Analisi costi dovuti alla prima operazione 3

Potere entrare all'interno delle sale

Con ridondanza:

<-> 1 lettura in "Biglietto"

<-> 1 lettura in entrare

<->1 lettura in " Sala"

$3 \cdot 1v/mese = 3$ accessi/mese

Senza Ridondanza:

<-> 1 lettura in "Cliente"

<-> 1 lettura in compra

<->1 lettura in "Dipendente"

$3 \cdot 1v/mese = 3$ accessi/mese

Analisi costi dovuti alla prima operazione 4

Aprire le sale

Con ridondanza:

<-> 1 lettura in "Dipendenti"

<-> 1 lettura in far entrare

<-> 1 scrittura in "Biglietto"

$4 * 5v/mese = 20$ accessi/mese

Senza Ridondanza:

<-> 1 scrittura in "Sale"

$2 * 5v/mese = 10$ accessi/mese

Analisi costi dovuti alla prima operazione 5

Visualizzare l'orario di inizio dei film

Con ridondanza:

<-> 1 lettura in "film"

$1 * 20v/mese = 20$ accessi/mese

Senza Ridondanza:

<-> 1 scrittura in "Sale"

$1 * 20v/mese = 20$ accessi/mese

Analisi costi dovuti alla prima operazione 6

Potere decidere di modificare la sala di un film

Con ridondanza:

<-> 1 lettura in "film"

<-> 1 lettura in "proietta"

<-> 1 scrittura in "Sala"

$4 * 5v/mese = 20$ accessi/mese

Senza Ridondanza:

<-> 1 scrittura in "Sale"

$2 * 5v/mese = 10$ accessi/mese

Svolgeremo le varie operazioni per capire se tenere la ridondanza oppure no all'interno del progetto

Conseguenze delle analisi dei costi

Procediamo sommando il numero di e valutando se conviene mantenere o togliere possibili attributi ridondanti:

Con ridondanza:

<->Operazione 1:230;

<->Operazione 2:230;

<->Operazione 3:3

<->Operazione 4:20;

<->Operazione 5:20;

<->Operazione 6:20;

Totale $230*2+20*3+3=523$ accessi/mese

Senza ridondanza:

<->Operazione 1:230;

<->Operazione 2:230;

<->Operazione 3:3

<->Operazione 4:10

<->Operazione 5:20

<->Operazione 6:10

Totale $230*2+10*2+20+3=503/X$ accessi/mese

Valutazione finale della ridondanza per lo schema ER

Per valutare se conviene mantenere o togliere la ridondanza si risolve la disuguaglianza:

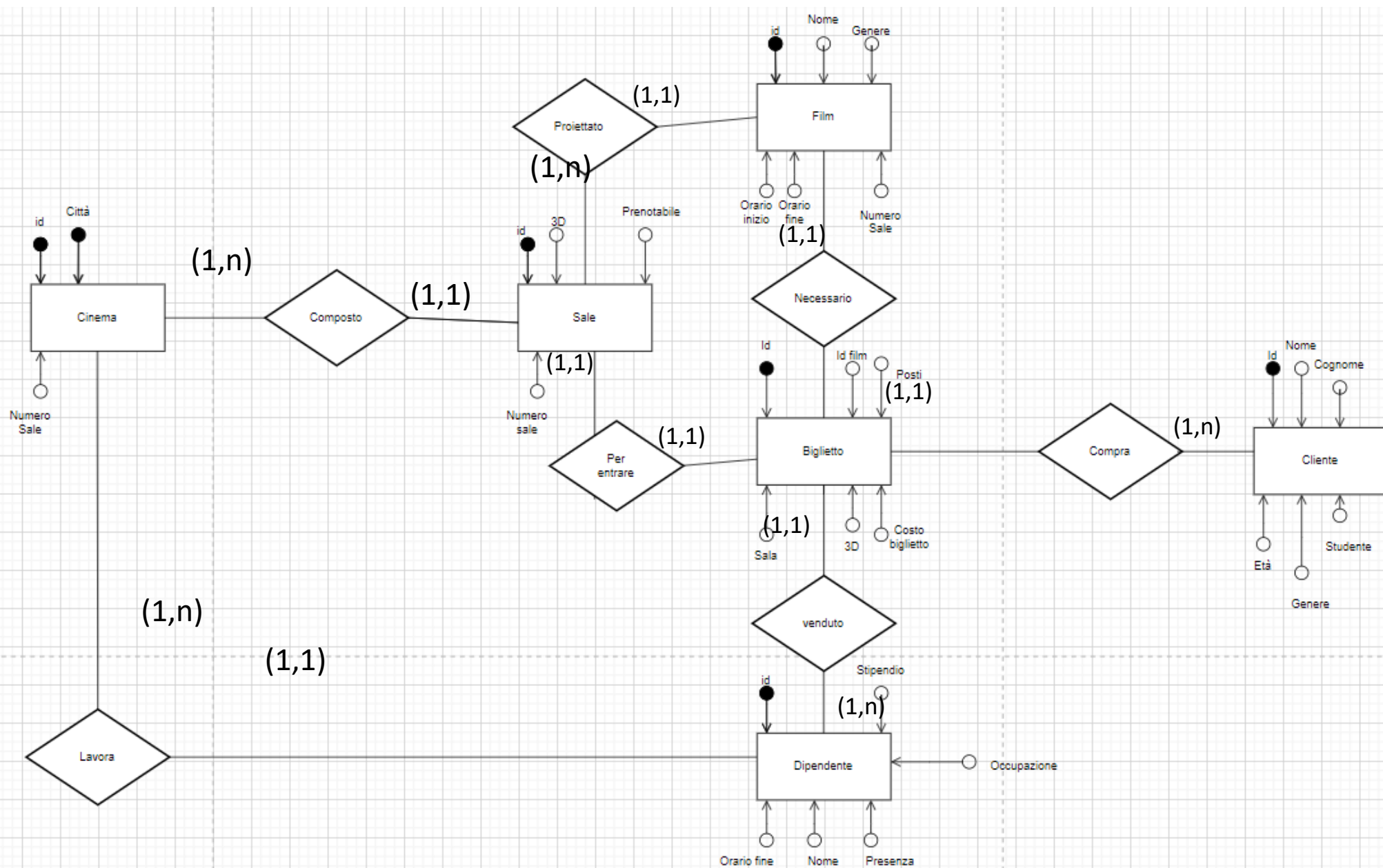
$$503/X \geq 523$$

$$X \leq 503 * 523$$

Dato che X rappresenta il numero di biglietti venduti in un mese, che è minore del secondo membro e quindi la disequazione è sempre verificata.

In definitiva conviene mantenere la ridondanza e quindi non si necessita di rimodellare lo schema ER.

Ristrutturazione dello schema ER



Implementazione database in Sql

```
CREATE TABLE IF NOT EXISTS 'database'.'Cinema'{  
    'id' INT NOT NULL,  
    'città' VARCHAR(15) NULL DEFAULT NULL,  
    'Numero Sale' INT NOT NULL,  
    PRIMARY KEY('Id'),  
    PRIMARY KEY('città'),  
    CONSTRAINT 'città'  
    ON DELETE NO ACTION  
    ON UPDATE NO ACTION  
}  
ENGINE= InnoDB  
DEFAULT CHARACTER SET= UTF8MB4;
```

```
CREATE TABLE IF NOT EXISTS 'database'.'Sala'{S  
    'id sala' INT NOT NULL,  
    'Aperta e/o Chiusa' VARCHAR(15) NULL DEFAULT NULL,  
    'Numero Posti' INT NOT NULL,  
    'Prenotabile' VARCHAR(13) NULL DEFAULT NULL,  
    PRIMARY KEY('Id sala'),  
    FOREIGN KEY('id sala')  
    REFERENCES 'database'.'Film'  
    ON DELETE NO ACTION  
    ON UPDATE NO ACTION  
}  
ENGINE= InnoDB  
DEFAULT CHARACTER SET= UTF8MB4;
```

```
CREATE TABLE IF NOT EXISTS 'database'.'film'{  
    'id film' INT NOT NULL,  
    'sala' INT NOT NULL,  
    'nome' VARCHAR(15) NULL DEFAULT NULL,  
    'Orario inizio' VARCHAR(15) NULL DEFAULT NULL,  
    'Orario fine' VARCHAR(15) NULL DEFAULT NULL,  
    'Data di uscita' VARCHAR(15) NULL DEFAULT NULL,  
    'Durata' VARCHAR(15) NULL DEFAULT NULL,  
    'Data di ritiro' VARCHAR(15) NULL DEFAULT NULL,  
    'genere' VARCHAR(13) NULL DEFAULT NULL,  
    PRIMARY KEY('Id film'),  
    FOREIGN KEY('id film')  
    REFERENCES 'database'.'Biglietto'  
    ON DELETE NO ACTION  
    ON UPDATE NO ACTION  
}  
ENGINE= InnoDB  
DEFAULT CHARACTER SET= UTF8MB4;
```


Implementazione database in Sql

```
CREATE TABLE IF NOT EXISTS 'database'.'Biglietto'{
  'id biglietto' INT NOT NULL,
  'id del film' INT NOT NULL,
  '3D' VARCHAR(15) NULL DEFAULT NULL,
  'Posti' VARCHAR(15) NULL DEFAULT NULL,
  'Costo Biglietto' INT NOT NULL,
  'Nome' VARCHAR(15) NULL DEFAULT NULL,

  PRIMARY KEY('Id biglietto'),
  FOREIGN KEY('id del film')
  REFERENCES 'database'.'Biglietto'
  FOREIGN KEY('nome')
  REFERENCES 'database'.'Biglietto'
  ON DELETE NO ACTION
  ON UPDATE NO ACTION
}

ENGINE= InnoDB
DEFAULT CHARACTER SET= UTF8MB4;
```

```
CREATE TABLE IF NOT EXISTS 'database'.'Dipendenti'{
  'id ' INT NOT NULL,
  'id cinema' INT NOT NULL,
  'Nome' VARCHAR(15) NULL DEFAULT NULL,
  'Cognome' VARCHAR(15) NULL DEFAULT NULL,
  'Occupazione' VARCHAR(13) NULL DEFAULT NULL,
  'Stipendio ' INT NOT NULL,
  'Orario Inizio' VARCHAR(13) NULL DEFAULT NULL,
  'Orario Fine' VARCHAR(13) NULL DEFAULT NULL,
  'Presenza' VARCHAR(13) NULL DEFAULT NULL,
  PRIMARY KEY('Id '),
  FOREIGN KEY('id cinema')
  REFERENCES 'database'.'Cinema'
  ON DELETE NO ACTION
  ON UPDATE NO ACTION
}

ENGINE= InnoDB
DEFAULT CHARACTER SET= UTF8MB4;
```

```
CREATE TABLE IF NOT EXISTS 'database'.'Clienti'{
  'id biglietto' INT NOT NULL,
  'id ' INT NOT NULL,
  'Nome' VARCHAR(15) NULL DEFAULT NULL,
  'Cognome' VARCHAR(15) NULL DEFAULT NULL,
  'età' INT NOT NULL,
  'Genere' VARCHAR(15) NULL DEFAULT NULL,

  PRIMARY KEY('Id '),
  FOREIGN KEY('id biglietto')
  REFERENCES 'database'.'Biglietto'
  FOREIGN KEY('Genere')
  REFERENCES 'database'.'film'
  ON DELETE NO ACTION
  ON UPDATE NO ACTION
}

ENGINE= InnoDB
DEFAULT CHARACTER SET= UTF8MB4;
```

Varie Interrogazioni in Sql

```
SELECT c, b
FROM Clienti AS c
JOIN Biglietto AS b ON c.id_biglietto = b.id_biglietto
WHERE c.id = <id_cliente>;
```

<->Ottenere i dettagli di un cliente ed il relativo biglietto

<->Ottenere i dettagli di un film ed la relativo sala

```
SELECT f, s
FROM film AS f
JOIN Sala AS s ON f.sala = s.id_sala
WHERE f.id_film = <id_film>;
```

```
SELECT c.id AS id_cinema, c.nome AS nome_cinema, d.nome AS nome_dirigente
FROM Cinema AS c
JOIN Dipendenti AS d ON c.id = d.id_cinema
WHERE d.occupazione = 'Dirigente';
```

<->Interrogazione per ottenere quali sono i Dirigenti per ogni cinema

Trigger Sql

Si riportano due esempi di trigger inerenti al database creato; il primo trigger si attiverà per dare uno sconto ad ogni cliente con età inferiore a 15 o superiore a 69 (Figura A); mentre il secondo trigger si attiverà per aumentare lo stipendio di un dipendente nel caso in cui la Presenza superi 20 (Figura B)

```
CREATE TRIGGER cliente_insert_trigger
BEFORE INSERT ON Biglietto
FOR EACH ROW
BEGIN
    DECLARE sconto INT;
    IF (SELECT età FROM Clienti WHERE id = NEW.id_cliente) < 15 OR (SELECT età FROM Clienti WHERE id = NEW.id_cliente) > 68 THEN
        SET sconto = 4;
    ELSE
        SET sconto = 0;
    END IF;
    SET NEW.Costo_Biglietto = NEW.Costo_Biglietto - sconto;
END
```

A

```
CREATE TRIGGER dipendente_presenza_trigger
BEFORE UPDATE ON Dipendenti
FOR EACH ROW
BEGIN
    IF NEW.Presenza = 10 THEN
        SET NEW.Stipendio = NEW.Stipendio + 20;
    END IF;
END
```

B

Implementazione DataBase in XML

```
<database>
  <table name="Cinema">
    <column name="id" type="INT" nullable="false" />
    <column name="città" type="VARCHAR(15)" nullable="true" default="NULL" />
    <column name="Numero_Sale" type="INT" nullable="false" />
  </table>

  <table name="Sala">
    <column name="id_sala" type="INT" nullable="false" />
    <column name="Aperta_e/o_Chiusa" type="VARCHAR(15)" nullable="true" default="NULL" />
    <column name="Numero_Posti" type="INT" nullable="false" />
    <column name="Prenotabile" type="VARCHAR(13)" nullable="true" default="NULL" />
  </table>

  <table name="film">
    <column name="id_film" type="INT" nullable="false" />
    <column name="sala" type="INT" nullable="false" />
    <column name="nome" type="VARCHAR(15)" nullable="true" default="NULL" />
    <column name="Orario_inizio" type="VARCHAR(15)" nullable="true" default="NULL" />
    <column name="Orario_fine" type="VARCHAR(15)" nullable="true" default="NULL" />
    <column name="Data_di_uscita" type="VARCHAR(15)" nullable="true" default="NULL" />
    <column name="Durata" type="VARCHAR(15)" nullable="true" default="NULL" />
    <column name="Data_di_ritiro" type="VARCHAR(15)" nullable="true" default="NULL" />
    <column name="genere" type="VARCHAR(13)" nullable="true" default="NULL" />
  </table>
```

Implementazione DataBase in XML

```
<table name="Biglietto">
  <column name="id_biglietto" type="INT" nullable="false" />
  <column name="id_del_film" type="INT" nullable="false" />
  <column name="id_cliente" type="INT" nullable="false" />
  <column name="3D" type="VARCHAR(15)" nullable="true" default="NULL" />
  <column name="Posti" type="VARCHAR(15)" nullable="true" default="NULL" />
  <column name="Costo_Biglietto" type="INT" nullable="false" />
  <column name="Nome" type="VARCHAR(15)" nullable="true" default="NULL" />
</table>

<table name="Clienti">
  <column name="id_biglietto" type="INT" nullable="false" />
  <column name="id" type="INT" nullable="false" />
  <column name="Nome" type="VARCHAR(15)" nullable="true" default="NULL" />
  <column name="Cognome" type="VARCHAR(15)" nullable="true" default="NULL" />
  <column name="età" type="INT" nullable="false" />
  <column name="Genere" type="VARCHAR(15)" nullable="true" default="NULL" />
</table>

<table name="Dipendenti">
  <column name="id" type="INT" nullable="false" />
  <column name="id_cinema" type="INT" nullable="false" />
  <column name="Nome" type="VARCHAR(15)" nullable="true" default="NULL" />
  <column name="Cognome" type="VARCHAR(15)" nullable="true" default="NULL" />
  <column name="Occupazione" type="VARCHAR(13)" nullable="true" default="NULL" />
  <column name="Stipendio" type="INT" nullable="false" />
  <column name="Orario_Inizio" type="VARCHAR(13)" nullable="true" default="NULL" />
  <column name="Orario_Fine" type="VARCHAR(13)" nullable="true" default="NULL" />
  <column name="Presenza" type="VARCHAR(13)" nullable="true" default="NULL" />
</table>
</database>
```