



Università
di Catania

2022/2023

PROGETTO DI BASI DI DATI

Database relativo alla gestione
di un cinema

Stefan Andrei Daniel

Specifiche sui dati

Si vuole progettare una base di dati per la gestione di un cinema che tenga conto delle seguenti informazioni:

Le sale di proiezione (5) sono identificate da un codice univoco, dal film che deve essere proiettato, dal corrispondente numero di spettatori e dal numero di posti disponibili (100). Le sale possono essere occupate da nuovi spettatori esclusivamente se ci sono posti disponibili e quindi che la capienza massima non sia raggiunta. Per ogni sala deve essere possibile leggere il numero di spettatori.

La cassa è identificata da un codice identificativo e per ognuna di esse è necessario rappresentare l'incasso in un determinato giorno. I prodotti venduti possono essere: biglietto, che servirà per poter visionare il film, oppure snack. Per ognuno dei prodotti venduti si decide di conservare un codice identificativo e per ogni vendita il relativo codice e la data. In media si effettuano circa 250 vendite al giorno.

Inoltre, il biglietto conserva anche i seguenti dati: prezzo e una sala alla quale fa riferimento. I biglietti possono essere acquistati esclusivamente se ci sono i posti disponibili per una determinata proiezione visto che non è consentita la visione di un film stando in piedi.

Per gli snack è necessario identificare anche il nome che può essere, per esempio, "Pop-Corn", "Coca-Cola", "Patatine", e il relativo prezzo. È necessario consentire la modifica del prezzo dello snack e tale modifica dovrà avvenire due volte al giorno.

L'operatore è rappresentato dal suo codice fiscale, nome, cognome, data di nascita e ognuno di loro lavora in determinate orari di lavoro in una determinata cassa. Al fine di migliorare la gestione della base di dati si è deciso di rappresentare i vari turni con un codice univoco, una fascia oraria e il relativo dipendente.

Per un film si vuole rappresentare il codice identificativo, titolo e genere. I generi proiettati nel cinema in questione sono solamente "Comico", "Horror" e "Romantico".

Per quanto riguarda il feedback, esso rappresenta la recensione anonima di un cliente e viene rappresentato univocamente da un id, il film al quale fa riferimento e un livello che viene memorizzato con un valore numerico da 1 a 5.

Vista la possibilità di cambi di turno fra i vari operatori del cinema, deve essere possibile inserire un nuovo turno di lavoro (2 volte al giorno) e, per facilitare il calcolo dei guadagni, la base di dati dovrebbe dare la possibilità di accedere facilmente al totale ricavato in un determinato giorno (1 volta al giorno).

Visto che i film possono essere scelti in base alla loro recensione effettuata da parte di terzi, è necessario poter visualizzare le informazioni relative alla media dei feedback per un film (1 volta al giorno) e avere la possibilità di aggiungere una nuova valutazione (100 volte al giorno).

Glossario dei termini

TERMINE	DESCRIZIONE	SINONIMO	TERMINI COLLEGATI
Cassa	Luogo adibito alla vendita		Vendita, Operatore
Vendita	Beni che sono acquistati alla cassa		Cassa, Biglietto, Snack
Sala	Luogo dove viene proiettato il film		Film, Biglietto
Operatore	Persona che lavora alla cassa	Dipendente	Cassa, Turno
Feedback	Livello di soddisfazione del cliente	Valutazione, Recensione	Film
Turno	Periodo di lavoro di un operatore	Orario di lavoro	Operatore

Dati di carattere generale

Si vuole progettare una base di dati per la gestione di un cinema relativo alle sue vendite e ai suoi operatori.

Dati sulle sale

Per ogni sala si vuole rappresentare il codice identificativo, il film in proiezione, dal numero di spettatori presenti per la visione di un determinato film e dal numero di posti ancora disponibili.

Dati sulla cassa

Per la cassa si decide di memorizzare l'id della cassa, un numero che rappresenta il totale ricavato in un giorno e la relativa data.

Dati sul biglietto

Il biglietto è identificato da id, prezzo e la relativa sala. Esso può essere acquistato solo se ci sono posti disponibili nella sala oppure se la capienza della sala lo permette.

Dati sugli snack

Uno snack è identificato da un id, dal nome e dal relativo prezzo.

Dati sull'operatore

Un operatore è identificato da un codice che indica il codice fiscale, un nome, un cognome, una data di nascita. Ogni operatore, inoltre, lavora presso una cassa in vari turni di lavoro.

Dati sul film

Per ogni film si decide di memorizzare i seguenti dati: id del film, titolo e genere. In questo cinema vengono proiettati film di genere "Comico", "Horror" oppure "Romantico".

Dati sul feedback

Il feedback è memorizzato tenendo conto dell'id, il film di riferimento e la rispettiva valutazione che è rappresentato da un numero compreso fra 1 e 5. Molto spesso ne viene calcolata la media, il valore massimo e vengono aggiunte nuove feedback.

Dati sul turno

Un turno è rappresentato da una fascia oraria, un codice identificativo e l'operatore che è assegnato a tale turno. Nella base di dati si può sempre inserire un nuovo turno scegliendo fra i vari operatori.

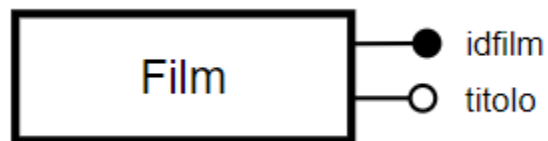
Dati sulla vendita

Per ogni vendita effettuata si vuole memorizzare un identificativo, la data della vendita, il relativo prodotto venduto e la relativa cassa che ha effettuato la vendita.

Strategia

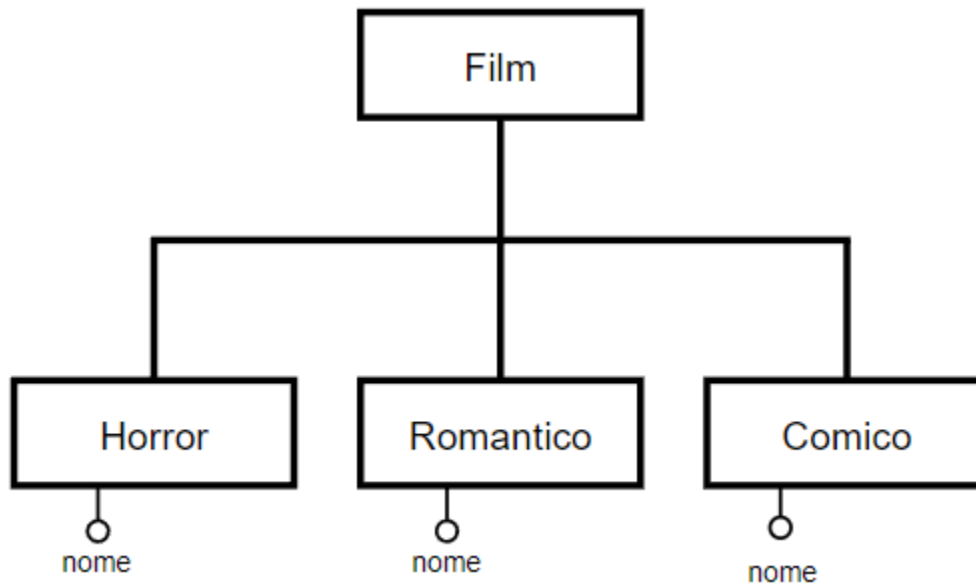
Per la base di dati in questione si sceglie una strategia bottom-up dove vengono individuate parti individuali del sistema specificate in dettaglio e poi vengono connesse tra loro in modo da formare componenti più complesse. Queste ultime sono a loro volta connesse fra di loro fino a realizzare uno schema completo.

PASSO 1



- Trasformazione T1: Dal concetto di film è stata individuata l'entità Film
- Trasformazione T4: Gli attributi dell'entità Film sono *idfilm*, *titolo*

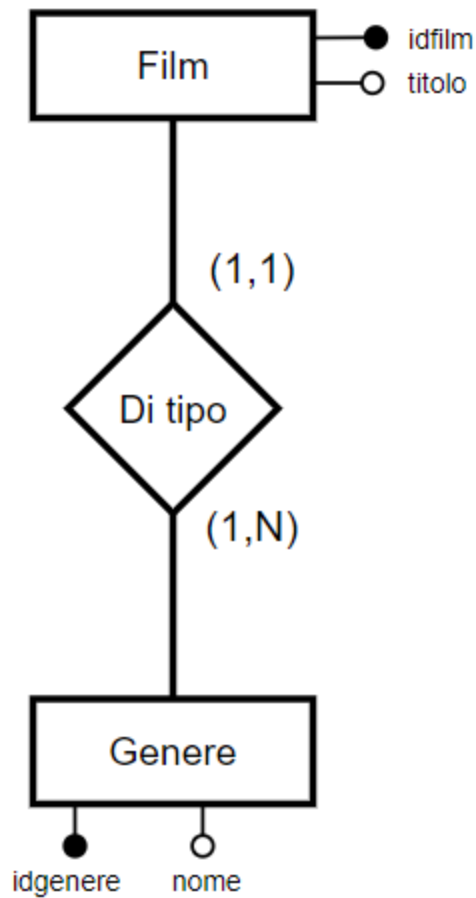
PASSO 2



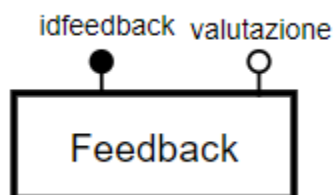
- Trasformazione T1: Dal concetto di horror è individuata l'entità Horror.
- Trasformazione T4: L'attributo dell'entità Horror è *nome*
- Trasformazione T1: Dal concetto di romantico è individuata l'entità Romantico.
- Trasformazione T4: L'attributo dell'entità Romantico è *nome*
- Trasformazione T1: Dal concetto di comico è individuata l'entità Comico.
- Trasformazione T4: L'attributo dell'entità Comico è *nome*
- Trasformazione T3: È stata individuata la generalizzazione fra l'entità Film e le entità figlie Horror, Romantico, Comico

Visto che la generalizzazione è di tipo totale ed esclusiva, si decide di creare l'entità Genere per identificare il genere di Film.

- Trasformazione T4: Gli attributi dell'entità Genere sono *idgenere*, *nome*

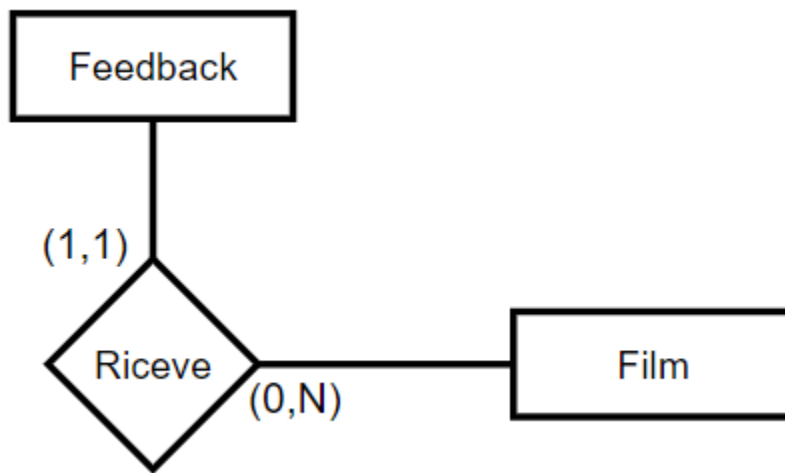


PASSO 3



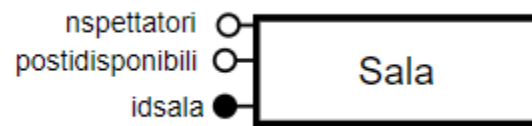
- Trasformazione T1: Dal concetto di feedback è stata individuata l'entità Feedback
- Trasformazione T4: Gli attributi dell'entità Feedback sono *idfeedback*, *valutazione*

PASSO 4



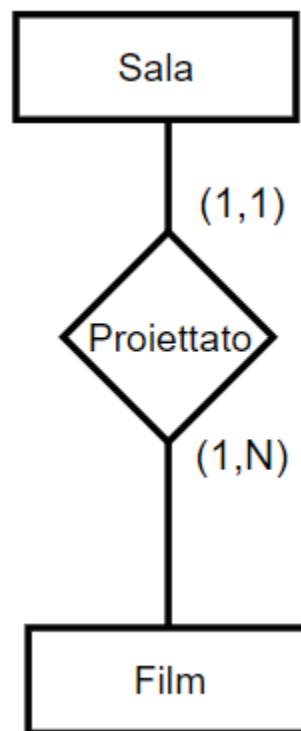
- Trasformazione T2: È stata individuata la relazione fra Feedback e Film

PASSO 5



- Trasformazione T1: Dal concetto di sala è stata individuata l'entità Sala
- Trasformazione T4: Gli attributi dell'entità Sala sono *idsala*, *nspettatori*, *postidisponibili*

PASSO 6



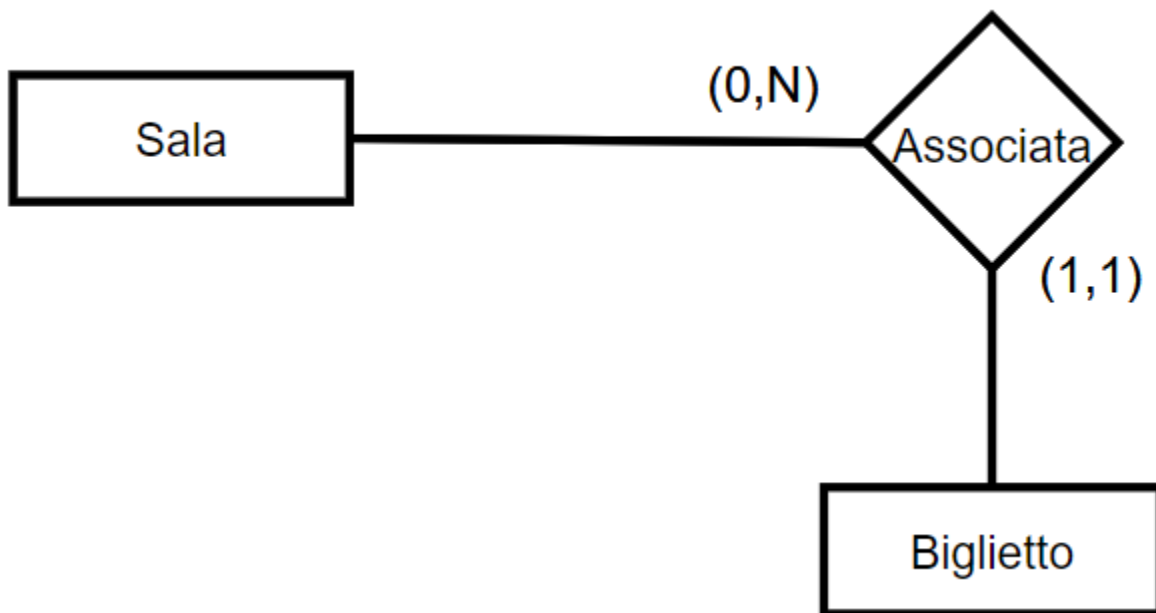
- Trasformazione T2: È stata individuata la relazione fra Film e Sala

PASSO 7



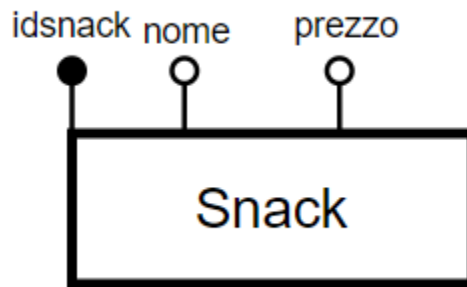
- Trasformazione T1: Dal concetto di biglietto è stata individuata l'entità Biglietto
- Trasformazione T4: Gli attributi dell'entità Biglietto sono *idbiglietto*, *prezzo*

PASSO 8



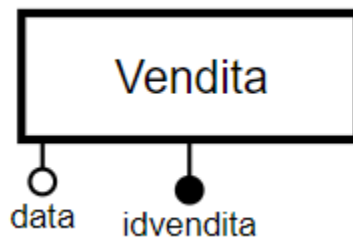
- Trasformazione T2: È stata individuata la relazione fra Sala e Biglietto

PASSO 9



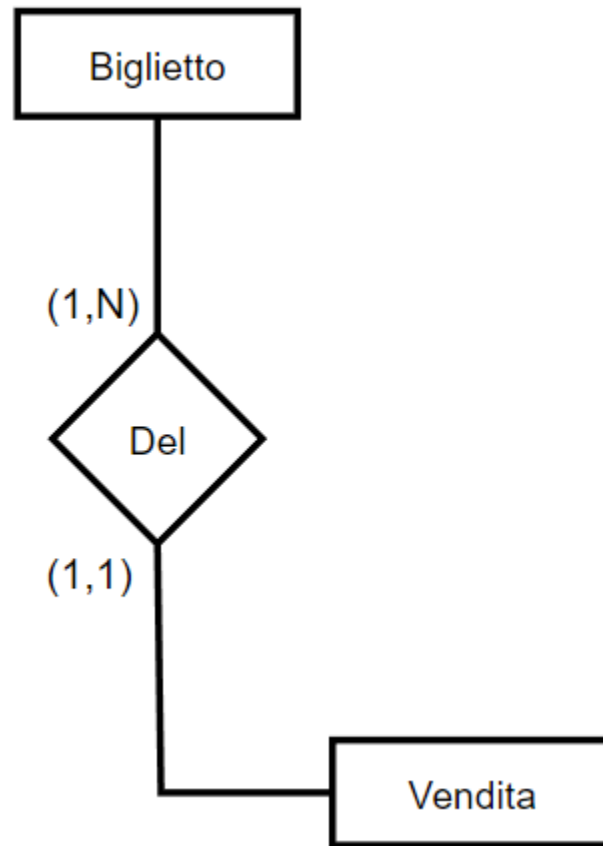
- Trasformazione T1: Dal concetto di snack è stata individuata l'entità Snack
- Trasformazione T4: Gli attributi dell'entità Snack sono *idsnack*, *nome*, *prezzo*

PASSO 10



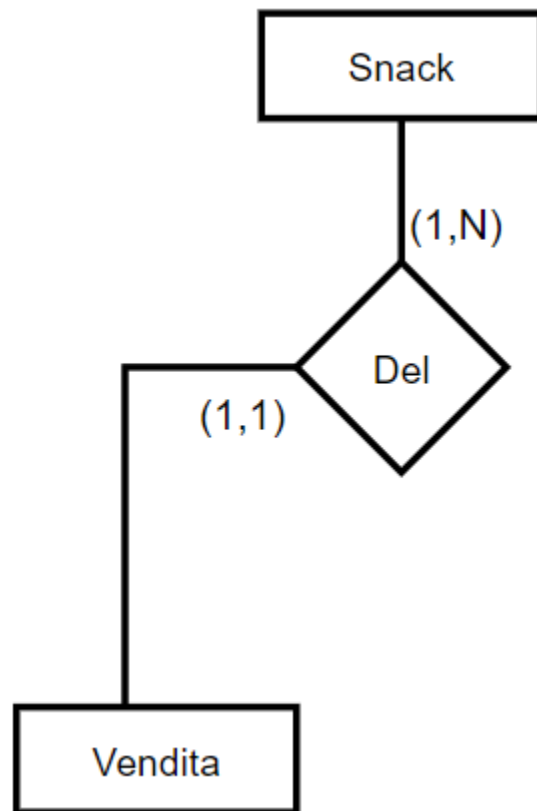
- Trasformazione T1: Dal concetto di vendita di un prodotto è stata individuata l'entità Vendita
- Trasformazione T4: Gli attributi dell'entità Vendita sono *data*, *idvendita*

PASSO 11.0



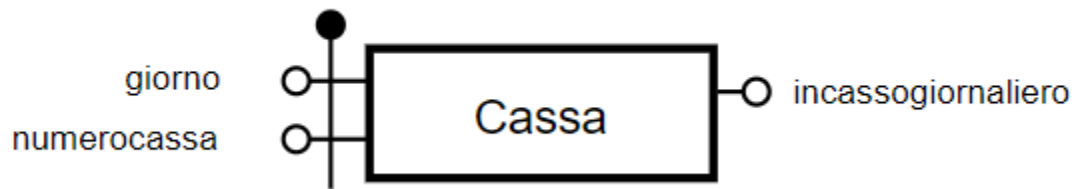
- Trasformazione T2: È stata individuata l'associazione fra l'entità Biglietto e l'entità Vendita

PASSO 11.1



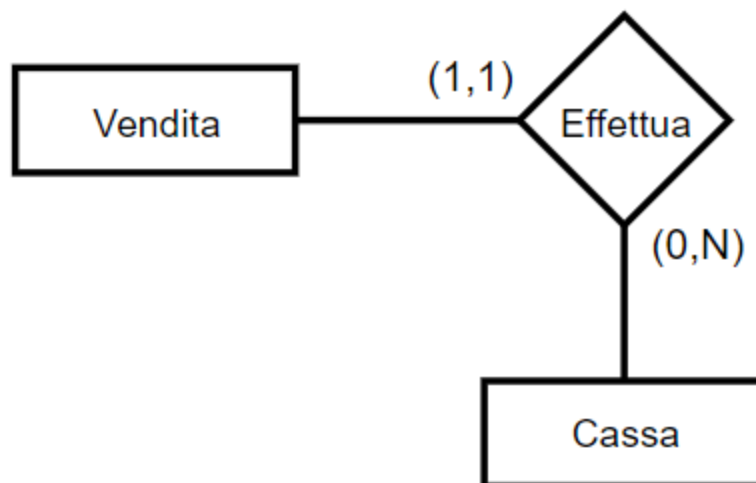
- Trasformazione T2: È stata individuata l'associazione fra l'entità **Snack** e l'entità **Vendita**

PASSO 12



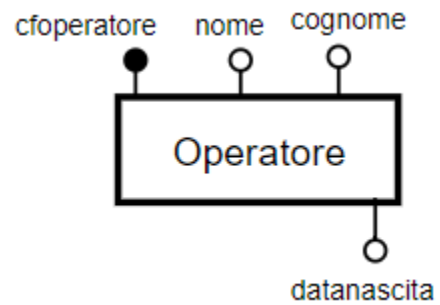
- Trasformazione T1: Dal concetto di cassa è stata individuata l'entità Cassa.
- Trasformazione T4: Gli attributi dell'entità Cassa sono *giorno*, *numerocassa*, *incassogiornaliero*

PASSO 13



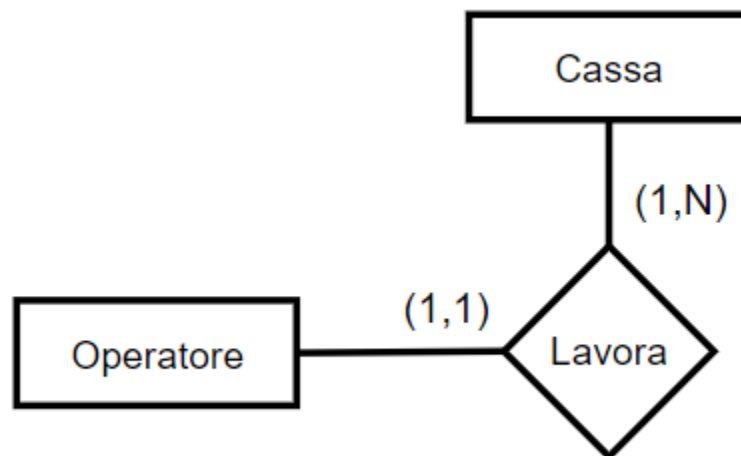
- Trasformazione T2: È stata individuata l'associazione fra Vendita e Cassa

PASSO 14



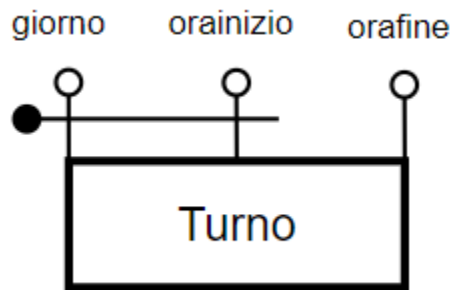
- Trasformazione T1: Dal concetto di operatore è stata individuata l'entità Operatore
- Trasformazione T4: Gli attributi dell'entità Operatore sono *cfoperatore*, *datanascita*, *nome*, *cognome*

PASSO 15



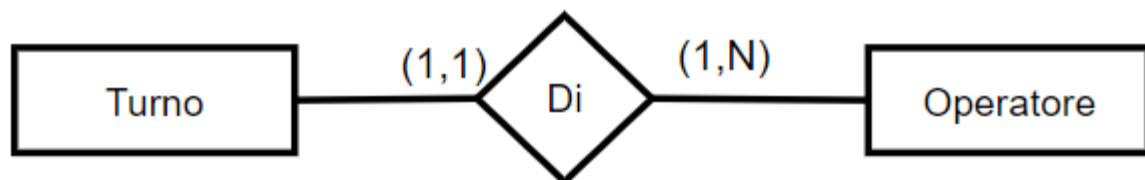
- Trasformazione T2: È stata individuata l'associazione fra Operatore e Cassa

PASSO 16



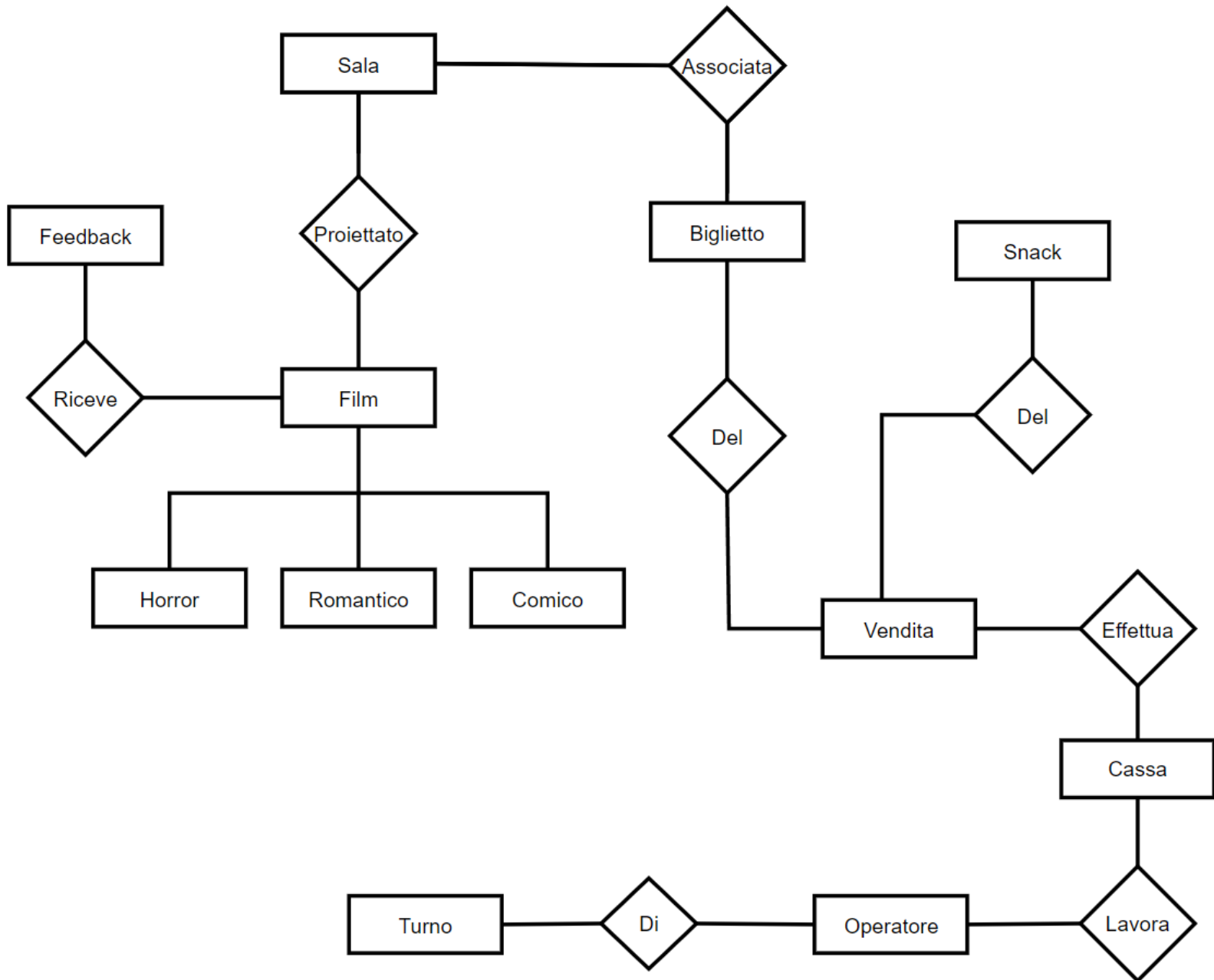
- Trasformazione T1: Dal concetto di turno è stata individuata l'entità **Turno**
- Trasformazione T4: Gli attributi dell'entità **Turno** sono *giorno*, *orainizio*, *orafine*

PASSO 17



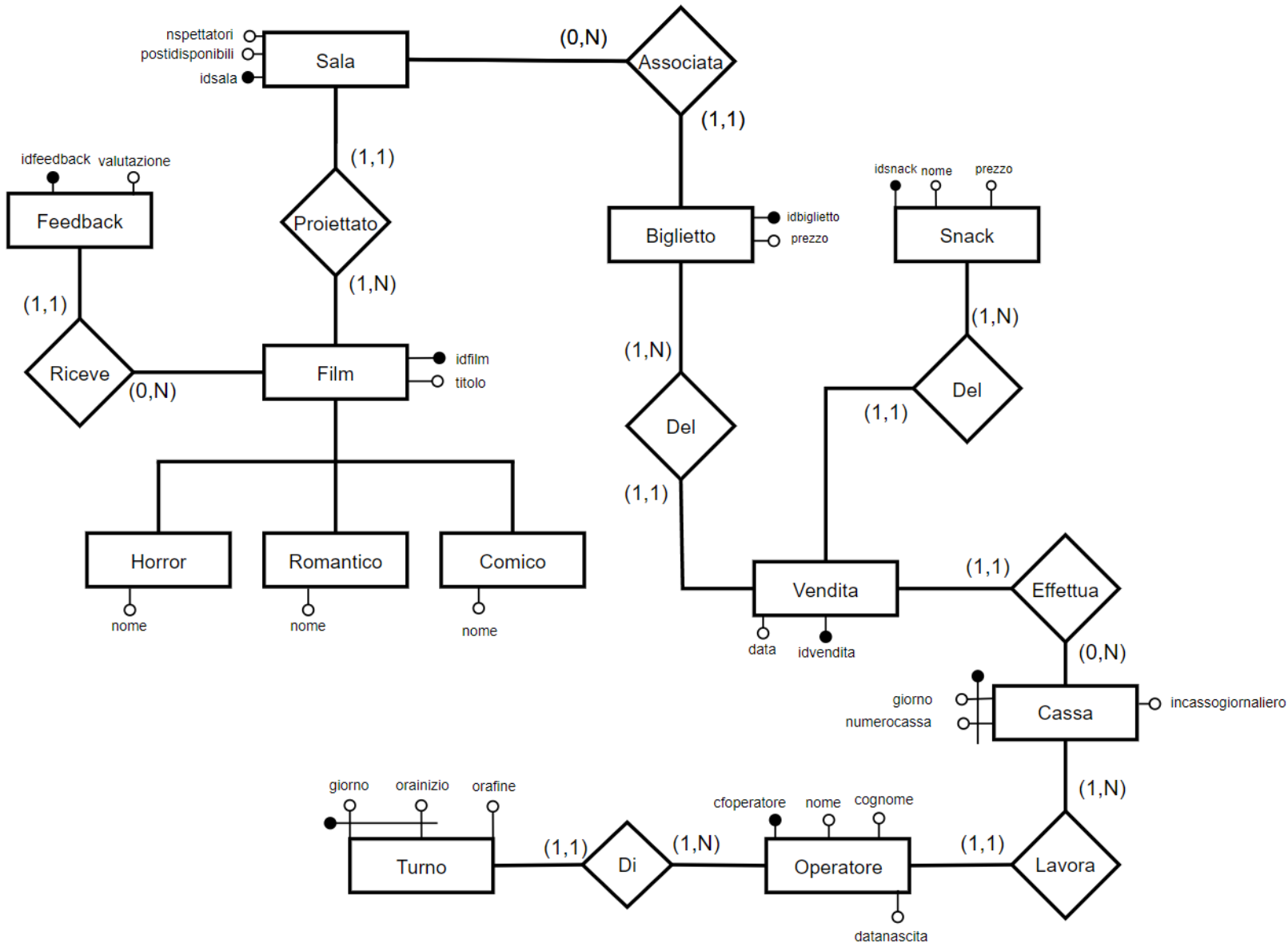
- Trasformazione T2: È stata individuata la relazione fra Operatore e Turno

SCHEMA SCHELETRO / 1



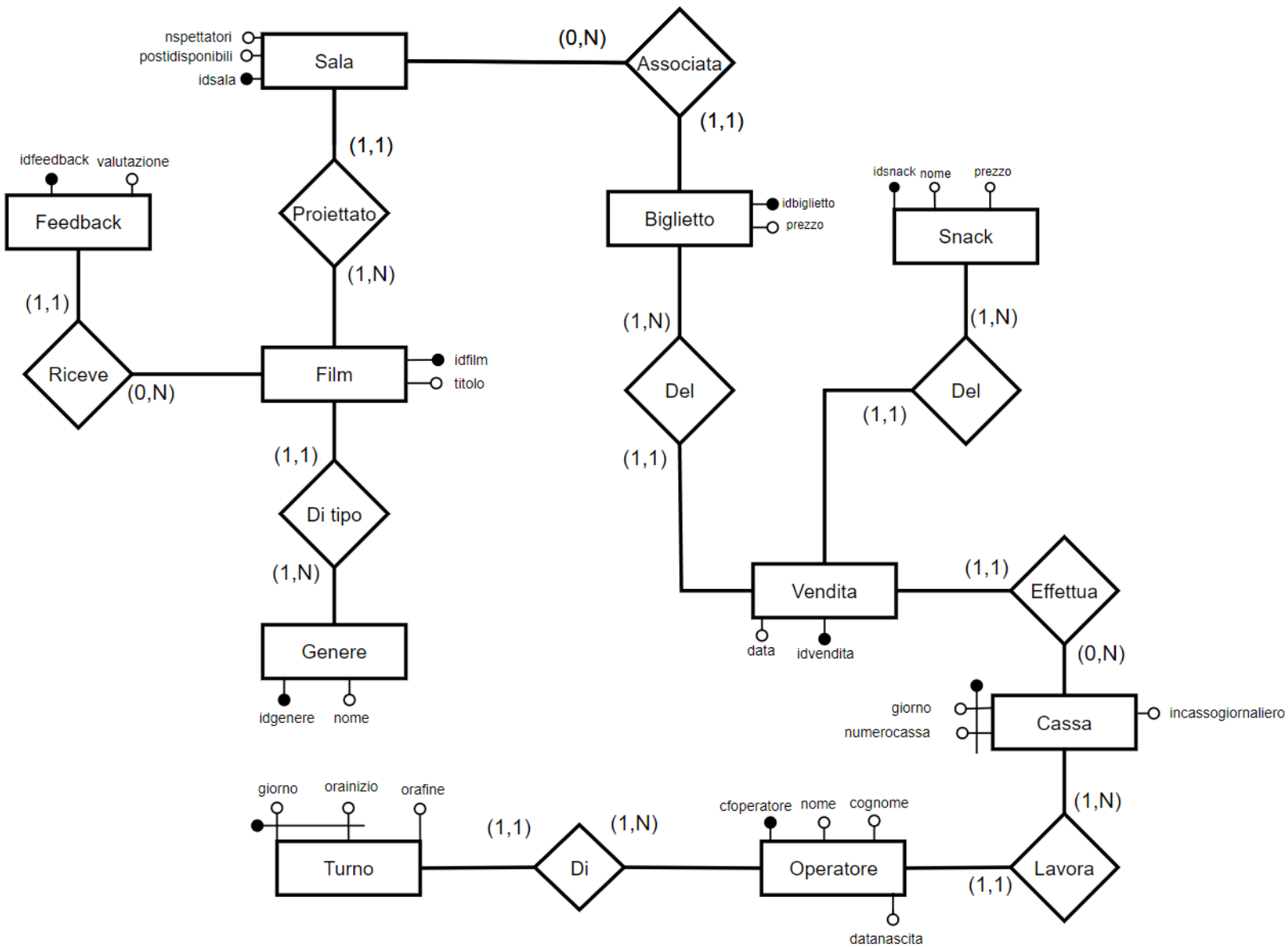
SCHEMA SCHELETRO / 2

Vengono aggiunti gli attributi e le cardinalità.



SCHEMA FINALE

Vengono eliminate le gerarchie



DIZIONARIO DEI DATI

ENTITA'	ATTRIBUTI	IDENTIFICATORE
Feedback	Idfeedback, valutazione, idfilm	idfeedback
Genere	idgenere, nome	idgenere
Film	idfilm, titolo, genere	idfilm
Sala	idsala, idfilm, nspettatori, postidisponibili	idsala
Biglietto	idbiglietto, prezzo, idsala	idbiglietto
Snack	idsnack, nome, prezzo	idsnack
Vendita	idvendita, idcassa, idbiglietto, idsnack, data	idvendita
Cassa	numerocassa, incassogiornaliero, giorno	numerocassa, giorno
Operatore	cfoperatore, datanascita, nome, cognome	cfoperatore
Turno	giorno, orainizio, orafine, cfoperatore	giorno, orainizio

RELAZIONE	ENTITÀ PARTECIPANTI
Riceve	Feedback e Film
Di tipo	Film e Genere
Proiettato	Sala e Film
Associata	Biglietto e Sala
Del	Vendita e Biglietto
Del	Vendita e Snack
Effettua	Vendita e Cassa
Lavora	Cassa e Operatore
Di	Operatore e Turno

Vincoli non esprimibili dallo schema E/R

- Non è possibile acquistare un biglietto se non ci sono posti disponibili in quella sala.
- Quando si vende un prodotto Snack o Biglietto, tale prezzo va sommato all'incasso giornaliero per quella determinata cassa.
- Quando viene venduto un biglietto, il numero di posti disponibili per quella determinata sala viene decrementato di 1 e il numero di spettatori viene incrementato di 1.
- Prima di aggiungere una nuova vendita è necessario che venga registrata la data per quella determinata cassa con il relativo incasso giornaliero

Specifiche sulle operazioni

Per la gestione del cinema è previsto l'uso di alcune operazioni, di cui vengono riportate le descrizioni e il relativo carico previsto per ognuna di esse:

- **O1:** Inserimento di un nuovo turno di lavoro (2 volte/giorno)
- **O2:** Modifica del prezzo di uno snack (2 volte/giorno)
- **O3:** Lettura dell'incasso totale giornaliero (1 volta/giorno)
- **O4:** Lettura della media di feedback per un film (1 volta/giorno)
- **O5:** Inserimento di un nuovo feedback (100 volte/giorno)
- **O6:** Lettura del numero di posti in una sala (150 volte/giorno)

ANALISI DELE RIDONDANZE

Gli attributi che seguono risultano ridondanti:

1. **incassogiornaliero** su Cassa che indica il totale ricavato in un determinato giorno.
2. **nspettatori** in Sala che indica il numero di spettatori in una sala

Le operazioni interessate sono:

1. **O3**: Lettura dell'incasso totale giornaliero (1 volta/giorno)
2. **O6**: Lettura del numero di posti in una sala (150 volte/giorno)

Tabella dei volumi

CONCETTO	TIPO	VOLUME
Sala	E	5
Vendita	E	250
Biglietto	E	50

Tabella delle operazioni

CONCETTO	TIPO	FREQUENZA
O1	I	2 volte / giorno
O2	I	2 volte / giorno
O3	I	1 volta / giorno
O4	I	1 volta / giorno
O5	I	100 volte / giorno
O6	I	150 volte / giorno

Analisi dell'attributo incassogiornaliero

CON RIDONDANZA

Tavola degli accessi dell'operazione O3

CONCETTO	COSTRUTTO	ACCESSI	TIPO
Cassa	Entità	1	L

Visto che l'operazione **O3** viene fatta 1 volta al giorno, allora il risultato di tale analisi è: $1 / \text{giorno} * 1 \text{ L} = 1 \text{ L} / \text{giorno}$

SENZA RIDONDANZA

Tavola degli accessi dell'operazione O3

CONCETTO	COSTRUTTO	ACCESSI	TIPO
Vendita	Entità	250	L

Visto che l'operazione **O3** viene fatta 1 volta al giorno e visto che in media ci sono 250 vendite, allora per trovare l'incasso giornaliero si devono sommare tutte le vendite per un determinato giorno. Il risultato di tale analisi è:

$1 / \text{giorno} * 250 \text{ L} = 250 \text{ L} / \text{giorno}$

In conclusione, **CONVIENE MANTENERE LA RIDONDANZA** *incassogiornaliero*

Analisi dell'attributo nspettatori

CON RIDONDANZA

Tavola degli accessi dell'operazione O6

CONCETTO	COSTRUTTO	ACCESSI	TIPO
Sala	E	1	L

Visto che l'operazione **O6** viene fatta 150 volte al giorno per ogni sala, allora il risultato di tale analisi è: $150 / \text{giorno} * 1 \text{ L} = 150 \text{ L} / \text{giorno}$

SENZA RIDONDANZA

Tavola degli accessi dell'operazione O6

Bisogna trovare il numero di biglietti mediamente venduti. Tale valore si trova applicando la seguente operazione:

$$\text{Vendite} / \text{Sale} = 250 / 5 = 50$$

CONCETTO	COSTRUTTO	ACCESSI	TIPO
Sala	E	1	L
Associata	R	1	L
Biglietto	E	1	L
Del	R	1	L
Vendita	E	50	L

Visto che l'operazione **O6** viene fatta 150 volte al giorno, allora il risultato di tale analisi è: $150 / \text{giorno} * 54 \text{ L} = 8100 \text{ L} / \text{giorno}$

In conclusione, **CONVIENE MANTENERE LA RIDONDANZA nspettatori**

CREAZIONE DEL DATABASE E TABELLE

```
DROP DATABASE IF EXISTS Cinema;  
CREATE DATABASE IF NOT EXISTS `Cinema`;  
USE `Cinema`;
```

```
CREATE TABLE IF NOT EXISTS Genere(  
    idgenere SMALLINT PRIMARY KEY,  
    nome TEXT NOT NULL CHECK (nome IN ('Horror', 'Romantico', 'Comico'))  
);
```

```
CREATE TABLE IF NOT EXISTS Film(  
    idfilm SMALLINT,  
    titolo TEXT,  
    genere SMALLINT NOT NULL,  
    PRIMARY KEY (idfilm),  
    FOREIGN KEY (genere) REFERENCES Genere(idgenere)  
);
```

```
CREATE TABLE IF NOT EXISTS Feedback(  
    idfeedback SMALLINT PRIMARY KEY,  
    valutazione SMALLINT NOT NULL DEFAULT 1 CHECK (valutazione >0 AND valutazione <6),  
    idfilm SMALLINT,  
    FOREIGN KEY (idfilm) REFERENCES Film(idfilm)  
);
```

```
CREATE TABLE IF NOT EXISTS Sala(  
    idsala SMALLINT,  
    idfilm SMALLINT,  
    nspettatori SMALLINT DEFAULT 0 CHECK (nspettatori <= 100),  
    postidisponibili SMALLINT DEFAULT 100 CHECK (postidisponibili >=0),  
    PRIMARY KEY (idsala),  
    FOREIGN KEY (idfilm) REFERENCES Film(idfilm)  
);
```

```
CREATE TABLE IF NOT EXISTS Biglietto(  
    idbiglietto SMALLINT,  
    prezzo FLOAT NOT NULL,  
    idsala SMALLINT NOT NULL,  
    PRIMARY KEY (idbiglietto),  
    FOREIGN KEY (idsala) REFERENCES Sala(idsala)  
);
```

```
CREATE TABLE IF NOT EXISTS Snack(  
    idsnack SMALLINT PRIMARY KEY,  
    nome VARCHAR(30) NOT NULL,  
    prezzo FLOAT NOT NULL  
);
```

```
CREATE TABLE IF NOT EXISTS Cassa (  
    numerocassa SMALLINT,  
    incassogiornaliero FLOAT,  
    giorno DATE,  
    PRIMARY KEY (numerocassa, giorno)  
);
```

```
CREATE TABLE IF NOT EXISTS Vendita(  
    idvendita SMALLINT AUTO_INCREMENT,  
    idcassa SMALLINT,  
    idbiglietto SMALLINT,  
    idsnack SMALLINT,  
    data DATE,  
    PRIMARY KEY (idvendita),  
    FOREIGN KEY (idsnack) REFERENCES Snack(idsnack),  
    FOREIGN KEY (idbiglietto) REFERENCES Biglietto(idbiglietto),  
    FOREIGN KEY (idcassa) REFERENCES Cassa(numerocassa)  
);
```

CREATE TABLE IF NOT EXISTS Operatore (
 cfoperatore VARCHAR(16) NOT NULL PRIMARY KEY,
 datanascita DATE NOT NULL,
 nome VARCHAR(30) NOT NULL,
 cognome VARCHAR(30) NOT NULL,
 idcassa SMALLINT,
 FOREIGN KEY (idcassa) REFERENCES Cassa(numerocassa) ON DELETE SET NULL
);

CREATE TABLE IF NOT EXISTS Turno (
 giorno DATE NOT NULL,
 orainizio TIME NOT NULL,
 orafine TIME NOT NULL,
 cfoperatore VARCHAR(16),
 PRIMARY KEY (giorno,orainizio,orafine),
 FOREIGN KEY (cfoperatore) REFERENCES Operatore(cfoperatore)
);

TRIGGER con esempio

Aggiornamento dell'incasso giornaliero

Ogni volta che si inserisce una nuova vendita per è necessario aggiornare, di conseguenza, l'incasso giornaliero. Ciò dipende dal tipo di prodotto acquistato e dalla cassa che effettua la vendita.

```
DELIMITER $$
CREATE TRIGGER aggiorna_incasso
AFTER INSERT ON Vendita
FOR EACH ROW
BEGIN
    DECLARE snack_prezzo FLOAT;
    DECLARE biglietto_prezzo FLOAT;

    SET snack_prezzo = (SELECT prezzo
                        FROM Snack s
                        WHERE s.idsnack = NEW.idsnack);

    SET biglietto_prezzo = (SELECT prezzo
                           FROM Biglietto b
                           WHERE b.idbiglietto = NEW.idbiglietto);

    UPDATE Cassa SET incassogiornaliero = incassogiornaliero +
        COALESCE(snack_prezzo, 0) + COALESCE(biglietto_prezzo, 0)
    WHERE numerocassa = NEW.idcassa AND giorno = NEW.data;
END $$
DELIMITER ;
```

Per far funzionare il trigger verranno usati dei dati *di esempio* presenti nelle tabelle rispettivamente elencate successivamente:

Biglietto

idbiglietto	prezzo	idsala
111	10.5	9
222	7	8
333	9	7
444	5	6

Snack

idsnack	nome	prezzo
1	Popcorn	4
2	Bibita	2.5
3	Nachos	5
4	Hot dog	3.5
5	Patatine	2

Cassa

numerocassa	incassogiornaliero	giorno
1	440	2023-03-01
1	400	2023-03-10
2	200	2023-03-09
3	500	2023-03-08
3	252	2023-03-11

Di seguito viene registrata una nuova vendita dello snack con id 5 effettuata presso la cassa 1 del giorno 2023-03-11:

```
INSERT INTO Vendita (idcassa, idbiglietto, idsnack, data) VALUES  
(3, NULL , 5 , '2023-03-11');
```

E di seguito viene aggiornato l'incasso giornaliero corrispettivo della cassa numero 3. Per semplicità si visualizzano gli incassi giornalieri per la cassa numero 3 con la seguente istruzione:

```
SELECT *  
FROM Cassa  
WHERE numerocassa = 3;
```

Cassa

numerocassa	incassogiornaliero	giorno
3	500	2023-03-08
3	254	2023-03-11

Aggiornamento dei posti in sala in seguito ad una vendita

Ogni volta che si esegue una vendita di un biglietto è necessario incrementare il numero di spettatori per quella determinata sala e decrementare opportunamente il numero di posti disponibili.

```
DELIMITER $$
CREATE TRIGGER modifica_posti
AFTER INSERT ON Vendita
FOR EACH ROW
BEGIN
    UPDATE Sala SET postidisponibili = postidisponibili - 1, nspettatori = nspettatori + 1
    WHERE idsala = (SELECT idsala
                    FROM Biglietto
                    WHERE idbiglietto = NEW.idbiglietto);
END $$

DELIMITER ;
```

Per testare il trigger verrà simulata una vendita. In particolare, la vendita verrà registrata in data 09-03-2023 alla cassa 2 e il biglietto acquistato sarà quello avente codice 222.

Con la seguente istruzione si preleva l'informazione relativa alla sala:

```
SELECT idbiglietto, idsala
FROM biglietto
WHERE idbiglietto = 222;
```

Il risultato di questa query è il seguente:

Biglietto

idbiglietto	idsala
222	2

Il numero della sala è 2, per cui è possibile vedere il numero di posti disponibili e il numero di spettatori attualmente in possesso di un biglietto:

```
SELECT idsala, nspettatori, postidisponibili
FROM sala
WHERE idsala = 2;
```

Produce il seguente risultato:

Sala

idsala	nspettatori	postidisponibili
2	70	30

Adesso è possibile registrare una nuova vendita:

```
INSERT INTO Vendita (idcassa, idbiglietto, idsnack, data) VALUES
(2, 222, NULL, '2023-03-09');
```

Questa query produce l'attivazione del trigger che modifica i relativi posti. Il risultato lo visualizzo con la stessa **SELECT** scritta precedentemente:

Sala

idsala	nspettatori	postidisponibili
2	71	29

Inserimento in cassa dei dati relativi alla nuova vendita

```
DELIMITER $$
CREATE TRIGGER controlla_cassa
BEFORE INSERT ON Vendita
FOR EACH ROW
BEGIN
    IF NOT EXISTS (SELECT *
                   FROM Cassa
                   WHERE giorno = NEW.data)
    THEN
        INSERT INTO Cassa (numerocassa, incassogiornaliero, giorno)
        VALUES (NEW.idcassa, 0, NEW.data);
    END IF;
END $$
DELIMITER ;
```

Il trigger verrà attivato con valori *di esempio*. Per prima cosa si visualizzano tutte le vendite e tutti i dati presenti nella cassa:

Vendita

idvendita	idcassa	idbiglietto	idsnack	data
1	1	111	1	2023-03-11
2	2	222	2	2023-03-11
3	3	333	3	2023-03-08
4	2	222	4	2023-03-12
5	2	222	NULL	2023-03-09

Cassa

numerocassa	incassogiornaliero	giorno
1	440	2023-03-01
1	400	2023-03-10
2	207	2023-03-09
3	500	2023-03-08
3	250	2023-03-11

Snack

idsnack	nome	prezzo
1	Popcorn	4
2	Bibita	2.5
3	Nachos	5
4	Hot dog	3.5
5	Patatine	2

Adesso verrà registrata una nuova vendita dello snack con id 3 in data 2023-01-01 presso la cassa 2:

```
INSERT INTO Vendita (idcassa, idbiglietto, idsnack, data) VALUES  
(2, NULL, 3, '2023-01-01');
```

La vendita viene registrata nella tabella Vendita:

Vendita

idvendita	idcassa	idbiglietto	idsnack	data
1	1	111	1	2023-03-11
2	2	222	2	2023-03-11
3	3	333	3	2023-03-08
4	2	222	4	2023-03-12
5	2	222	NULL	2023-03-09
6	2	NULL	3	2023-01-01

Alla cassa viene aggiunta la data della vendita visto che non era presente ed è possibile visualizzare tale dato con l'istruzione:

```
SELECT *  
FROM Cassa  
WHERE giorno = '2023-01-01';
```

Cassa

numerocassa	incassogiornaliero	giorno
2	5	2023-01-01

OPERAZIONI SQL

Inserimento di un nuovo turno di lavoro

```
INSERT INTO Turno (giorno, orainizio, orafine, cfooperatore)
VALUES ('2023-01-13', '09:00:00', '17:00:00', 'CF12345678900010');
```

Che produrrà il risultato:

giorno	orainizio	orafine	cfooperatore
2023-01-02	09:00:00	17:00:00	CF12345678900010
2023-01-13	09:00:00	17:00:00	CF12345678900010

Modifica del prezzo di uno snack

```
UPDATE Snack
SET prezzo = 3.50
WHERE idsnack = 1;
```

Che produrrà il seguente risultato:

idsnack	nome	prezzo
1	Popcorn	3.5

Lettura dell'incasso totale giornaliero

```
SELECT giorno, incassogiornaliero
FROM Cassa
WHERE giorno = '2023-03-08';
```

giorno	incassogiornaliero
2023-03-08	500

Lettura della media dei feedback per un film

Dati i seguenti film:

idfilm	titolo	genere
1	La casa infestata	1
2	La sposa perfetta	2
3	Benvenuti al sud	3
4	Il mostro in cantina	1
5	La cena	3
6	Tra me e te	2
7	Paranormal	1
8	Beautiful	2
9	Ho perso la macchina	3
10	Non aprire quella porta	1

E i seguenti Feedback:

idfeedback	valutazione	idfilm
124	4	1
243	3	2
301	4	3
438	2	1
500	5	5

Usando la seguente query si può rispondere alla richiesta:

```
SELECT fi.idfilm, fi.titolo, g.nome as genere, AVG(fe.valutazione) AS media_feedback
FROM Feedback fe
JOIN Film fi ON fe.idfilm = fi.idfilm
JOIN Genere g ON fi.genere = g.idgenere
WHERE fi.idfilm = 1;
```

idfilm	titolo	genere	media_feedback
1	La casa infestata	horror	3.0000

Inserimento di un nuovo feedback

```
INSERT INTO Feedback (idfeedback, valutazione, idfilm)
VALUES (300, 3, 5);
```

OPERAZIONI UTILI

Lettura degli incassi giornalieri per ogni cassa e per ogni data

Si registrano i seguenti dati:

```
INSERT INTO Vendita (idcassa, idbiglietto, idsnack, data) VALUES
  (2, 444, 3, '2023-01-01'),
  (2, NULL, 5, '2023-01-01'),
  (2, 222, 2, '2023-01-01');
```

Dopodiché si può leggere il dato interessato:

```
SELECT numerocassa, giorno, SUM(incassogiornaliero) AS incasso_giornaliero
FROM Cassa
GROUP BY numerocassa, giorno;
```

Inserimento di una nuova vendita

```
INSERT INTO Vendita (idcassa, idbiglietto, idsnack, data) VALUES
  (3, 444, 3, '2023-03-12'),
  (3, NULL, 5, '2023-03-12'),
  (3, NULL, 2, '2023-03-12');
```

PROGETTAZIONE LOGICA

Turno (**giorno**, **orainizio**, **orafine**, cfooperatore)

Operatore (**cfooperatore**, data nascita, nome, cognome, idcassa)

Cassa (**numerocassa**, incassogiornaliero, **giorno**)

Vendita (**idvendita**, idcassa, idbiglietto, idsnack, data)

Biglietto (**idbiglietto**, prezzo, idsala)

Snack (**idsnack**, nome, prezzo)

Sala (**idsala**, idfilm, nspettatori, postidisponibili)

Film (**idfilm**, titolo, genere)

Feedback (**idfeedback**, valutazione, idfilm)

Genere (**idgenere**, nome)

PROGETTAZIONE FISICA

