

Progetto di Basi di Dati:

Recensioni Film

Daniele Cocuzza

1000014622

INDICE DEL DOCUMENTO

Descrizione e specifiche	2
Glossario dei termini	2
Dati generali	2
Specifiche sulle operazioni	3
Schema scheletro	4
Schema intermedio	5
Schema finale	6
Dizionario dei dati – Entità	7
Dizionario dei dati - Relazioni	7
Vincoli non esprimibili dallo schema E-R	7
Tavola dei volumi	8
Tavola delle frequenze	8
Eliminazione gerarchie	9
Traduzione verso il modello relazionale	10
Progettazione fisica	11
Implementazione in SQL delle tabelle	12
Implementazione in SQL delle operazioni	14

Descrizione e specifiche

Si vuole realizzare un database per la gestione di recensioni su film. Un utente si registra in piattaforma inserendo il nome, il cognome, l'e-mail e la password. Un utente può recensire più film dando come valutazione un voto o un commento o entrambi. Un film ha un titolo, una trama, una data di uscita ed uno o più generi, dove ogni genere ha una sua descrizione. La formazione di ogni film è data dai collaboratori. La piattaforma permette di trovare per ogni attore i film in cui ha recitato e di trovare i film appartenenti ad un dato genere.

Glossario dei termini

TERMINE	DESCRIZIONE	SINONIMI	TERMINI COLLEGATI
Utente	Individuo che è registrato nella piattaforma e può fare recensioni	Individuo	Recensione
Recensione	Giudizio di un utente su un film	Giudizio, Critica, Commento	Utente, Film
Film	Pellicola in piattaforma	Pellicola, Proiezione	Recensione, Genere, Cast
Genere	Categoria a cui appartiene un film	Categoria, Tipo	Film
Cast	Collaboratori relativi alla realizzazione di un film	Troupe	Film

Dati generali

Piattaforma per la gestione delle recensioni di film da parte degli utenti.

Dati sugli utenti

Gli utenti sono identificati dal nome, cognome, e-mail. La password servirà per il login.

Dati sulle recensioni

La recensione è composta da un voto numerico ed un commento testuale.

Dati sui film

Un film ha un titolo ed una trama.

Dati sui generi

Il genere ha l'insieme delle categorie a cui può appartenere un film.

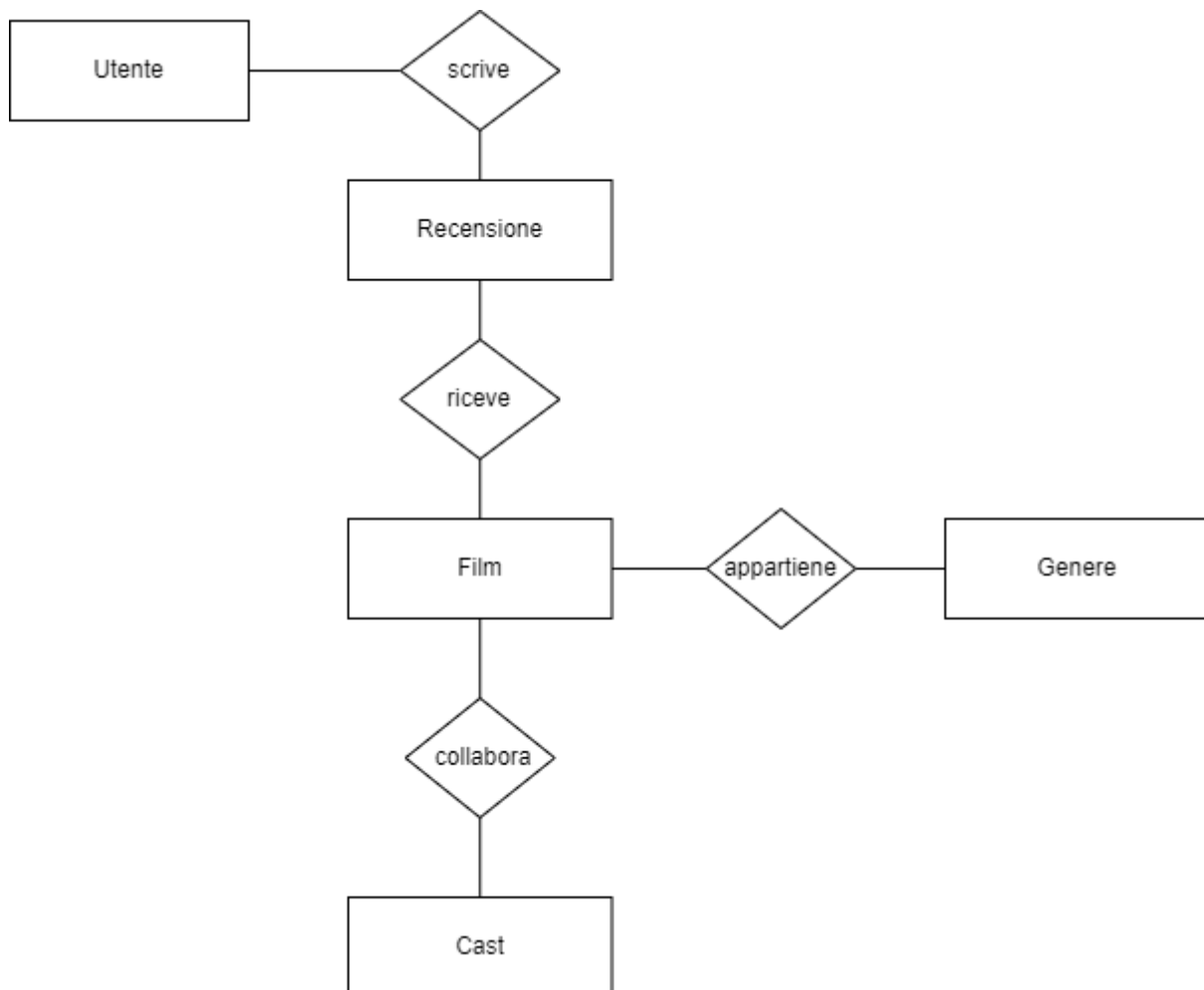
Dati sui cast

Identifica chi collabora alla realizzazione di un film, identificato dal nome, cognome, data di nascita ed il ruolo che ha in un determinato film.

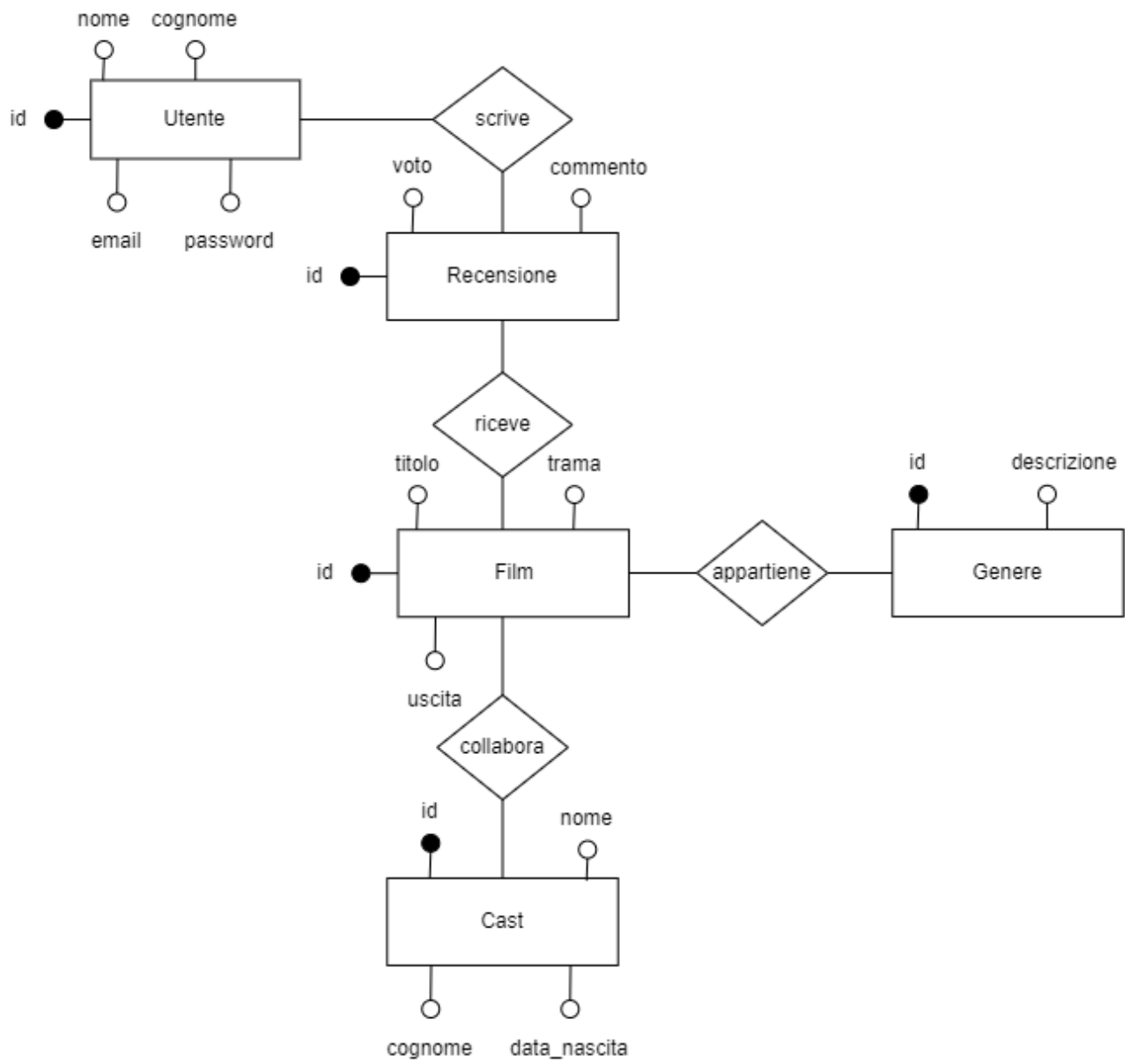
SPECIFICHE SULLE OPERAZIONI

- O1** Registrazione di un nuovo utente.
- O2** Inserimento di un nuovo film.
- O3** Scrittura recensione di un film da parte di un utente.
- O4** Stampa informazioni su un film.
- O5** Stampa recensioni di un film.
- O6** Stampa media dei voti dei film.
- O7** Dato un attore trovare i film in cui ha recitato.
- O8** Elencare tutti i film fantasy.

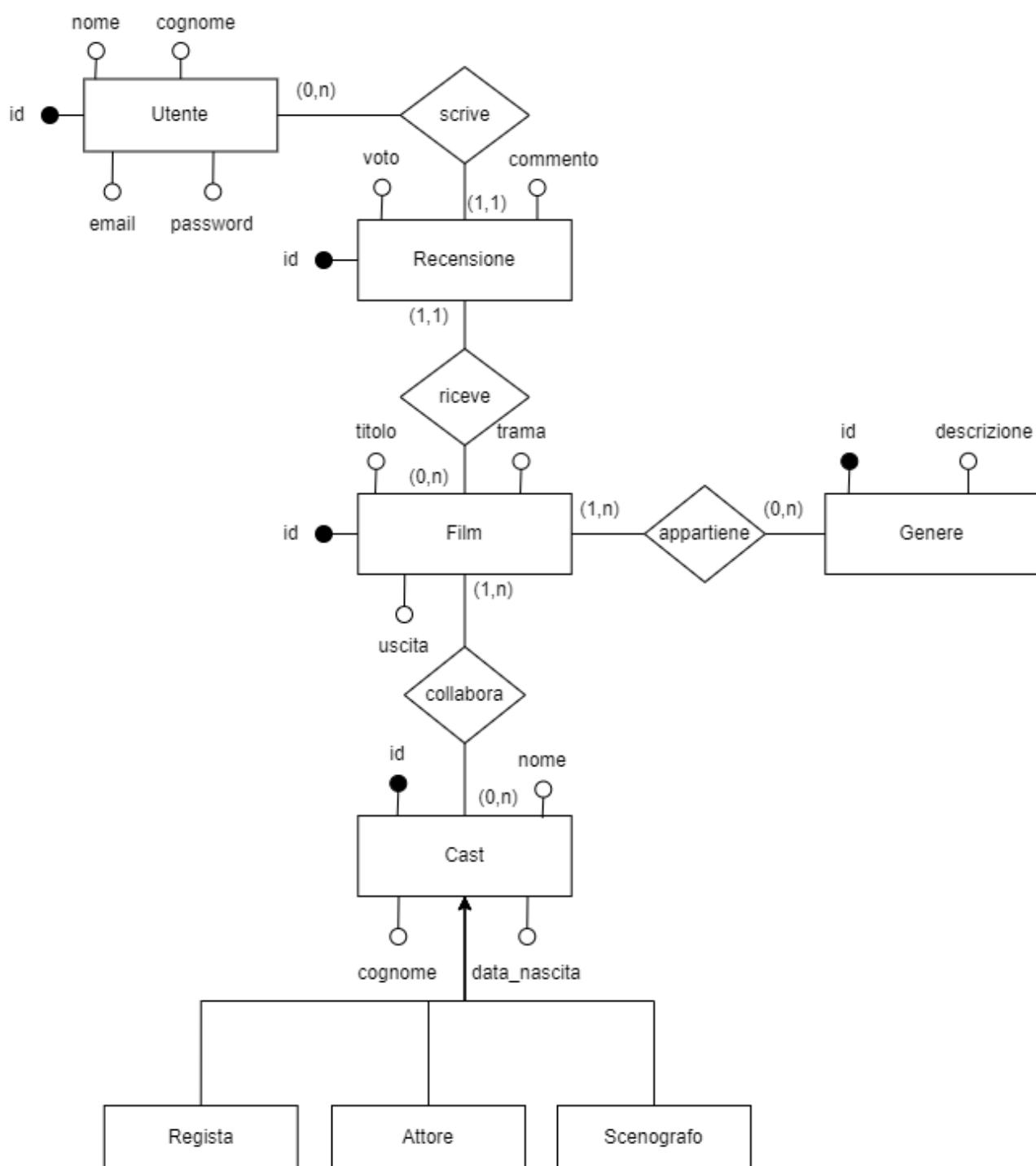
Schema scheletro



Schema intermedio



Schema finale



Dizionario dei dati – Entità

ENTITÀ	DESCRIZIONE	ATTRIBUTI	IDENTIFICATORE
Utente	Individuo che può fare recensioni	Nome, cognome, e-mail, password	id
Recensione	Giudizio di un utente su un film	Voto, commento	id
Film	Pellicola in piattaforma	Titolo, trama, uscita	id
Genere	Categoria a cui appartiene un film	descrizione	id
Cast	Collaboratori relativi alla realizzazione di un film	Nome, cognome, data_nascita	id

Dizionario dei dati – Relazioni

RELAZIONE	ENTITÀ PARTECIPANTE	DESCRIZIONE
Scrive	Utente, Recensione	Un utente può scrivere una o più recensioni. Una recensione deve essere scritta da un solo utente
Riceve	Recensione, Film	Un film può ricevere una o più recensioni. Una recensione deve essere relativa ad un solo film
Appartiene	Film, Genere	Un film deve appartenere ad uno o più generi. Un genere può appartenere ad uno o più film.
Collabora	Film, Cast	Ad un film devono collaborare uno o più cast. Un cast può collaborare con uno o più film

Vincoli non esprimibili dallo schema E-R

- Il voto della recensione è compreso tra 1 e 10.
- La piattaforma dà la possibilità di stampare le informazioni di un film.
- L'inserimento dei nuovi film viene fatto dagli amministratori della piattaforma.
- Possono esserci generi registrati ma possono non essere associati a nessun film.

Tavola dei volumi

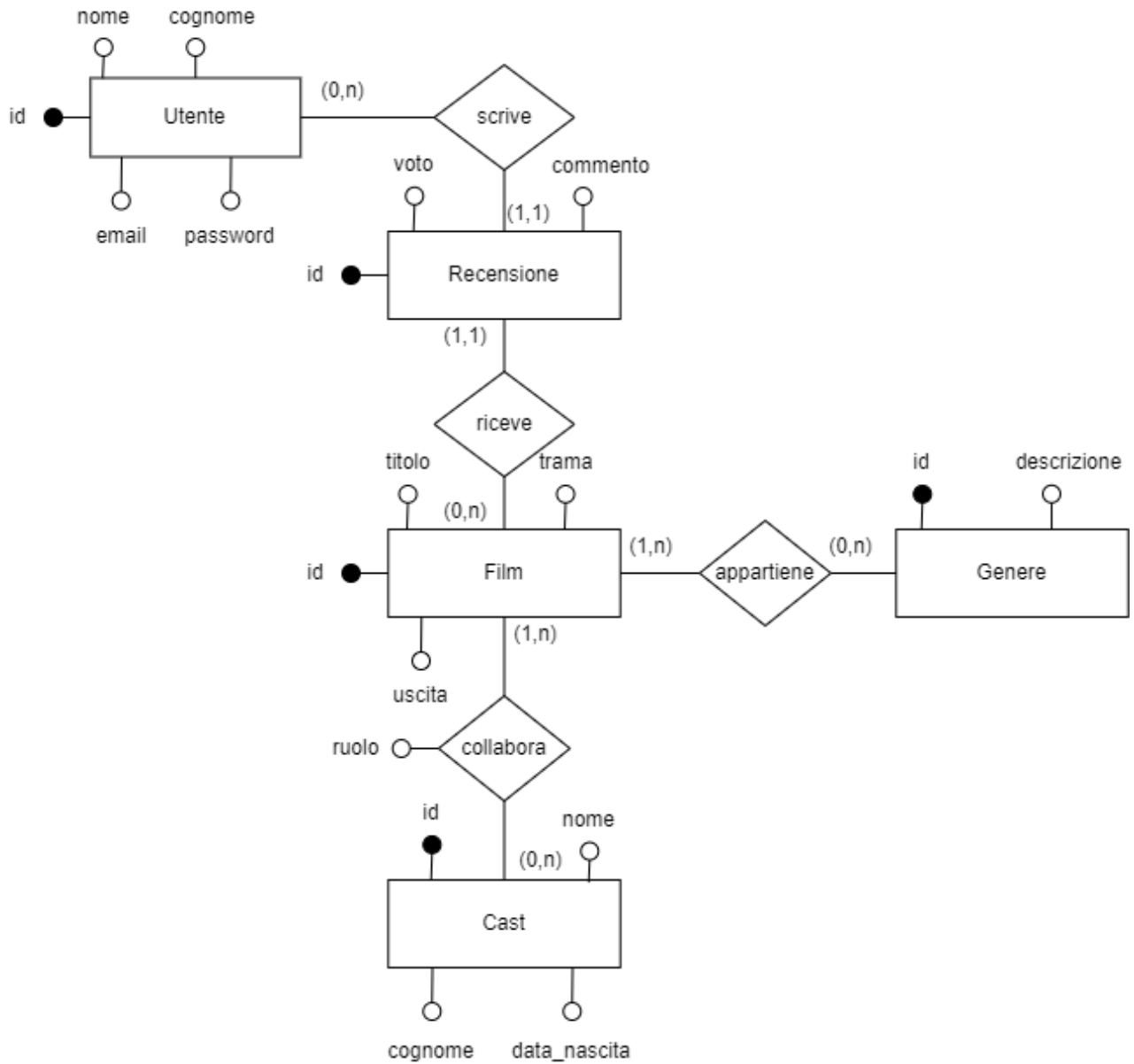
CONCETTO	TIPO	VOLUME
Utente	E	1000
Recensione	E	10000
Film	E	500
Genere	E	10
Cast	E	2000
Scrive	R	10000
Riceve	R	10000
Appartiene	R	800
Collabora	R	3000

Tavola delle frequenze

OPERAZIONE	DESCRIZIONE	FREQUENZA	TIPO
O1	Registrazione di un nuovo utente	5/mese	B
O2	Inserimento di un nuovo film	4/mese	B
O3	Scrittura recensione di un film da parte di un utente	15/giorno	I
O4	Stampa informazioni su un film	50/giorno	I
O5	Stampa recensioni di un film	100/giorno	I
O6	Stampa media dei voti dei film	1/giorno	I
O7	Dato un attore trovare i film in cui ha recitato	2/settimana	B
O8	Elencare tutti i film fantasy	2/giorno	I

Eliminazione delle gerarchie, schema ristrutturato

-Eliminiamo la gerarchia su Cast aggiungendo un attributo alla relazione "collabora" per specificare il ruolo nel Cast del Film.



Descrizione dei dati – Relazione “Collabora” aggiornato dopo l’eliminazione della gerarchia

RELAZIONE	ENTITÀ PARTECIPANTE	DESCRIZIONE	ATTRIBUTI
Collabora	Film, Cast	Ad un film devono collaborare uno o più cast. Un cast può collaborare con uno o più film	ruolo

Traduzione verso il modello relazionale: le associazioni appartiene e collabora essendo delle molti a molti diventano delle relazioni, rispettivamente GenereFilm e RuoloFilm.

Utente(id, nome, cognome, email, password)

Recensione(id, idutente, idfilm, voto, commento)

Film(id, titolo, trama, uscita)

GenereFilm(id, idgenere, idfilm)

Genere(id, descrizione)

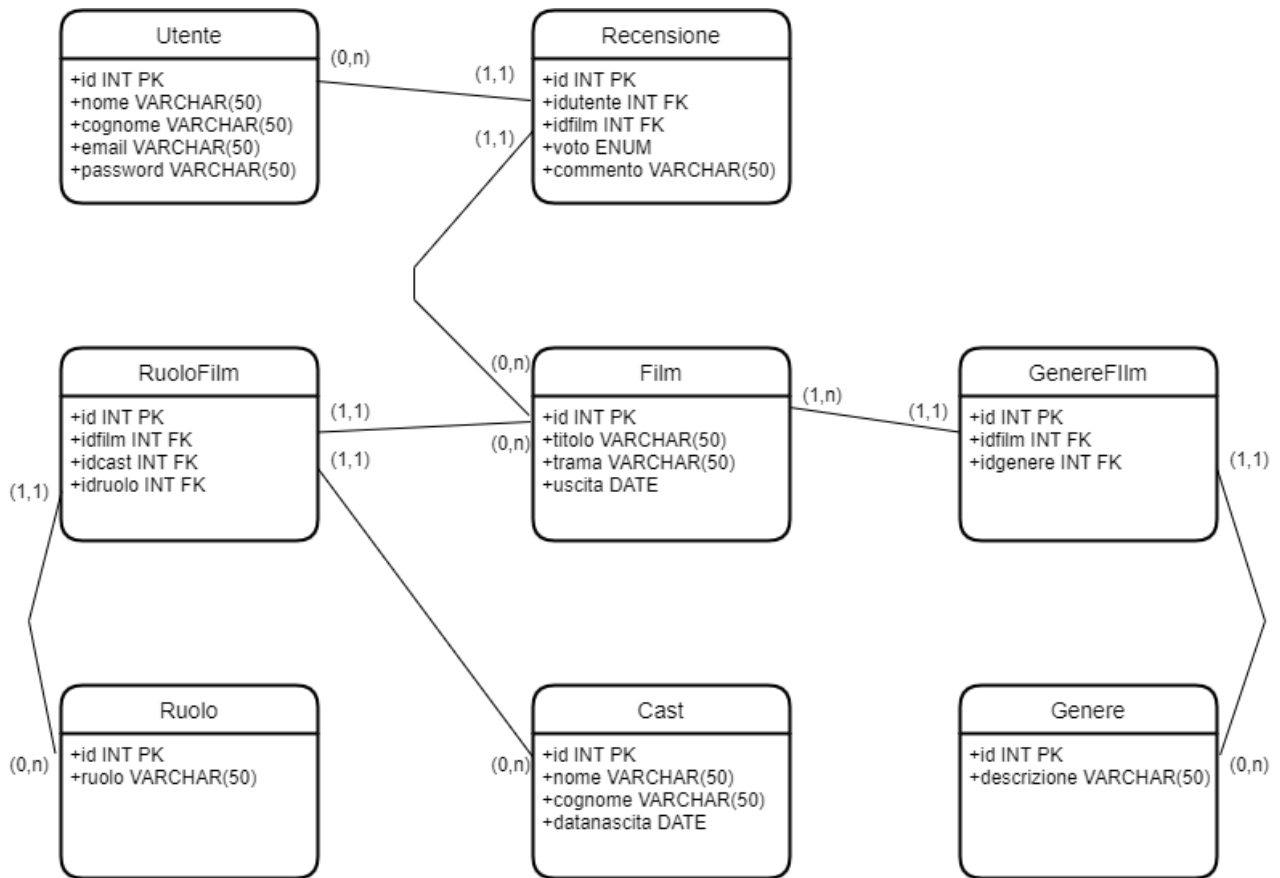
RuoloFilm(id, idcast, idfilm, idruolo)

Ruolo(id, ruolo)

Cast(id, nome, cognome, datanascita)

- 1) La sottolineatura indica la chiave primaria
- 2) La tratteggiatura indica la chiave esterna

Progettazione fisica



Implementazione in SQL delle tabelle

```
CREATE TABLE IF NOT EXISTS `cast` (  
  `id` int(11) NOT NULL AUTO_INCREMENT,  
  `nome` varchar(50) NOT NULL DEFAULT '0',  
  `cognome` varchar(50) NOT NULL DEFAULT '0',  
  `datanascita` date DEFAULT NULL,  
  PRIMARY KEY (`id`)  
) ENGINE=InnoDB AUTO_INCREMENT=6 DEFAULT CHARSET=utf8mb4;
```

```
CREATE TABLE IF NOT EXISTS `film` (  
  `id` int(11) NOT NULL AUTO_INCREMENT,  
  `titolo` varchar(50) DEFAULT NULL,  
  `trama` varchar(500) DEFAULT NULL,  
  `uscita` date DEFAULT NULL,  
  PRIMARY KEY (`id`)  
) ENGINE=InnoDB AUTO_INCREMENT=4 DEFAULT CHARSET=utf8mb4;
```

```
CREATE TABLE IF NOT EXISTS `genere` (  
  `id` int(11) NOT NULL AUTO_INCREMENT,  
  `descrizione` varchar(50) DEFAULT NULL,  
  PRIMARY KEY (`id`)  
) ENGINE=InnoDB AUTO_INCREMENT=7 DEFAULT CHARSET=utf8mb4;
```

```
CREATE TABLE IF NOT EXISTS `generefilm` (  
  `id` int(11) NOT NULL AUTO_INCREMENT,  
  `idfilm` int(11) NOT NULL,  
  `idgenere` int(11) NOT NULL,  
  PRIMARY KEY (`id`),  
  KEY `FK_generefilm_film` (`idfilm`),  
  KEY `FK_generefilm_genere` (`idgenere`),  
  CONSTRAINT `FK_generefilm_film` FOREIGN KEY (`idfilm`) REFERENCES `film` (`id`)  
ON DELETE CASCADE ON UPDATE CASCADE,  
  CONSTRAINT `FK_generefilm_genere` FOREIGN KEY (`idgenere`) REFERENCES `genere`  
(`id`) ON DELETE CASCADE ON UPDATE CASCADE  
) ENGINE=InnoDB AUTO_INCREMENT=5 DEFAULT CHARSET=utf8mb4;
```

```

CREATE TABLE IF NOT EXISTS `recensione` (
  `id` int(11) NOT NULL AUTO_INCREMENT,
  `idutente` int(11) NOT NULL DEFAULT 0,
  `idfilm` int(11) NOT NULL DEFAULT 0,
  `voto` enum('0','1','2','3','4','5','6','7','8','9','10') NOT NULL,
  `commento` varchar(50) NOT NULL DEFAULT '',
  PRIMARY KEY (`id`),
  KEY `FK_recensione_utente` (`idutente`),
  KEY `FK_recensione_film` (`idfilm`),
  CONSTRAINT `FK_recensione_film` FOREIGN KEY (`idfilm`) REFERENCES `film` (`id`)
ON DELETE CASCADE ON UPDATE CASCADE,
  CONSTRAINT `FK_recensione_utente` FOREIGN KEY (`idutente`) REFERENCES `utente`
(`id`) ON DELETE CASCADE ON UPDATE CASCADE
) ENGINE=InnoDB AUTO_INCREMENT=4 DEFAULT CHARSET=utf8mb4;

```

```

CREATE TABLE IF NOT EXISTS `ruolo` (
  `id` int(11) NOT NULL AUTO_INCREMENT,
  `ruolo` varchar(50) NOT NULL,
  PRIMARY KEY (`id`)
) ENGINE=InnoDB AUTO_INCREMENT=6 DEFAULT CHARSET=utf8mb4;

```

```

CREATE TABLE IF NOT EXISTS `ruolofilm` (
  `id` int(11) NOT NULL AUTO_INCREMENT,
  `idfilm` int(11) DEFAULT NULL,
  `idcast` int(11) DEFAULT NULL,
  `idruolo` int(11) DEFAULT NULL,
  PRIMARY KEY (`id`),
  KEY `FK_castfilm_film` (`idfilm`),
  KEY `FK_castfilm_cast` (`idcast`),
  KEY `FK_castfilm_ruolo` (`idruolo`),
  CONSTRAINT `FK_castfilm_cast` FOREIGN KEY (`idcast`) REFERENCES `cast` (`id`) ON
DELETE CASCADE ON UPDATE CASCADE,
  CONSTRAINT `FK_castfilm_film` FOREIGN KEY (`idfilm`) REFERENCES `film` (`id`) ON
DELETE CASCADE ON UPDATE CASCADE,
  CONSTRAINT `FK_castfilm_ruolo` FOREIGN KEY (`idruolo`) REFERENCES `ruolo` (`id`)
ON DELETE NO ACTION ON UPDATE NO ACTION
) ENGINE=InnoDB AUTO_INCREMENT=6 DEFAULT CHARSET=utf8mb4;

```

```
CREATE TABLE IF NOT EXISTS `utente` (
  `id` int(11) NOT NULL AUTO_INCREMENT,
  `nome` varchar(50) NOT NULL,
  `cognome` varchar(50) NOT NULL,
  `email` varchar(50) NOT NULL,
  `password` varchar(50) NOT NULL,
  PRIMARY KEY (`id`) USING BTREE
) ENGINE=InnoDB AUTO_INCREMENT=4 DEFAULT CHARSET=utf8mb4;
```

Implementazione delle operazioni in SQL

O1 Registrazione di un nuovo utente

```
INSERT INTO `progettobasididati`.`utente` (`nome`, `cognome`, `email`, `password`) VALUES ($nome, $cognome, $email, $password);
```

O2 Inserimento di un nuovo film

```
INSERT INTO `progettobasididati`.`film` (`titolo`, `trama`, `uscita`) VALUES ($titolo, $trama, &uscita);
```

O3 Scrittura recensione di un film da parte di un utente

```
INSERT INTO `progettobasididati`.`recensione` (`idutente`, `idfilm`, `voto`, `commento`) VALUES ($idutente, $idfilm, $voto, $commento);
```

O4 Stampa informazioni su un film

```
SELECT cast.nome, cast.cognome, ruolo
FROM cast, ruolo, ruolofilm
WHERE ruolofilm.idcast = cast.id AND ruolofilm.idruolo = ruolo.id AND ruolofilm.idfilm = $idfilm
```

O5 Stampa recensioni di un film

```
SELECT nome, cognome, voto, commento
FROM recensione, utente
WHERE recensione.idutente = utente.id AND idfilm = $idfilm
```

O6 Stampa media dei voti dei film

```
SELECT titolo, AVG(voto)
FROM film, recensione
WHERE film.id = recensione.idfilm
GROUP BY titolo
```

O7 Dato un attore trovare i film in cui ha recitato

```
SELECT distinct titolo
FROM film, ruolofilm, ruolo, cast
WHERE film.id = ruolofilm.idfilm AND ruolo = 'attore' AND ruolofilm.idcast = $idcast
```

O8 Elencare tutti i film fantasy

```
SELECT titolo
FROM film, generefilm, genere
WHERE film.id = generefilm.idfilm AND genere.id = generefilm.idgenere AND genere.descrizione = "fantasy"
```