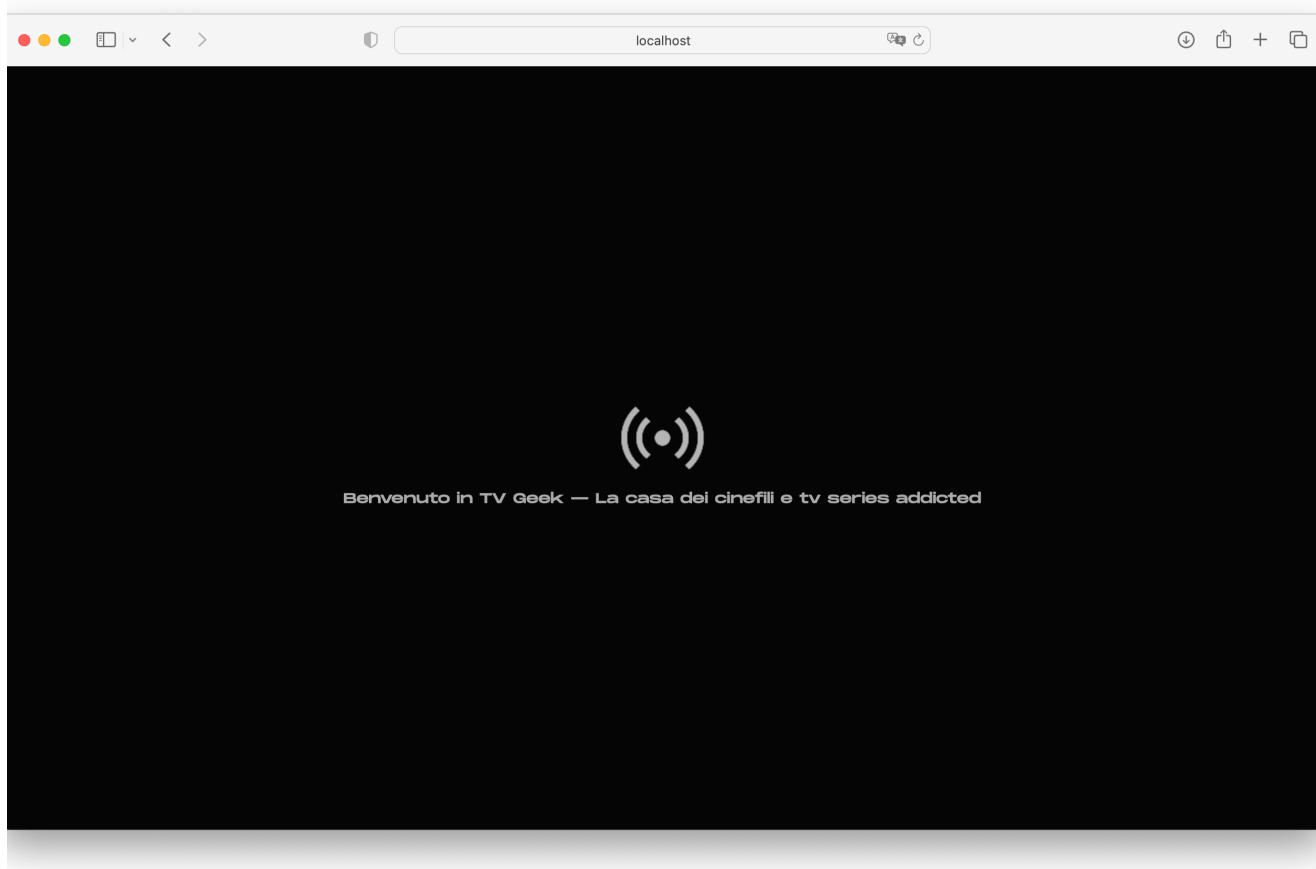

TVGeek

Progetto Basi di Dati – A.A. 2022/23

Martina Contestabile, mat. 731044



INTRODUZIONE	3
ANALISI DEI REQUISITI	4
DESCRIZIONE	4
SICUREZZA	6
DATASET	6
SCHEMA CONCETTUALE E-R	7
COMMENTI SU ALCUNI ATTRIBUTI, ENTITÀ E RELAZIONI	7
all_movies_details_clear	7
all_movies_casting_raw	8
language_to_country	8
spielberg_awards	8
awards_by_directors	8
acclaimed_directors_awards	9
favorite_directors_summary	9
most_common_language_by_director	9
users	10
TRADUZIONE NEL MODELLO RELAZIONALE	11
all_movies_details_clear	11
all_movies_casting_raw	11
language_to_country	11
spielberg_awards	12
awards_by_directors	12
acclaimed_directors_awards	12
favorite_directors_summary	12
most_common_language_by_director	12
users	12
SCHEMA LOGICO	13
QUERY E COMANDI SQL	14
RECUPERO PASSWORD	14
REGISTRAZIONE A TVGEEK	14
ACCESSO A TVGEEK	15
SERVIZIO TVGEEK	15

INTRODUZIONE

TGeek è il sito web che porta nel mondo della televisione e del cinema come mai prima d'ora. Sfruttando la potenza di HTML e SQL, TGeek offre un'esperienza coinvolgente e personalizzata, consentendo di scoprire nuovi contenuti da guardare sulla base dei parametri di ricerca d'interesse per l'utente e di approfondire informazioni dettagliate sui film e le serie TV suggerite, il tutto previa registrazione gratuita al servizio.

Se l'utente è stufo di passare ore a cercare qualcosa di interessante da guardare, TGeek è qui per semplificargli la vita. Grazie alla sua intuitiva interfaccia utente, potrà filtrare e cercare i contenuti in base a una vasta gamma di criteri, come il genere, l'anno di uscita, il punteggio di valutazione e molto altro ancora. Che l'utente sia un appassionato di azione, un fan del mistero o un amante delle commedie romantiche, TGeek troverà il materiale perfetto per soddisfare i suoi gusti e, magari, anche quelli che non conosceva.

Ma TGeek non si ferma solo a fornire un semplice elenco di titoli. Infatti, è pensato per mostrare informazioni approfondite su tutto ciò che viene consigliato. Attraverso un database completo e aggiornato regolarmente, è possibile avere accesso a informazioni dettagliate sugli attori, i registi e gli sceneggiatori coinvolti in ogni progetto. Scoprire le biografie dei registi, i premi vinti, i loro progetti precedenti e molto altro ancora. Con TGeek, ci si potrà immergere completamente nell'universo di film e delle serie TV, conoscendo le menti brillanti dietro le quinte.

TGeek è progettato per offrire la migliore esperienza di navigazione possibile. Il team di sviluppatori lavora costantemente per migliorare il sito, garantendo la velocità, l'affidabilità e la sicurezza che gli utenti meritano. TGeek auspica ad essere un ambiente online in cui si può esplorare, scoprire e connettersi con la propria passione per il cinema e la televisione.

Quindi, che tu sia un cinefilo incallito, un appassionato di serie TV o semplicemente qualcuno in cerca di nuovi contenuti da guardare, TGeek è qui per te. Entra nel mondo dei tuoi film e delle tue serie TV preferite e scopri un universo di possibilità.

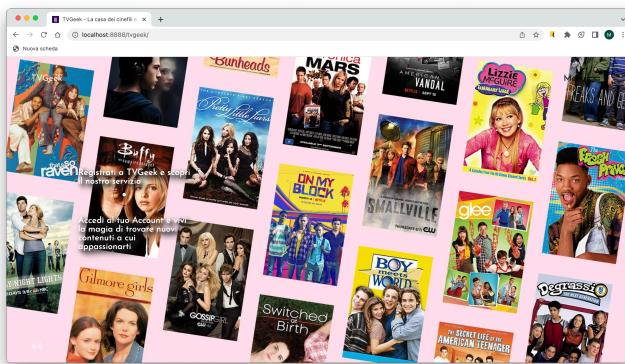
ANALISI DEI REQUISITI

DESCRIZIONE

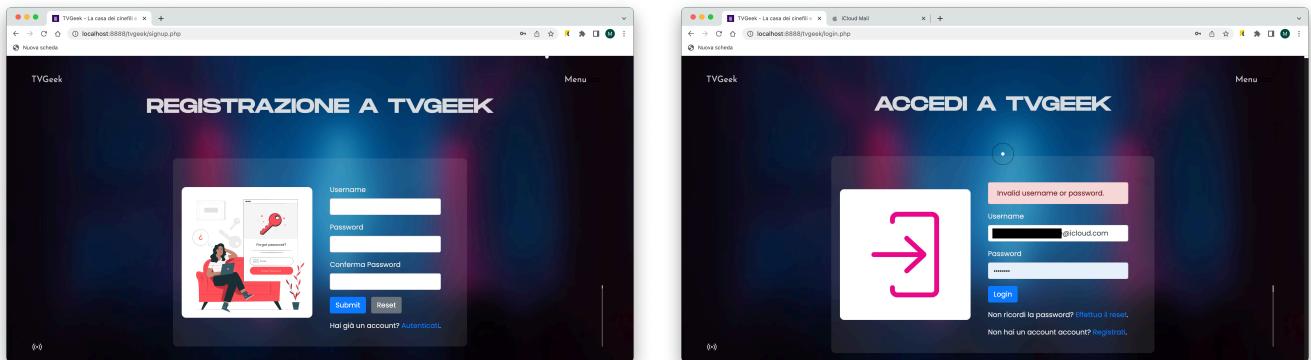
L'applicativo fa uso di Php 8.2.0 e SQL nel back-end, mentre HTML, CSS e JavaScript per il front-end attraverso il Web server Apache. Durante lo sviluppo di TVGeek ho fatto uso MAMP, stack di tecnologie utilizzabili per creare un server web completo.

TVGeek prevede come unica tipologia di utente quello che fa uso del servizio TVGeek. Per quanto riguarda la gestione della base di dati, al momento viene effettuata attraverso phpMyAdmin, ma non escludo la possibilità in futuro di consentire all'amministratore di accedere tramite interfaccia HTML per amministrare l'aggiunta/rimozione di tuple dalla base di dati.

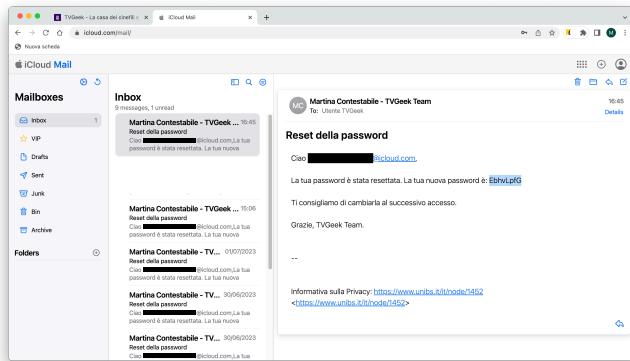
TVGeek presenta una schermata principale di benvenuto, dal quale l'utente può registrarsi, se è la prima volta che visita il sito web, oppure effettuare il login, se si è registrato in precedenza.



Nel caso in cui debba registrarsi, questa è la schermata che si presenta, ma c'è comunque un riferimento alla pagina di login, nel caso in cui si sia sbagliato.

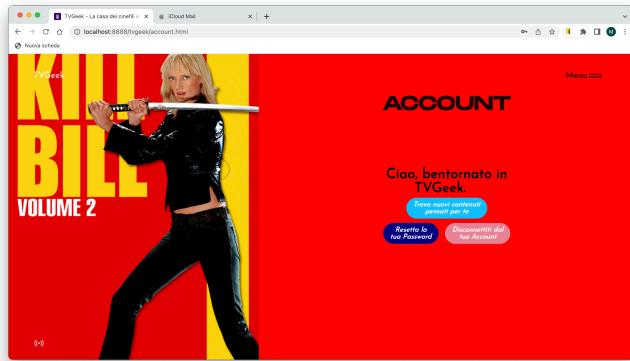


In entrambi i casi, viene effettuato un controllo sulla TG.users, in modo tale da registrare un nuovo utente oppure controllare che le credenziali siano corrette – se non lo sono, viene mostrato un messaggio come in figura. Se, per qualsivoglia ragione, l'utente non ricorda la password, è possibile recuperarla attraverso il form di recupero password, dove, inserendo il proprio indirizzo email, che corrisponde allo username, si riceverà una mail con la password temporanea, da cambiare al successivo accesso.

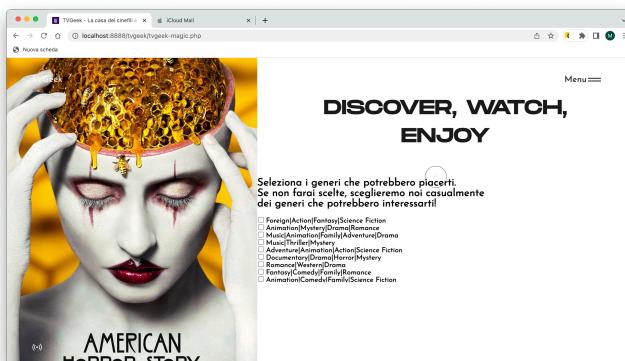


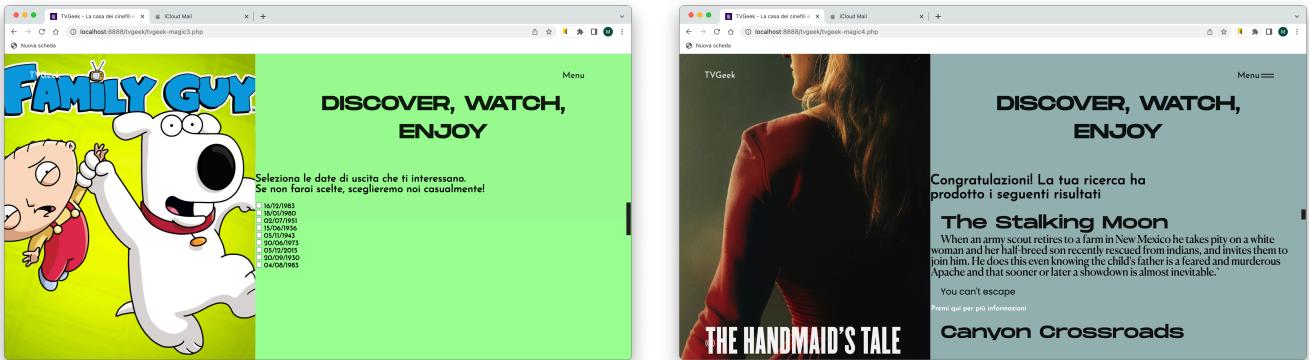
Per l'invio di tali email ho sfruttato la libreria [PHPMailer](#), mentre la mail utilizzata per l'invio è quella universitaria [@studenti.unibs.it](#).

Proseguendo, la schermata che si presenta all'utente una volta effettuato l'accesso è la seguente

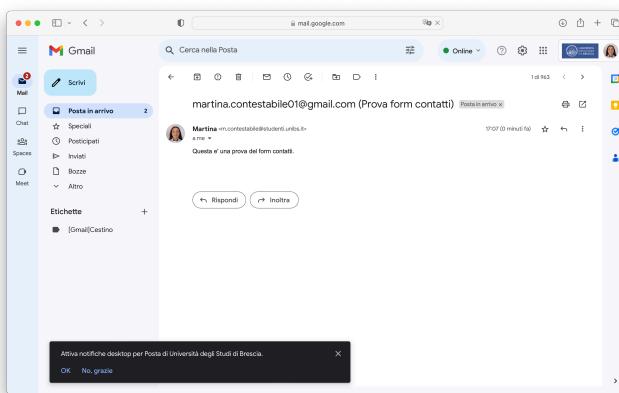


nonché la schermata in cui l'utente può gestire le proprie credenziali, disconnettersi oppure usufruire del servizio TVGeek, il cui funzionamento a livello back-end verrà analizzato nei paragrafi successivi. La prima scelta che viene fatta è quella dei generi, la seconda quella dei paesi e la terza quella delle date di uscita. Per ciascuna checkbox è possibile scegliere più di un valore, alla fine delle opzioni è presente il bottone «proseguì», che consente di passare alla pagina successiva. Una volta mostrati i risultati, è possibile accedere alle informazioni più approfondite sul contenuto e sul regista che ci ha lavorato.





Infine, è stata messa a disposizione un form contatti, accessibile da ogni pagina dell'applicativo attraverso il Menu a tendina, dove è possibile inviare una mail alla casella preposta alla ricezione di messaggi, dove è possibile chiedere qualsiasi informazione.



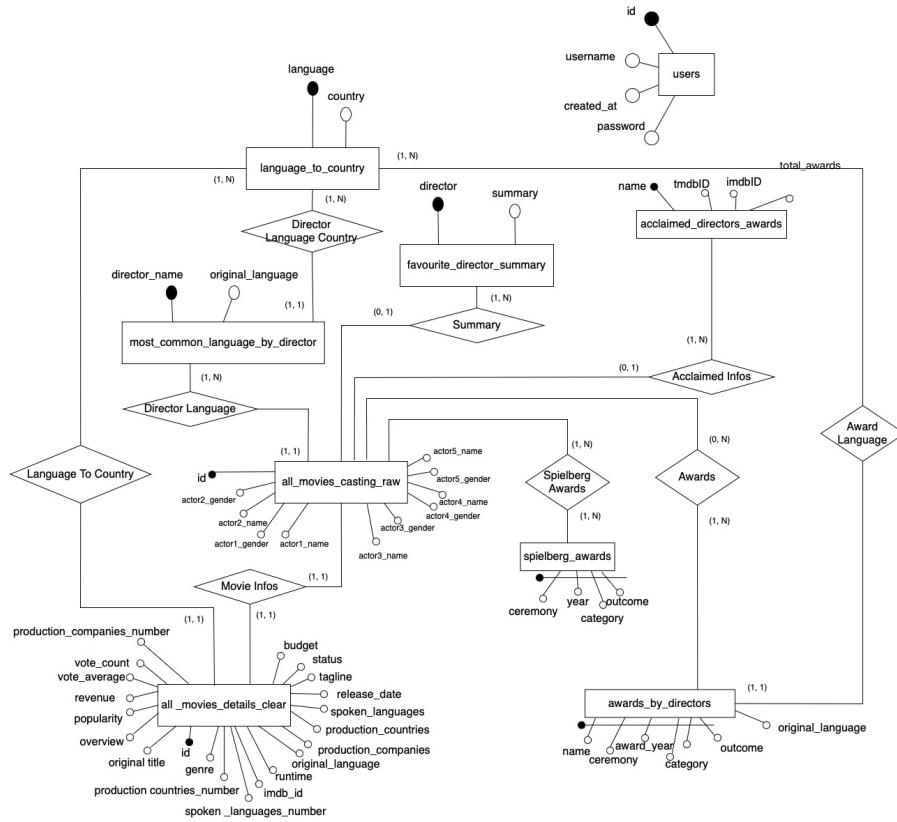
SICUREZZA

TVGeek è dotato di un database formato da 9 tabelle. Per quanto riguarda la tabella users, che contiene informazioni sensibili, si è fatto in modo che avvenisse il salvataggio criptato delle password.

DATASET

Chiaramente, reperire manualmente il database con un quantitativo valori tali da consentire scelta non era cosa pensabile, di conseguenza ho sfruttato il dataset reperito su Kaggle [a questo link](#).

SCHEMA CONCETTUALE E-R



Dall'analisi del dataset, ho elaborato lo schema concettuale in figura. Essendo il diagramma Entity-Relations un tipo di diagramma di flusso che illustra come le *entità*, quali persone, oggetti o concetti, si relazionano tra loro all'interno di un sistema, ho elaborato la seguente struttura concettuale determinando come l'entità principale sia *all_movies_details_clear*. Infatti, dall'analisi del dataset, ho compreso come tale entità sia quella che permette di creare collegamenti logici con le altre.

COMMENTI SU ALCUNI ATTRIBUTI, ENTITÀ E RELAZIONI

all_movies_details_clear

Entità che raggruppa le informazioni principali, sono quelle su cui poi l'utente va ad effettuare delle scelte. Una nota su *imdb_id* è il fatto che il codice qui memorizzato è l'identificativo univoco del contenuto, trovabile su [IMDb](#). Tuttavia, l'identificatore qui è *id*, un identificativo univoco del contenuto, nonostante *imdb_id* possa far parte di una ipotetica superchiave in un modello logico. La ragione è che, sebbene *imdb_id* sia un valore unico, è sufficiente identificare un contenuto, film o serie tv che sia, con il suo *id*. A livello di presentazione, però, *imdb_id* è necessario per presentare all'utente anche un'altra fonte di informazione sul contenuto suggerito.

Infine, con l'entità `language_to_country` c'è una relazione 1 a 1 in quanto ad una lingua corrisponde un paese, mentre in verso opposto è 1 a molti, in quanto ad una lingua possono corrispondere più contenuti.

`all_movies_casting_raw`

Entità che va a completare il pacchetto informativo fornito da `all_movies_details_clear`. Anche qui ritroviamo l'attributo `id` e, per questa ragione, fra `all_movies_details_clear` e `all_movies_casting_raw` ritroviamo una relazione 1 a 1 in ambo le direzioni.

`language_to_country`

Entità che rappresenta la corrispondenza fra lingua parlata e paese di provenienza.

Ritroviamo una relazione 1 a molti con `all_movies_details_clear`, in quanto ad una lingua possono corrispondere più contenuti, mentre dal punto di vista di `all_movies_details_clear` abbiamo una relazione 1 a 1, perché ad un contenuto corrisponde una ed una sola `original_language`. Per le altre due relazioni che presenta, invece, abbiamo una relazione 1 a molti con `most_common_language_by_director`, in quanto ad una lingua possono corrispondere più registi, mentre 1 a molti nell'altro senso, dato che ad un regista corrisponde una sola lingua. Infine, con `awards_by_directors` abbiamo sempre una relazione 1 a molti per via del fatto che una lingua può corrispondere a più `original_language`, mentre 1 a 1 nell'altro senso dato che un' `original_language` corrisponde ad una e una sola lingua.

`spielberg_awards`

Entità che rappresenta i premi vinti dal regista Steven Spielberg, necessita di un'entità a sé a causa del fatto che è un regista pluripremiato e, di conseguenza, possiede un numero di valori elevato. L'identificatore è costituito da `name`, `ceremony`, `award_year`, `category` e `outcome`, in quanto per avere una corrispondenza univoca serve considerare tutti questi attributi.

Questa entità è in relazione con `all_movies_casting_raw` con una relazione 1 a molti, in quanto a Steven Spielberg possono corrispondere più film – l'entità non registra il contenuto per cui è stato premiato, tuttavia è utile per informare l'utente su quali siano stati i premi vinti dal regista. Anche il viceversa è molti a molti, dato che a Steven Spielberg corrispondono più premi, precisamente tutti quelli in `spielberg_awards`.

`awards_by_directors`

Entità che rappresenta i premi vinti da vari registi. Anche qui l'identificatore è formato da `name`, `ceremony`, `award_year`, `category` e `outcome`, ciò è dovuto alle ragioni esplicitate per l'entità `spielberg_awards` – esistono registi pluripremiati come registi mai premiati, ma anche registi che in più anni hanno gareggiato e/o vinto per la stessa categoria dello stesso premio.

Quest'entità si ritrova anch'essa in relazione 1 a molti con *all_movies_casting_raw* per le stesse ragioni viste con *spielberg_awards*, mentre 0 a molti nel senso opposto, dato che un regista potrebbe non aver vinto alcun premio come averne vinti molti.

acclaimed_directors_awards

Entità che rappresenta una sorta di anagrafica dei registi più acclamati in campo cinematografico e televisivo, contenendo l'informazione sul totale dei premi vinti nella loro carriera attraverso l'attributo *total_awards*. L'attributo che identifica unicamente l'entità è il nome del regista, dato che in campo cinematografico in tale entità non si sono riscontrate omonimie. Tuttavia, è possibile sfruttare altri parametri, in caso di reingegnerizzazione del database, come *tmdbID* e *imdbID*, che sono i codici univoci delle pagine dei registi su [IMDb](#) e [TMDB](#).

In quest'entità troviamo una relazione 1 a molti con *all_movies_casting_raw*, dato che ad un regista premiato possono corrispondere più contenuti, mentre in verso opposto la relazione è di tipo 0 a molti, in quanto un regista di un contenuto potrebbe non risultare nella categoria «acclamati».

favorite_directors_summary

Entità che rappresenta il contenuto informativo della biografia, fonte [Wikipedia](#), dei registi preferiti, contenuta nell'attributo *summary*. L'identificatore è il nome del regista, utilizzabile in quanto pure qui non si sono verificati casi di omonimia.

La relazione che possiamo vedere è quella con *all_movies_casting_raw*, che è 1 a molti in quanto un regista può aver prodotto più contenuti, mentre nell'altro verso è 0 a 1 dato che ad un regista corrisponde alcuna oppure una sola biografia.

most_common_language_by_director

Entità che rappresenta il contenuto informativo sulla lingua (tipicamente) più utilizzata nei contenuti diretti dal regista.

Tale entità possiede due relazioni. La prima che vediamo è quella con *all_movies_casting_raw*, 1 a molti per sottolineare come un regista possa aver prodotto più contenuti e 1 a 1 nell'altro verso dato che il regista è uno solo. L'altra, invece, è quella con *language_to_country* di tipo 1 a 1, in quanto è utile sottolineare come ad una lingua parlata dal regista corrisponda un solo paese, mentre nell'altro verso è 1 a molti dato che una stessa lingua può essere parlata da più registi.

users

Entità che rappresenta gli utenti iscritti al servizio TVGeek. Tale entità ha come identificatore l'id, che viene generato al momento dell'iscrizione al servizio. Tuttavia, sarebbe possibile avere come identificatore anche username, dato che è una email e, come è noto, le mail sono univoche ed associate ad una sola persona. L'attributo *created_at* è un'informazione per gli amministratori di sistema su quando l'account è stato creato.

TRADUZIONE NEL MODELLO RELAZIONALE

all_movies_details_clear

```
all_movies_details_clear (id, budget, genres, imdb_id, original_language, original_title,  
overview, popularity, production_companies, production_countries, release_date, revenue,  
runtime, spoken_languages, status, tagline, title, vote_average, vote_count,  
production_companies_number, production_countries_number, spoken_languages_number)
```

```
CONSTRAINT `MovieInfos` FOREIGN KEY (`id`) REFERENCES `all_movies_casting_raw`  
(`id`) ON DELETE CASCADE ON UPDATE CASCADE
```

```
CONSTRAINT `OgLanguage` FOREIGN KEY (`original_language`) REFERENCES  
`language_to_country` (`language`) ON DELETE NO ACTION ON UPDATE NO ACTION
```

all_movies_casting_raw

```
all_movies_casting_raw (id, actor1_name, actor1_gender, actor2_name, actor2_gender,  
actor3_name, actor3_gender, actor4_name, actor4_gender, actor5_name, actor5_gender,  
actor_number, director_name, director_gender, director_number, producer_name,  
producer_number, screenplay_name, editor_name)
```

```
CONSTRAINT `AcclaimedInfos` FOREIGN KEY (`director_name`) REFERENCES  
`acclaimed_directors_awards` (`name`) ON DELETE NO ACTION ON UPDATE NO ACTION
```

```
CONSTRAINT `Awards` FOREIGN KEY (`director_name`) REFERENCES  
`awards_by_directors` (`director_name`) ON DELETE NO ACTION ON UPDATE NO ACTION
```

```
CONSTRAINT `DirectorLanguage` FOREIGN KEY (`director_name`) REFERENCES  
`most_common_language_by_director` (`director_name`) ON DELETE NO ACTION ON UPDATE NO  
ACTION
```

```
CONSTRAINT `SpielbergAwards` FOREIGN KEY (`director_name`) REFERENCES  
`spielberg_awards` (`name`) ON DELETE NO ACTION ON UPDATE NO ACTION,
```

```
CONSTRAINT `Summary` FOREIGN KEY (`director_name`) REFERENCES  
`favorite_directors_summary` (`director`) ON DELETE NO ACTION ON UPDATE NO ACTION
```

language_to_country

```
language_to_country (language, country)
```

spielberg_awards

spielberg_awards (name, ceremony, award year, category, outcome, simplified_outcome)

awards_by_directors

awards_by_directors (director name, ceremony, award year, category, outcome,
original language)

CONSTRAINT `Country` FOREIGN KEY (`original_language`) REFERENCES
`language_to_country` (^language) ON DELETE NO ACTION ON UPDATE NO ACTION

acclaimed_directors_awards

acclaimed_directors_awards (name, tmdbID, imdbID, total_awards)

favorite_directors_summary

favorite_directors_summary (director, summary)

most_common_language_by_director

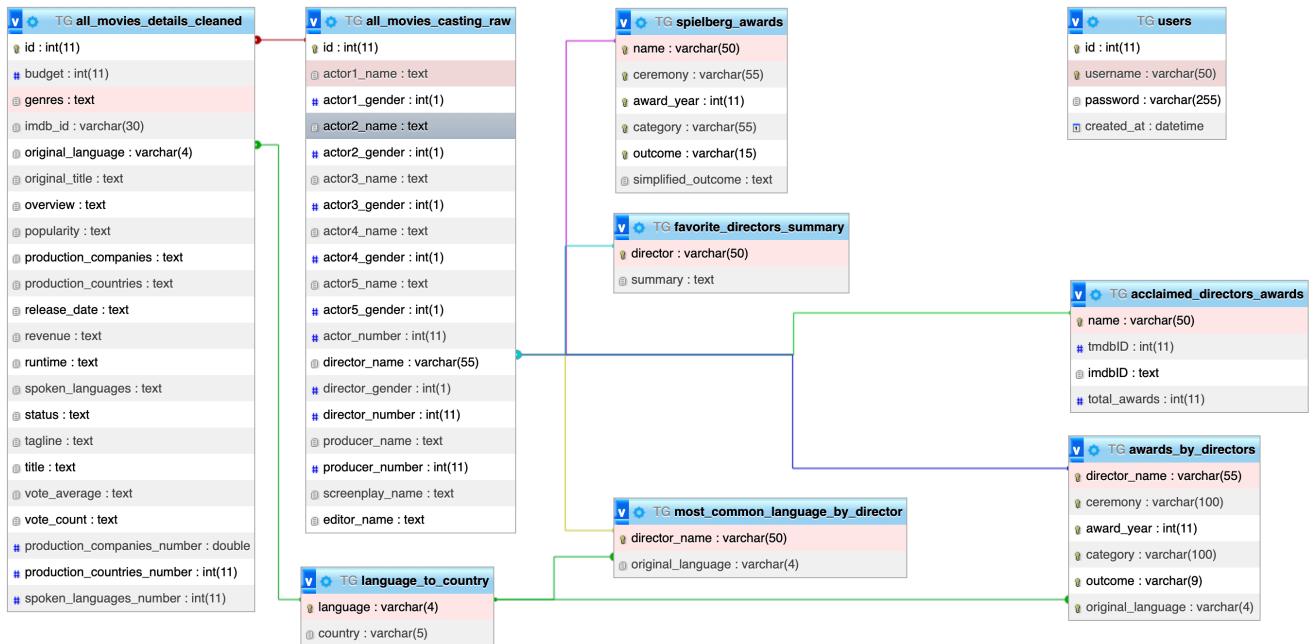
most_common_language_by_director (director name, original_language)

CONSTRAINT `DirectorLanguageCountry` FOREIGN KEY (`original_language`) REFERENCES
`language_to_country` (^language) ON DELETE NO ACTION ON UPDATE NO ACTION

users

users (id, username, password, created_at)

SCHEMA LOGICO



QUERY E COMANDI SQL

In questo paragrafo riporterò e commenterò alcune delle query più significative che permettono il funzionamento di TVGeek.

RECUPERO PASSWORD

A tutti può capitare di dimenticarsi di salvare o ricordarsi la propria password, TVGeek, quindi, ha implementata la funzionalità di recupero password attraverso la propria mail. Ovviamente, la mail deve essere presente nel Data Base TVGeek, altrimenti viene segnalato il fatto che l'email non corrisponde ad alcun account esistente. In alternativa, l'email viene inviata con un messaggio standard ed una password da cambiare (preferibilmente) al primo accesso. Tale password viene generata randomicamente, dove la query per aggiornare la password con la password temporanea è la seguente

```
UPDATE users SET password = '$hashedPassword' WHERE username = '$email'
```

REGISTRAZIONE A TVGEEK

TVGeek mette a disposizione un form d'iscrizione HTML che ha alla base uno script php per creare l'account.

Invece, per verificare che il valore immesso come username abbia il formato di una email, faccio uso dell'espressione regolare

```
/^[\w-]+(\.[\w-]+)*@[ \w-]+\.\+[a-zA-Z]{2,7}$/
```

Se, oltre all'username, sono corretti i campi password e conferma password, avverrà l'esecuzione di memorizzazione dell'utente nella tabella users con la seguente query

```
INSERT INTO users (username, password) VALUES (?, ?)
```

Come si può evincere dalla sintassi, per farlo ho sfruttato PDO e Prepared Statement, ovvero delle istruzioni SQL precedentemente costruite a meno di alcuni parametri da inserire in seguito utilizzando metodi appropriati. Le Prepared Statement consentono di separare l'istruzione SQL dalla sua input data. Invece di concatenare direttamente i valori nella query SQL, si crea un'istruzione precompilata con segnaposto per i valori che verranno forniti successivamente. Tutto ciò comporta dei vantaggi, quali:

1. Sicurezza: Le Prepared Statement prevengono gli attacchi di SQL-injection. Poiché i valori dei parametri sono forniti separatamente dalla query SQL, non è possibile inserire del codice SQL malevolo direttamente all'interno della query.
2. Prestazioni: Le Prepared Statement possono essere precompilate e ottimizzate dal database, consentendo di riutilizzarle più volte con diversi valori di input. Questo può portare a un miglioramento delle prestazioni in confronto alle query SQL tradizionali.
3. Leggibilità: Utilizzando segnaposto per i valori dei parametri, è possibile rendere le query SQL più leggibili e manutenibili, separando la logica SQL dai dati specifici.
4. Compatibilità: Le Prepared Statement sono supportate da quasi tutti i database relazionali e sono implementate nelle librerie di connessione al database per molti linguaggi di programmazione.

ACCESSO A TVGEEK

TVGeek presenta il form di accesso con una query che controlla che i dati immessi siano effettivamente esistenti nel database. Per verificare che l'username esista effettua la query

```
SELECT id, username, password FROM users WHERE username = ?
```

Anche qui, ho fatto uso di Prepared Statement.

La password dell'utente è crittografata con una hash function. Se l'username esiste, si controlla che la password dell'utente combaci con quella memorizzata nel database; in caso affermativo, l'utente ha accesso al proprio account TVGeek.

SERVIZIO TVGEEK

Una volta che l'utente decide di trovare nuovi contenuti, il sistema TVGeek fa uso di varie query per portare l'utente al risultato/ai risultati finali.

Come prima cosa, viene proposto di scegliere il genere/i generi da guardare. Inizialmente, pensavo di mostrare tutta la colonna *genres* della tabella *all_movies_details_clean*, ma, eseguendo la query, mi sono resa conto che fosse infattibile per un utente scegliere fra oltre 7000 risultati

```
Showing rows 0 - 24 (7619 total, Query took 0.0005 seconds.)  
.....  
SELECT DISTINCT genres FROM all_movies_details_cleaned WHERE genres IS NOT NULL;
```

Inoltre, era poco proficuo per il sistema far vedere, potenzialmente, moltissimi contenuti fra cui scegliere, implicherebbe un utente che non riesce effettivamente a flirtare le informazioni e

non essere soddisfatto del servizio. Di conseguenza, ho deciso di impostare la query nel seguente modo

```
SELECT DISTINCT genres
FROM all_movies_details_clear
WHERE genres IS NOT NULL ORDER BY RAND() LIMIT 50
```

genres
Drama Horror Action Thriller Crime Foreign
History Action Drama Romance
Thriller Science Fiction Horror
TV Movie Drama Crime Mystery
TV Movie Drama Romance Thriller
Adventure Romance Drama
Fantasy Adventure Animation Comedy

Alcune delle 50 righe ritornate dalla query

Così l'utente visualizzerà, ordinati in maniera casuale, 50 differenti generi cinematografici e/o televisivi. La selezione per l'utente avviene mediante checkbox e, una volta fatto, preme il bottone *proseguì*, che si occupa di inviare il risultato alla pagina successiva. Tuttavia, ho realizzato che un utente, per qualsivoglia ragione, possa non effettuare una scelta, di conseguenza ho riformulato la stessa query

```
SELECT DISTINCT genres
FROM all_movies_details_clear
WHERE genres IS NOT NULL ORDER BY RAND() LIMIT 20
```

Questa volta con 20 risultati, in quanto l'obiettivo di TVGeek è quello di ridurre, mano a mano che l'utente si addentra nella determinazione dei parametri, il numero di contenuti con cui l'utente è affine. Il risultato, che provenga da una o dall'altra query, serve per poter eseguire la seguente query

```
SELECT DISTINCT production_countries
FROM all_movies_details_clear
WHERE genres IN ($genreConditions)
    AND production_countries IS NOT NULL
```

L'obiettivo questa volta è recuperare i paesi di produzione per permettere all'utente di selezionarli con una checkbox e, come nel caso dei generi, se non viene effettuata almeno una scelta al momento dell'invio con *proseguì*, verrà eseguita una query che genera risultati in modo random

```
SELECT DISTINCT production_countries
FROM all_movies_details_clear
WHERE genres IN ($genreConditions) AND production_countries IS NOT NULL
ORDER BY RAND() LIMIT 15
```

production_countries
Finland
United States of America
France
Italy
United Kingdom
Brazil
Portugal
Germany
Australia
Japan
Hong Kong

Alcune delle righe ritornate dalla query, con generi scelti Drama|Crime, Mystery|Drama, Action|Crime e Science Fiction|Drama|Romance

sempre con l'obiettivo di limitare i danni di una non-scelta, ma comunque basandosi sulle preferenze espresse sui generi, o comunque quelli ritornati dalla query che li estrae random.

La scelta di ordinare anche i paesi randomicamente è quella di non mostrare i paesi sempre nello stesso ordine, ma dare all'utente la possibilità di scoprire paesi di produzione che magari non conosce e lo incuriosiscono.

Infine, i risultati delle query appena descritte servono per andare ad elaborare la seguente query, ossia quella che estrae le date di rilascio del contenuto sulla base dei valori di generi e paese di produzione

```
SELECT DISTINCT formatted_release_date
FROM (
    SELECT DATE_FORMAT(STR_TO_DATE(release_date, '%d/%m/%Y'), '%d/%m/%Y')
        AS formatted_release_date
    FROM all_movies_details_clear
    WHERE genres IN ($genreConditions)
        AND production_countries IN ($prodCCConditions)
        AND release_date IS NOT NULL
) AS subquery
ORDER BY STR_TO_DATE(formatted_release_date, '%d/%m/%Y') ASC;
```

formatted_release_date
01/09/1905
07/10/1905
29/03/1906
10/11/1906
02/03/1907
19/06/1908
28/01/1909
20/12/1913
22/02/1915
03/06/1916
30/06/1916
30/06/1917
04/05/1918
01/06/1919
14/12/1919
25/12/1919
29/02/1920
23/08/1920
19/12/1920
01/12/1921
01/03/1922
03/09/1922
28/02/1924
31/05/1924
01/06/1924

Sulla base della query nell'immagine precedente e la scelta di 'United Kingdom', 'Italy', 'United States of America', 'Hungary' e 'Norway', questi sono i risultati

Questa volta, oltre ad aver aggiunto i paesi di produzione alla query, l'ordine di presentazione dei valori è in ordine crescente, in modo tale da dare un ordine cronologico – se un utente vuole vedere qualcosa di recente, non deve stare a cercare il periodo che gli interessa fra valori disposti in maniera casuale. Tuttavia, per farlo è necessario fare una nested-query, poiché release_date è salvata nel formato text, ma se si ordina in modo crescente è valutata come carattere, non come data.

A questo punto, per l'utente è possibile visualizzare i contenuti consigliati, grazie alla query

```
SELECT t1.*, t2.actor1_name, t2.actor2_name, t2.actor3_name,
       t2.actor4_name, t2.actor5_name, t2.actor_number, t2.director_name,
       t2.producer_name, t2.screenplay_name, t2.editor_name,
       (
           SELECT country
           FROM language_to_country
           WHERE language_to_country.language = t2.original_language
       ) AS country
FROM all_movies_casting_raw AS t1
LEFT JOIN all_movies_details_clear AS t2 ON t1.id = t2.id
WHERE t1.id = $id;
```

Da cui si recuperano tutti gli attributi che definiscono il contenuto, grazie al vincolo d'integrità dato dall'attributo id. Qui è presente una nested-query per indicare il paese di provenienza della lingua parlata nel film. Nella tabella *all_movies_details_clear* è presente il

Inoltre, grazie al vincolo d'integrità referenziale con t2.director_name, si ha la possibilità di visualizzare le informazioni su biografia e premi vinti dal regista, se l'informazione è disponibile.

La query che recupera le informazioni è la seguente

```
SELECT t1.summary, t2.original_language, t3.tmdbID,
       t3.imdbID, t3.total_awards
  FROM favorite_directors_summary AS t1
       LEFT JOIN most_common_language_by_director AS t2
      ON t1.director = t2.director_name
       LEFT JOIN acclaimed_directors_awards AS t3 ON t1.director = t3.name
 WHERE director = '$director_name';
```

L'obiettivo iniziale era recuperare, tramite il codice IMDb oppure TMBD, un'immagine del regista. Tuttavia, per ragioni di copyright ciò non è stato possibile, ma comunque all'utente vengono mostrati i valori, nel caso in cui voglia recuperare ulteriori informazioni da tali siti web.

In conclusione, per recuperare i singoli premi vinti dal regista viene fatta una discriminazione. Se il regista è Steven Spielberg, la query eseguita è

```
SELECT award_ceremony, award_year, award_category, outcome
  FROM (
    SELECT DISTINCT adb.ceremony AS award_ceremony,
                   adb.award_year AS award_year,
                   adb.category AS award_category, adb.outcome AS outcome
      FROM all_movies_casting_raw AS amcr
           LEFT JOIN awards_by_directors AS adb
          ON amcr.director_name = adb.director_name
     WHERE amcr.director_name = 'Steven Spielberg'

    UNION

    SELECT DISTINCT sa.ceremony AS award_ceremony,
                   sa.award_year AS award_year,
                   sa.category AS award_category, sa.outcome AS outcome
      FROM all_movies_casting_raw AS amcr
           LEFT JOIN spielberg_awards AS sa ON amcr.director_name = sa.name
     WHERE amcr.director_name = 'Steven Spielberg'
  ) AS combined_results;
```

Recuperando tutti i valori distinti presenti nelle tabelle contenenti elenchi di premi – non si esclude l'ipotetica presenza di valori in entrambe le tabelle, cosa accertata in fase di progettazione.

award_ceremony	award_year	award_category	outcome
AACTA International Awards	2013	AACTA International Award	Nominated
AARP Movies for Grownups Awards	2003	Movies for Grownups Award	Nominated
AARP Movies for Grownups Awards	2006	Movies for Grownups Award	Won
AARP Movies for Grownups Awards	2013	Movies for Grownups Award	Won
Academy Awards, USA	1978	Oscar	Nominated
Academy Awards, USA	1982	Oscar	Nominated
Academy Awards, USA	1983	Oscar	Nominated
Academy Awards, USA	1986	Oscar	Nominated

Alcuni dei 342 risultati della query

In alternativa, la query viene elaborata nel seguente modo

```
SELECT DISTINCT adb.ceremony AS award_ceremony, adb.award_year AS award_year,
adb.category AS award_category, adb.outcome AS adb_outcome,
adb.original_language AS ol
FROM all_movies_casting_raw AS amcr
LEFT JOIN awards_by_directors AS adb
ON amcr.director_name = adb.director_name
WHERE amcr.director_name = '$director_name';
```

award_ceremony	award_year	award_category	adb_outcome	ol
AFI Fest	2014	Audience Award	Won	ab
Athens International Film Festival	2014	Golden Athena	Nominated	ab
Cinema Writers Circle Awards, Spain	2015	CEC Award	Nominated	ab
European Film Awards	2014	European Discovery of the award_year	Nominated	ab
Feroz Awards, ES	2015	Feroz Award	Nominated	ab
Gaudi Awards	2015	Gaudi Award	Nominated	ab
Gaudi Awards	2015	Gaudi Award	Won	ab
Gaudi Awards	2016	Gaudi Award	Won	ab
Goya Awards	2015	Goya	Won	ab
Málaga Spanish Film Festival	2014	Golden Biznaga	Won	ab
Málaga Spanish Film Festival	2014	Silver Biznaga	Won	ab
Oldenburg Film Festival	2014	German Independence Award - Audience Award	Nominated	ab
Palm Springs International Film Festival	2015	Cine Latino Award	Nominated	ab
Sant Jordi Awards	2015	Sant Jordi	Won	ab
Seattle International Film Festival	2014	New Director's Showcase Award	Won	ab
Stockholm Film Festival	2014	Telia Film Award	Won	ab
SXSW Film Festival	2014	SXSW Grand Jury Award	Nominated	ab
Transilvania International Film Festival	2015	Transilvania Trophy	Nominated	ab
Turia Awards	2015	Turia Award	Won	ab

Risultato della query con il regista Carlos Marques-Marcet

Ovviamente, potrebbero esserci registi non premiati, quindi potrebbe anche non ritornare righe.