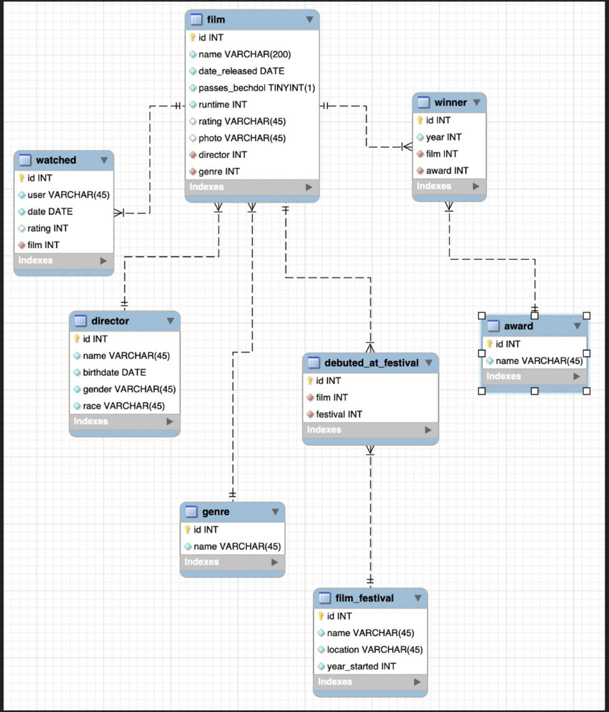
cinefile

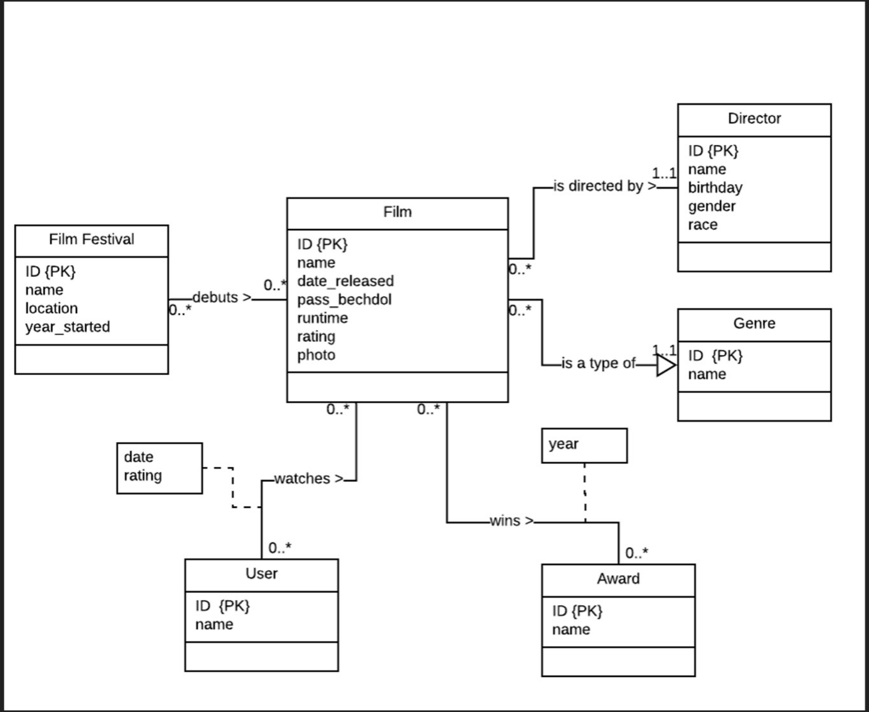
An app to locally store films that you have watched

Abby Carr, Natalie Duerr, and Kia Zafar
CS3200 - Section 1
CarrAbbDuerrNatZafarKia
Project Presentation

Project Schema



EER Diagram



UML Diagram

This is our schema.

The center is the *Film* entity.

Film is directed by *Director*.

Film debuts at *Film Festival(s)*.

Film wins *Award(s)*.

Film is a type of *Genre*.

Additionally, *Users* can watch and rate *Films*.

How it works

- MySQL database
- React front-end
- Material UI library
- Node.js to connect front-end and MySQL

What the user can do

- Go to **Add Film** to add a new film to the database.
- Go to **Account** to view the list of your watched films, ratings, and data visualizations about them.
    - There is an option at the top of the page for the user to log-out, and an option at the bottom for the user to delete their account and watched list.

What the user can do

- Log in with a unique username
- **Explore** all available films in the database
  - Users can explore all films, or films by a specific director, that have won a specific award, or debuted at a specific festival. The director, any festivals, and any awards are linked on Film Detail pages to create this navigation
- Click on a film to view information about it (including awards it has won, festivals it debuted at, and if and when film was watched).
  - To log a film they have watched, they select the date they saw the film and add an optional rating out of 5.
  - They can also update an existing “watched” entry of that film with a different date and/or rating.

CRUD Examples

C: Create

- Creating New Film

R: Read

- Reading all films

U: Update

- Updating a “Watched” entry

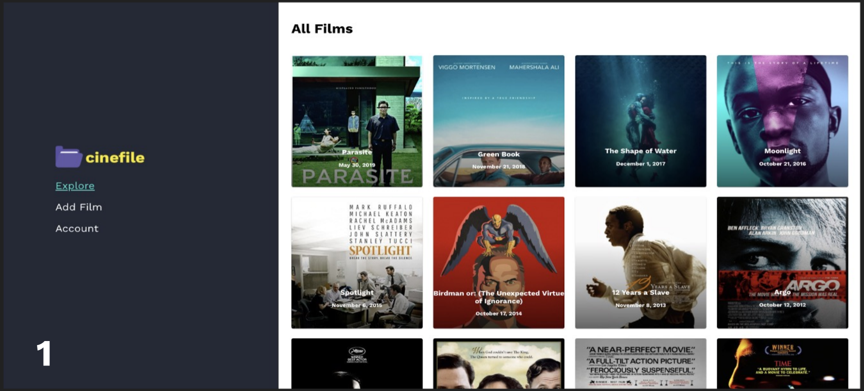
D: Delete

- Deleting a user

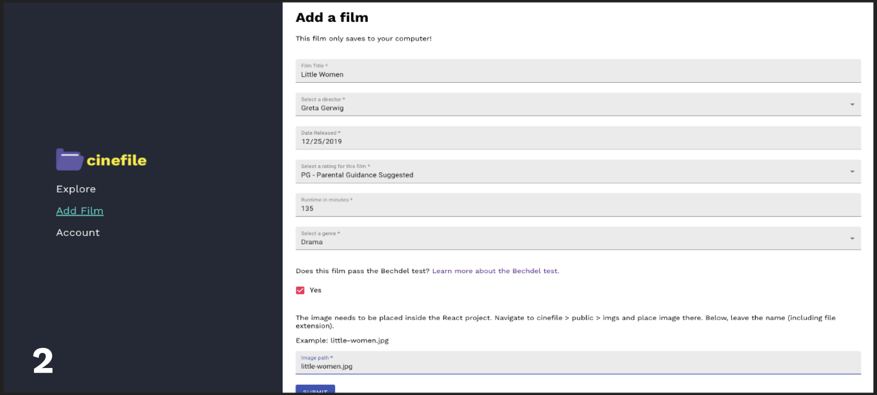
Create New Film

- CREATES new tuple in the film table.
- The film Little Women will be added:
    - **Name**: Little Women
    - **Director**: Greta Gerwig
    - **Release Date**: 12/25/19
    - **Rating**: PG
    - **Genre**: Drama
    - **Passes Bechdel**: TRUE
    - **Image Path**: /imgs/little-women.jpg

Create New Film - App


cinefile
Explore
Add Film
Account
All Films
Parasite May 30, 2019
Green Book November 21, 2018
The Shape of Water December 1, 2017
Moonlight October 21, 2016
Spotlight November 6, 2015
Birdman or: (The Unexpected Virtue of Ignorance) October 17, 2014
12 Years a Slave November 8, 2013
Argo October 12, 2012
1

User is viewing all films (Explore)



Add a film

This film only saves to your computer!

Film Title *
Little Women

Select a director *
Greta Gerwig

Date Released *
12/25/2019

Select a rating for this film *
PG - Parental Guidance Suggested

Runtime in minutes *
135

Select a genre *
Drama

Does this film pass the Bechdel test? Learn more about the Bechdel test.

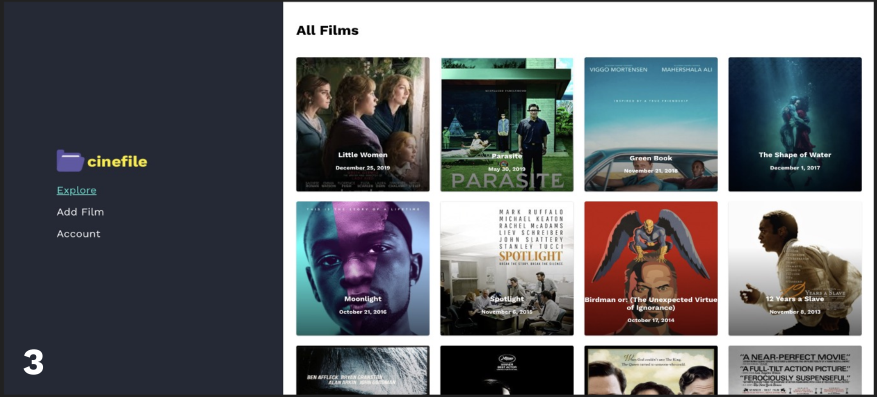
✓ Yes

The image needs to be placed inside the React project. Navigate to cinefile > public > imgs and place image there. Below, leave the name (including file extension).

Example: little-women.jpg

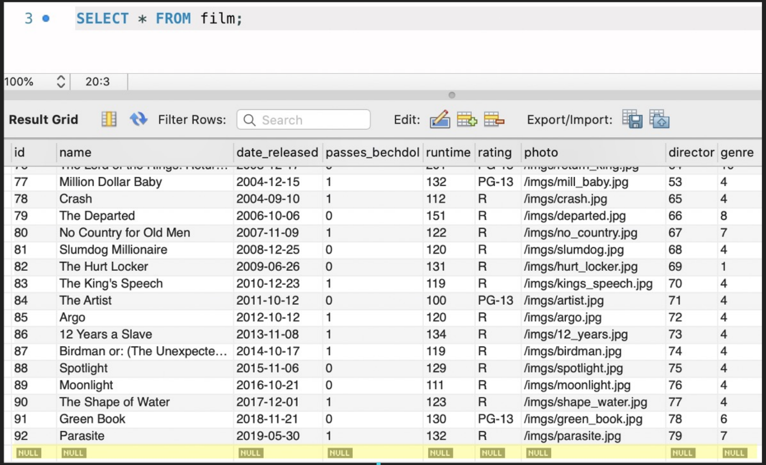
Image path *
little-women.jpg

User goes to “Add Film” and enters info for Little Women and clicks Submit


cinefile
Explore
Add Film
Account
All Films
Little Women December 25, 2019
Parasite May 30, 2019
Green Book November 21, 2018
The Shape of Water December 1, 2017
Moonlight October 21, 2016
Spotlight November 6, 2015
Birdman or: (The Unexpected Virtue of Ignorance) October 17, 2014
12 Years a Slave November 8, 2013
3

Little Women now appears in all films (Explore)

Create New Film - Database

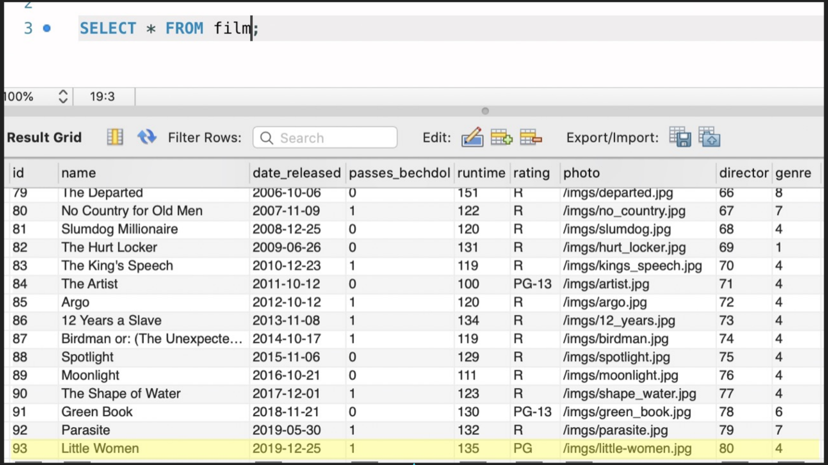


```sql
SELECT * FROM film;
```

100% 20:3

Result Grid

id	name	date_released	passes_bechdol	runtime	rating	photo	director	genre
77	Million Dollar Baby	2004-12-15	1	132	PG-13	/imgs/mill_baby.jpg	53	4
78	Crash	2004-09-10	1	112	R	/imgs/crash.jpg	65	4
79	The Departed	2006-10-06	0	151	R	/imgs/departed.jpg	66	8
80	No Country for Old Men	2007-11-09	1	122	R	/imgs/no_country.jpg	67	7
81	Slumdog Millionaire	2008-12-25	0	120	R	/imgs/slumdog.jpg	68	4
82	The Hurt Locker	2009-06-26	0	131	R	/imgs/hurt_locker.jpg	69	1
83	The King's Speech	2010-12-23	1	119	R	/imgs/kings_speech.jpg	70	4
84	The Artist	2011-10-12	0	100	PG-13	/imgs/artist.jpg	71	4
85	Argo	2012-10-12	1	120	R	/imgs/argo.jpg	72	4
86	12 Years a Slave	2013-11-08	1	134	R	/imgs/12_years.jpg	73	4
87	Birdman or: (The Unexpecte...	2014-10-17	1	119	R	/imgs/birdman.jpg	74	4
88	Spotlight	2015-11-06	0	129	R	/imgs/spotlight.jpg	75	4
89	Moonlight	2016-10-21	0	111	R	/imgs/moonlight.jpg	76	4
90	The Shape of Water	2017-12-01	1	123	R	/imgs/shape_water.jpg	77	4
91	Green Book	2018-11-21	0	130	PG-13	/imgs/green_book.jpg	78	6
92	Parasite	2019-05-30	1	132	R	/imgs/parasite.jpg	79	7
NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

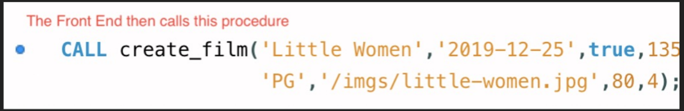


```sql
SELECT * FROM film;
```

100% 19:3

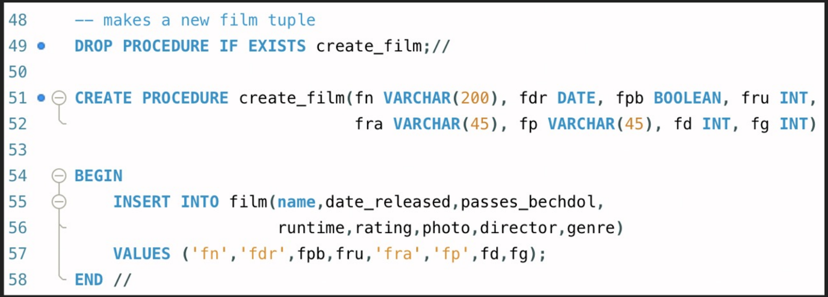
Result Grid

id	name	date_released	passes_bechdol	runtime	rating	photo	director	genre
79	The Departed	2006-10-06	0	151	R	/imgs/departed.jpg	66	8
80	No Country for Old Men	2007-11-09	1	122	R	/imgs/no_country.jpg	67	7
81	Slumdog Millionaire	2008-12-25	0	120	R	/imgs/slumdog.jpg	68	4
82	The Hurt Locker	2009-06-26	0	131	R	/imgs/hurt_locker.jpg	69	1
83	The King's Speech	2010-12-23	1	119	R	/imgs/kings_speech.jpg	70	4
84	The Artist	2011-10-12	0	100	PG-13	/imgs/artist.jpg	71	4
85	Argo	2012-10-12	1	120	R	/imgs/argo.jpg	72	4
86	12 Years a Slave	2013-11-08	1	134	R	/imgs/12_years.jpg	73	4
87	Birdman or: (The Unexpecte...	2014-10-17	1	119	R	/imgs/birdman.jpg	74	4
88	Spotlight	2015-11-06	0	129	R	/imgs/spotlight.jpg	75	4
89	Moonlight	2016-10-21	0	111	R	/imgs/moonlight.jpg	76	4
90	The Shape of Water	2017-12-01	1	123	R	/imgs/shape_water.jpg	77	4
91	Green Book	2018-11-21	0	130	PG-13	/imgs/green_book.jpg	78	6
92	Parasite	2019-05-30	1	132	R	/imgs/parasite.jpg	79	7
93	Little Women	2019-12-25	1	135	PG	/imgs/little-women.jpg	80	4



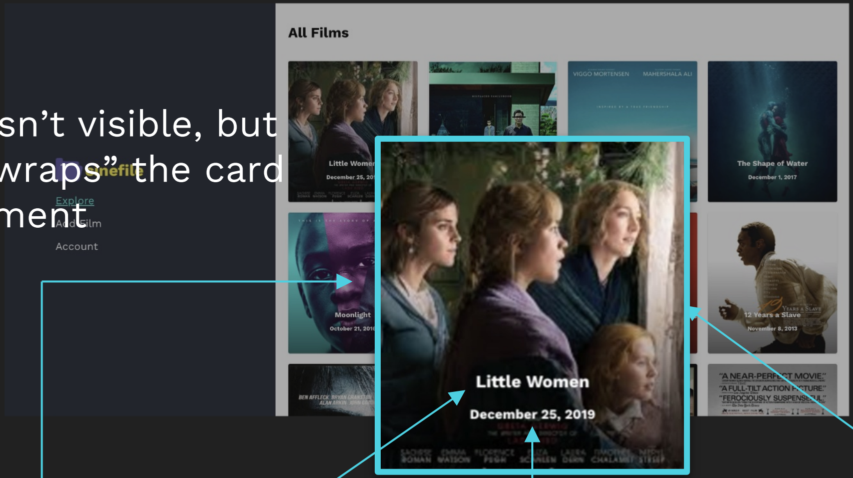
The Front End then calls this procedure

```sql
CALL create_film('Little Women','2019-12-25',true,135
                 'PG','/imgs/little-women.jpg',80,4);
```

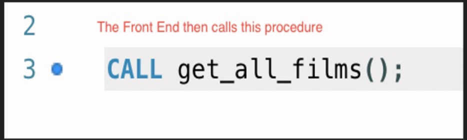


```sql
    -- makes a new film tuple
    DROP PROCEDURE IF EXISTS create_film;//

    CREATE PROCEDURE create_film(fn VARCHAR(200), fdr DATE, fpb BOOLEAN, fru INT,
                                 fra VARCHAR(45), fp VARCHAR(45), fd INT, fg INT)

    BEGIN
        INSERT INTO film(name,date_released,passes_bechdol,
                         runtime,rating,photo,director,genre)
        VALUES ('fn','fdr',fpb,fru,'fra','fp',fd,fg);
    END //
```

Reading All Films

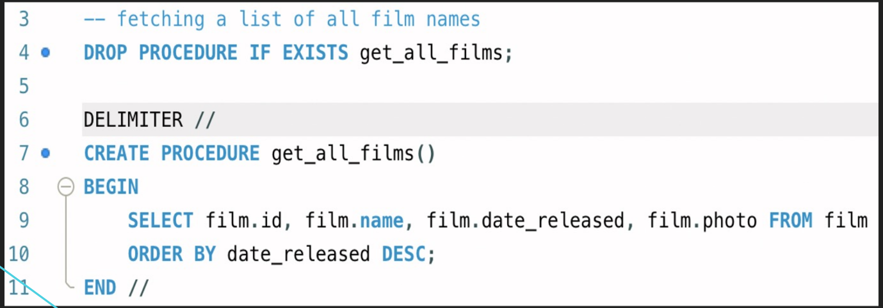


ID isn't visible, but it "wraps" the card element

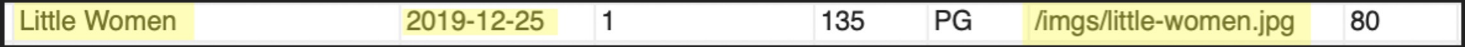


The Front End then calls this procedure

```
CALL get_all_films();
```



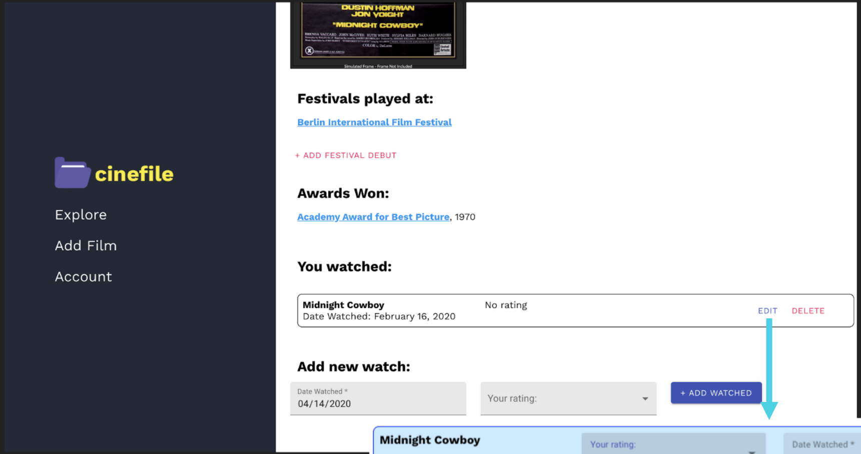
```
-- fetching a list of all film names
DROP PROCEDURE IF EXISTS get_all_films;

DELIMITER //
CREATE PROCEDURE get_all_films()
BEGIN
    SELECT film.id, film.name, film.date_released, film.photo FROM film
    ORDER BY date_released DESC;
END //
```



Little Women	2019-12-25	1	135	PG	/imgs/little-women.jpg	80

Updating a “Watched” Entry

- UPDATES existing tuple in the watched table.
- User “natalie” updates her “watched” entry of “Midnight Cowboy” to have rating of 4.


cinefile

Explore

Add Film

Account

Festivals played at:

Berlin International Film Festival

+ ADD FESTIVAL DEBUT

Awards Won:

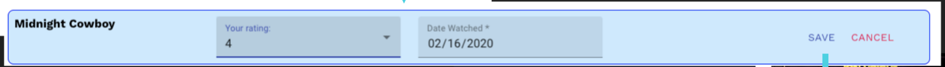
Academy Award for Best Picture, 1970

You watched:

Midnight Cowboy
Date Watched: February 16, 2020
No rating

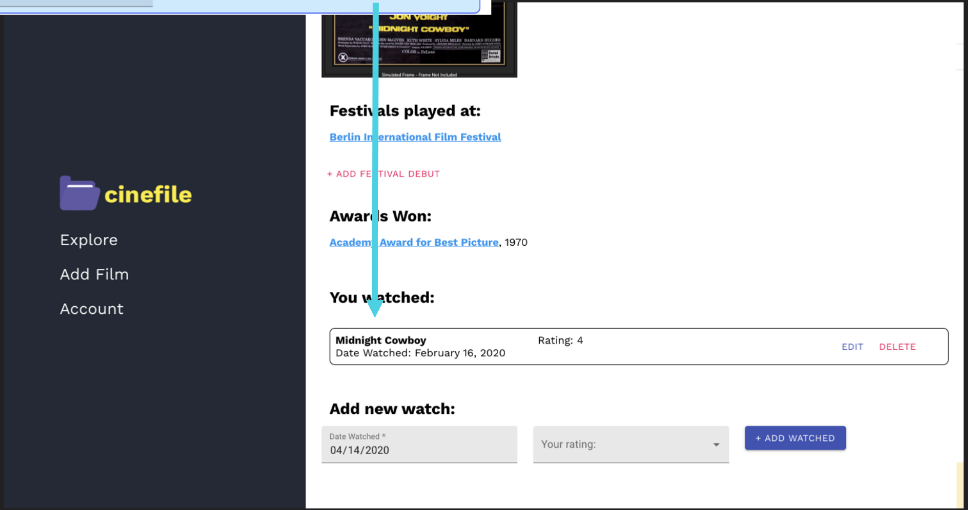
Add new watch:

Date Watched *
04/14/2020


Midnight Cowboy

Your rating:
4

Date Watched *
02/16/2020


cinefile

Explore

Add Film

Account

Festivals played at:

Berlin International Film Festival

+ ADD FESTIVAL DEBUT

Awards Won:

Academy Award for Best Picture, 1970

You watched:

Midnight Cowboy
Date Watched: February 16, 2020
Rating: 4

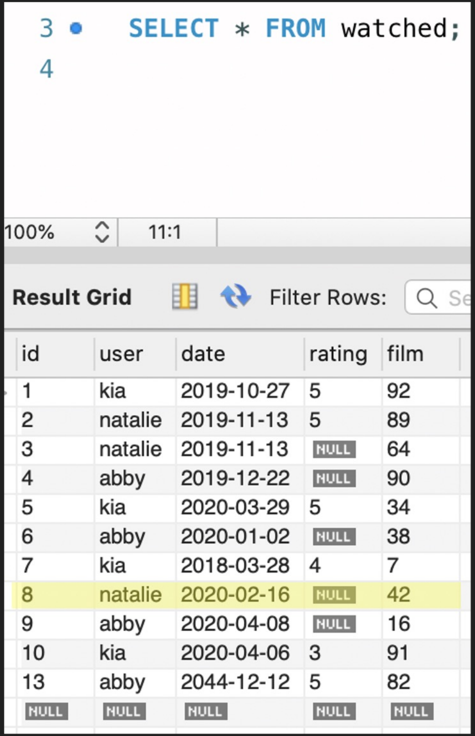
Add new watch:

Date Watched *
04/14/2020

Updating a “Watched” Entry - App

Updating a “Watched” Entry - Database

UPDATES existing tuple in the watched table.

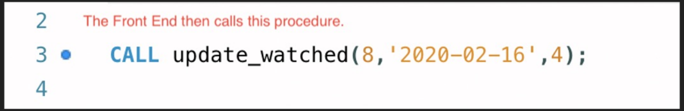


```
SELECT * FROM watched;
```

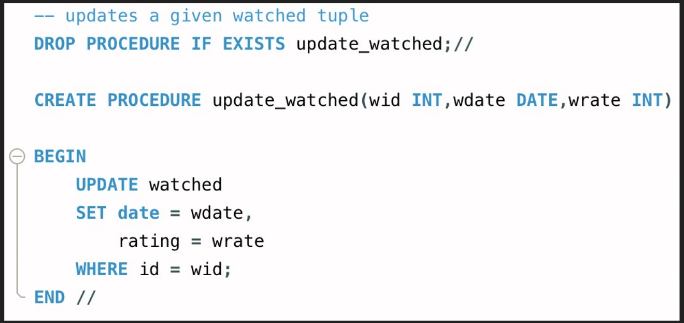
100% 11:1

Result Grid Filter Rows:

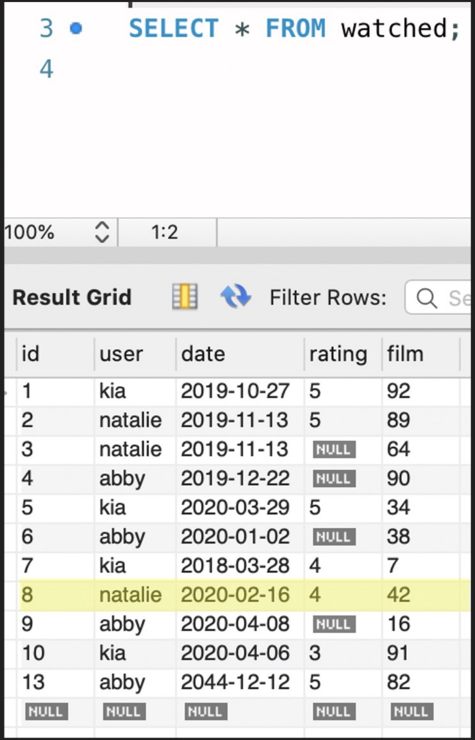
id	user	date	rating	film
1	kia	2019-10-27	5	92
2	natalie	2019-11-13	5	89
3	natalie	2019-11-13	NULL	64
4	abby	2019-12-22	NULL	90
5	kia	2020-03-29	5	34
6	abby	2020-01-02	NULL	38
7	kia	2018-03-28	4	7
8	natalie	2020-02-16	NULL	42
9	abby	2020-04-08	NULL	16
10	kia	2020-04-06	3	91
13	abby	2044-12-12	5	82
NULL	NULL	NULL	NULL	NULL



```
The Front End then calls this procedure.
CALL update_watched(8,'2020-02-16',4);
```



```
-- updates a given watched tuple
DROP PROCEDURE IF EXISTS update_watched;//

CREATE PROCEDURE update_watched(wid INT,wdate DATE,wrate INT)

BEGIN
    UPDATE watched
    SET date = wdate,
        rating = wrate
    WHERE id = wid;
END //
```



```
SELECT * FROM watched;
```

100% 1:2

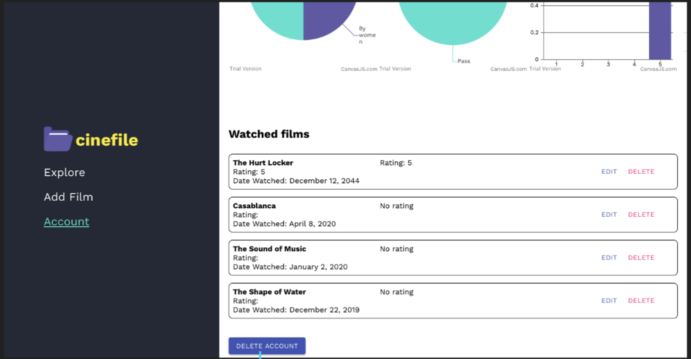
Result Grid Filter Rows:

id	user	date	rating	film
1	kia	2019-10-27	5	92
2	natalie	2019-11-13	5	89
3	natalie	2019-11-13	NULL	64
4	abby	2019-12-22	NULL	90
5	kia	2020-03-29	5	34
6	abby	2020-01-02	NULL	38
7	kia	2018-03-28	4	7
8	natalie	2020-02-16	4	42
9	abby	2020-04-08	NULL	16
10	kia	2020-04-06	3	91
13	abby	2044-12-12	5	82
NULL	NULL	NULL	NULL	NULL

Deleting a User

- DELETES all tuples in the watched table for a specified user
- Deleting user “abby”

Deleting a User - App



Watched films

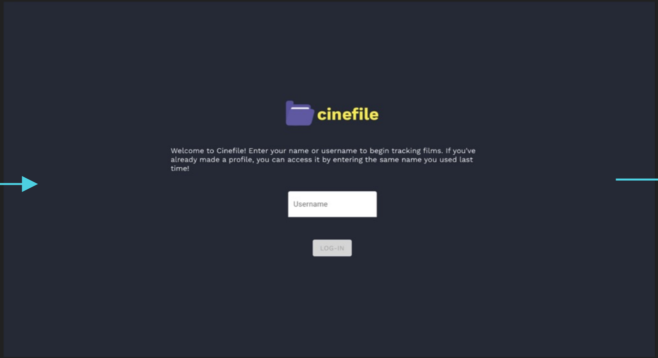
The Hurt Locker Rating: 5
Rating: 5
Date Watched: December 12, 2044

Casablanca No rating
Rating:
Date Watched: April 8, 2020

The Sound of Music No rating
Rating:
Date Watched: January 2, 2020

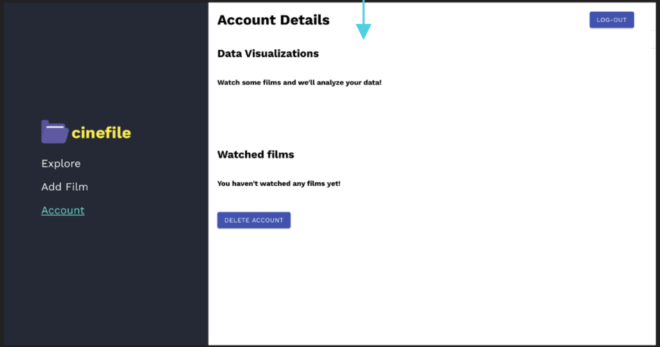
The Shape of Water No rating
Rating:
Date Watched: December 22, 2019

User “abby” clicks “Delete Account”. Their account is deleted and they are brought back to log-in



Welcome to Cinefile! Enter your name or username to begin tracking films. If you've already made a profile, you can access it by entering the same name you used last time!

If the user logs back in as “abby”, their previous watched history is gone



Account Details

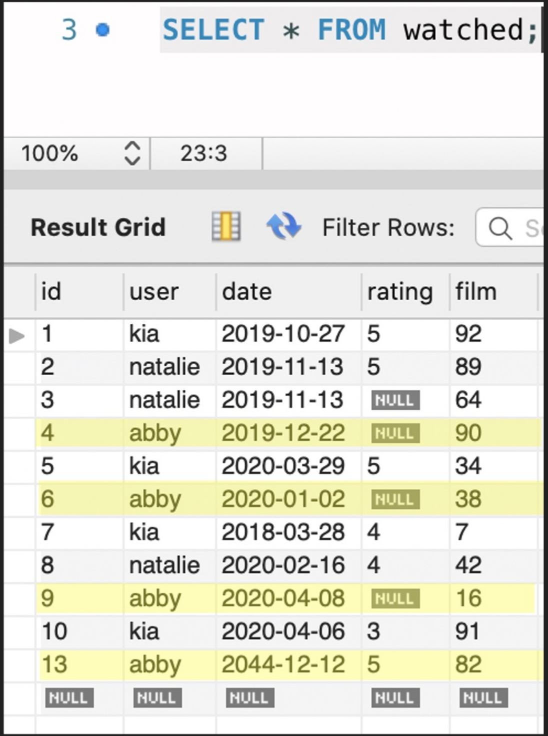
Data Visualizations

Watch some films and we'll analyze your data!

Watched films

You haven't watched any films yet!

Deleting a User - Database



```
SELECT * FROM watched;
```

100% 23:3

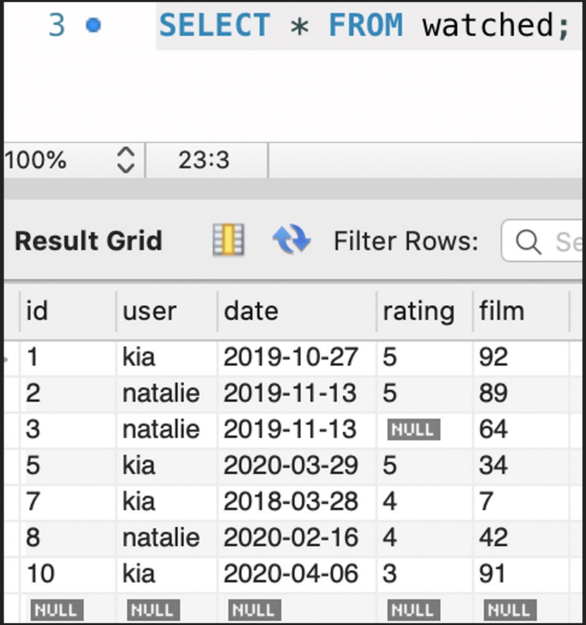
Result Grid Filter Rows:

id	user	date	rating	film
1	kia	2019-10-27	5	92
2	natalie	2019-11-13	5	89
3	natalie	2019-11-13	NULL	64
4	abby	2019-12-22	NULL	90
5	kia	2020-03-29	5	34
6	abby	2020-01-02	NULL	38
7	kia	2018-03-28	4	7
8	natalie	2020-02-16	4	42
9	abby	2020-04-08	NULL	16
10	kia	2020-04-06	3	91
13	abby	2044-12-12	5	82
NULL	NULL	NULL	NULL	NULL

The Front End then calls this procedure

```
CALL delete_user('abby');
```

```
-- deletes a user from the watched table
DROP PROCEDURE IF EXISTS delete_user;//

CREATE PROCEDURE delete_user(username VARCHAR(45))

BEGIN
    DELETE FROM watched
    WHERE watched.user = username
    AND watched.id > -1;
END //
```



```
SELECT * FROM watched;
```

100% 23:3

Result Grid Filter Rows:

id	user	date	rating	film
1	kia	2019-10-27	5	92
2	natalie	2019-11-13	5	89
3	natalie	2019-11-13	NULL	64
5	kia	2020-03-29	5	34
7	kia	2018-03-28	4	7
8	natalie	2020-02-16	4	42
10	kia	2020-04-06	3	91
NULL	NULL	NULL	NULL	NULL

Thank you!

Questions? Reach out to us:

- **Abby Carr**: carr.ab@husky.neu.edu
- **Natalie Duerr**: duerr.n@husky.neu.edu
- **Kia Zafar**: zafar.k@husky.neu.edu