

Database Final Report

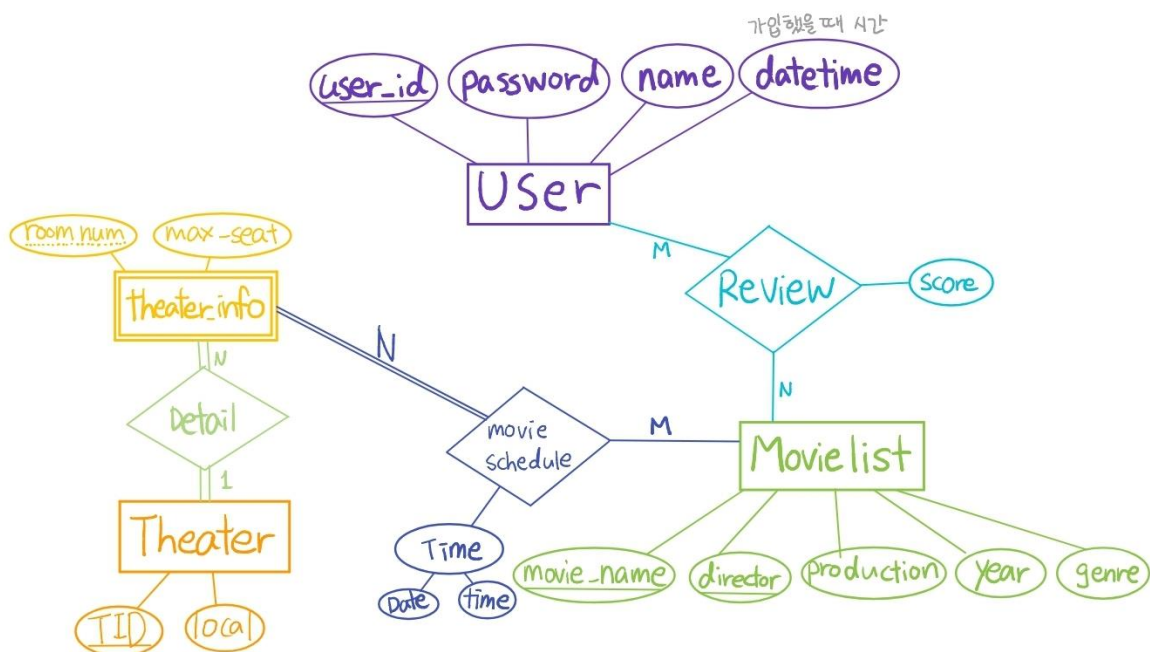
2016311981 유재영

Overview

Topic : Movie Portal(영화 페이지)

이번 데이터베이스 프로젝트의 5가지 주제 중 4번째 주제인 영화 페이지를 제작하였습니다. 시중의 영화 페이지에는 영화의 정보, 영화관의 정보, 영화 예매, 영화 리뷰 등의 서비스를 제공하고 있으며 사람들은 해당 페이지를 이용하여 영화를 보다 편리하게 즐길 수 있습니다. 이번 프로젝트에서는 해당 사이트들을 모방하여 프로젝트 주제에 주어진 기능들 중 5가지 기능; 로그인 관련 기능, 영화의 포스터&트레일러, 영화에 대한 리뷰 및 리뷰의 작성, 전체 영화를 장르나 기준을 통한 정렬, 영화관에 따라 상영을 하고 있는 영화 리스트 기능을 구현해보았습니다. 해당 프로젝트는 데이터베이스의 정보들을 어떻게 이용하고 실제로 어떤 식으로 사용하는지, 어떻게 적용하고 구현하는 지를 배우는 데에 목적으로 하고 있습니다. 다양한 프론트엔드 기술들(html, css, javascript), PHP, MySQL을 연동하여 프로젝트를 진행하였으며 이번 프로젝트를 통해 데이터베이스 명령어에 대한 이해도를 높이고 부가적으로 웹 관련 기술들도 습득할 수 있었습니다.

ER diagram



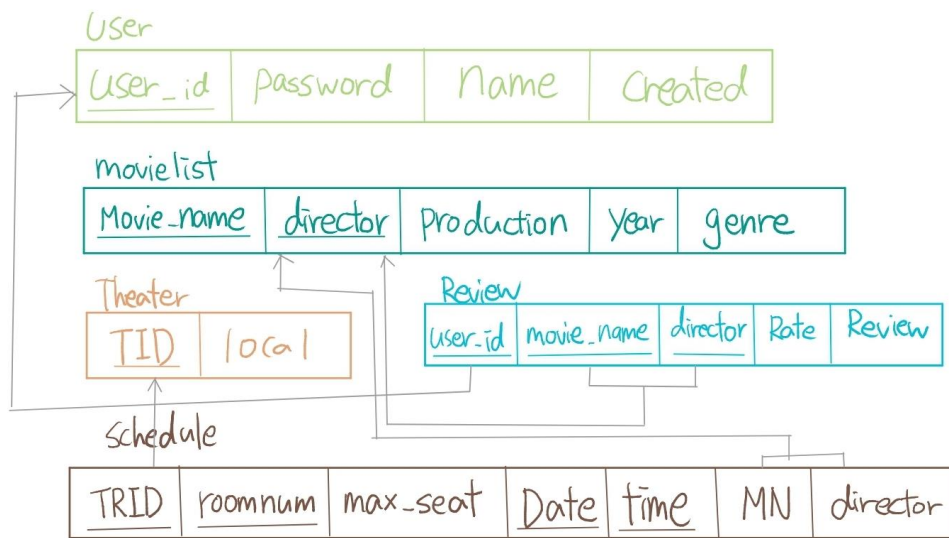
User Entity: 사용자의 정보를 담아두는 entity로 로그인 기능을 구현할 때 이용할 attribute들로 구성된 entity이다. 사용자는 영화를 보고 리뷰를 쓸 수 있으므로 movie list entity와 review에 대해서 relation을 가진다. 다양한 사용자가 다양한 영화에 대한 리뷰를 남길 수 있으므로 M:N(partial) 관계를 가진다. 사용자의 id는 겹치면 안되기 때문에 이 attribute가 key attribute가 된다.

Movielist Entity: 영화의 정보를 담는 entity이다. 영화 이름은 같을 수 있으나 영화를 만든 감독은 달라야 하고(ex 알라딘(존 머스커), 알라딘(가이 리치)) 한 감독이 여러 영화를 만들 수 있기 때문에 영화 이름과 감독, 두 attribute가 key attribute가 된다. 영화는 사용자에 대해서 리뷰 relation을 가진다. 영화관에 대해서는 영화관에서 시간 스케줄에 따라 영화가 상영되므로 movie schedule의 relation을 가진다. 이때 한 영화관에서 다양한 영화가 상영될 수 있고 상영되지 않는 영화가 있을 수 있다. 또한 한 영화에 대해서도 여러 영화관에서 상영될 수 있으므로 M:N relation을 가진다. 한 영화관에 대해서 상영관이 1개도 없는 경우는 없으므로 weak entity에 대해 1:N total relation을 가진다.

Theater Entity: 영화관의 정보를 담고 있는 entity이다. 영화관의 위치를 알려주며 (ex CGV 강남점, CGV 수원점) 이름이 겹치는 경우도 있을 수 있으므로 (경기도 광주, 광주 광역시) 점포의 ID를 따로 지정해주고 이를 key attribute로 삼는다. 한 영화관에 상영관이 하나도 없는 경우는 없기 때문에 weak relation에 대해 total participation을 가진다.

Theater_info Entity: 영화관의 세부적인 정보를 가지고 있는 entity이다. 각 영화관에 대한 상영관의 번호와 해당 상영관의 좌석 수를 가지고 있다. 또한 상영관의 번호는 겹치면 안되므로 room number와 의존하고 있는 Theater의 TID를 key attribute로 가지게 된다. 해당 entity는 영화 상영 스케줄에 대해 영향을 끼치므로 movie list entity와 schedule relation으로 엮여 있다. 상영관에서 어느 영화도 상영하지 않는 경우는 없으므로 total participation을 가진다. 그리고 한 상영관에서 시간대에 따라 다른 영화를 방영할 수 있으므로 M:N relation을 가진다.

Relational Data Model



user와 movie list, theater entity는 그대로 relational data model로 만든다.

weak entity인 theater_info는 본인의 roomnum과 TID를 따와서 TRID attribute를 key로 한다. 그리고 movie와의 relation 관계에서의 Time(date와 time), 그리고 관계를 맺은 entity의 key인 MN(movie_name)과 director를 포함하여 relational model 'schedule'을 만든다. 이때 실제로 영화 스케줄을 고려해봤을 때, 영화관의 상영관마다 날짜와 시간이 unique해야하므로 date와 time도 key로 만들어준다. 다른 entity에서 가져온 attribute인 TRID는 Foreign Key로써 Theater의 TID를 래퍼런스한다. 또한 MN과 director도 movie list에서 가져온 attribute이므로 Foreign key로써 movie list의 Primary key인 movie_name과 director를 래퍼런스 한다.

movielist와 user entity 사이의 관계인 Review는 따로 relational model로 만든다. 각 entity의 key attribute와 score를 세부적으로 평점인 rate와 글 형식의 review(text)로 나눠서 포함시킨다. 또한 이 attribute들도 관계를 가지는 entity에서 가져온 attribute이므로 Foreign key로써 각 entity를 래퍼런스한다.

Implementation and Results

실제로 데이터베이스를 구현했을 때, user table에 편의를 위해서 PID를 추가했습니다. 단, PID와 user_id가 Primary key 지정되어 있으면 (1, yulyoo97) (2, yulyoo97)의 경우가 들어갈 수 있으므로, user_id가 중복되지 못하도록 unique key를 설정해줍니다.

```
mysql> alter table user
-> add unique key (user_id);
Query OK, 0 rows affected (0.35 sec)
Records: 0 Duplicates: 0 Warnings: 0

mysql> desc user;
```

Field	Type	Null	Key	Default	Extra
PID	int(11)	NO	PRI	NULL	auto_increment
user_id	varchar(15)	NO	PRI	NULL	
password	varchar(15)	NO		NULL	
name	varchar(15)	NO		NULL	
created	datetime	NO		NULL	

```
mysql> insert into user(user_id, password, name, created) values(
-> 'yulyoo97', 7777, '유쟁', NOW());
ERROR 1062 (23000): Duplicate entry 'yulyoo97' for key 'user_id'
mysql>
```

movielist table 역시 편의를 위해 MID를 primary key로 지정해줬습니다. 기존의 primary key인 (movie_name, director)는 unique보장을 위해 unique key 설정을 해주었고 실제로 PK-FK 관계를 맺을 때 MID만 사용하기 위해서 Primary key를 해제해주었습니다..

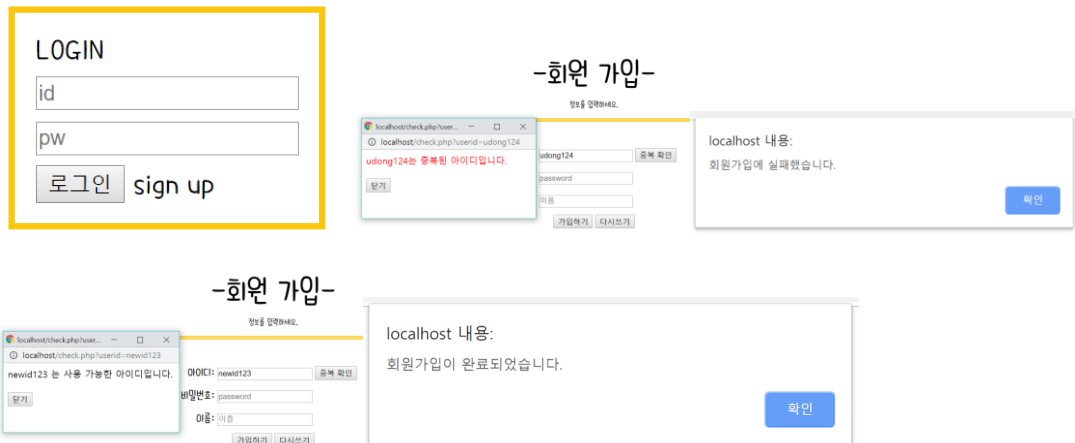
```
mysql> desc movielist;
```

Field	Type	Null	Key	Default	Extra
MID	int(11)	NO	PRI	NULL	auto_increment
movie_name	varchar(50)	NO	MUL	NULL	
production	varchar(20)	NO		NULL	
year	int(4)	YES		NULL	
genre	varchar(10)	NO		NULL	
director	varchar(30)	NO		NULL	

```
6 rows in set (0.01 sec)

mysql> alter table movielist add constraint
-> UNIQUE KEY(movie_name, director);
Query OK, 0 rows affected, 1 warning (0.19 sec)
Records: 0 Duplicates: 0 Warnings: 1
```

1) 로그인 관련 기능



[아이디 중복확인 기능]

```
<input type="text" name="userid" id="uid" placeholder="ID">&nbsp;<input type="button" value="중복 확인" onClick="check_id()" class=m_box><br>
```

```
<script>
function check_id()
{
    var userid = document.getElementById("uid").value;
    if(userid){
        url="check.php?userid="+userid;
        window.open(url,"chkid","width=300, height=100");
    }else{
        alert("아이디를 입력하세요");
    }
}
</script>
```

중복확인의 버튼을 누르면 자바 스크립트의 check_id()함수를 실행하게 된다. 이 함수는 중복확인 버튼 앞의 아이디를 칠 수 있는 인풋의 html id값인 uid를 받아와서 check.php로 해당 아이디를 GET방식으로 넘긴다. 이때 페이지가 넘어가는 것이 아니라 window.open을 통해 작은 창을 열게 된다. 만약 아이디 input이 비어있으면 alert를 통해 아이디를 입력하라는 경고창의 띄운다.

```
$uid = $_GET["userid"];
$sql = "select * from user where user_id='".$uid."'";
$member = mysqli_query($conn, $sql);
if(!$member->num_rows)
```

check.php에는 GET방식으로 넘어온 userid를 통해 쿼리를 만들 수 있고 해당 아이디를 user table에서 찾게 된다. 이 때 결과로 나온 row의 개수에 따라 중복된 아이디인지 아닌지 알 수 있다. row가 존재하지 않는다면 등록되어 있지 않은 아이디므로 사용 가능하다는 문구를 내보내고, 존재한다면 이미 등록되어 있으므로 중복된 아이디라는 문구를 보낸다.

가입하기 버튼을 누르면 signup_create.php로 POST형식으로 넘어가게 된다.

[회원가입 기능]

```
$sql = "
    INSERT INTO user
      (user_id, password, name, created)
    VALUES(
      '{$_POST['userid']}',
      '{$_POST['pw']}',
      '{$_POST['user_name']}',
      NOW()
    )
  ";
```

해당 쿼리를 데이터베이스에 등록하도록 하는 mysqli_query() 함수를 이용하는데 쿼리가 적용되지 않는다면 false를 리턴한다. 이러한 경우 false를 받았을 경우 '회원가입에 실패했습니다.'라는 alert창을 띄우고 전 페이지로 돌아간다. 만약 쿼리가 정상적으로 잘 적용이 됐을 시엔 true를 리턴하므로 '회원가입이 완료되었습니다.'라는 alert창을 띄우고 mainpage.php로 돌아갑니다.

[로그인 기능]

로그인 창에 아이디와 비밀번호를 작성 후 로그인 버튼을 누르면 auto.php로 POST형식으로 넘어가게 된다.

```

$sqlquery = "SELECT * FROM user WHERE user_id='$user_id' AND password='$pw'";
$result = mysqli_query($conn, $sqlquery);

if(mysqli_num_rows($result) == 1){ //값이 하나면
    $_SESSION['user_id'] = $user_id;
    $sqlname = "SELECT * FROM user WHERE user_id='$user_id'";
    $result = mysqli_query($conn, $sqlname);
    $row = mysqli_fetch_array($result);
    $_SESSION['user_name'] = $row[3];
    $_SESSION['user_pw'] = $row[2];

<script>
    alert('로그인 성공');
    history.back(-1);
</script>

```

제일 윗줄의 쿼리를 통해 결과를 받아오게 되면 session에다가 해당 아이디 값과 해당 아이디를 가지는 사용자의 이름(비밀번호까지) session에 저장하고 로그인을 성공한다는 창을 띄운다. 실제 코드에서는 이후에 로그아웃의 버튼의 값을 받아올 시, unset()을 통해 session값 설정을 취소시키고 로그아웃 과정을 거치게 된다.

2) 영화 정렬 기능

localhost/mainpage.php

Jang's Movie

현재 상영중인 영화

정렬: | 장르: |

Movie Name	Production	Movie Director	Year	ETC
Frozen2	Disney	크리스 벅	2019	
Avengers: Endgame	Marvel	앤소니 루소	2019	
Aladdin	Disney	가이 리치	2019	
Spider-Man: Far From Home	Marvel	존 왓츠	2019	
Joker	DC	2019		
기생충	CJ ENM	봉준호	2019	
극한직업	About Film	이병헌	2019	
엑시트	외유내강	이상근	2019	
Weathering With You	Comics Wave	신카이 마코토	2019	

<- 기본값

localhost/mainpage.php?Sorted_By=Name&Genre=0&sorted=1

Jang's Movie

현재 상영중인 영화

평점 높은 순 | 장르 | 확인

Movie Name	Production	Movie Director	Year	ETC
Avengers: Endgame	Marvel	앤소니 루소	2019	Action/5,000
Spider-Man: Far From Home	Marvel	존 어츠	2019	Action/4,500
가생종	CJ ENM	봉준호	2019	Drama/4,500
Frozen2	Disney	크리스 벅	2019	Animation/4,000
국한직업	About Film	이병헌	2019	Comedy/2,000
Aladdin	Disney	개이 러차	2019	Fantasy/평점 없음
Joker	DC	토드 필립스	2019	Thriller/평점 없음
엑시트	위유패	이상근	2019	Comedy/평점 없음
Weathering With You	Comics Wave	신카이 마코토	2019	Animation/평점 없음
Avengers: Infinity War	Marvel	앤소니 루소	2018	Action/평점 없음

현재 상영중인 영화

평점 높은 순 | 장르 | 확인

Movie Name	Production	Movie Director	Year	ETC
Frozen2	Disney	크리스 벅	2019	Animation/4,000
Weathering With You	Comics Wave	신카이 마코토	2019	Animation/평점 없음

(평점 높은 순서&모든 장르)

(평점 높은 순서&Animation 장르)

(select box의 값을 저장해서 넘기는 기능을 구현하지 않아서 확인 버튼을 누르면 다시 디폴트 값으로 넘어옵니다. 값을 세팅하고 확인을 누르면 정상적으로 잘 작동합니다.)

```
<form action="mainpage.php" method="GET">
  <select name="Sorted_By" style="text-align: right" id="sort">
    <option value="0">평점 높은 순</option>
    <option value="movie_name">이름 순</option>
    <option value="year">최근 순</option>
  </select>
  <select name="Genre" style="text-align: right" id="genre">
    <option value="0">장르</option>
    <option value="Fantasy">판타지</option>
    <option value="Action">액션</option>
    <option value="Thriller">스릴러</option>
    <option value="Comedy">코미디</option>
    <option value="Animation">애니메이션</option>
    <option value="Drama">드라마</option>
  </select>
  <input type="submit" name="sorted" value="확인">
</form>
```

GET형식으로 현재 페이지로 정렬값과 장르값을 넘기게 된다.(노란 하이라이트 부분) 그리고 정렬값과 장르값의 조합에 따라 쿼리문을 다르게 한다.

```
if($sort_value == year){
  if($genre_value === '0'){
    $sqlquery = "SELECT * FROM movielist ORDER BY $sort_value DESC";
  }else{
    $sqlquery = "SELECT * FROM movielist WHERE genre='$genre_value' ORDER BY $sort_value DESC";
  }

  $result = mysqli_query($conn, $sqlquery);
  while($row = mysqli_fetch_array($result)){
    echo '<tr><td>'. $row['movie_name']. '</td><td>'. $row['production']. '</td><td>'. $row['director']. '</td><td>'. $row['year']. '</td><td>'. $row['genre']. '</td></tr>';
  }
}
```

년도 순으로 정렬하는 경우에서 장르값이 기본값인 경우와 설정된 경우의 쿼리이다.


```

}elseif ($sort_value == movie_name) {
    if($genre_value === '0'){
        $sqlquery = "SELECT * FROM movielist ORDER BY $sort_value";
    }else{
        $sqlquery = "SELECT * FROM movielist WHERE genre='$genre_value' ORDER BY $sort_value";
    }
}

$result = mysqli_query($conn, $sqlquery);
while($row = mysqli_fetch_array($result)){
    echo "<tr><td>".$row['movie_name'].</td><td>".$row['production'].</td><td>".$row['director'].</td><td>".$row['year'].</td><td>".$row['genre'].</td></tr>";
}

```

영화이름 순으로 정렬하는 경우에서 장르값이 기본값인 경우와 설정된 경우의 쿼리이다.

```

}else{
    if($genre_value === '0'){
        $sqlquery = "SELECT M.movie_name, M.production, M.director, M.year, M.genre, avg(R.rate)
        FROM movielist M
        LEFT JOIN review R
        ON M.MID = R.MID
        GROUP BY M.movie_name, M.production, M.director, M.year, M.genre, R.MID
        ORDER BY avg(R.rate) desc";
    }else{
        $sqlquery = "SELECT M.movie_name, M.production, M.director, M.year, M.genre, avg(R.rate)
        FROM movielist M
        LEFT JOIN review R
        ON M.MID = R.MID
        WHERE M.genre='$genre_value'
        GROUP BY M.movie_name, M.production, M.director, M.year, M.genre, R.MID
        ORDER BY avg(R.rate) desc";
    }
}
$result = mysqli_query($conn, $sqlquery);
while($row = mysqli_fetch_array($result)){
    if( is_null($row['avg(R.rate)'])){
        echo "<tr><td>".$row['movie_name'].</td><td>".$row['production'].</td><td>".$row['director'].</td><td>".$row['year'].</td><td style='font-size:15px'>".$row['genre'].</td><td>./평점 없음</td></tr>";
    }else{
        echo "<tr><td>".$row['movie_name'].</td><td>".$row['production'].</td><td>".$row['director'].</td><td>".$row['year'].</td><td style='font-size:15px'>".$row['genre'].</td><td>./".$row['avg(R.rate)'].</td></tr>";
    }
}
}

```

```

mysql> SELECT M.movie_name, M.production, M.director, M.year, M.genre, avg(R.rate) FROM movielist M
LEFT JOIN review R ON M.MID = R.MID GROUP BY M.movie_name, M.production, M.director, M.year, M.genre,
R.MID ORDER BY avg(R.rate) desc;

```

movie_name	production	director	year	genre	avg(R.rate)
Avengers: Endgame	Marvel	안소니 루소	2019	Action	5.0000
Spider-Man: Far From Home	Marvel	존 왓츠	2019	Action	4.5000
기생충	CJ ENM	봉준호	2019	Drama	4.5000
Frozen2	Disney	크리스 벅	2019	Animation	4.2500
극한직업	About Film	이병헌	2019	Comedy	2.0000
Aladdin	Disney	가이 리치	2019	Fantasy	NULL
Joker	DC	토드 필립스	2019	Thriller	NULL
엑시트	외유내강	이상근	2019	Comedy	NULL
Weathering With You	Comics Wave	신카이 마코토	2019	Animation	NULL
Avengers: Infinity War	Marvel	안소니 루소	2018	Action	NULL

웹페이지 상에 띄워줄 attribute들을 select한다. 이 경우에 2가지 테이블을 쓰게 된다. 영화 정보를 담고 있는 'movielist' 테이블과 평점의 정보를 담고 있는 'review' 테이블을 써야 한다. 이때 아직 평점이 작성되지 않은 영화들도 보여주기 위해서 LEFT JOIN을 썼다.

aggregate인 avg()를 썼기 때문에 나머지 attribute list들을 GROUP BY 해 주어야 하고 영화에 따라 평점이 보여지므로 GROUP BY에 R.MID도 포함시켜준다. 그리고 정렬은 평점의 내림차순이므로 ORDER BY에 avg(R.rate) desc를 적어준다.

3) 영화 검색 기능 & 트레일러, 포스터 기능

moviepage.php에서는 영화를 검색하고 영화 이름을 누를 시 트레일러와 포스터, 리뷰를 볼 수 있다.

영화를 검색하세요.

Movie Name	Production	Movie Director	Year
Frozen2	Disney	크리스 벅	2019
Avengers: Endgame	Marvel	안소니 루소	2019
Aladdin	Disney	가이 리치	2019
Spider-Man: Far From Home	Marvel	존 왓츠	2019
Joker	DC	토드 필립스	2019
기생충	CJ ENM	봉준호	2019
극한직업	About Film	이병헌	2019
엑시트	외유내강	이상근	2019
Weathering With You	Comics Wave	신카이 마코토	2019
Avengers: Infinity War	Marvel	안소니 루소	2018

전체 영화 검색

영화를 검색하세요.

Movie Name	Production	Movie Director	Year
Avengers: Endgame	Marvel	안소니 루소	2019
Avengers: Infinity War	Marvel	안소니 루소	2018

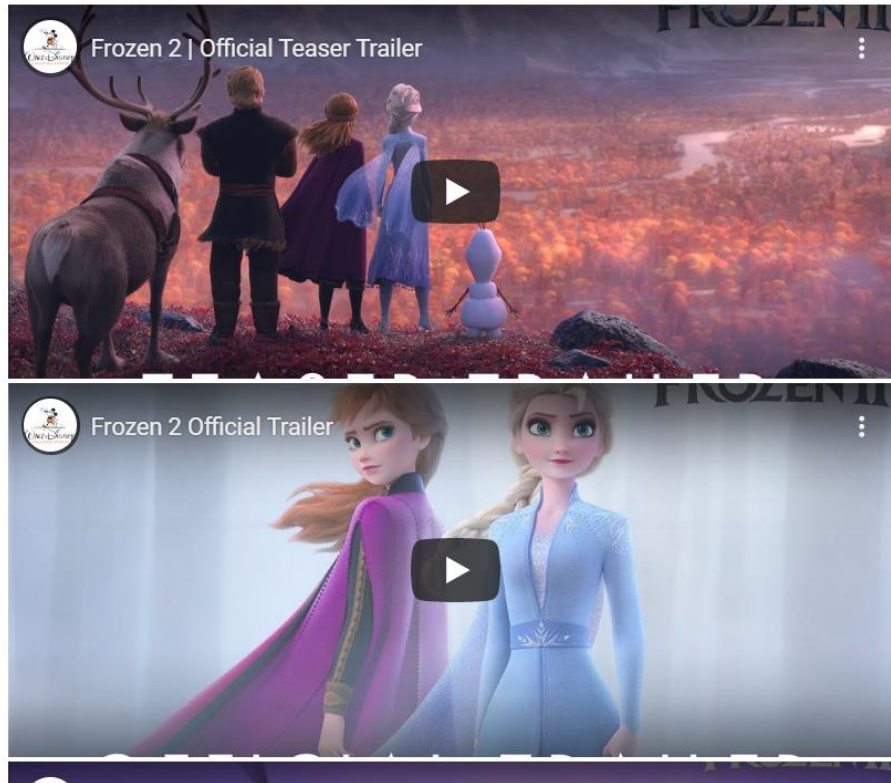
'aven' 검색

```
$sqlquery = "SELECT * FROM movielist WHERE movie_name LIKE '%$ms%'";
$result = mysqli_query($conn, $sqlquery);
while($row = mysqli_fetch_array($result)){
    echo '<tr><td><a href="moviepage.php?id='.$row['movie_name'].'">'.$row['movie_name'].'</td><td>'.$row['production'].'</td><td>'.$row['director'].'</td><td>'.$row['year'].'</td></tr>';
}
```

LIKE를 이용해서 해당 단어를 포함하는 영화를 웹 페이지에 표현한다.

echo의 html코드를 보게 되면 하이퍼링크에 id값을 담아 GET방식으로 영화 이름을 해당 페이지에 보내게 된다.

Movie Name	Production	Movie Director	Year
Frozen2	Disney	크리스 벅	2019



<localhost/moviepage.php?id=Frozen2>

영화에 해당하는 트레일러와 포스터가 나온다. GET으로 받은 id값을 통해 id에 해당하는 문서파일(트레일러와 포스터가 담겨있음)을 열어준다.

```
$sqlquery = "SELECT * FROM movielist WHERE movie_name = '$ms'";
$result = mysqli_query($conn, $sqlquery);
$row = mysqli_fetch_array($result);
echo '<tr><td><a href="moviepage.php?id='.$row['movie_name'].'">'. $row['movie_name'].'</td><td>'. $row['production'].'</td><td>'. $row['director'].'</td><td>'. $row['year'].'</td></tr>';

</table>
</div>
</div>

echo file_get_contents("data/".$row['movie_name']);
```

4) 리뷰와 평점, 리뷰 작성 기능

영화 포스터 아래에 Review기능이 존재하며, 로그인이 되어 있지 않은 상태면 리뷰를 작성할 수 없고 볼 수만 있습니다.

```
mysql> desc review;
+-----+-----+-----+-----+-----+-----+
| Field | Type          | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+-----+
| UID   | int(11)       | NO   | PRI | NULL    |       |
| user_id | varchar(15)   | NO   | PRI | NULL    |       |
| MID   | int(11)       | NO   | PRI | NULL    |       |
| Rate  | int(3)        | NO   |     | NULL    |       |
| Review | text          | YES  |     | NULL    |       |
+-----+-----+-----+-----+-----+-----+
5 rows in set (0.00 sec)
```

user의 primary key인 PID(해당 테이블에서는 UID)와 user_id, movielist의 primary key인 MID(편의를 위해 movie_name과 director대신 설정한 값) Review 테이블의 primary key를 보면 한 아이디 당 한 영화에는 리뷰를 한 번만 작성할 수 있도록 구성했다.

Review

0점 ▾

Review this movie.

리뷰쓰기

평점 - 4.5000

udong124

제시카송이 기억에 남아요. 자꾸 맴돌아요. 곧~

mizil234

제 인생작품이에요.

<- 로그인 상태에서의 영화 Review

평점 - 4.2500

yulyoo97

OST가 너무 좋아요!

udong124

겨울영국이 더 재밌었어요.

mizil234

재밌게 잘 봤습니다.

newid123

너무 재밌어요.

<- 로그아웃 상태에서의 영화 Review

```

if($_SESSION['user_id'] and isset($_GET['id'])){
    $movie_name=$_GET['id']

<hr color="#fac60e" size="10">
<div>Review</div>
<div style="text-align:center">
    <form action="moviepost.php" method="POST">
        <select name="number" id="select">
            <option value="0">0점</option>
            <option value="1">1점</option>
            <option value="2">2점</option>
            <option value="3">3점</option>
            <option value="4">4점</option>
            <option value="5">5점</option>
        </select>
        <textarea name="review" placeholder="Review this movie." style="width:400px; height:20px"></textarea>
        <?php echo '<input type="hidden" name="movie" value="'. $movie_name.' ">' ?>
        <input type="submit" value="리뷰쓰기" name="review_post">
    </form>
</div>
<div>

```

user_id값이 세션에 저장되어 있는

경우(로그인 한 상태)에만 점수 셀렉트 박스와 리뷰 작성 칸이 뜨도록 만들었다. 리뷰를 작성하게 되면 moviepost.php로 넘어가게 된다.

```

$uid = mysqli_fetch_array($result_uid)[0];
$mid = mysqli_fetch_array($result_mid)[0];

$insert_review = "INSERT INTO review VALUES($uid, '$review_id', $mid,
$review_score, '$review_text')";
$result_review = mysqli_query($conn, $insert_review);

```

```

$result_review = mysqli_query($conn, $insert_review);
if($result_review === false){

    <script type="text/javascript">
        alert('리뷰등록에 실패했습니다. ');
        history.back(-1);
    </script>

}else{

    <script type="text/javascript">
        alert('리뷰등록에 성공했습니다. ');
        history.back(-1);
    </script>

```

해당쿼리를 데이터베이스에 보내는데 이때, 한 아이디가 같은 영화에 리뷰를 쓸 수 없으므로 실패하는 경우에는 리뷰를 등록할 수 없다는 경고창을 띄운다.

```
$reviewquery = "SELECT * FROM review R WHERE MID IN (SELECT MID FROM movielist WHERE movie_name = '$movie_name')";
$review_result = mysqli_query($conn, $reviewquery);
$ratequery = "SELECT AVG(rate) FROM review WHERE MID = (SELECT MID FROM movielist WHERE movie_name = '$movie_name')";
$rate_result = mysqli_query($conn, $ratequery);
echo '<tr><th id="userid"></th><th id="userreview">평점 - '.mysqli_fetch_array($rate_result)[0].'\</th></tr>';
while($row = mysqli_fetch_array($review_result)){
    echo '<tr><td id="userid">'. $row['user_id'].'\</td><td id="userreview"> '. $row['Review'].'\</td></tr>';
}
```

두 쿼리를 이용하여 평점을 띄우고 사용자들이 적은 리뷰를 전부 보여준다.

5) 영화관과 상영 가능한 영화 리스트 기능

CINEMA: 상영관	
CINEMA	MOVIE LIST
수원	상영 가능한 영화 Aladdin Avengers: Endgame Frozen2 Joker Spider-Man: Far From Home 극한직업 가생종 엑시트
판교	상영 가능한 영화 Aladdin Avengers: Endgame Frozen2

```
mysql> select * from schedule;
```

TRID	roomnum	date	time	MID	max_seat
1	1관	2019-11-21	11:00:00	1	120
1	1관	2019-11-21	13:00:00	1	100
1	1관	2019-11-21	15:00:00	1	120
1	1관	2019-11-21	17:00:00	1	120
1	1관	2019-11-21	19:00:00	1	120
1	1관	2019-11-22	09:00:00	1	120

영화관에 대해 상영 중인 영화를 볼 수 있다. 영화관 내의 상영관에 대한 정보도 테이블에 저장되어 있으나 시간 부족으로 예매 기능을 구현하지 못했다. theater의 weak entity인 theater_info와 relation이었던 schedule을 합친 이유는 실제로 사용자가 영화를 예매할 때 사용되는 정보들을 빠르고 간편하게 접근할 수 있도록 중복되는 값이 존재하더라도 하나의 table로 만드는 선택을 했다.

```
mysql> SELECT T.TID, T.local, COUNT(DISTINCT S.MID)
-> FROM theater T, schedule S
-> WHERE T.TID = S.TRID
-> GROUP BY T.TID, T.local;
```

TID	local	COUNT(DISTINCT S.MID)
1	수원	8
2	판교	7
3	홍대	5

3 rows in set (0.00 sec)

해당 쿼리를 이용하여 각 지역의 영화관에서 상영하고 있는 영화의 수를 알아낸다. 해당 숫자를 이용하여 <td rowspan='영화관 마다 상영되는 영화의 수'>를 설정하여 html <table>을 세팅할 수 있다.

```
$movielist_query = 'SELECT DISTINCT T.TID, T.local, M.movie_name
FROM theater T, movielist M, schedule S WHERE S.TRID = T.TID AND
S.MID = M.MID AND T.TID = ' . $rsrow["TID"] . ' GROUP BY T.TID,
T.local, M.movie_name';
```

각 영화관 고유번호 TID에 해당하는 영화를 보여준다.

```
mysql> SELECT DISTINCT T.TID, T.local, M.movie_name
-> FROM theater T, movielist M, schedule S
-> WHERE S.TRID = T.TID
-> AND S.MID = M.MID
-> AND T.TID = 1
-> GROUP BY T.TID, T.local, M.movie_name;
```

TID	local	movie_name
1	수원	Aladdin
1	수원	Avengers: Endgame
1	수원	Frozen2
1	수원	Joker
1	수원	Spider-Man: Far From Home
1	수원	극한직업
1	수원	기생충
1	수원	엑시트

8 rows in set (0.00 sec)

schedule table에는 영화관의 상영관과 날짜와 시간에 따라서 같은 영화가 중복되기 때문에 DISTINCT를 써서 중복을 없애준다.