

# 最終課題

1029289895 尾崎翔太

2019/01/28

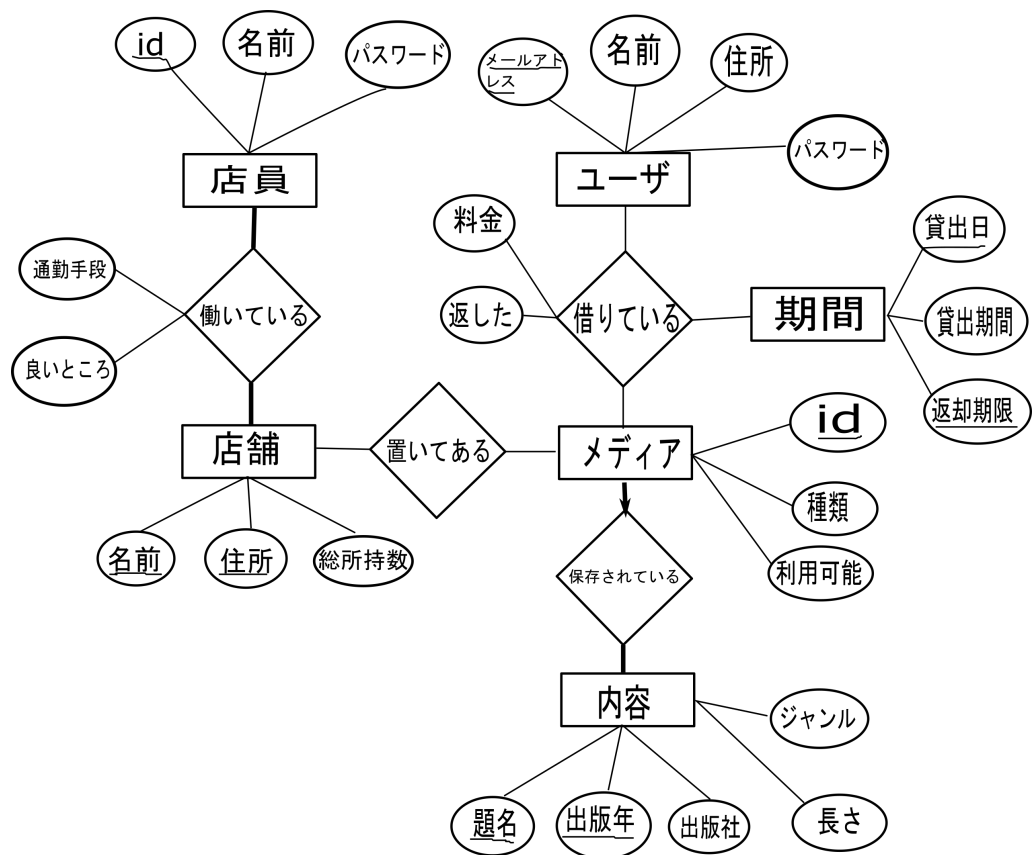


図 1: 実体関連図

## 1 システム概要

レンタルビデオ屋のシステムを作った。役割は課題 1 と同じだが、機能は課題 1 とは大きく変わっている。機能としてはユーザによる店舗検索と作品検索と履歴閲覧、店員によるユーザ登録と貸出/返却、管理者による店員の登録とメディアの登録と店の登録/削除及び店員がどの店で働くか、メディアをどの店に置くかの管理がある。

## 2 実体関連図

実体関連図は図 1 のようになった。基本的には課題 1 のものと同じである。しかし、異なる点もいくつかある。

### パスワードの追加

ユーザと店員の属性に「パスワード」を追加した。これはインターフェー

スを分けるためにログインする必要があるためである。

#### 関連「借りている」の三項関連化

課題 1 のものと同じユーザが同じメディアを二回借りることができなかったのも、日付や期間に関する情報を属性ではなく実体として三項関連にした。

#### 属性「返した」と「利用可能」の追加

ひとつのメディアが同時に借りられることのないように追加した。

#### 属性「最大数」と「数」の削除

これは「メディア」ではなく「内容」に関する情報であるため、「置いてある」の属性として不適切であったため削除した。

#### 属性「通勤手段」と「良いところ」の追加

自明でない多値従属性を作るための属性である。

### 3 関係スキーマ

関係スキーマは以下ようになった。

- ユーザ (メールアドレス, ユーザ名, ユーザ住所, ユーザパスワード)
- メディア (mid, 種類, 利用可能)
- 内容 (題名, 発売年, 長さ, 出版社, ジャンル)
- 店舗 (店舗名, 店舗住所, 総所持数)
- 店員 (eid, 店員名, 店員パスワード)
- 付属情報 (通勤手段, 店の良い所)
- 借りている (メールアドレス, mid, 料金, 貸出日, 返却日, 返した)
- 期間 (貸出日, 貸出期間, 返却日)
- 置いてある (mid, 店舗名, 店舗住所)
- 働いている 1 (eid, 店舗名, 店舗住所, 通勤手段)
- 働いている 2 (eid, 店舗名, 店舗住所, 店の良い所)
- 保存されている (mid, 題名, 発売年)

課題 3 のものに第 2 章で述べた変更点を反映させただけである。全ての自明でない関数従属性の左辺は超キーであるから Boyce-Codd 正規形である。それぞれのデータ例を以下に示す。

ユーザ (メールアドレス, ユーザ名, ユーザ住所, ユーザパスワード)

mail	username	useraddress	userpw
yamada@example.jp	山田	A 市	yamada
takahashi@example.jp	高橋	B 市	takahashi
yoshida@example.jp	吉田	C 市	yoshida
baba@example.jp	馬場	D 市	baba
hotta@example.jp	堀田	E 市	hotta

メディア (mid, 種類, 利用可能)

mid	type	available
1	VHS	yes
2	DVD	yes
3	DVD	yes
4	Blu-ray	yes
5	Blu-ray	yes

内容 (題名, 発売年, 長さ, 出版社, ジャンル)

title	published_year	length	publisher	genre
ABC	2000	01 時間 30 分 00 秒	A	movie
ABD	2010	00 時間 50 分 30 秒	B	drama
ACD	2005	02 時間 00 分 00 秒	C	variety
BCD	2001	02 時間 00 分 00 秒	D	anime
BE	2008	02 時間 10 分 00 秒	E	sport

店舗 (店舗名, 店舗住所, 総所持数)

shopname	shopaddress	total_media
AAA	A 市	1
ABD	B 市	2
CAD	C 市	2
EDC	D 市	2
BCD	E 市	3

店員 (eid, 店員名, 店員パスワード)

eid	clerkname	clerkpw
1	山田	yamada
2	高橋	takahashi
3	吉田	yoshida
4	馬場	baba
5	堀田	hotta

付属情報 (通勤手段, 店の良い所)

commute_method	good_point_of_shop
walk	A
walk	B
bicycle	A
bicycle	B
bus	A

借りている (メールアドレス, mid, 料金, 貸出日, 返却日, 返した)

mail	mid	rental_fee	rental_date	return_date	finished
yamada@example.jp	1	300	2019/01/04	2019/01/05	yes
takahashi@example.jp	2	300	2019/01/05	2019/01/06	yes
yoshida@example.jp	3	400	2019/01/06	2019/01/08	yes
baba@example.jp	4	400	2019/01/07	2019/01/09	yes
hotta@example.jp	5	500	2019/01/08	2019/01/11	yes

期間 (貸出日, 貸出期間, 返却日)

rental_date	rental_duration	return_date
2019/01/04	1 日	2019/01/05
2019/01/05	1 日	2019/01/06
2019/01/06	2 日	2019/01/08
2019/01/07	2 日	2019/01/09
2019/01/08	3 日	2019/01/11

置いてある (mid, 店舗名, 店舗住所)

	mid	shopname	shopaddress
	---	-----	-----
1	AAA	A 市	
2	ABD	B 市	
3	CAD	C 市	
4	EDC	D 市	
5	BCD	E 市	

働いている 1(eid, 店舗名, 店舗住所, 通勤手段)

	eid	shopname	shopaddress	commute_method
	---	-----	-----	-----
1	AAA	A 市	walk	
2	ABD	B 市	walk	
3	CAD	C 市	bicycle	
4	EDC	D 市	bicycle	
5	BCD	E 市	bus	

働いている 2(eid, 店舗名, 店舗住所, 店の良い所)

	eid	shopname	shopaddress	good_point_of_shop
	---	-----	-----	-----
1	AAA	A 市	A	
2	ABD	B 市	B	
3	CAD	C 市	A	
4	EDC	D 市	B	
5	BCD	E 市	A	

保存されている (mid, 題名, 発売年)

	mid	title	published_year
	---	-----	-----
1	ABC	2000	
2	ABD	2010	
3	ACD	2005	
4	BCD	2001	
5	BE	2008	

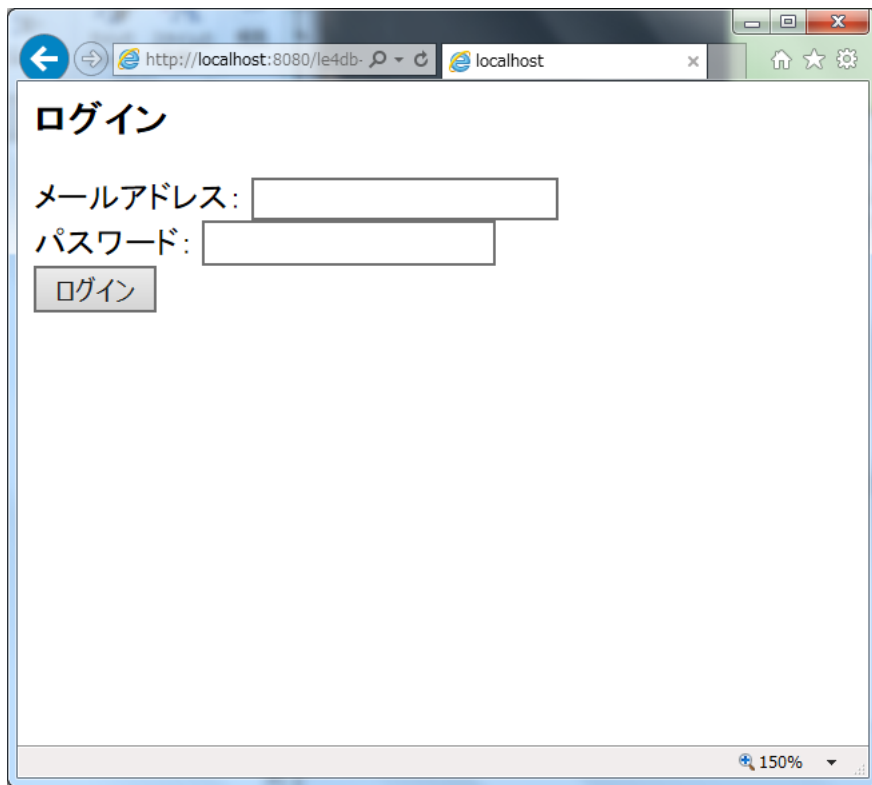


図 2: ログイン画面

## 4 機能・インタフェース

### 4.1 ログイン

画面は図 2 である。メールアドレスの欄に、ユーザならメールアドレス、店員なら eid, 管理者なら「supervisor」を入力してパスワードを入力するとログインできる。メールアドレスや eid を用いて user テーブルや clerk テーブルから対応するパスワードを得て、それが入力されたパスワードと一致するかどうかをチェックしている。

### 4.2 ユーザによる店舗検索

画面は図 3 と図 4 である。検索は店舗名と住所の AND 検索である。何も入力されていない場合は、その項目に関してはすべて表示される。入力された店舗名や住所を WHERE の後ろに並べて shop テーブルを検索している。また、検索結果のソートもできて、これは ORDER BY で行っている。検索で出

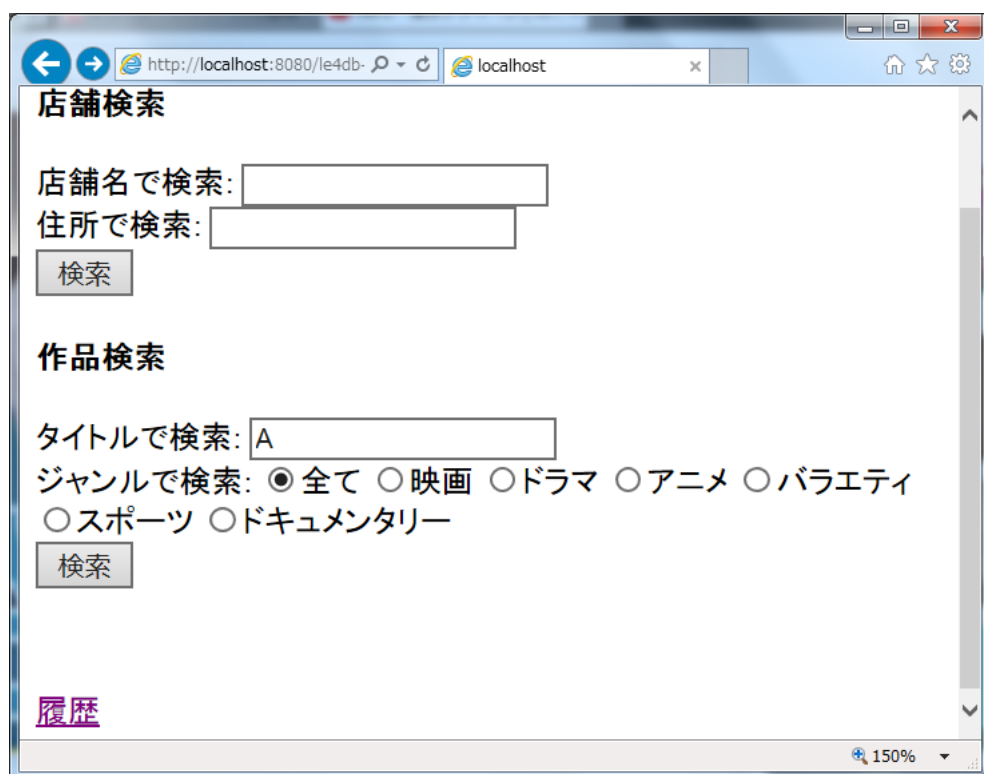


図 3: 検索画面



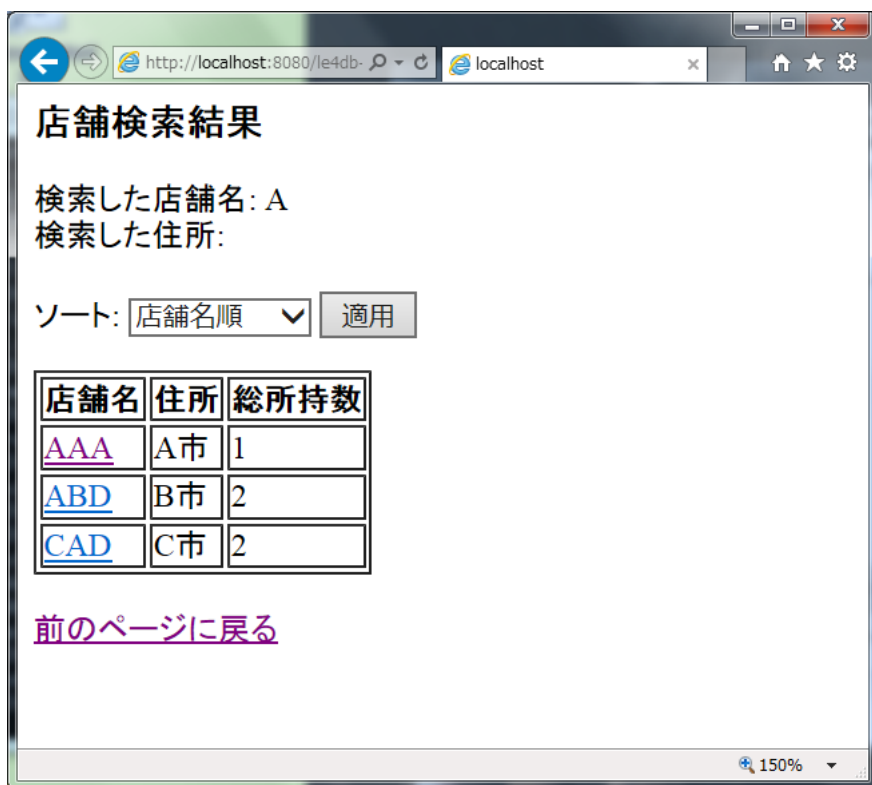


図 4: 検索結果 (店舗)

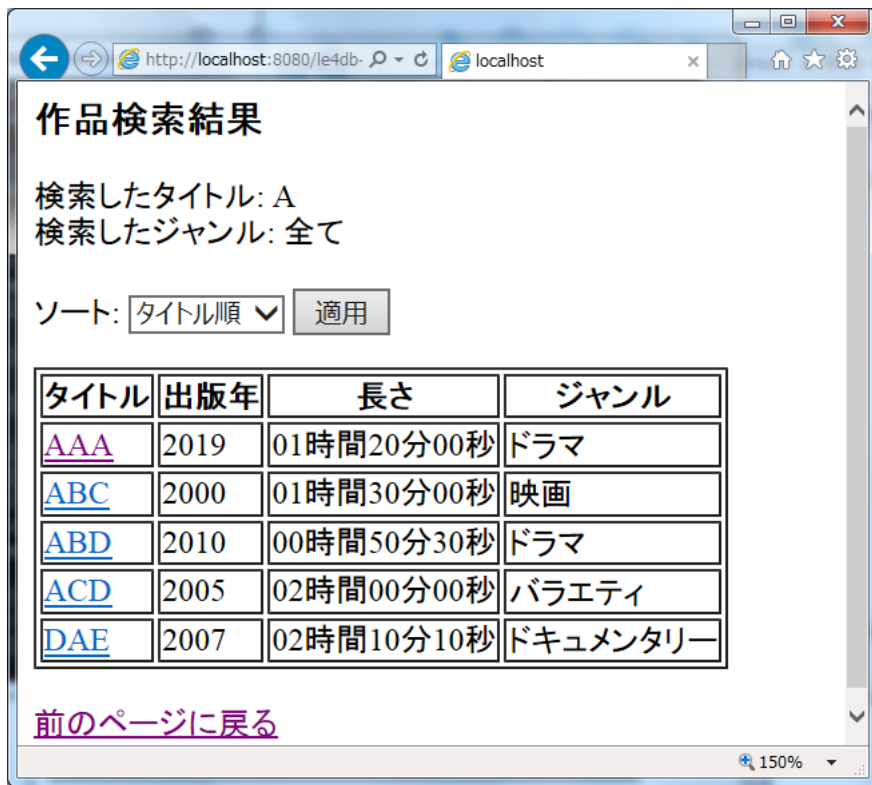


図 5: 検索結果 (作品)

てきた店舗をクリックすると、その店舗に置いてある作品のリストが表示されるようになっている。作品をクリックすると詳細情報が見られる。

#### 4.3 ユーザによる作品検索

画面は図 3 と図 5 である。基本的に店舗検索と同じである。media テーブルと store テーブルと content テーブルと put テーブルの四つのテーブルを自然結合したものを検索している。なぜ put テーブルを結合しているかというと、店舗検索からとべる店舗に置いてある作品のリストは、図 5 と同じページであるから、そちらから来た場合に必要になるからである。

#### 4.4 ユーザによる履歴閲覧

画面は図 6 である。これは rent テーブルと store テーブルを結合したものに対して、このユーザのメールアドレスを用いて検索している。

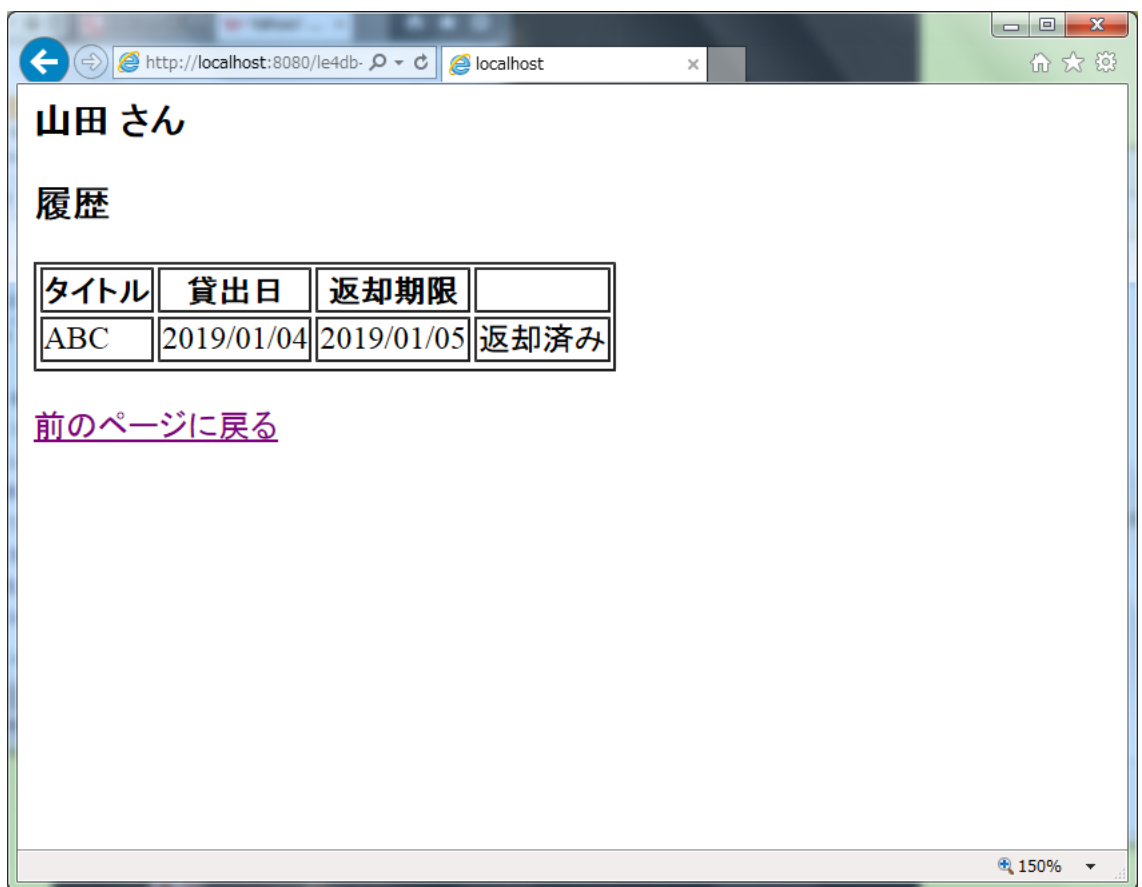


図 6: 履歴画面

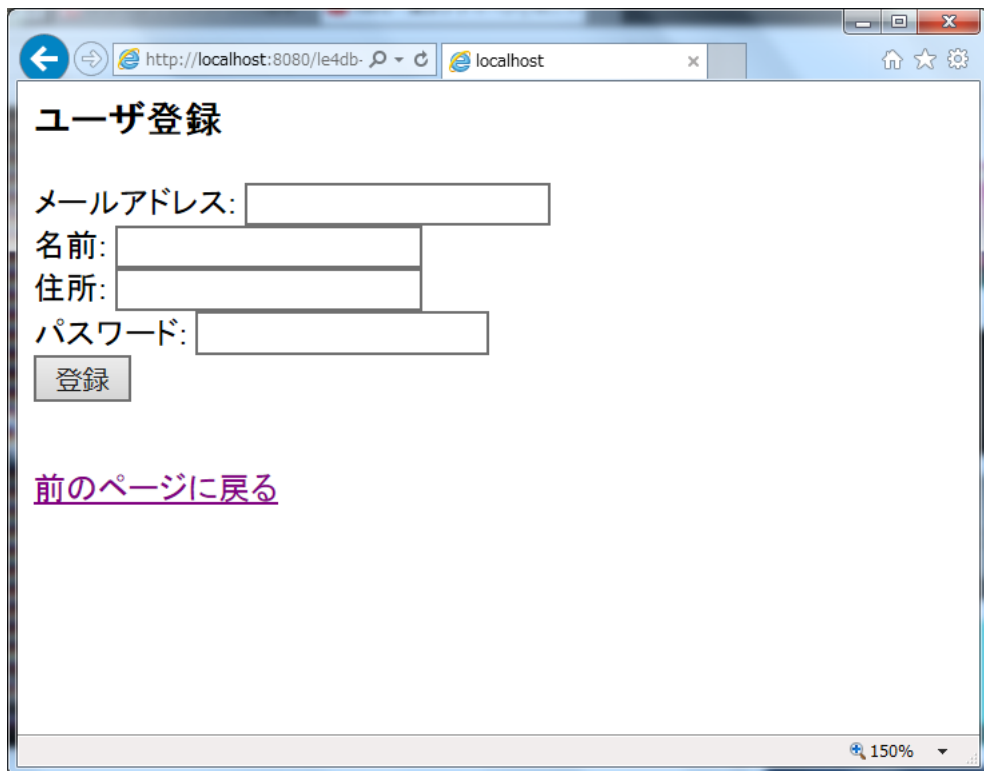


図 7: ユーザ登録画面

#### 4.5 店員によるユーザ登録

画面は図 7 である。すべての情報を入力しないとエラーになる。また、メールアドレスも@がなかったり、すでにいるユーザと同じであるとエラーになる。内部的には、user テーブルにそのメールアドレスの行がないことを確認してから挿入している。

#### 4.6 店員による貸出/返却

画面は図 8 と図 9 である。貸出については、空欄や不適切な値に対してエラーを出すようになっている。さらに、user テーブルや media テーブル、put テーブルを確認して、ユーザが存在するか、メディアがこの店に置いてあるか、メディアが利用可能かを調べてから、rent テーブルに挿入を行っている。また、このとき duration テーブルにも同時に挿入を行っている。ただし、挿入しようとしている行が存在しているかどうかを確認して、存在していない場合だけ挿入する SQL 文を実行している。さらに、media テーブルの available を no に更新している。削除については、図 9 にある「返却する」をクリックす

レンタル

メールアドレス:

mid:

料金:

貸出年月日: 2019年1月27日

貸出期間: 1日 ▼

[前のページに戻る](#)

図 8: 貸出画面

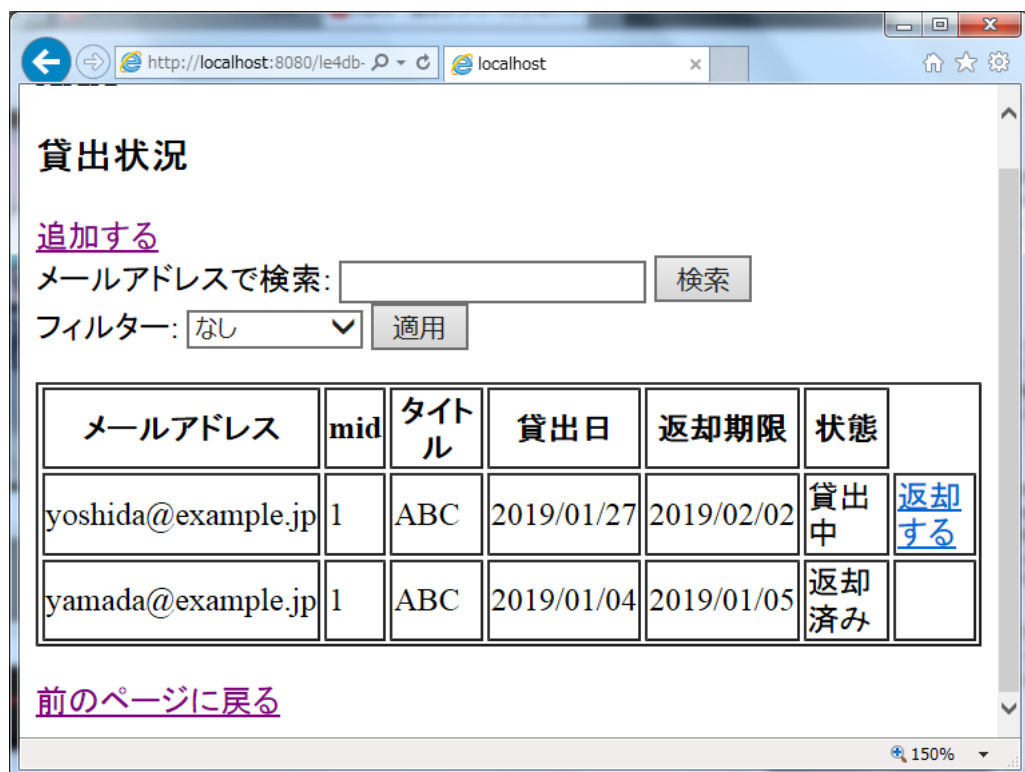


図 9: 貸出状況確認画面

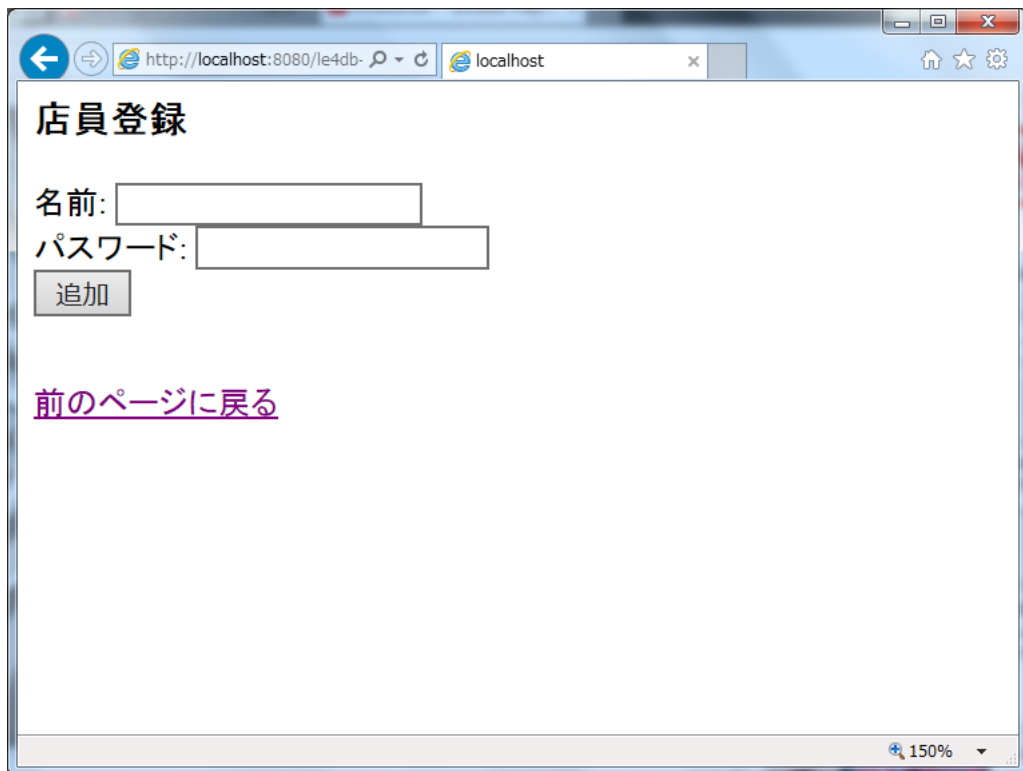


図 10: 店員登録画面

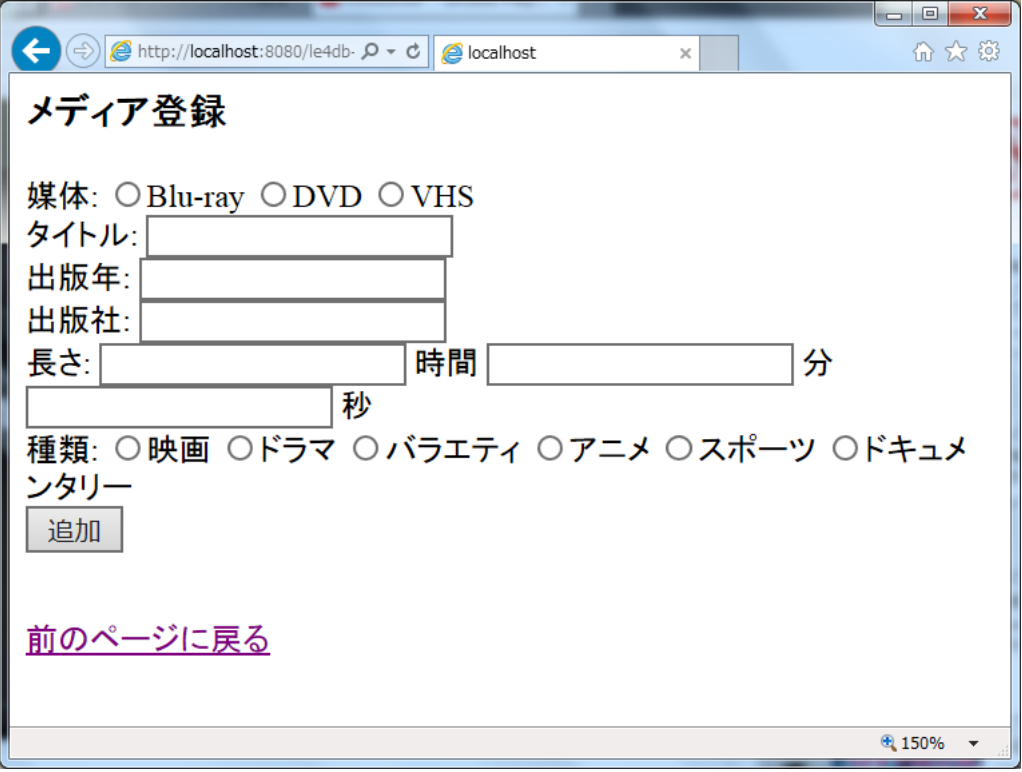
ることで行っている。内部的には, rent テーブルの finished を yes に更新している。このとき, 同時に media テーブルの available も yes に更新している。

#### 4.7 管理者による店員登録

画面は図 10 である。基本的にはユーザ登録と同じである。ただ, 店員は eid で管理しているので値の重複は起らない。一方, 新たな eid を得るために, 現在存在する eid の最大値を clerk テーブルより得て, それに 1 を加えたものを用いて挿入を行っている。

#### 4.8 管理者によるメディア登録

画面は図 11 である。基本的には店員登録と同じである。異なるのは, media テーブルと store テーブルと content テーブルの 3 つのテーブルに挿入を行う点である。



The screenshot shows a web browser window with the address bar displaying 'http://localhost:8080/le4db-'. The page title is 'メディア登録' (Media Registration). The form contains the following fields and options:

- 媒体: ☐ Blu-ray ☐ DVD ☐ VHS
- タイトル:
- 出版年:
- 出版社:
- 長さ:  時間  分  秒
- 種類: ☐ 映画 ☐ ドラマ ☐ バラエティ ☐ アニメ ☐ スポーツ ☐ ドキュメンタリー
- 
- [前のページに戻る](#)

The browser's status bar at the bottom right shows a magnification level of 150%.

図 11: メディア登録画面



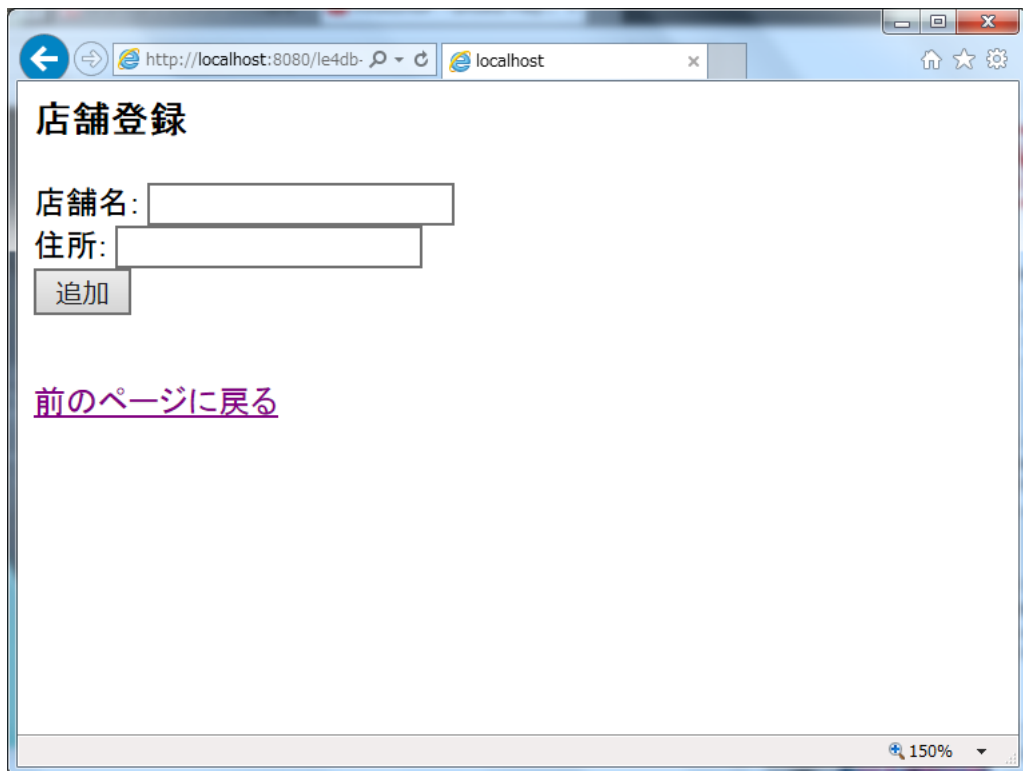


図 12: 店舗登録画面

#### 4.9 管理者による店舗登録/削除

画面は図 12 と図 13 である。登録についてはユーザ登録と同じである。削除については、返却と同様に「削除」をクリックすることで行っている。ただ、こちらは DELETE 文が走るようになっている。また、同時に実体「店舗」が参加していた関連に対応するテーブルにおいても DELETE 文が走るようになっている。この削除によってどこにも勤めていない店員が現れた場合、自動的にその店員は削除される。

#### 4.10 管理者による店員の管理

画面は図 14 と図 15 である。ある店舗における店員の管理が行える。追加については、基本的にはユーザ登録と同じである。ただ、存在しない eid を入力するとエラーを出す。削除については、店舗の削除と同様に「削除」をクリックすることで行える。内部的には、work1 テーブルと work2 テーブルにおいて対応する行を削除している。この削除によってどこにも勤めていない店員が現れた場合にもその店員は削除される。



図 13: 店舗確認画面

店員追加

eid:

通勤手段: ☐ 徒歩 ☐ 自転車 ☐ 車 ☐ バス ☐ 電車

店の良いところ:

[前のページに戻る](#)

図 14: 店員追加画面

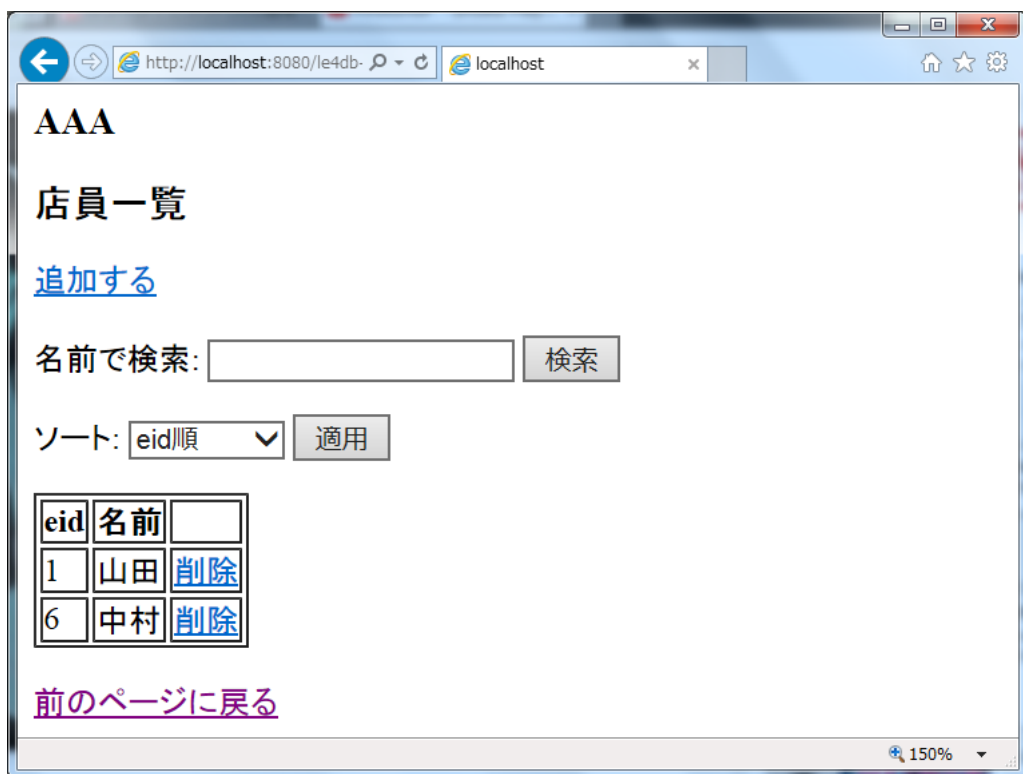


図 15: 店員確認画面

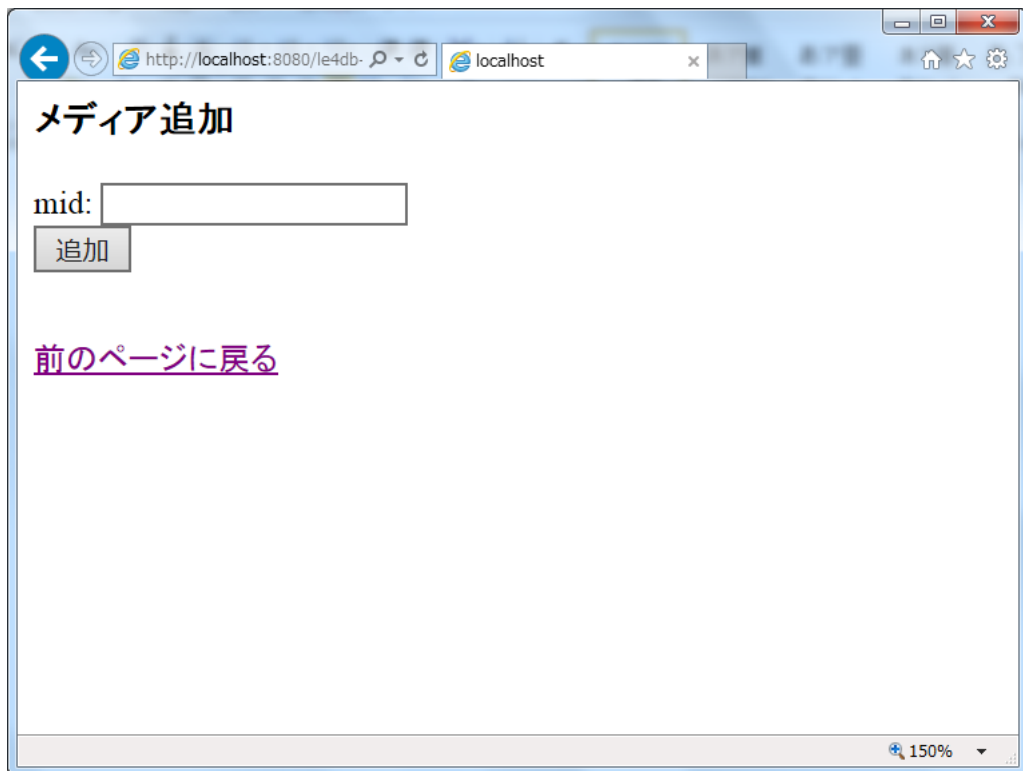


図 16: メディア追加画面

#### 4.11 管理者によるメディアの管理

画面は図 16 と図 17 である。基本的には店員の管理と同じである。異なる点は 2 つある。1 つはこれでどこにも置かれないメディアが現れても、そのメディアは削除されないという点である。もう 1 つは、メディアの追加/削除に応じて shop テーブルの total\_media の値を増減させているという点である。これは、単純に total\_media の値を得て、それに  $\pm 1$  をした値で更新している。

## 5 工夫点

SELECT 文以外の SQL 文には PreparedStatement を用いてインジェクション対策をしている。また、値チェックや同時更新などで一度のアクションで複数回テーブルを触る場合はトランザクションを用いている。ほかには、店舗削除に伴う削除や、店員を店舗から外した場合に起こる削除についてはトリガーを用いている。そのトリガーは以下の通りである。

```
CREATE TRIGGER deleteshop DELETE ON shop
```

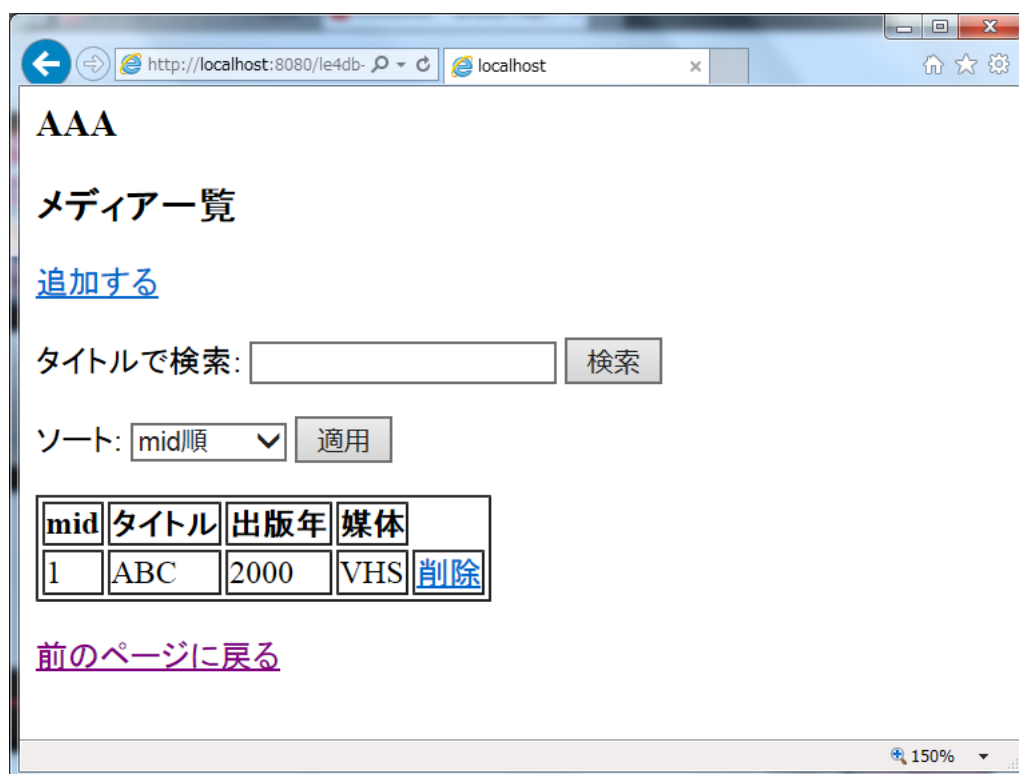


図 17: メディア確認画面

```

BEGIN
    DELETE FROM put WHERE shopname = old.shopname and shopaddress = old.shopaddress;
    DELETE FROM work1 WHERE shopname = old.shopname and shopaddress = old.shopaddress;
    DELETE FROM work2 WHERE shopname = old.shopname and shopaddress = old.shopaddress;
END;
CREATE TRIGGER deletework AFTER DELETE ON work1
BEGIN
    DELETE FROM clerk WHERE (SELECT count(*) FROM work1 WHERE clerk.eid = work1.eid) = 0
END;

```

データベース以外の点では、一部ではあるけれどもソートやフィルターをできるようにしたり、前のページに戻るをクリックしたときに、検索やソートの状態を残すようにしたりした。

## 6 感想

いろいろと難しかったと思った。データベースを設計するのも勿論初めてだったが、Web アプリケーションを作ったのも初めてで手間取った。最終的にできたのを見ると、同じようなページがいくつもできたので、そのあたりを上手くまとめて記述できればもっと楽だったのだろうと思った。ほかには、制約の付け方は難しかった。例えば、「店舗」の「働いている」への参加制約はシステムの方では実現できていない。これは、直感的には従業員のいない店舗は存在しないが、実際にシステムで扱う際には誰が働くかということを決めずに、まず店舗を作成したくなるからである。このあたりはどうすれば良いのかなと思った。授業全体で言うと、自明でない関数従属性や多値従属性を作りたいものに自然に落とし込むことは困難だと思った。僕のシステムでも、貸出日と貸出期間と返却日で関数従属性を作ったが、数学的に従属しているので、本当は貸出期間は必要ない。これを用いて正規形するという点でも、それをするために課題1の時点では「借りている」の属性としてこれらを使用しており、結果として、同じ人が同じメディアを複数回借りることができなくなっている。最後に、データベースは本当に様々なところに現れるので、データベースをプログラムから操作することを学べたことは今後生きてくると思うので、とても有意義な実験だったと思った。