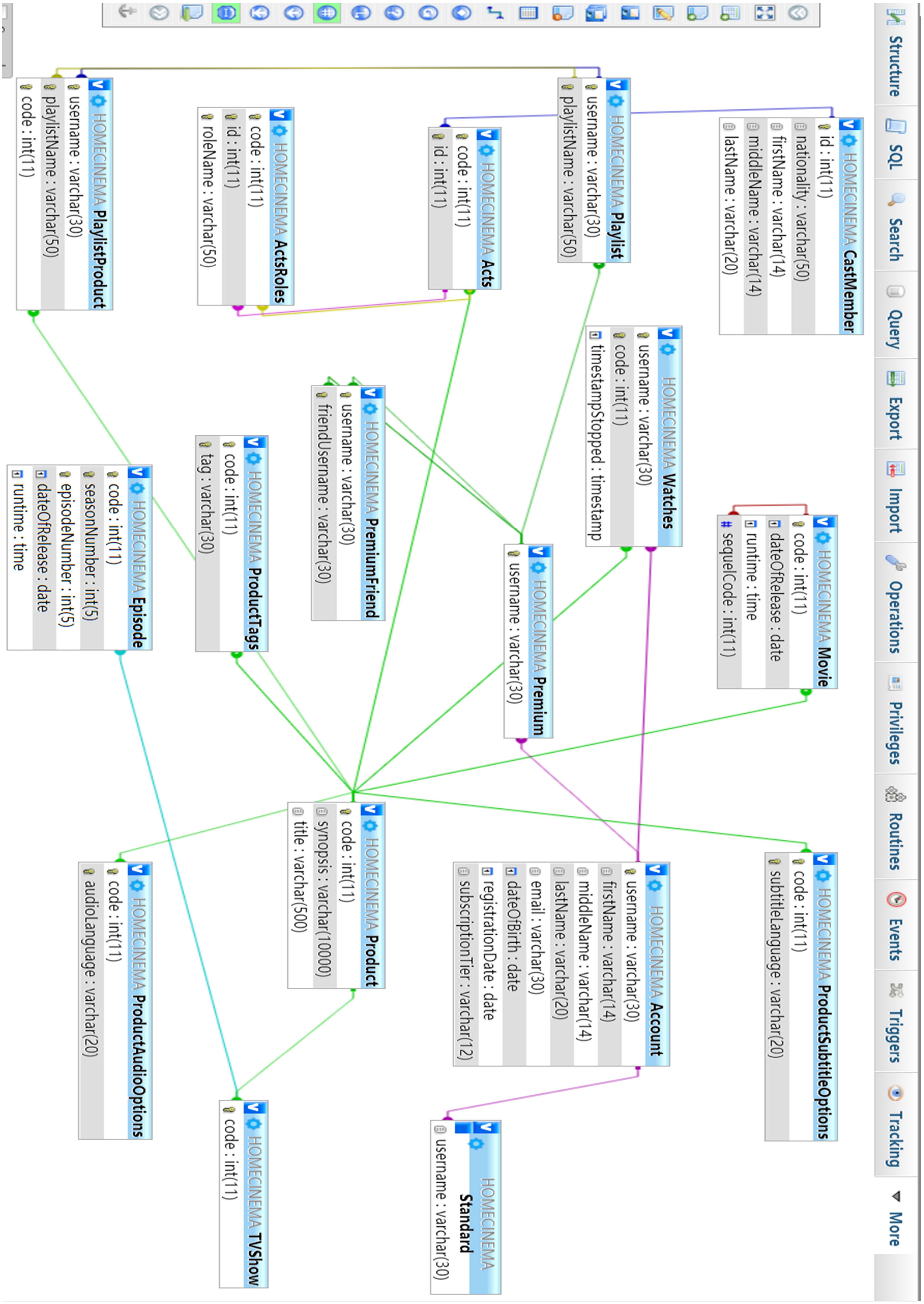
Assignment 2

Name: Xiaoyu Dong

Student Number: 45576999

SCREENSHOT OF DESIGNER VIEW FROM PHPMYADMIN (You may put this in sideways if required)



HOMECINEMA CastMember
- id : int(11)
- nationality : varchar(50)
- firstName : varchar(14)
- middleName : varchar(14)
- lastName : varchar(20)

HOMECINEMA Playlist
- username : varchar(30)
- playlistName : varchar(50)

HOMECINEMA PlaylistProduct
- username : varchar(30)
- playlistName : varchar(50)
- code : int(11)

HOMECINEMA Acts
- code : int(11)
- id : int(11)

HOMECINEMA ActsRoles
- code : int(11)
- id : int(11)
- roleName : varchar(50)

HOMECINEMA Watches
- username : varchar(30)
- code : int(11)
- timestampStopped : timestamp

HOMECINEMA Movie
- code : int(11)
- dateOfRelease : date
- runtime : time
- sequelCode : int(11)

HOMECINEMA PremiumFriend
- username : varchar(30)
- friendUsername : varchar(30)

HOMECINEMA Premium
- username : varchar(30)

HOMECINEMA ProductTags
- code : int(11)
- tag : varchar(30)

HOMECINEMA Episode
- code : int(11)
- seasonNumber : int(5)
- episodeNumber : int(5)
- dateOfRelease : date
- runtime : time

HOMECINEMA Product
- code : int(11)
- synopsis : varchar(10000)
- title : varchar(500)

HOMECINEMA ProductSubtitleOptions
- code : int(11)
- subtitleLanguage : varchar(20)

HOMECINEMA Account
- username : varchar(30)
- firstName : varchar(14)
- middleName : varchar(14)
- lastName : varchar(20)
- email : varchar(30)
- dateOfBirth : date
- registrationDate : date
- subscriptionTier : varchar(12)

HOMECINEMA ProductAudioOptions
- code : int(11)
- audioLanguage : varchar(20)

HOMECINEMA TVShow
- code : int(11)

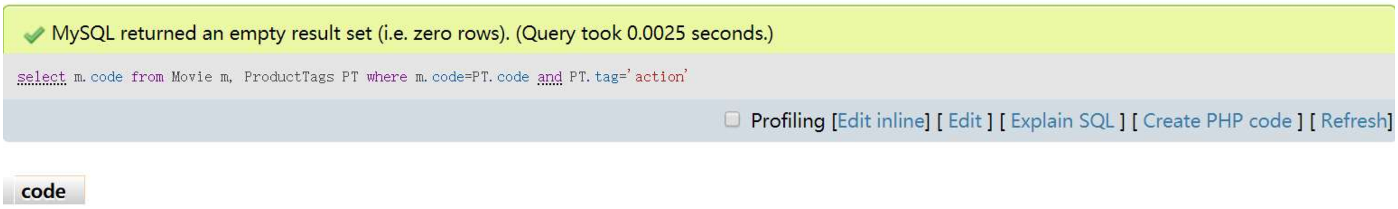
HOMECINEMA Standard
- username : varchar(30)

Query 1: Find all movies with the tag ‘Action’

QUERY SQL

```
select m.code

from Movie m, ProductTags PT

where m.code=PT.code and PT.tag='action'
```

QUERY OUTPUT (SCREENSHOT)



✓ MySQL returned an empty result set (i.e. zero rows). (Query took 0.0025 seconds.)

```
select m.code from Movie m, ProductTags PT where m.code=PT.code and PT.tag='action'
```

□ Profiling [Edit inline] [Edit] [Explain SQL] [Create PHP code] [Refresh]

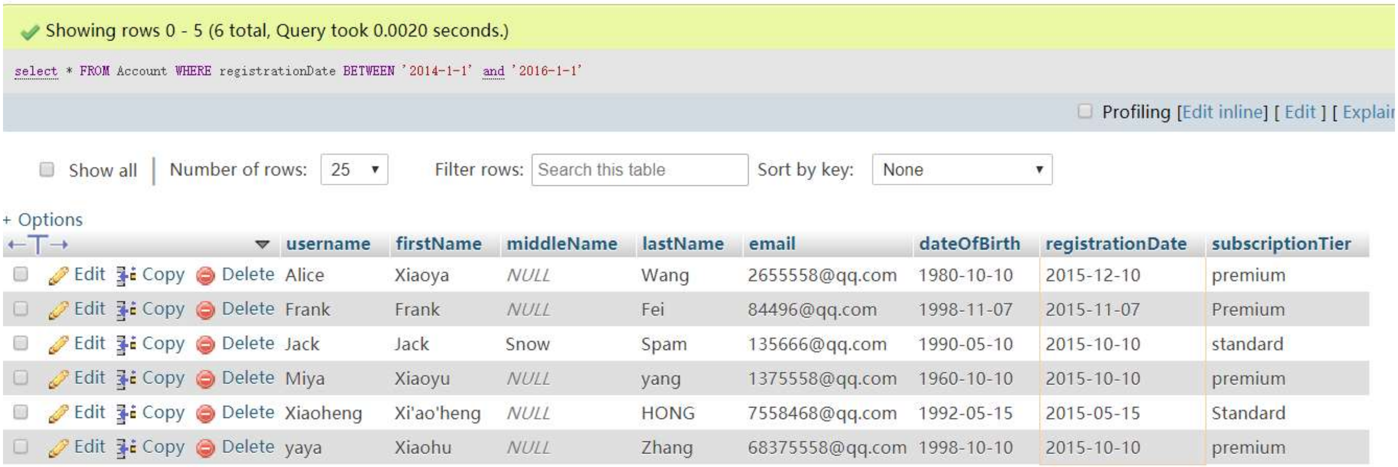
code

Query 2: Find all accounts registered between January 1st, 2014 and January 1st, 2016

QUERY SQL

```
select *

FROM Account

WHERE registrationDate BETWEEN '2014-1-1' and '2016-1-1'
```

QUERY OUTPUT (SCREENSHOT)



✓ Showing rows 0 - 5 (6 total, Query took 0.0020 seconds.)

```
select * FROM Account WHERE registrationDate BETWEEN '2014-1-1' and '2016-1-1'
```

□ Profiling [Edit inline] [Edit] [Explai

□ Show all | Number of rows: 25 ▼ Filter rows: Search this table Sort by key: None ▼

+ Options

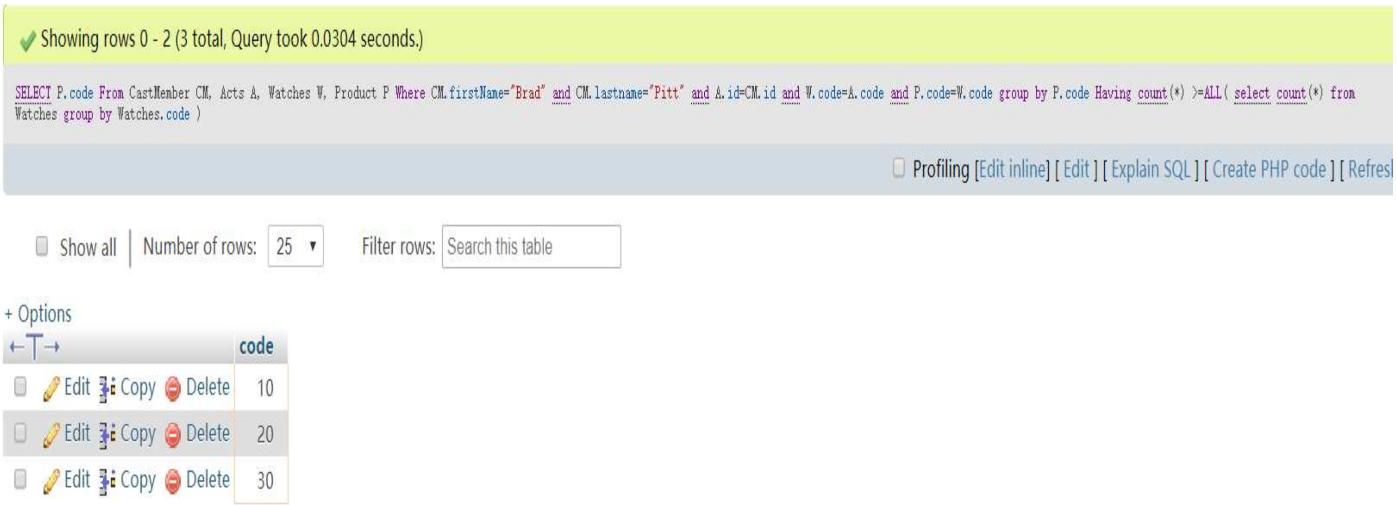
←T→ ▼	username	firstName	middleName	lastName	email	dateOfBirth	registrationDate	subscriptionTier
□ Edit Copy Delete	Alice	Xiaoya	*NULL*	Wang	2655558@qq.com	1980-10-10	2015-12-10	premium
□ Edit Copy Delete	Frank	Frank	*NULL*	Fei	84496@qq.com	1998-11-07	2015-11-07	Premium
□ Edit Copy Delete	Jack	Jack	Snow	Spam	135666@qq.com	1990-05-10	2015-10-10	standard
□ Edit Copy Delete	Miya	Xiaoyu	*NULL*	yang	1375558@qq.com	1960-10-10	2015-10-10	premium
□ Edit Copy Delete	Xiaoheng	Xi'ao'heng	*NULL*	HONG	7558468@qq.com	1992-05-15	2015-05-15	Standard
□ Edit Copy Delete	yaya	Xiaohu	*NULL*	Zhang	68375558@qq.com	1998-10-10	2015-10-10	premium

Query 3: Find the product, which has casted ‘Brad Pitt’, that has been watched the most times

QUERY SQL

```
SELECT  P.code

From CastMember CM, Acts A, Watches W, Product P

Where CM.firstName="Brad" and CM.lastname="Pitt" and A.id=CM.id and W.code=A.code and P.code=W.code

group by P.code

Having count(*) >=ALL(

                              select count(*)

                              from Watches

                              group by Watches.code

                                                            )
```

QUERY OUTPUT (SCREENSHOT)



✓ Showing rows 0 - 2 (3 total, Query took 0.0304 seconds.)

```
SELECT P.code From CastMember CM, Acts A, Watches W, Product P Where CM.firstName="Brad" and CM.lastname="Pitt" and A.id=CM.id and W.code=A.code and P.code=W.code group by P.code Having count(*) >=ALL( select count(*) from Watches group by Watches.code )
```

□ Profiling [Edit inline] [Edit] [Explain SQL] [Create PHP code] [Refresl

□ Show all | Number of rows: 25 ▾ Filter rows: Search this table

+ Options

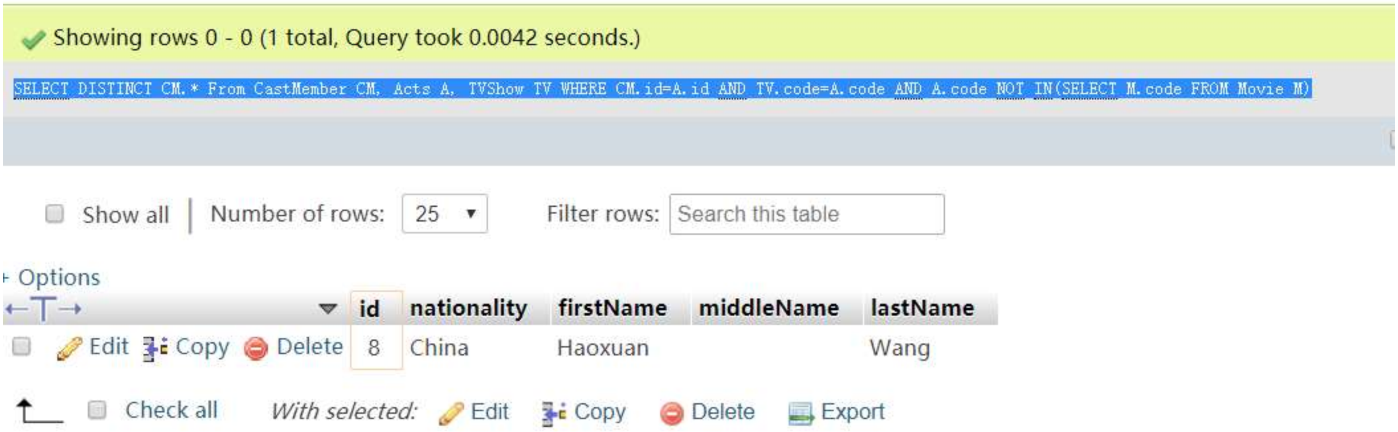
←T→	code
□ Edit Copy Delete	10
□ Edit Copy Delete	20
□ Edit Copy Delete	30

Query 4: Find all cast members who have acted in a TV show episode, but have not acted in a movie

QUERY SQL

```
SELECT DISTINCT CM.*
From CastMember CM, Acts A, TVShow TV
WHERE CM.id=A.id AND TV.code=A.code AND A.code NOT IN
                                        (SELECT M.code FROM Movie M)
```

QUERY OUTPUT (SCREENSHOT)



✓ Showing rows 0 - 0 (1 total, Query took 0.0042 seconds.)

```
SELECT DISTINCT CM.* From CastMember CM, Acts A, TVShow TV WHERE CM.id=A.id AND TV.code=A.code AND A.code NOT IN(SELECT M.code FROM Movie M)
```

□ Show all | Number of rows: 25 ▾ Filter rows: Search this table

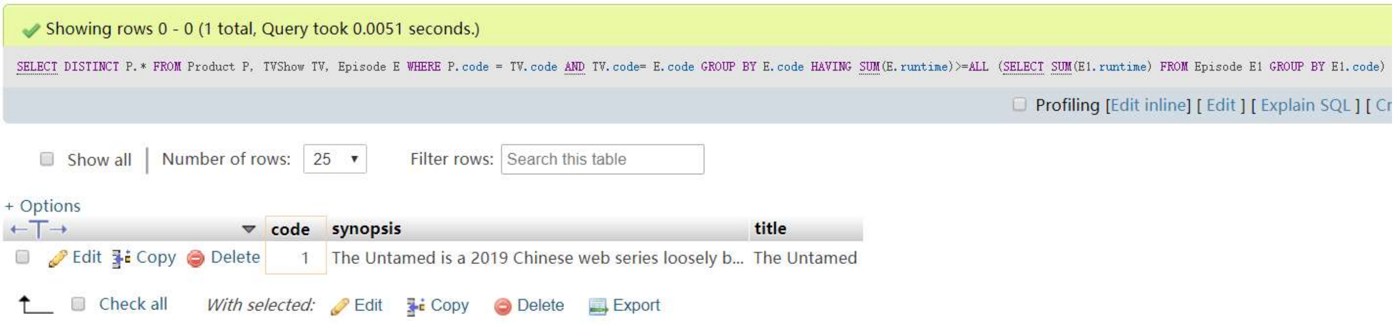
id	nationality	firstName	middleName	lastName
8	China	Haoxuan		Wang

Query 5: Find the TV show with the longest cumulative runtime

QUERY SQL

```
SELECT DISTINCT P.*
FROM Product P, TVShow TV, Episode E
WHERE P.code = TV.code AND TV.code= E.code
GROUP BY E.code
HAVING SUM(E.runtime)>=
        ALL (
           SELECT SUM(E1.runtime)
           FROM Episode E1
           GROUP BY E1.code)
```

QUERY OUTPUT (SCREENSHOT)



✓ Showing rows 0 - 0 (1 total, Query took 0.0051 seconds.)

```
SELECT DISTINCT P.* FROM Product P, TVShow TV, Episode E WHERE P.code = TV.code AND TV.code= E.code GROUP BY E.code HAVING SUM(E.runtime)>=ALL (SELECT SUM(E1.runtime) FROM Episode E1 GROUP BY E1.code)
```

□ Profiling [Edit inline] [Edit] [Explain SQL] [Cr

□ Show all | Number of rows: 25 ▾ Filter rows: Search this table

+ Options

	code	synopsis	title
□ Edit Copy Delete	1	The Untamed is a 2019 Chinese web series loosely b...	The Untamed

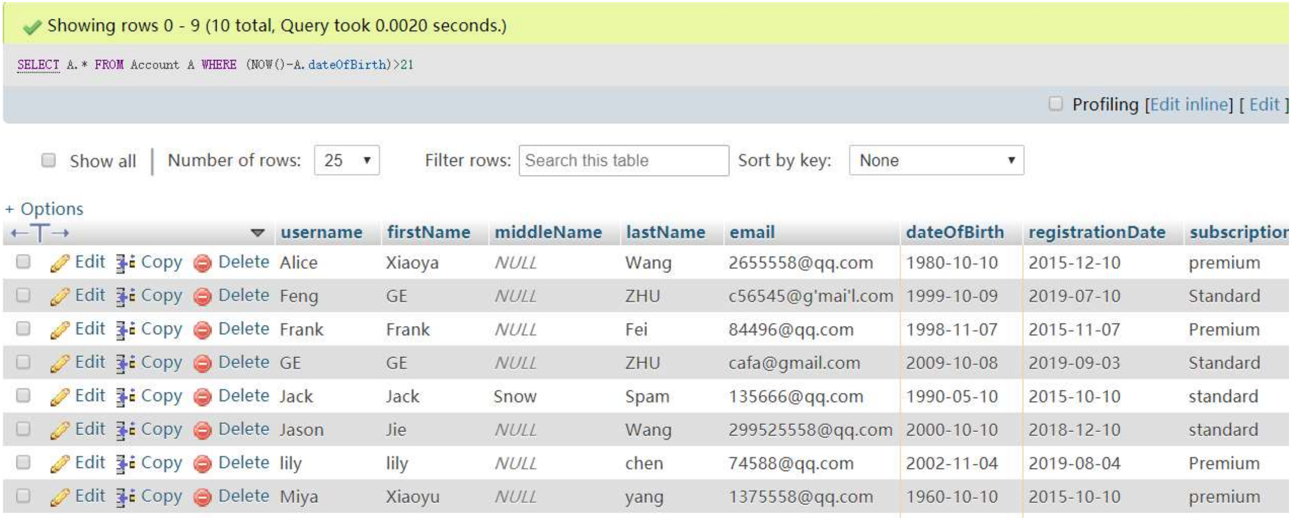
□ Check all *With selected:* Edit Copy Delete Export

Query 6: Find all accounts where the owner is older than 21 years old. (Note: This need to be correct at the time the query is run)

QUERY SQL

```
SELECT A.*
FROM Account A
WHERE (NOW()-A.dateOfBirth)>21
```

QUERY OUTPUT (SCREENSHOT)



✓ Showing rows 0 - 9 (10 total, Query took 0.0020 seconds.)

```
SELECT A.* FROM Account A WHERE (NOW()-A.dateOfBirth)>21
```

□ Profiling [Edit inline] [Edit]

□ Show all | Number of rows: 25 ▾ Filter rows: Search this table Sort by key: None ▾

+ Options

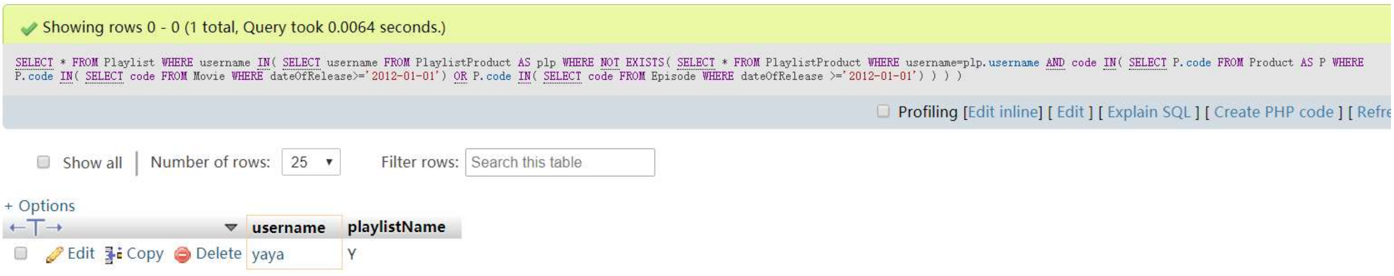
	username	firstName	middleName	lastName	email	dateOfBirth	registrationDate	subscriptior
□ Edit Copy Delete	Alice	Xiaoya	*NULL*	Wang	2655558@qq.com	1980-10-10	2015-12-10	premium
□ Edit Copy Delete	Feng	GE	*NULL*	ZHU	c56545@g'mai'l.com	1999-10-09	2019-07-10	Standard
□ Edit Copy Delete	Frank	Frank	*NULL*	Fei	84496@qq.com	1998-11-07	2015-11-07	Premium
□ Edit Copy Delete	GE	GE	*NULL*	ZHU	cafa@gmail.com	2009-10-08	2019-09-03	Standard
□ Edit Copy Delete	Jack	Jack	Snow	Spam	135666@qq.com	1990-05-10	2015-10-10	standard
□ Edit Copy Delete	Jason	Jie	*NULL*	Wang	299525558@qq.com	2000-10-10	2018-12-10	standard
□ Edit Copy Delete	lily	lily	*NULL*	chen	74588@qq.com	2002-11-04	2019-08-04	Premium
□ Edit Copy Delete	Miya	Xiaoyu	*NULL*	yang	1375558@qq.com	1960-10-10	2015-10-10	premium

Query 7: Find all playlists that do not contain a product that aired after January 1st, 2012

QUERY SQL

```
SELECT *
FROM Playlist
WHERE username IN(
                SELECT username
                FROM PlaylistProduct AS plp
                WHERE NOT EXISTS(
                                SELECT *
                                FROM PlaylistProduct
                                WHERE username=plp.username AND code IN(
                                SELECT P.code
                                FROM Product AS P
                                WHERE P.code IN(
                                                SELECT code
                                                FROM Movie
                                                WHERE dateOfRelease>='2012-01-01')
                                                OR P.code IN(
                                                SELECT code
                                                FROM Episode
                                                WHERE dateOfRelease >='2012-01-01')
                                                )
                                                )
                                                )
```

QUERY OUTPUT (SCREENSHOT)



✓ Showing rows 0 - 0 (1 total, Query took 0.0064 seconds.)

```
SELECT * FROM Playlist WHERE username IN( SELECT username FROM PlaylistProduct AS plp WHERE NOT EXISTS( SELECT * FROM PlaylistProduct WHERE username=plp.username AND code IN( SELECT P.code FROM Product AS P WHERE
P.code IN( SELECT code FROM Movie WHERE dateOfRelease>='2012-01-01') OR P.code IN( SELECT code FROM Episode WHERE dateOfRelease >='2012-01-01') ) ) )
```

□ Profiling [Edit inline] [Edit] [Explain SQL] [Create PHP code] [Refre
□ Show all | Number of rows: 25 ▾ Filter rows: Search this table
+ Options

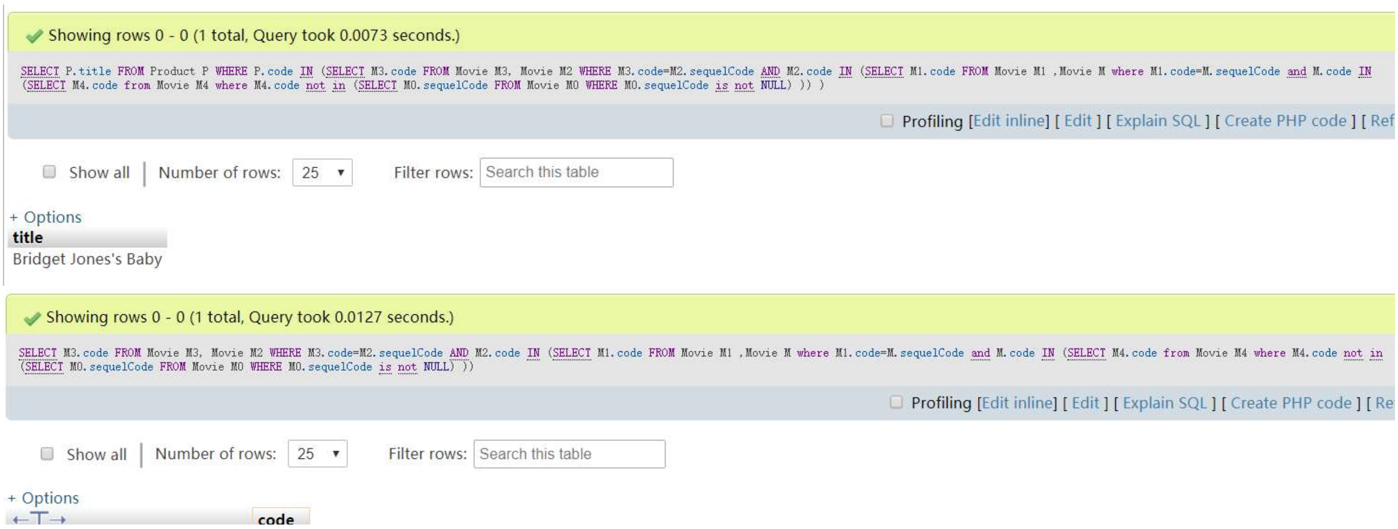
	username	playlistName
□ Edit Copy Delete	yaya	Y

Query 8: Find all movies that are the third move in their franchise (i.e. at least the sequel of a sequel)

QUERY SQL

```
SELECT P.title
FROM Product P
WHERE P.code IN
                    (SELECT M3.code
                    FROM Movie M3, Movie M2
                    WHERE M3.code=M2.sequelCode AND M2.code IN

                            (SELECT M1.code
                            FROM Movie M1 ,Movie M
                            where M1.code=M.sequelCode and M.code
                                       IN
                                    (SELECT M4.code from Movie M4
                                       where M4.code not in
                                       (SELECT M0.sequelCode
                                        FROM Movie M0
                                               WHERE M0.sequelCode is not NULL) ))
                                                                       )
```

QUERY OUTPUT (SCREENSHOT)



✓ Showing rows 0 - 0 (1 total, Query took 0.0073 seconds.)

```
SELECT P.title FROM Product P WHERE P.code IN (SELECT M3.code FROM Movie M3, Movie M2 WHERE M3.code=M2.sequelCode AND M2.code IN (SELECT M1.code FROM Movie M1 ,Movie M where M1.code=M.sequelCode and M.code IN
(SELECT M4.code from Movie M4 where M4.code not in (SELECT M0.sequelCode FROM Movie M0 WHERE M0.sequelCode is not NULL) )) )
```

□ Profiling [Edit inline] [Edit] [Explain SQL] [Create PHP code] [Ref
□ Show all | Number of rows: 25 ▾ Filter rows: Search this table
+ Options

title
Bridget Jones's Baby

✓ Showing rows 0 - 0 (1 total, Query took 0.0127 seconds.)

```
SELECT M3.code FROM Movie M3, Movie M2 WHERE M3.code=M2.sequelCode AND M2.code IN (SELECT M1.code FROM Movie M1 ,Movie M where M1.code=M.sequelCode and M.code IN (SELECT M4.code from Movie M4 where M4.code not in
(SELECT M0.sequelCode FROM Movie M0 WHERE M0.sequelCode is not NULL) ))
```

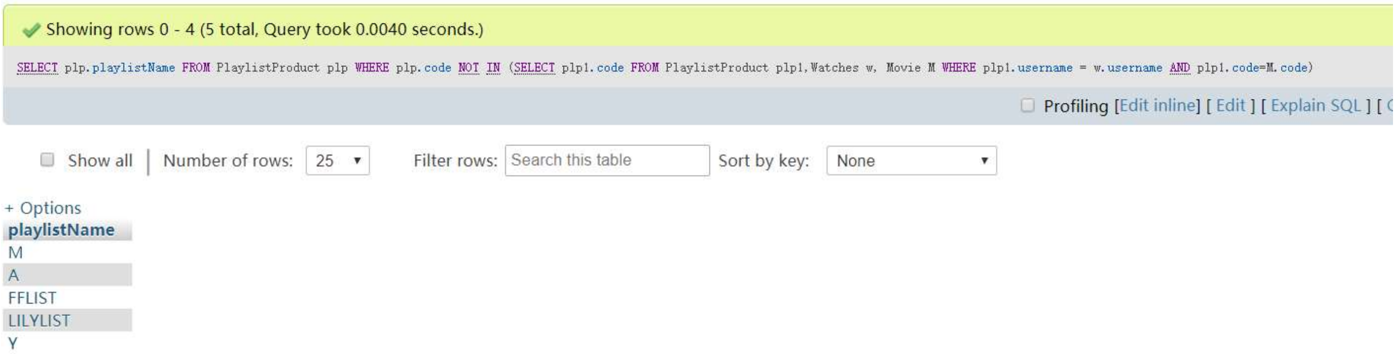
□ Profiling [Edit inline] [Edit] [Explain SQL] [Create PHP code] [Re
□ Show all | Number of rows: 25 ▾ Filter rows: Search this table
+ Options
←T→
code

Query 9: Find all playlists which contain a movie that the account owner has not watched

QUERY SQL

```
SELECT plp.playlistName

FROM PlaylistProduct plp

WHERE plp.code NOT IN

(SELECT plp1.code

 FROM PlaylistProduct plp1,Watches w, Movie M

 WHERE plp1.username = w.username AND plp1.code=M.code)
```

QUERY OUTPUT (SCREENSHOT)



✓ Showing rows 0 - 4 (5 total, Query took 0.0040 seconds.)

```
SELECT plp.playlistName FROM PlaylistProduct plp WHERE plp.code NOT IN (SELECT plp1.code FROM PlaylistProduct plp1,Watches w, Movie M WHERE plp1.username = w.username AND plp1.code=M.code)
```

□ Profiling [Edit inline] [Edit] [Explain SQL] [
□ Show all | Number of rows: 25 ▾ Filter rows: Search this table Sort by key: None ▾
+ Options

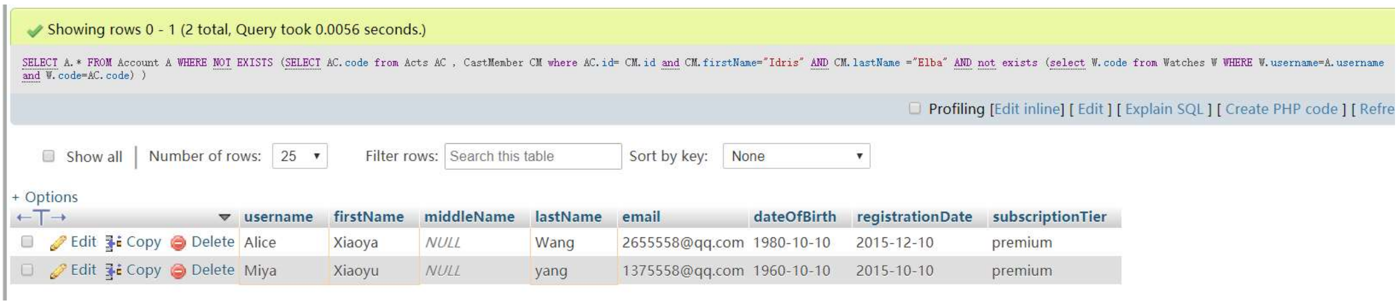
playlistName
M
A
FFLIST
LILYLIST
Y

Query 10: Find which account(s) have watched at least all the products that ‘Idris Elba’ has been cast in

QUERY SQL

```
SELECT A.*

FROM Account A

WHERE

NOT EXISTS

        (SELECT AC.code

         from Acts AC , CastMember CM

        where AC.id= CM.id and CM.firstName="Idris" AND CM.lastName ="Elba"

        AND not exists

                 (select W.code

                  from Watches W

                  WHERE W.username=A.username and W.code=AC.code)

                          )
```

QUERY OUTPUT (SCREENSHOT)



✓ Showing rows 0 - 1 (2 total, Query took 0.0056 seconds.)

```
SELECT A.* FROM Account A WHERE NOT EXISTS (SELECT AC.code from Acts AC , CastMember CM where AC.id= CM.id and CM.firstName="Idris" AND CM.lastName ="Elba" AND not exists (select W.code from Watches W WHERE W.username=A.username
and W.code=AC.code) )
```

□ Profiling [Edit inline] [Edit] [Explain SQL] [Create PHP code] [Refre
□ Show all | Number of rows: 25 ▾ Filter rows: Search this table Sort by key: None ▾
+ Options

	username	firstName	middleName	lastName	email	dateOfBirth	registrationDate	subscriptionTier
□ Edit Copy Delete	Alice	Xiaoya	*NULL*	Wang	2655558@qq.com	1980-10-10	2015-12-10	premium
□ Edit Copy Delete	Miya	Xiaoyu	*NULL*	yang	1375558@qq.com	1960-10-10	2015-10-10	premium