

## Taylor Herb Assignment 2

**Task 1 (15 points):** In database [your Pitt username], create the following entity tables:

### 1. Movies

```
CREATE TABLE `tah77`.`Movies` ( `movie_id` INT NOT NULL AUTO_INCREMENT , `title` VARCHAR(200) NOT NULL , `release_date` DATETIME NOT NULL , `plot_description` VARCHAR(4000) NOT NULL , PRIMARY KEY (`movie_id`)) ENGINE =InnoDB;
```

The screenshot shows the phpMyAdmin interface for the 'tah77' database. The 'Table: Movies' structure is displayed in 'Table structure' view. The table has four columns: 'movie\_id' (INT(11), PRIMARY KEY, AUTO\_INCREMENT), 'title' (VARCHAR(200)), 'release\_date' (DATETIME), and 'plot\_description' (VARCHAR(4000)). The 'Information' tab is selected, showing space usage and row statistics.

#	Name	Type	Collation	Attributes	Null	Default	Extra	Action
1	movie_id	int(11)			No	None	AUTO_INCREMENT	Change Drop Primary Unique Index Spatial More
2	title	varchar(200)	latin1_swedish_ci		No	None		Change Drop Primary Unique Index Spatial More
3	release_date	datetime			No	None		Change Drop Primary Unique Index Spatial More
4	plot_description	varchar(4000)	latin1_swedish_ci		No	None		Change Drop Primary Unique Index Spatial More

Space usage		Row statistics	
Data	16 KiB	Format	Compact
Index	0 B	Collation	latin1_swedish_ci
Total	16 KiB	Next autoindex	1
		Creation	Feb 19, 2019 at 01:51 PM

### 2. Actors

```
CREATE TABLE `tah77`.`actors` ( `actor_id` INT NOT NULL AUTO_INCREMENT , `first_name` VARCHAR(100) NOT NULL , `last_name` VARCHAR(100) NOT NULL , `birth_date` DATETIME NOT NULL , `biography` VARCHAR(1000) NOT NULL , PRIMARYKEY (`actor_id`)) ENGINE = InnoDB;
```

localhost/phpmyadmin/#PMAURL=8:tbl\_structure.php?db=tah77&table=actors&server=1&target=&token=8cb0f931c4dd606a25ded416428b2af7

Server: localhost Database: tah77 Table: actors

Table structure Relation view

#	Name	Type	Collation	Attributes	Null	Default	Extra	Action
1	actor_id	int(11)			No	None	AUTO_INCREMENT	Change Drop Primary Unique Index Spatial More
2	first_name	varchar(100)	latin1_swedish_ci		No	None		Change Drop Primary Unique Index Spatial More
3	last_name	varchar(100)	latin1_swedish_ci		No	None		Change Drop Primary Unique Index Spatial More
4	birth_date	datetime			No	None		Change Drop Primary Unique Index Spatial More
5	biography	varchar(1000)	latin1_swedish_ci		No	None		Change Drop Primary Unique Index Spatial More

Check All With selected: Browse Change Drop Primary Unique Index

Print view Propose table structure Move columns Improve table structure

Add 1 column(s) after biography Go

+ Indexes

Information

Space usage		Row statistics	
Data	16 KiB	Format	Compact
Index	0 B	Collation	latin1_swedish_ci
Total	16 KiB	Next autoindex	1
		Creation	Feb 19, 2019 at 01:59 PM

Console

### 3. Locations

```
CREATE TABLE `tah77`.`locations` ( `location_id` INT NOT NULL AUTO INCREMENT , `location_name` VARCHAR(100) NOT NULL , `street_address` VARCHAR(150) NOT NULL, `city` VARCHAR(100) NOT NULL , `state` CHAR(2) NOT NULL , `zip` VARCHAR(5) NOT NULL , PRIMARY KEY (`location_id`)) ENGINE = InnoDB;
```

The screenshot shows the phpMyAdmin interface for a database named 'tah77'. The 'locations' table is selected, and its structure is displayed. The table has the following columns:

#	Name	Type	Collation	Attributes	Null	Default	Extra	Action
1	location_id	int(11)			No	None	AUTO_INCREMENT	Change Drop Primary Unique Index Spatial More
2	location_name	varchar(100)	latin1_swedish_ci		No	None		Change Drop Primary Unique Index Spatial More
3	street_address	varchar(150)	latin1_swedish_ci		No	None		Change Drop Primary Unique Index Spatial More
4	city	varchar(100)	latin1_swedish_ci		No	None		Change Drop Primary Unique Index Spatial More
5	state	char(2)	latin1_swedish_ci		No	None		Change Drop Primary Unique Index Spatial More
6	zip	varchar(5)	latin1_swedish_ci		No	None		Change Drop Primary Unique Index Spatial More

Below the table structure, there are options to 'Add' columns or indexes. The 'Information' tab is active, showing 'Space usage' and 'Row statistics'.

Space usage		Row statistics	
Data	16 KiB	Format	Compact
Index	0 B	Collation	latin1_swedish_ci
Total	16 KiB	Next autoindex	1
		Creation	Feb 19, 2019 at 02:03 PM

Each table's logical structure should correspond to the descriptions provided in this assignment. Use CREATE TABLE statement.

**Task 2 (10 points):** In your database, create the following junction tables:

1. movies\_actors

Create table:

```
CREATE TABLE `tah77`.`movies_actors` ( `actor_id` INT NOT NULL , `movie_id` INT NOT NULL )
ENGINE = InnoDB;
```

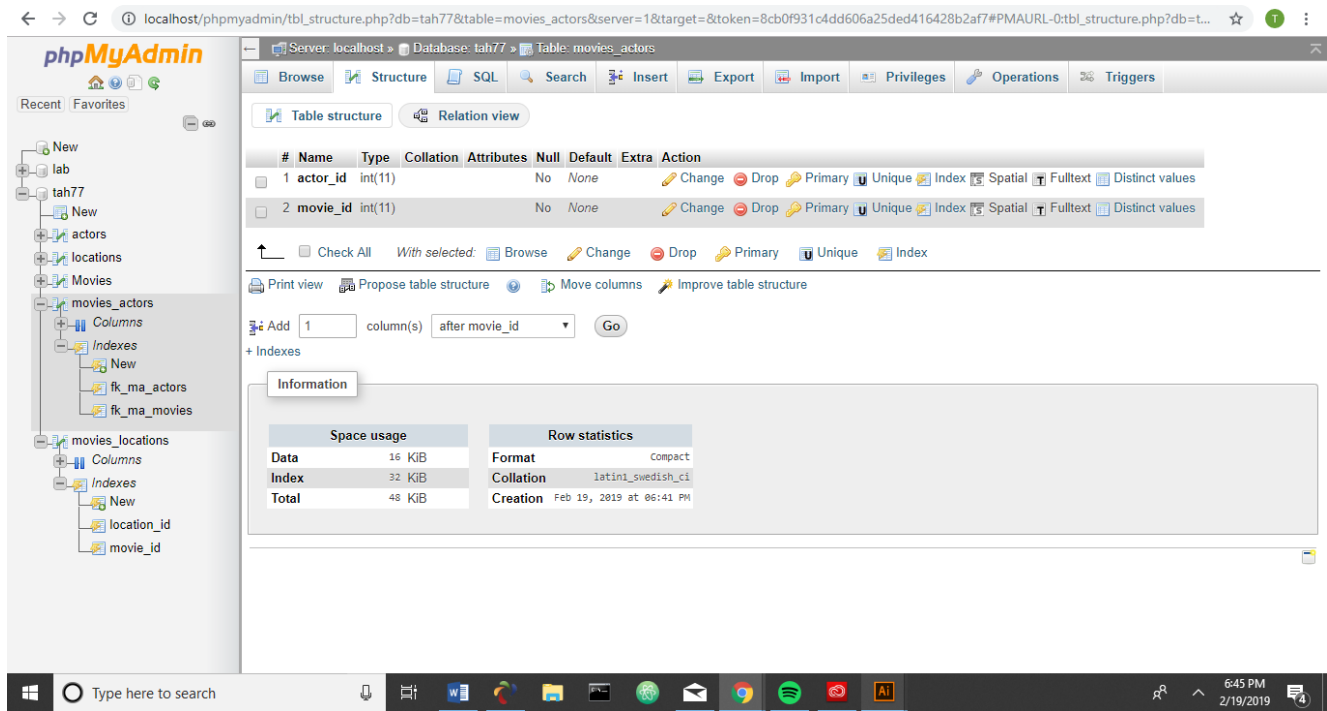
Add movie\_id foreign key:

```
ALTER TABLE movies_actors
ADD CONSTRAINT `fk_ma_movies`
FOREIGN KEY (`movie_id`) REFERENCES `Movies` (`movie_id`);
```

Add actor\_id foreign key:

```
ALTER TABLE movies_actors
ADD CONSTRAINT `fk_ma_actors`
FOREIGN KEY (`movie_actors`) REFERENCES `actors` (`actor_id`);
```

\*deleted & edited after original creation to add constraint 'fk\_' names



2.

## movies\_locations

Create table:

```
CREATE TABLE `tah77`.`movies_locations` ( `movie_id` INT NOT NULL , `location_id` INT NOT NULL ) ENGINE = InnoDB;
```

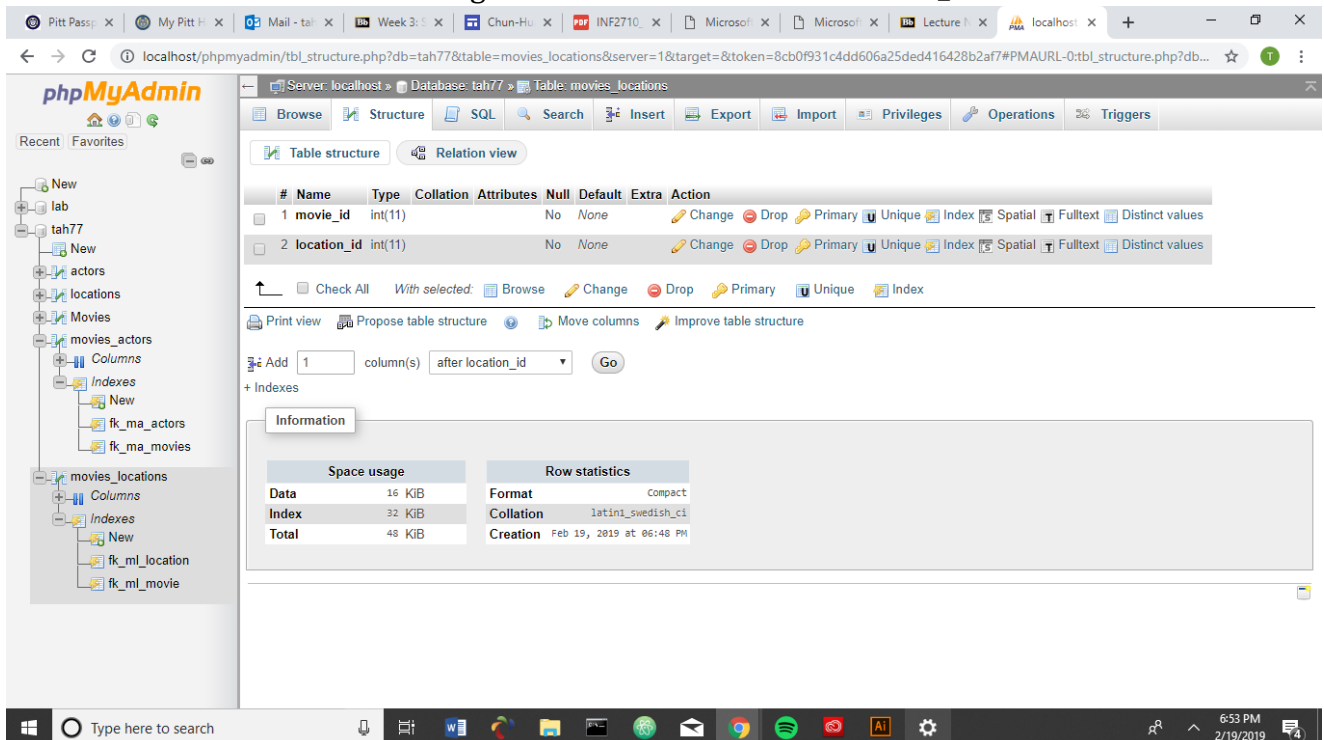
Add movie\_id foreign key:

```
ALTER TABLE movies_locations
ADD CONSTRAINT `fk_ml_movies`
FOREIGN KEY (`movie_id`) REFERENCES `Movies` (`movie_id`);
```

Add location\_id foreign key:

```
ALTER TABLE movies_locations
ADD CONSTRAINT `fk_ml_location`
FOREIGN KEY (`location_id`) REFERENCES `locations` (`location_id`);
```

\*deleted & edited after original creation to add constraint 'fk\_' names



Use CREATE TABLE statement to create junction tables. Make sure to create appropriate foreign keys – each table will have two foreign keys. Use ALTER TABLE statement to create foreign keys.

**Task 3 (15 points):** For each entity table, insert at least 5 rows using INSERT statement:

At least 5 movies in the **movies** table

insert into Movies values ('1', 'movietitle', '2018-02-10 10:34:34', 'description');

insert into Movies values ('2', 'movietwo', '2004-12-03 10:24:04', 'a movie about things');

insert into Movies values ('3', 'another\_movie', '1976-03-13 02:24:03', 'insert plot');

insert into Movies values ('4', 'movie\_title', '2013-09-12 01:12:07', 'the plot of the movie');

insert into Movies values ('5', 'the\_fifth\_movie', '2009-05-17 10:02:07', 'description of the plot');

The screenshot shows the phpMyAdmin interface for a database named 'tah77'. The 'Movies' table is selected, and its structure is visible. The table has four columns: 'movie\_id', 'title', 'release\_date', and 'plot\_description'. Five rows of data are displayed, corresponding to the INSERT statements provided in the task. The interface includes a sidebar with a tree view of the database structure, a top navigation bar with tabs for Browse, Structure, SQL, Search, Insert, Export, Import, Privileges, Operations, and Triggers, and a bottom status bar showing the number of rows and query execution time.

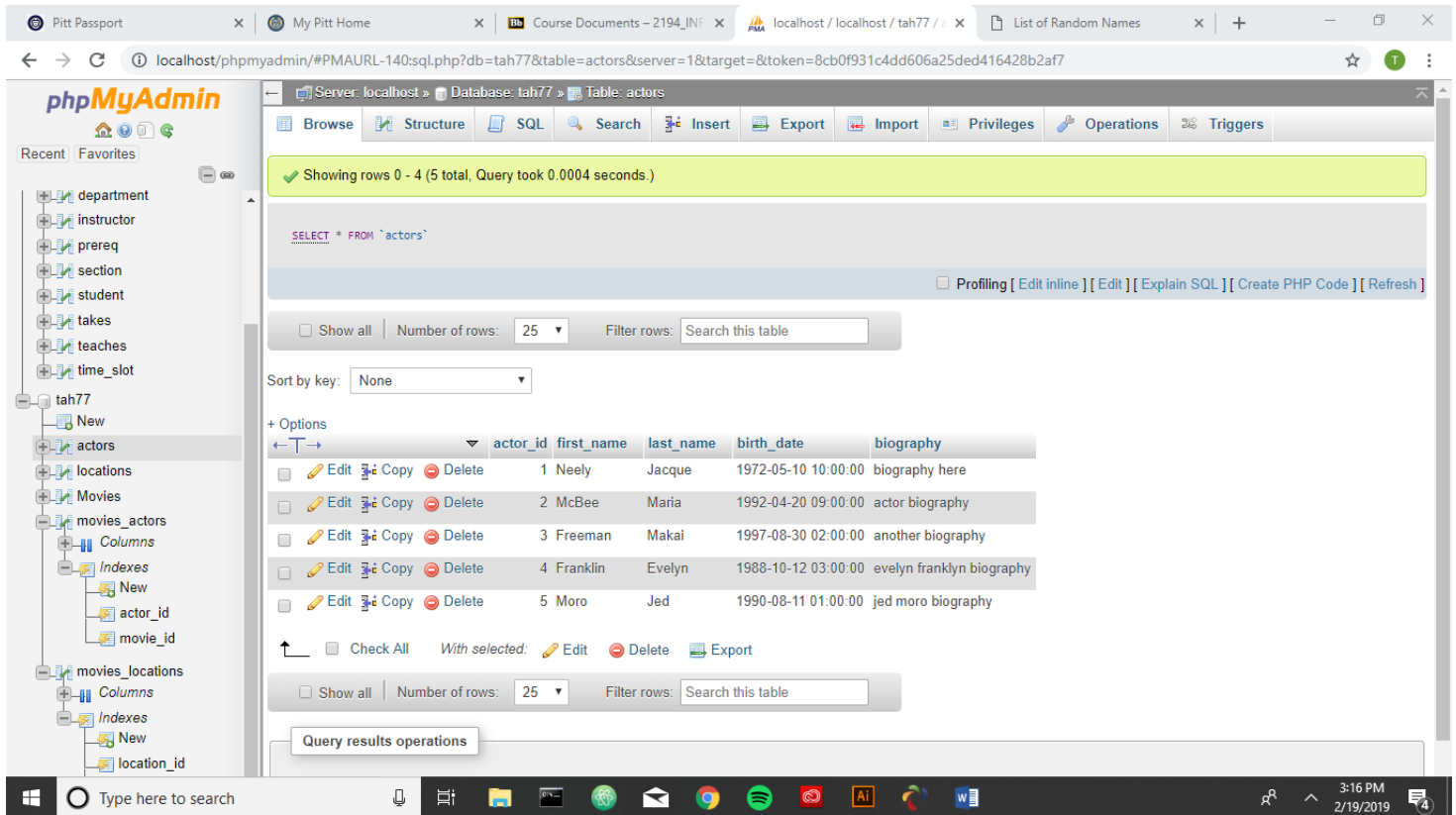
movie_id	title	release_date	plot_description
1	movietitle	2018-02-10 10:34:34	description
2	movietwo	2004-12-03 10:24:04	a movie about things
3	another_movie	1976-03-13 02:24:03	insert plot
4	movie_title	2013-09-12 01:12:07	the plot of the movie
5	the_fifth_movie	2009-05-17 10:02:07	description of the plot

1. At least 5 actors in the **actors** table

insert into actors values ('1', 'Neely', 'Jacque', '1972-05-10 10:00:00', 'biography here');

insert into actors values ('2', 'McBee', 'Maria', '1992-04-20 09:00:00', 'actor biography');

insert into actors values ('3', 'Freeman', 'Makai', '1997-08-30 02:00:00', 'another biography');  
insert into actors values ('4', 'Franklin', 'Evelyn', '1988-10-12 03:00:00', 'evelyn franklyn biography');  
insert into actors values ('5', 'Moro', 'Jed', '1990-08-11 01:00:00', 'jed moro biography');



2. At least 5 locations in the **locations** table  
insert into locations values ('1', 'movie location', '123 street ave', 'new york', 'NY', '88888');  
insert into locations values ('2', 'another place', '44 second ave', 'pittsburgh', 'PA', '55555');  
insert into locations values ('3', 'a big building', '77 place rd', 'houston', 'TX', '33333');  
insert into locations values ('4', 'the park', '543 brick st', 'san diego', 'CA', '24313');

insert into locations values ('5', 'niagara falls', '22 falls dr', 'niagara', 'NY', '99999');

The screenshot shows the phpMyAdmin interface for a database named 'tah77'. The 'locations' table is selected, and its structure is displayed. The table has 6 columns: location\_id, location\_name, street\_address, city, state, and zip. There are 5 rows of data in the table.

location_id	location_name	street_address	city	state	zip
1	movie location	123 street ave	new york	NY	88888
2	another place	44 second ave	pittsburgh	PA	55555
3	a big building	77 place rd	houston	TX	33333
4	the park	543 brick st	san diego	CA	24313
5	niagara falls	22 falls dr	niagara	NY	99999

**Task 4 (10 points): For each junction table, create at least five relationships (insert at least five rows of proper IDs).**

### **movies\_actors**

insert into movies\_actors (movie\_id, actor\_id) values (1, 2)

insert into movies\_actors (movie\_id, actor\_id) values (1, 3)

insert into movies\_actors (movie\_id, actor\_id) values (2, 1)

insert into movies\_actors (movie\_id, actor\_id) values (2, 5)

insert into movies\_actors (movie\_id, actor\_id) values (3, 4)

Current selection does not contain a unique column. Grid edit, checkbox, Edit, Copy and Delete features are not available.

Showing rows 0 - 4 (5 total. Query took 0.0004 seconds.)

```
SELECT * FROM `movies_actors`
```

Profiling [ Edit inline ] [ Edit ] [ Explain SQL ] [ Create PHP Code ] [ Refresh ]

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

Sort by key: None

+ Options

movie_id	actor_id
1	2
1	3
2	1
2	5
3	4

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

Query results operations

Print view | Print view (with full texts) | Export | Display chart | Create view

### **movies\_locations:**

insert into movies\_locations (movie\_id, location\_id) values (1, 5);  
insert into movies\_locations (movie\_id, location\_id) values (2, 5);  
insert into movies\_locations (movie\_id, location\_id) values (3, 2);  
insert into movies\_locations (movie\_id, location\_id) values (3, 4);  
insert into movies\_locations (movie\_id, location\_id) values (5, 3);



localhost/phpmyadmin/sql.php?db=tah77&table=actors&pos=0&sql\_query=SELECT+%2A+FROM+%60tah77%60.%60actors%60+WHERE+%60actor\_id%60+%3D+5&token=8c...

Server: localhost » Database: tah77 » Table: movies\_locations

Current selection does not contain a unique column. Grid edit, checkbox, Edit, Copy and Delete features are not available.

Showing rows 0 - 4 (5 total, Query took 0.0004 seconds)

```
SELECT * FROM `movies_locations`
```

Profiling [ Edit inline ] [ Edit ] [ Explain SQL ] [ Create PHP Code ] [ Refresh ]

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

Sort by key: None

+ Options

movie_id	location_id
1	5
2	5
3	2
3	4
5	3

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

Query results operations

Print view | Print view (with full texts) | Export | Display chart | Create view

**Task 5 (5 points): Write a SELECT statement to display top 2 actors sorted by actor's last name.**

**SELECT \* FROM `actors` GROUP BY `last\_name` LIMIT 2;**

localhost/phpmyadmin/sql.php?db=tah77&table=actors&pos=0&sql\_query=SELECT+%2A+FROM+%60tah77%60.%60actors%60+WHERE+%60actor\_id%60+%3D+5&token=8c...

Server: localhost » Database: tah77 » Table: actors

Showing rows 0 - 1 (2 total, Query took 0.0005 seconds)

```
SELECT * FROM `actors` GROUP BY `last_name` LIMIT 2
```

Profiling [ Edit inline ] [ Edit ] [ Explain SQL ] [ Create PHP Code ] [ Refresh ]

Sort by key: None

+ Options

actor_id	first_name	last_name	birth_date	biography
4	Evelyn	Franklin	1988-10-12 03:00:00	evelyn franklin biography
3	Makai	Freeman	1997-08-30 02:00:00	another biography

Check All | With selected: Edit | Delete | Export

Query results operations

Print view | Print view (with full texts) | Export | Display chart | Create view

**Task 6 (5 points): Write a SELECT statement to display location name, street address, and city sorted by location name in descending order.**

SELECT location\_name, street\_address, city FROM locations ORDER BY location\_name DESC

The screenshot shows the phpMyAdmin interface for a database named 'tah77'. The 'locations' table is selected. The SQL query entered is: `SELECT location_name, street_address, city FROM locations ORDER BY location_name DESC`. The results are displayed in a table with 5 rows. The first row is highlighted in green.

location_name	street_address	city
the park	543 brick st	san diego
niagara falls	22 falls dr	niagara
movie location	123 street ave	new york
another place	44 second ave	pittsburgh
a big building	77 place rd	houston

**Task 7 (5 points): Write a SELECT statement to display movies released between two dates of your choice.**

select \* from Movies WHERE (release\_date BETWEEN '2000-04-01' AND '2018-04-1')

localhost/phpmyadmin/sql.php?db=tah77&table=actors&pos=0&sql\_query=SELECT+%2A+FROM+%60tah77%60.%60actors%60+WHERE+%60actor\_id%60+%3D+4&token=8c...

Server: localhost » Database: tah77 » Table: Movies

Showing rows 0 - 7 (8 total, Query took 0.0284 seconds.)

```
select * from Movies WHERE (release_date BETWEEN '2000-04-01' AND '2018-04-1')
```

Profiling [ Edit inline ] [ Edit ] [ Explain SQL ] [ Create PHP Code ] [ Refresh ]

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

Sort by key: None

+ Options

	movie_id	title	release_date	plot_description
<input type="checkbox"/> Edit Copy Delete	1	movietitle	2018-02-10 10:34:34	description
<input type="checkbox"/> Edit Copy Delete	2	movietwo	2004-12-03 10:24:04	a movie about things
<input type="checkbox"/> Edit Copy Delete	4	movie_title	2013-09-12 01:12:07	the plot of the movie
<input type="checkbox"/> Edit Copy Delete	5	the_fifth_movie	2009-05-17 10:02:07	description of the plot
<input type="checkbox"/> Edit Copy Delete	7	Green Destiny	2017-02-15 03:00:00	Sddfdjdjd jajaiwoeu asdlkajlsdj lksjd
<input type="checkbox"/> Edit Copy Delete	8	The Academy Edge	2008-03-16 09:00:00	wowowoow waoiudoasudosudi kwkaskdksdjsl
<input type="checkbox"/> Edit Copy Delete	9	Moon of Serpent	2012-06-03 00:00:00	asldkasjdklsdj asldjslkdsjkd iaouwieuwoieuwoeu as...
<input type="checkbox"/> Edit Copy Delete	12	The Bridge of the Witches	2010-08-08 02:00:00	aldk sjdlkasjd sadk lsa wioeuwoieuwoieu asjsjs als...

Check All | With selected: Edit Delete Export

**Task 8 (5 points): Write an UPDATE statement to update zip code for all locations to 15217**  
**UPDATE locations SET zip='15217';**

localhost/phpmyadmin/sql.php?db=tah77&table=actors&pos=0&sql\_query=SELECT+%2A+FROM+%60tah77%60.%60actors%60+WHERE+%60actor\_id%60+%3D+4&token=8c...

Server: localhost » Database: tah77 » Table: locations

Showing rows 0 - 4 (5 total, Query took 0.0004 seconds.)

```
SELECT * FROM `locations`
```

Profiling [ Edit inline ] [ Edit ] [ Explain SQL ] [ Create PHP Code ] [ Refresh ]

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

Sort by key: None

+ Options

	location_id	location_name	street_address	city	state	zip
<input type="checkbox"/> Edit Copy Delete	1	movie location	123 street ave	new york	NY	15217
<input type="checkbox"/> Edit Copy Delete	2	another place	44 second ave	pittsburgh	PA	15217
<input type="checkbox"/> Edit Copy Delete	3	a big building	77 place rd	houston	TX	15217
<input type="checkbox"/> Edit Copy Delete	4	the park	543 brick st	san diego	CA	15217
<input type="checkbox"/> Edit Copy Delete	5	niagara falls	22 falls dr	niagara	NY	15217

Check All | With selected: Edit Delete Export

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

Query results operations

**Task 9 (5 points): Write a statement that deletes one row of your choice from the actor's table. Be careful – make sure to write a correct WHERE clause.**  
**DELETE FROM `actors` WHERE actor\_id = '13';**

The screenshot shows the phpMyAdmin interface for a database named 'tah77'. The 'actors' table is selected, and its data is displayed in a table view. The table has 13 rows. The columns are: actor\_id, first\_name, last\_name, birth\_date, and biography. Row 13 is highlighted, showing actor\_id 13, first\_name bob, last\_name smith, birth\_date 2000-02-12 00:00:00, and biography djflladkldfjldj.

actor_id	first_name	last_name	birth_date	biography
1	Neely	Jacque	1972-05-10 10:00:00	biography here
2	Maria	McBee	1992-04-20 09:00:00	actor biography
3	Makai	Freeman	1997-08-30 02:00:00	another biography
4	Evelyn	Franklin	1988-10-12 03:00:00	evelyn franklin biography
5	Jed	Moro	1990-08-11 01:00:00	jed moro biography
6	Twanda	Ribiero	1970-02-20 00:00:00	akdjadlfdkfdj idfjlafj iuwoaeiwauu
7	Eda	Kovar	1968-01-14 04:00:00	a biography for eda
8	Jacinto	Caberra	2019-06-08 07:00:00	blah blah blah blah
9	Eliza	Malson	1992-06-06 02:00:00	eliza malson bio
10	Rashad	Paff	2018-10-16 10:00:00	things that happened
11	Raquele	Gularte	1984-08-10 05:00:00	ssiuftidaifu kslafjdj thtthttht
12	Ayanna	Ortman	2018-10-16 07:00:00	oooooooooooooooooooooooooooo
13	bob	smith	2000-02-12 00:00:00	djflldkldfjldj

**Before (actor\_id = 13 exists in table)**

The screenshot shows the phpMyAdmin interface for a database named 'tah77'. The 'actors' table is selected, and its data is displayed in a table view. The table has 12 rows. The columns are: actor\_id, first\_name, last\_name, birth\_date, and biography. Row 13 is missing.

actor_id	first_name	last_name	birth_date	biography
1	Neely	Jacque	1972-05-10 10:00:00	biography here
2	Maria	McBee	1992-04-20 09:00:00	actor biography
3	Makai	Freeman	1997-08-30 02:00:00	another biography
4	Evelyn	Franklin	1988-10-12 03:00:00	evelyn franklin biography
5	Jed	Moro	1990-08-11 01:00:00	jed moro biography
6	Twanda	Ribiero	1970-02-20 00:00:00	akdjadlfdkfdj idfjlafj iuwoaeiwauu
7	Eda	Kovar	1968-01-14 04:00:00	a biography for eda
8	Jacinto	Caberra	2019-06-08 07:00:00	blah blah blah blah
9	Eliza	Malson	1992-06-06 02:00:00	eliza malson bio
10	Rashad	Paff	2018-10-16 10:00:00	things that happened
11	Raquele	Gularte	1984-08-10 05:00:00	ssiuftidaifu kslafjdj thtthttht
12	Ayanna	Ortman	2018-10-16 07:00:00	oooooooooooooooooooooooooooo

**Task 10 (5 Points): Write a SELECT statement to display movies which are played by any actor/actress who is a Gemini (or pick one Horoscope Sign you like)**

**\*shows movies with pisces actors**

SELECT m.\* FROM movies as m, movies\_actors as a WHERE m.movie\_id = a.movie\_id and a.actor\_id IN (SELECT actor\_id FROM actors WHERE DATE\_FORMAT(birth\_date, '%m-%d') BETWEEN '02-19' AND '03-20')

The screenshot shows the phpMyAdmin interface for a database named 'tah77'. The 'movies' table is selected. A SQL query is entered in the query box, and the results are displayed below it. The query is:   
 SELECT m.\* FROM movies as m, movies\_actors as a WHERE m.movie\_id = a.movie\_id and a.actor\_id IN (SELECT actor\_id FROM actors WHERE DATE\_FORMAT(birth\_date, '%m-%d') BETWEEN '02-19' AND '03-20')   
 The results show two rows:   
 12 The Bridge of the Witches 2010-08-08 02:00:00 aldk sjldkasjd sadk lsa wioeuwoiuewoiue asjsj als...   
 11 Deep Night 1999-07-31 04:00:00 jafdljadjlf kwiiuivueue alksdjaksjdalsjd   
 The interface also shows the database structure on the left and various options for the query results.

**Task 11 (5 points): Write a SELECT statement to display movies which the main cast (all actor/actress) has the highest average age.**

**\*Shows top 3 movies with highest avg age**

SELECT \* FROM movies WHERE movie\_id in (SELECT ids.movie\_id FROM (SELECT ma.movie\_id as movie\_id, AVG(DATEDIFF(CURDATE(), a.birth\_date)) as avg\_age FROM actors a JOIN movies\_actors ma ON a.actor\_id = ma.actor\_id GROUP BY ma.movie\_id ORDER BY avg\_age DESC LIMIT 3) AS ids)

Server: localhost » Database: tah77 » Table: movies

Showing rows 0 - 2 (3 total, Query took 0.0011 seconds.)

```
SELECT * FROM movies WHERE movie_id in (SELECT ids.movie_id FROM (SELECT ma.movie_id as movie_id, AVG(DATEDIFF(CURDATE(), a.birth_date)) as avg_age FROM actors a JOIN movies_actors ma ON a.actor_id = ma.actor_id GROUP BY ma.movie_id ORDER BY avg_age DESC LIMIT 3) AS ids)
```

Profiling [ Edit inline ] [ Edit ] [ Explain SQL ] [ Create PHP Code ] [ Refresh ]

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

Sort by key: None

+ Options

	movie_id	title	release_date	plot_description
<input type="checkbox"/> Edit Copy Delete	11	Deep Night	1999-07-31 04:00:00	jafdljadljl kwluwueue alksdlaksjdlasjd
<input type="checkbox"/> Edit Copy Delete	5	the_fifth_movie	2009-05-17 10:02:07	description of the plot
<input type="checkbox"/> Edit Copy Delete	4	movie_title	2013-09-12 01:12:07	the plot of the movie

Check All With selected: Edit Delete Export

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

Query results operations

Print view Print view (with full texts) Export Display chart Create view

**Task 12 (5 points): Write a SELECT statement to display the Greatest Extra (<http://www.imdb.com/title/tt3087298/>) in your movie database, i.e., the actors who have appeared in the most movies.**

**\*shows list of actors in descending order of # of movies (actor\_id = 3 is the 'greatest extra' currently)**

SELECT a.\* FROM movies m, actors a, movies\_actors ma WHERE m.movie\_id = ma.movie\_id AND a.actor\_id = ma.actor\_id GROUP BY a.actor\_id ORDER BY SUM(m.movie\_id);

localhost/phpmyadmin/sql.php?db=tah77&table=movies\_actors&server=1&target=&token=8cb0f931c4dd606a25ded416428b2af7#PMAURL-8:tbl\_sql.php?db=tah77&table=loc...

phpMyAdmin

Recent Favorites

New  
lab  
tah77  
New  
actors  
locations  
Movies  
movies\_actors  
Columns  
Indexes  
New  
fk\_ma\_actors  
fk\_ma\_movies  
movies\_locations  
Columns  
Indexes  
New  
fk\_ml\_location  
fk\_ml\_movie

Server: localhost Database: tah77 Table: movies

Browse Structure SQL Search Insert Export Import Privileges Operations Triggers

Show query box

⚠ Current selection does not contain a unique column. Grid edit, checkbox, Edit, Copy and Delete features are not available.

✓ Showing rows 0 - 10 (11 total. Query took 0.0010 seconds.)

```
SELECT a.* FROM movies m, actors a, movies_actors ma WHERE m.movie_id = ma.movie_id AND a.actor_id = ma.actor_id GROUP BY a.actor_id ORDER BY SUM(m.movie_id)
```

☐ Profiling [ Edit inline ] [ Edit ] [ Explain SQL ] [ Create PHP Code ] [ Refresh ]

☐ Show all Number of rows: 25 Filter rows: Search this table

+ Options

actor_id	first_name	last_name	birth_date	biography
3	Makai	Freeman	1997-08-30 02:00:00	another biography
11	Raquele	Gularte	1984-08-10 05:00:00	ssiuftidaufu kslafjdj thtthttht
9	Eliza	Malson	1992-06-06 02:00:00	eliza malson bio
4	Evelyn	Franklin	1988-10-12 03:00:00	evelyn franklin biography
2	Maria	McBee	1992-04-20 09:00:00	actor biography
5	Jed	Moro	1990-08-11 01:00:00	jed moro biography
8	Jacinto	Caberra	1982-06-08 07:00:00	blah blah blah blah
10	Rashad	Paff	2001-10-16 10:00:00	things that happened
7	Eda	Kovar	1968-01-14 04:00:00	a biography for eda
6	Twanda	Ribiero	1970-02-20 00:00:00	akdjadjackdj idfjafj iuwoeiwaueu
1	Neely	Jacque	1972-05-10 10:00:00	biography here

Type here to search

8:56 PM 2/19/2019

**Task 13 (5 Points): Write a SELECT statement to display the number of actor occurrence count of each city, i.e., how many actors ever appear in each city? [Output: locations.city, "number of actor occurrence count"]**

SELECT COUNT(a.actor\_id), l.city FROM locations l, actors a, movies m, movies\_actors ma, movies\_locations ml WHERE ma.actor\_id = a.actor\_id AND l.location\_id = ml.location\_id AND ma.movie\_id = m.movie\_id AND ml.movie\_id = m.movie\_id GROUP BY l.city

localhost/phpmyadmin/sql.php?db=tah77&table=movies\_actors&server=1&target=&token=8cb0f931c4dd606a25ded416428b2af7#PMAURL-7:tbl\_sql.php?db=tah77&table=m...

phpMyAdmin

Recent Favorites

New  
lab  
tah77  
New  
actors  
locations  
Movies  
movies\_actors  
Columns  
Indexes  
New  
fk\_ma\_actors  
fk\_ma\_movies  
movies\_locations  
Columns  
Indexes  
New  
fk\_ml\_location  
fk\_ml\_movie

Server: localhost Database: tah77 Table: locations

Browse Structure SQL Search Insert Export Import Privileges Operations Triggers

Show query box

⚠ Current selection does not contain a unique column. Grid edit, checkbox, Edit, Copy and Delete features are not available.

✓ Showing rows 0 - 4 (5 total. Query took 0.0010 seconds.)

```
SELECT COUNT(a.actor_id), l.city FROM locations l, actors a, movies m, movies_actors ma, movies_locations ml WHERE ma.actor_id = a.actor_id AND l.location_id = ml.location_id AND ma.movie_id = m.movie_id AND ml.movie_id = m.movie_id GROUP BY l.city
```

☐ Profiling [ Edit inline ] [ Edit ] [ Explain SQL ] [ Create PHP Code ] [ Refresh ]

☐ Show all Number of rows: 25 Filter rows: Search this table

+ Options

COUNT(a.actor_id)	city
12	houston
6	new york
7	niagara
5	pittsburgh
2	san diego

☐ Show all Number of rows: 25 Filter rows: Search this table

Query results operations

Type here to search

8:52 PM 2/19/2019



**Task 14 (5 Points):** Write a **SELECT** statement to display an actor-to-actor network who ever act more than once in the same movie, i.e., to show a list of actor ID pair (ActorID1, ActorID2) that co-act more than one time.

SELECT a1.actor\_id,a2.actor\_id FROM movies\_actors a1, movies\_actors a2  
WHERE a1.movie\_id = a2.movie\_id AND a1.actor\_id < a2.actor\_id GROUP BY  
a1.actor\_id,a2.actor\_id HAVING COUNT(\*) > 1

The screenshot shows the phpMyAdmin interface for a database named 'tah77'. The 'movies\_actors' table is selected. A SQL query is entered in the 'Show query box' and executed. The query is: `SELECT a1.actor_id,a2.actor_id FROM movies_actors a1, movies_actors a2 WHERE a1.movie_id = a2.movie_id AND a1.actor_id < a2.actor_id GROUP BY a1.actor_id,a2.actor_id HAVING COUNT(*) > 1`. The results show 6 rows of actor ID pairs.

Server: localhost » Database: tah77 » Table: movies\_actors

Show query box

⚠ Current selection does not contain a unique column. Grid edit, checkbox, Edit, Copy and Delete features are not available.

✓ Showing rows 0 - 5 (6 total, Query took 0.0010 seconds.)

```
SELECT a1.actor_id,a2.actor_id FROM movies_actors a1, movies_actors a2 WHERE a1.movie_id = a2.movie_id AND a1.actor_id < a2.actor_id GROUP BY a1.actor_id,a2.actor_id HAVING COUNT(*) > 1
```

☐ Profiling [\[ Edit inline \]](#) [\[ Edit \]](#) [\[ Explain SQL \]](#) [\[ Create PHP Code \]](#) [\[ Refresh \]](#)

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

+ Options

actor_id	actor_id
1	5
1	6
1	7
1	8
6	7
6	8

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

Query results operations

**\*Hint: for some questions, you may need to insert more data so that you can see the meaningful output.**