

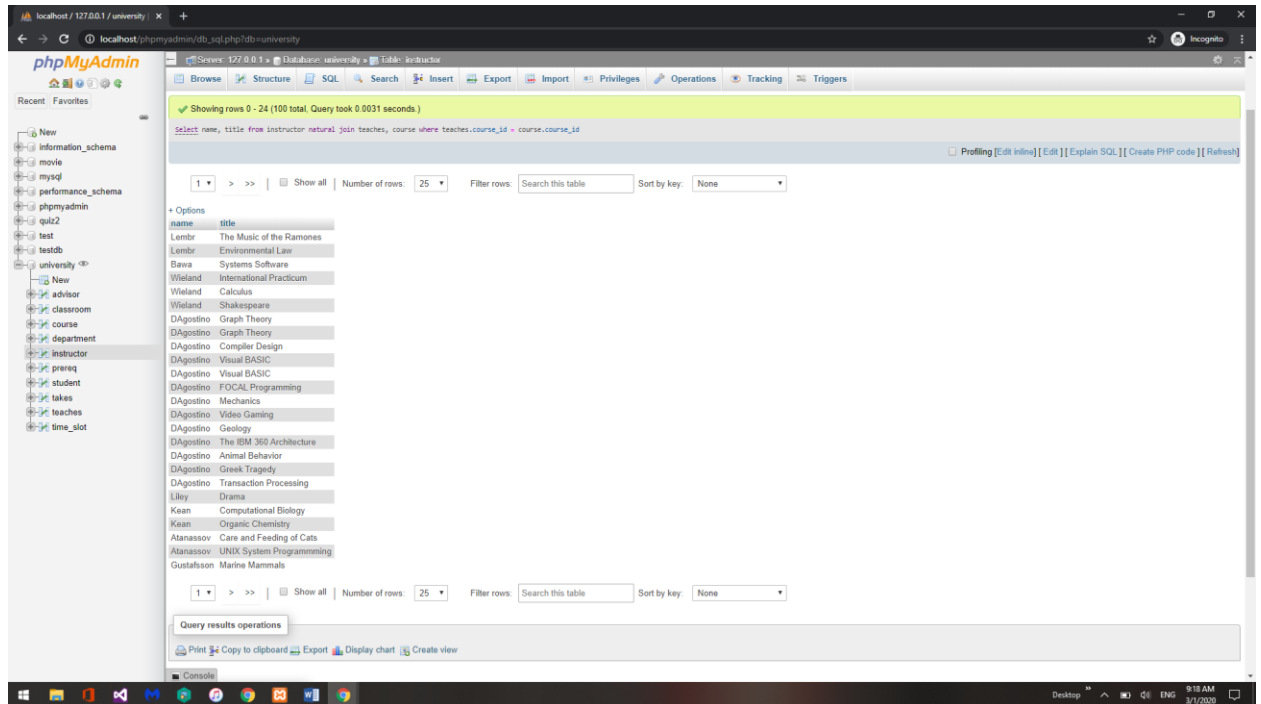
Chris Grimes  
CS 33007  
Homework 3  
Part1: Practice Queries [35 points]

Execute following queries and observe the results

1. List names of the instructors along with the title of the courses they have taught.

(observe results of these queries).

- Select name, title from instructor natural join teaches, course where teaches.course\_id = course.course\_id



Showing rows 0 - 24 (100 total, Query took 0.0031 seconds)

select name, title from instructor natural join teaches, course where teaches.course\_id = course.course\_id

name	title
Lembr	The Music of the Ramones
Lembr	Environmental Law
Bawa	Systems Software
Wieland	International Practicum
Wieland	Calculus
Wieland	Shakespeare
D'Agostino	Graph Theory
D'Agostino	Graph Theory
D'Agostino	Compiler Design
D'Agostino	Visual BASIC
D'Agostino	Visual BASIC
D'Agostino	FOCAL Programming
D'Agostino	Mechanics
D'Agostino	Video Gaming
D'Agostino	Geology
D'Agostino	The IBM 360 Architecture
D'Agostino	Animal Behavior
D'Agostino	Greek Tragedy
D'Agostino	Transaction Processing
Ullery	Drama
Kean	Computational Biology
Kean	Organic Chemistry
Atanassov	Care and Feeding of Cats
Atanassov	UNIX System Programming
Gustafsson	Marine Mammals

This query shows the name of the instructor and the title of the class said instructor is teaching.

Chris Grimes

CS 33007

Homework 3

b. Select name, title from instructor natural join teaches natural join course

The screenshot shows the phpMyAdmin interface with a query executed. The query is: `select name, title from instructor natural join teaches natural join course`. The result shows 24 rows. The table lists instructors and their associated course titles.

name	title
Lembr	The Music of the Ramones
Wieland	Calculus
Wieland	Shakespeare
D'Agostino	Graph Theory
D'Agostino	Graph Theory
D'Agostino	Compiler Design
D'Agostino	Visual BASIC
D'Agostino	Visual BASIC
D'Agostino	FOCAL Programming
D'Agostino	Mechanics
D'Agostino	Video Gaming
D'Agostino	Geology
D'Agostino	Animal Behavior
D'Agostino	Greek Tragedy
D'Agostino	Transaction Processing
Liley	Drama
Kean	Computational Biology
Kean	Organic Chemistry
Atanassov	Care and Feeding of Cats
Atanassov	UNIX System Programming
Gustafsson	Marine Mammals
Gustafsson	Marine Mammals
Gustafsson	The Music of Donovan
Gustafsson	Plasma Physics
Bourlier	Japanese

This query shows the same results as A since the natural joins is joining the tables based on common attributes.

c. Select name, title from instructor natural join teaches join course using (course\_id)

The screenshot shows the phpMyAdmin interface with a query executed. The query is: `select name, title from instructor natural join teaches join course using (course_id)`. The result shows 24 rows. The table lists instructors and their associated course titles.

name	title
Lembr	The Music of the Ramones
Lembr	Environmental Law
Bawa	Systems Software
Wieland	International Practicum
Wieland	Calculus
Wieland	Shakespeare
D'Agostino	Graph Theory
D'Agostino	Graph Theory
D'Agostino	Compiler Design
D'Agostino	Visual BASIC
D'Agostino	Visual BASIC
D'Agostino	FOCAL Programming
D'Agostino	Mechanics
D'Agostino	Video Gaming
D'Agostino	Geology
D'Agostino	The IBM 360 Architecture
D'Agostino	Animal Behavior
D'Agostino	Greek Tragedy
D'Agostino	Transaction Processing
Liley	Drama
Kean	Computational Biology
Kean	Organic Chemistry
Atanassov	Care and Feeding of Cats
Atanassov	UNIX System Programming
Gustafsson	Marine Mammals

This query show the same results but was specifically joined on the common attribute course\_id instead of letting the DBMS choose the common attribute

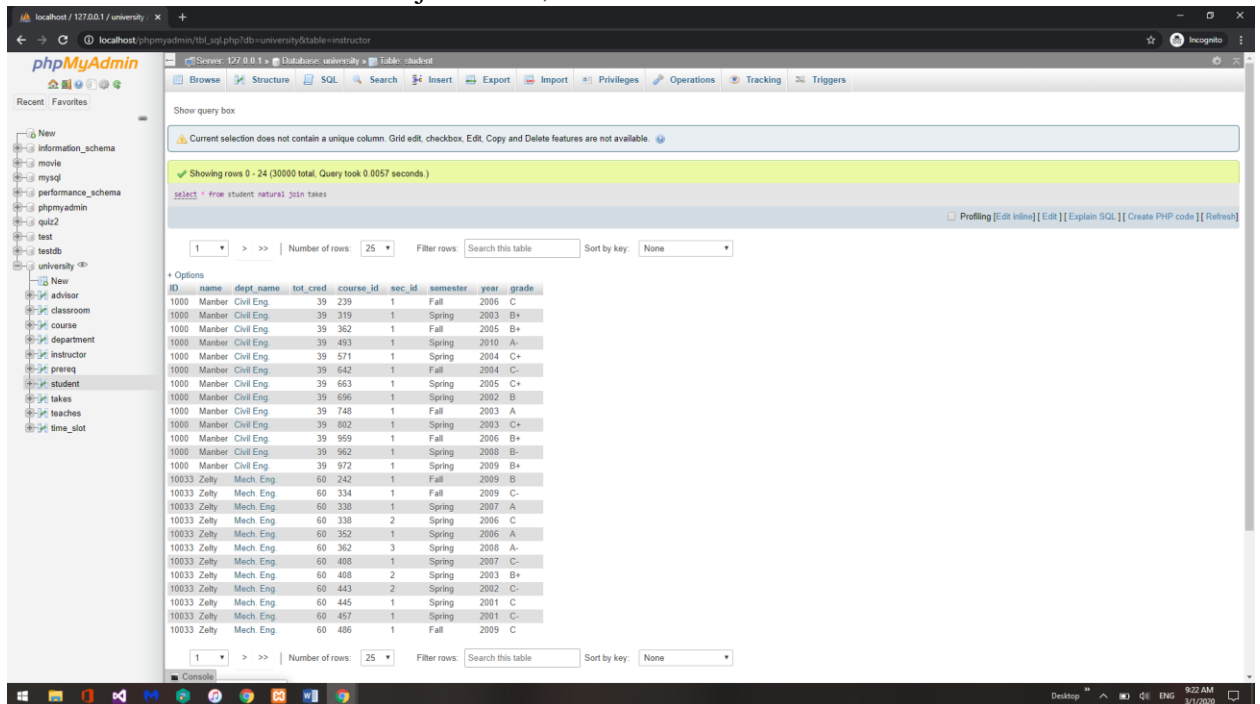
Chris Grimes

CS 33007

Homework 3

2. List of all students, displaying their ID, and name, dept name, and tot cred, along with the courses that they have taken (observe the differences in the result of the queries)

a. `select * from student natural join takes;`



The screenshot shows the phpMyAdmin interface with a query executed: `select * from student natural join takes;`. The result displays 25 rows of data, showing the natural join of the `student` and `takes` tables. The columns include `ID`, `name`, `dept_name`, `tot_cred`, `course_id`, `sec_id`, `semester`, `year`, and `grade`.

ID	name	dept_name	tot_cred	course_id	sec_id	semester	year	grade
1000	Manber	Civil Eng.	39	239	1	Fall	2006	C
1000	Manber	Civil Eng.	39	319	1	Spring	2003	B+
1000	Manber	Civil Eng.	39	362	1	Fall	2005	B+
1000	Manber	Civil Eng.	39	493	1	Spring	2010	A-
1000	Manber	Civil Eng.	39	571	1	Spring	2004	C+
1000	Manber	Civil Eng.	39	642	1	Fall	2004	C-
1000	Manber	Civil Eng.	39	663	1	Spring	2005	C+
1000	Manber	Civil Eng.	39	696	1	Spring	2002	B
1000	Manber	Civil Eng.	39	748	1	Fall	2003	A
1000	Manber	Civil Eng.	39	802	1	Spring	2003	C+
1000	Manber	Civil Eng.	39	959	1	Fall	2006	B+
1000	Manber	Civil Eng.	39	962	1	Spring	2008	B-
1000	Manber	Civil Eng.	39	972	1	Spring	2009	B+
10033	Zelby	Mech. Eng.	60	242	1	Fall	2009	B
10033	Zelby	Mech. Eng.	60	334	1	Fall	2009	C-
10033	Zelby	Mech. Eng.	60	338	1	Spring	2007	A
10033	Zelby	Mech. Eng.	60	338	2	Spring	2006	C
10033	Zelby	Mech. Eng.	60	352	1	Spring	2006	A
10033	Zelby	Mech. Eng.	60	362	3	Spring	2008	A-
10033	Zelby	Mech. Eng.	60	408	1	Spring	2007	C-
10033	Zelby	Mech. Eng.	60	408	2	Spring	2003	B+
10033	Zelby	Mech. Eng.	60	443	2	Spring	2002	C-
10033	Zelby	Mech. Eng.	60	445	1	Spring	2001	C
10033	Zelby	Mech. Eng.	60	457	1	Spring	2001	C
10033	Zelby	Mech. Eng.	60	486	1	Fall	2009	C

This query shows results from student being naturally joined by takes

Chris Grimes

CS 33007

Homework 3

b. select \* from student natural left outer join takes;

The screenshot shows the phpMyAdmin interface with a query executed: `SELECT * FROM student NATURAL LEFT OUTER JOIN takes`. The result set contains 24 rows. The columns are: ID, name, dept\_name, tot\_cred, course\_id, sec\_id, semester, year, and grade. The data is as follows:

ID	name	dept_name	tot_cred	course_id	sec_id	semester	year	grade
1000	Manber	Civil Eng.	39	239	1	Fall	2006	C
1000	Manber	Civil Eng.	39	319	1	Spring	2003	B+
1000	Manber	Civil Eng.	39	362	1	Fall	2005	B+
1000	Manber	Civil Eng.	39	493	1	Spring	2010	A-
1000	Manber	Civil Eng.	39	571	1	Spring	2004	C+
1000	Manber	Civil Eng.	39	642	1	Fall	2004	C-
1000	Manber	Civil Eng.	39	663	1	Spring	2005	C+
1000	Manber	Civil Eng.	39	696	1	Spring	2002	B
1000	Manber	Civil Eng.	39	748	1	Fall	2003	A
1000	Manber	Civil Eng.	39	802	1	Spring	2003	C+
1000	Manber	Civil Eng.	39	959	1	Fall	2006	B+
1000	Manber	Civil Eng.	39	962	1	Spring	2008	B-
1000	Manber	Civil Eng.	39	972	1	Spring	2009	B+
10033	Zelky	Mech. Eng.	60	242	1	Fall	2009	B
10033	Zelky	Mech. Eng.	60	334	1	Fall	2009	C-
10033	Zelky	Mech. Eng.	60	338	1	Spring	2007	A
10033	Zelky	Mech. Eng.	60	338	2	Spring	2006	C
10033	Zelky	Mech. Eng.	60	352	1	Spring	2006	A
10033	Zelky	Mech. Eng.	60	362	3	Spring	2008	A-
10033	Zelky	Mech. Eng.	60	408	1	Spring	2007	C-
10033	Zelky	Mech. Eng.	60	408	2	Spring	2003	B+
10033	Zelky	Mech. Eng.	60	443	2	Spring	2002	C-
10033	Zelky	Mech. Eng.	60	445	1	Spring	2001	C
10033	Zelky	Mech. Eng.	60	457	1	Spring	2001	C-
10033	Zelky	Mech. Eng.	60	486	1	Fall	2009	C

This query shows the same results as A since there are no empty columns in the previous query.

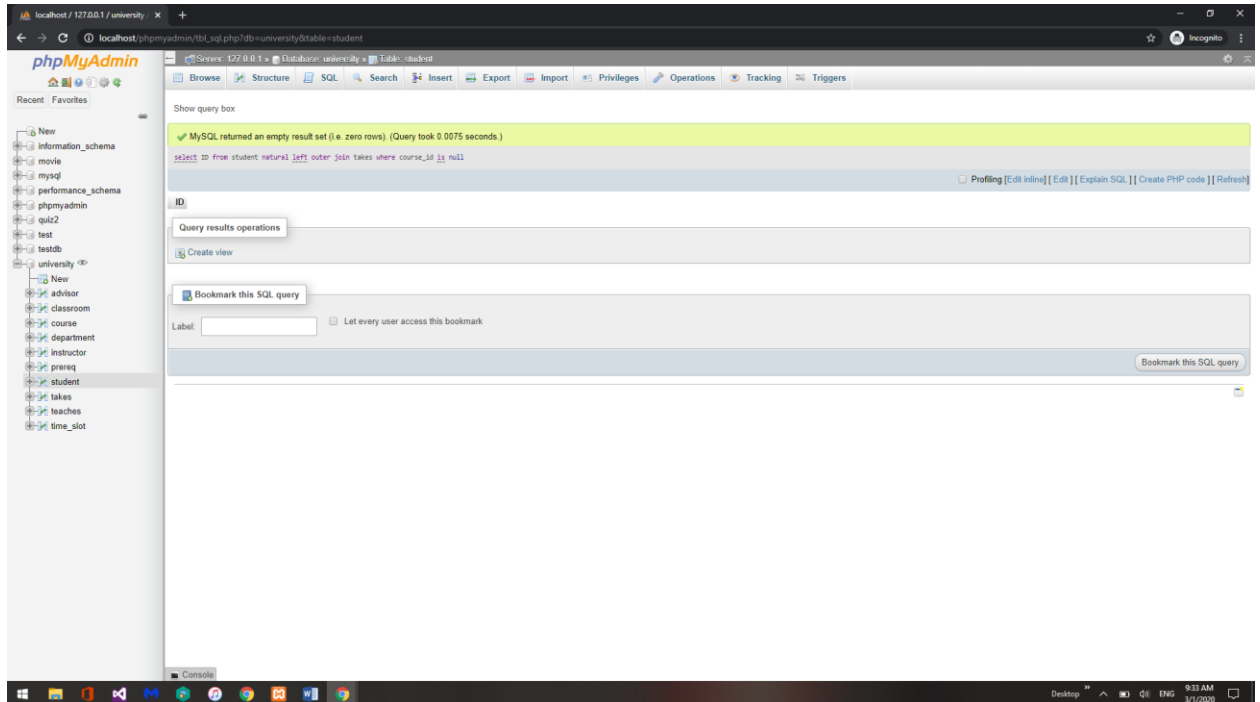
c. select \* from takes natural right outer join student;

The screenshot shows the phpMyAdmin interface with a query executed: `SELECT * FROM takes NATURAL RIGHT OUTER JOIN student`. The result set contains 24 rows, identical to the previous query. The columns are: ID, name, dept\_name, tot\_cred, course\_id, sec\_id, semester, year, and grade. The data is as follows:

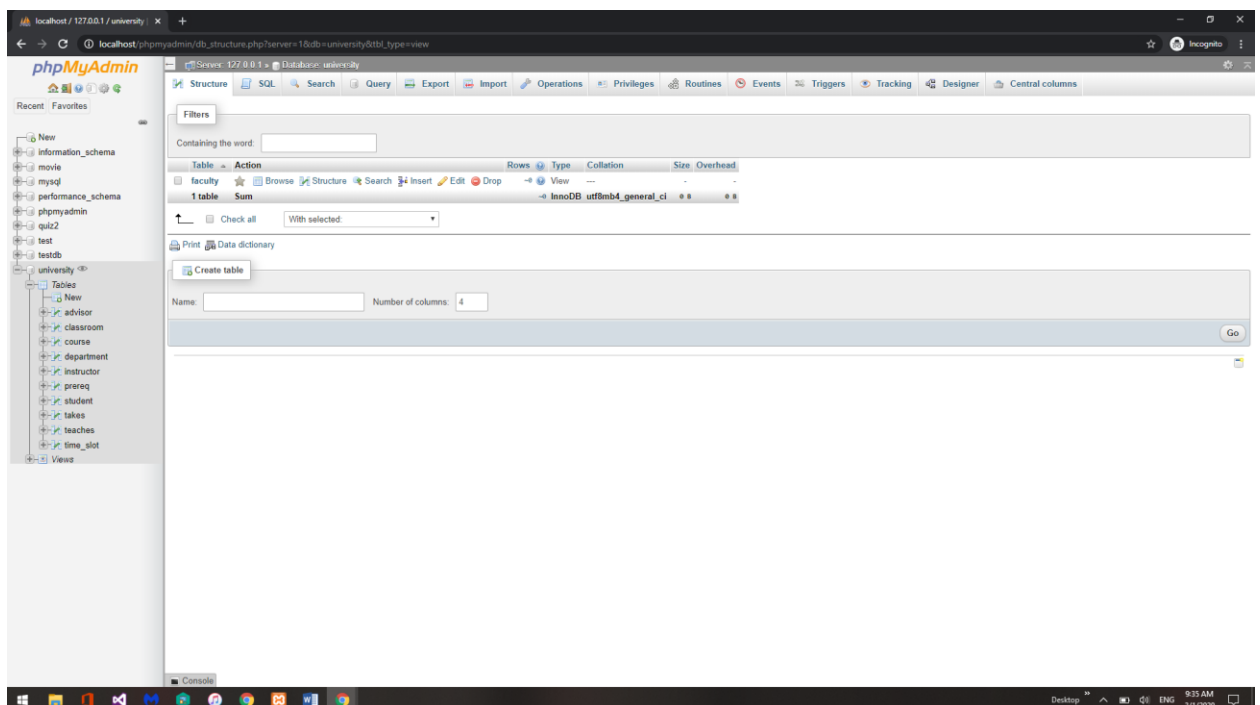
ID	name	dept_name	tot_cred	course_id	sec_id	semester	year	grade
1000	Manber	Civil Eng.	39	239	1	Fall	2006	C
1000	Manber	Civil Eng.	39	319	1	Spring	2003	B+
1000	Manber	Civil Eng.	39	362	1	Fall	2005	B+
1000	Manber	Civil Eng.	39	493	1	Spring	2010	A-
1000	Manber	Civil Eng.	39	571	1	Spring	2004	C+
1000	Manber	Civil Eng.	39	642	1	Fall	2004	C-
1000	Manber	Civil Eng.	39	663	1	Spring	2005	C+
1000	Manber	Civil Eng.	39	696	1	Spring	2002	B
1000	Manber	Civil Eng.	39	748	1	Fall	2003	A
1000	Manber	Civil Eng.	39	802	1	Spring	2003	C+
1000	Manber	Civil Eng.	39	959	1	Fall	2006	B+
1000	Manber	Civil Eng.	39	962	1	Spring	2008	B-
1000	Manber	Civil Eng.	39	972	1	Spring	2009	B+
10033	Zelky	Mech. Eng.	60	242	1	Fall	2009	B
10033	Zelky	Mech. Eng.	60	334	1	Fall	2009	C-
10033	Zelky	Mech. Eng.	60	338	1	Spring	2007	A
10033	Zelky	Mech. Eng.	60	338	2	Spring	2006	C
10033	Zelky	Mech. Eng.	60	352	1	Spring	2006	A
10033	Zelky	Mech. Eng.	60	362	3	Spring	2008	A-
10033	Zelky	Mech. Eng.	60	408	1	Spring	2007	C-
10033	Zelky	Mech. Eng.	60	408	2	Spring	2003	B+
10033	Zelky	Mech. Eng.	60	443	2	Spring	2002	C-
10033	Zelky	Mech. Eng.	60	445	1	Spring	2001	C
10033	Zelky	Mech. Eng.	60	457	1	Spring	2001	C-
10033	Zelky	Mech. Eng.	60	486	1	Fall	2009	C

This query shows the same results as A since there are no empty columns in the previous query.

3. Find all students who have not taken a course” select ID from student natural left outer join takes where course id is null;

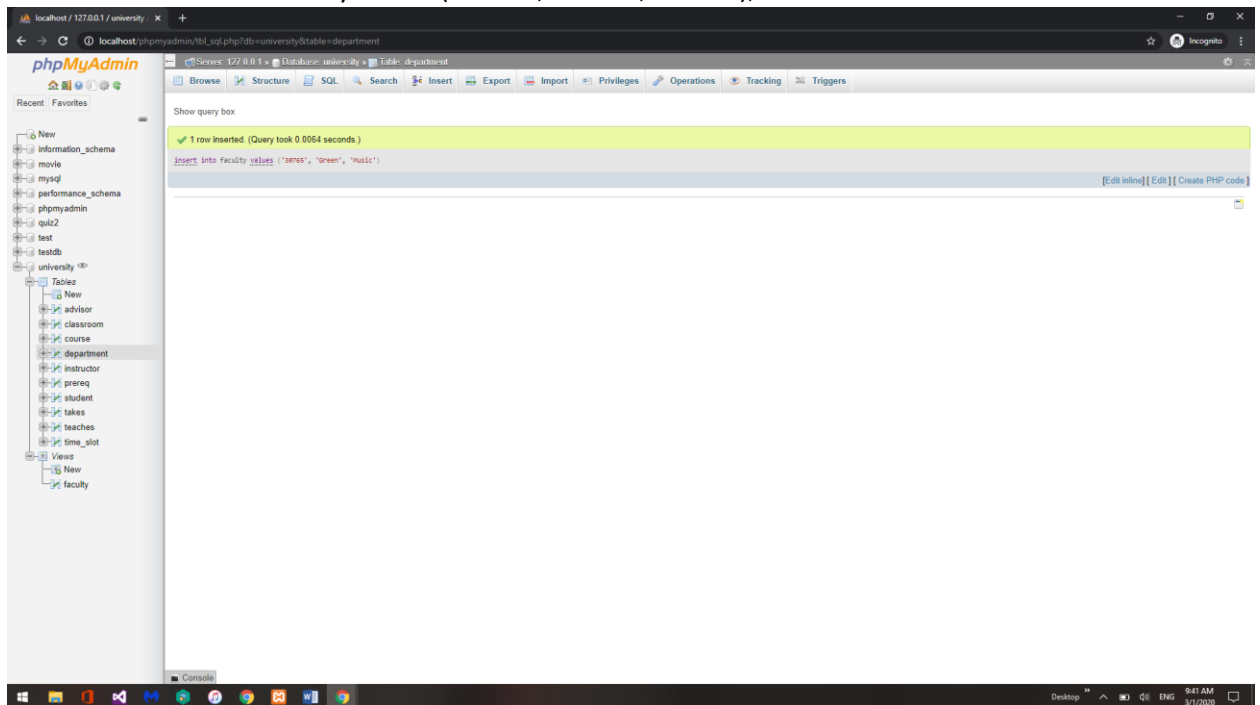


4. Create a view of instructors without their salary create view faculty as select ID, name, dept\_name from instructor

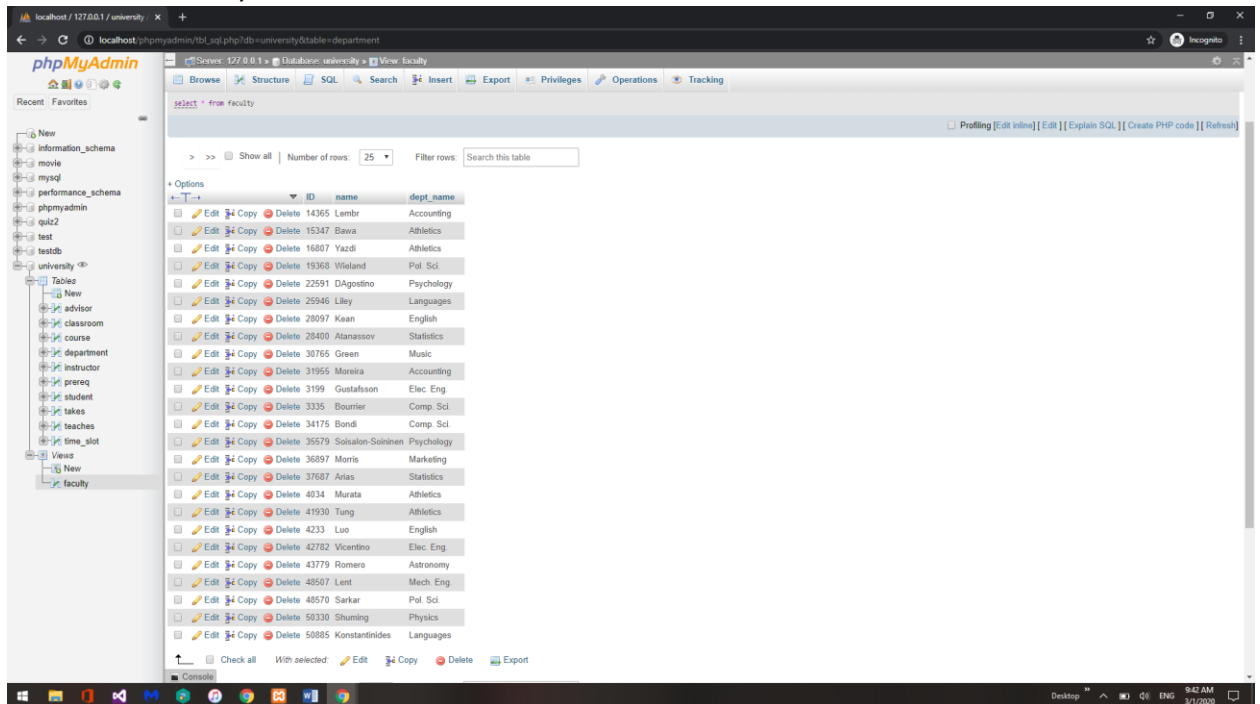


Chris Grimes  
CS 33007  
Homework 3

5. Insert a record into the view created in 4, then display the view to show that insertion was successful. insert into faculty values ('30765', 'Green', 'Music');



select \* from faculty

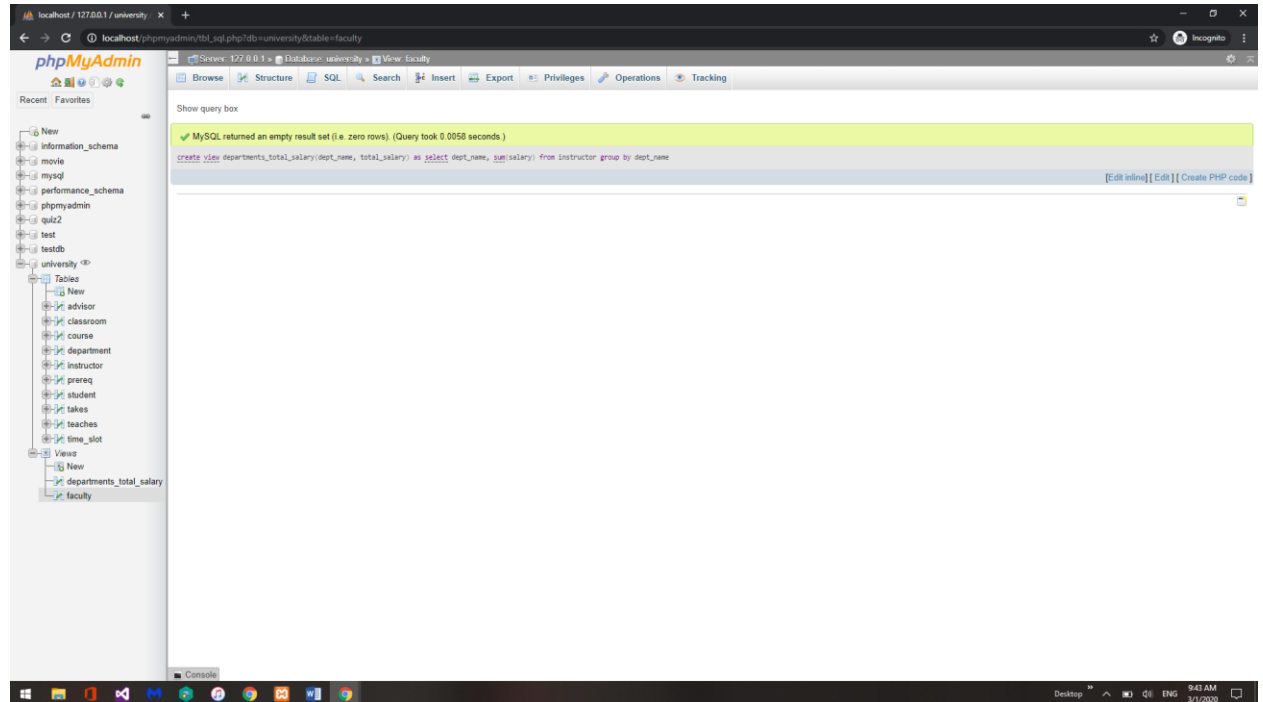


Chris Grimes

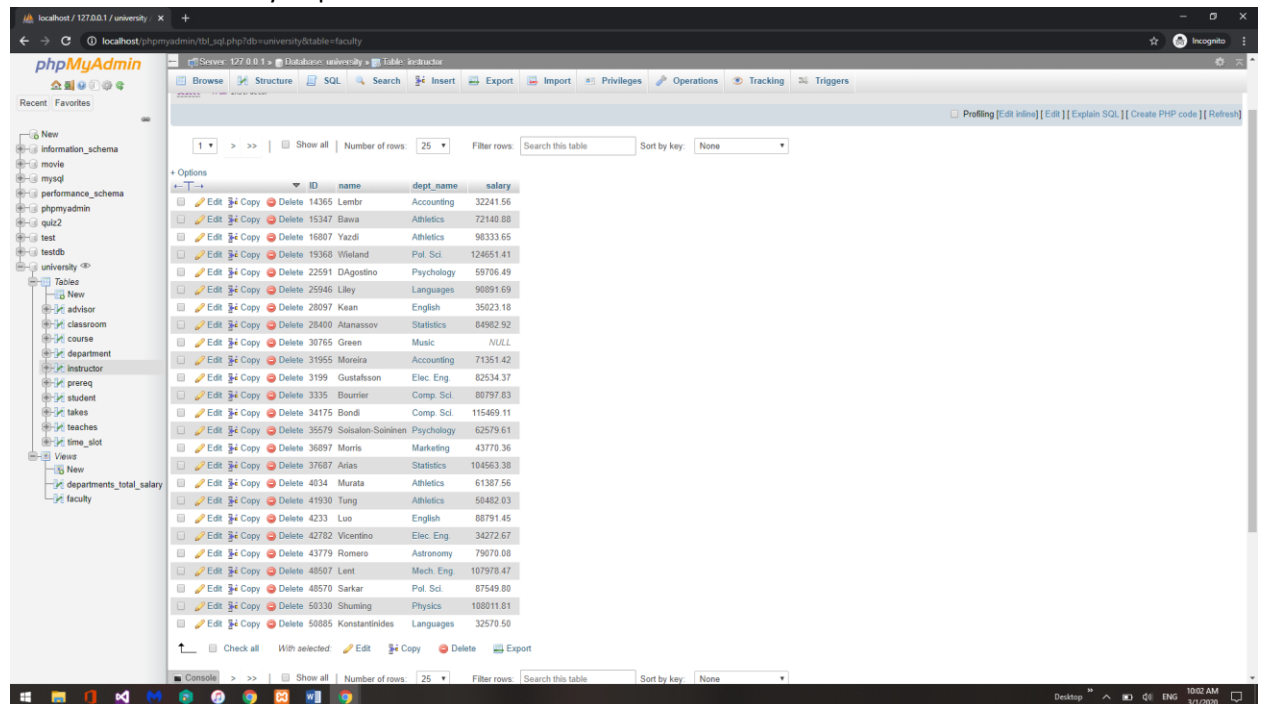
CS 33007

Homework 3

6. Create a view of department salary totals create view departments\_total\_salary(dept\_name, total\_salary) as select dept\_name, sum(salary) from instructor group by dept\_name;

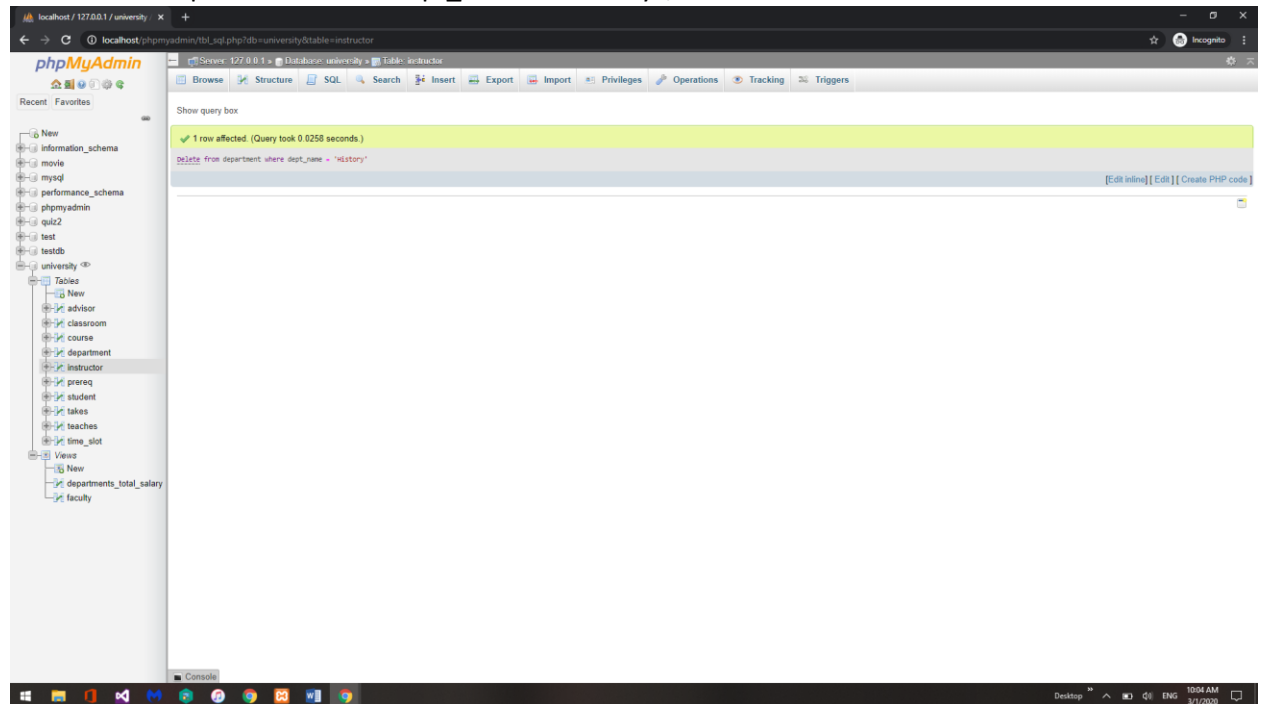


7. Delete a record from department table and show the effect on instructor table. ( Effect of integrity constraints)- report the changes. Select \* from instructor; // observe records of instructor from history department

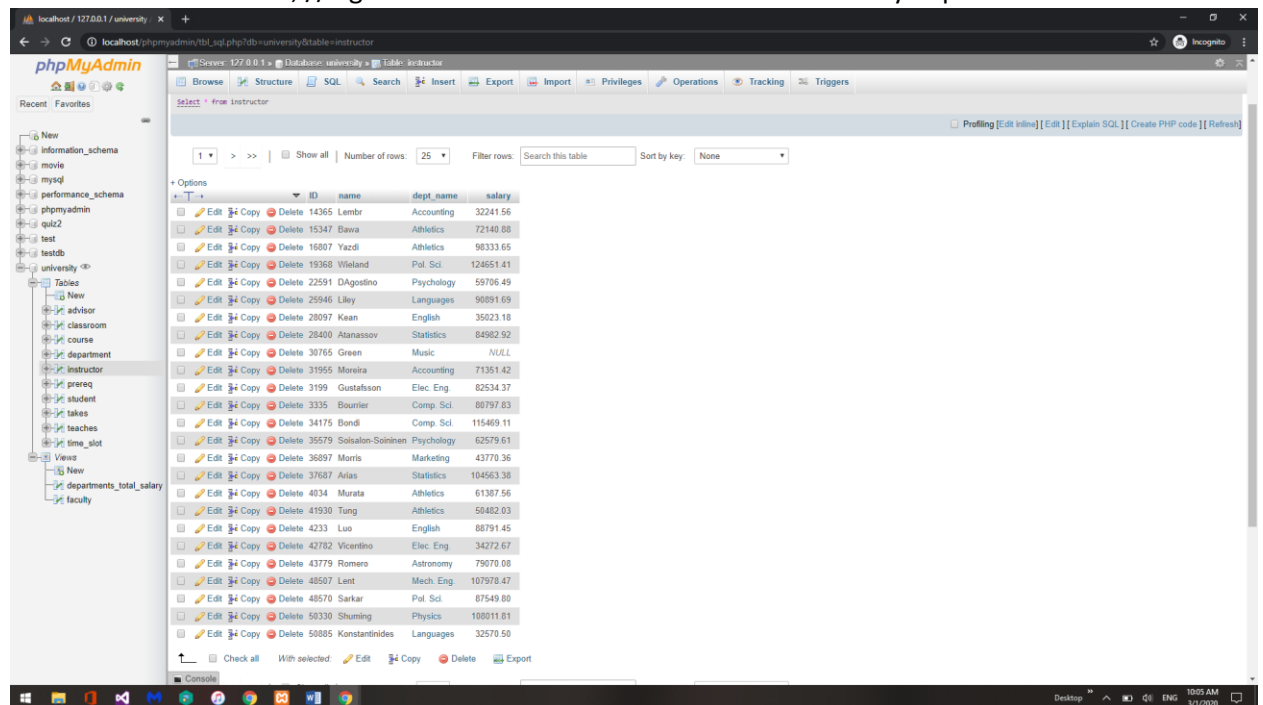


Chris Grimes  
CS 33007  
Homework 3

Delete from department where dept\_name = "History";



Select \* from instructor; // again observe records of instructor from history department



The changes made by this query are that there are no longer any records in the instructor table with an instructor from the history department.



Chris Grimes

CS 33007

Homework 3

8. Create a transaction table, insert few records, extract minutes from transaction time.

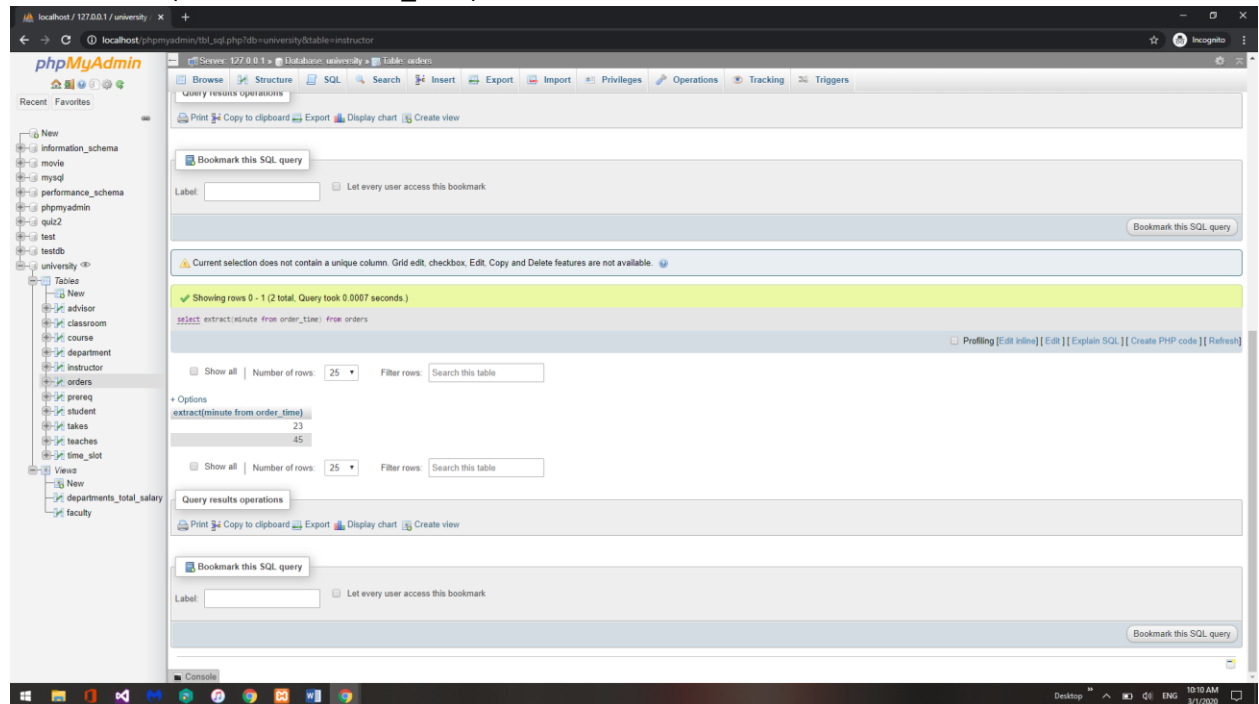
```
create table orders( order_id char(30), order_date date, order_time time, order_timestamp timestamp );
```

```
insert into orders values("345","2018-09-06","5:23:56", "2018-09-06 5:23:56.34")
```

```
insert into orders values("6778","2018-09-06","5:45:56", "2018-09-06 5:45:56.34")
```

```
select * from orders
```

```
select extract(minute from order_time) from orders
```



Chris Grimes  
CS 33007  
Homework 3

Part 2:

The top screenshot shows the phpMyAdmin interface at localhost/127.0.0.1. The 'User accounts overview' page is displayed, showing a table of users and their privileges. The table has columns: User name, Host name, Password, Global privileges, User group, Grant, and Action. The table lists several users, including 'Any', 'Matt', 'Sad', 'john', 'pma', 'root', and 'root' with various privileges and actions.

User name	Host name	Password	Global privileges	User group	Grant	Action
Any	%	No	USAGE	No		Edit privileges Export
Matt	localhost	Yes	USAGE	No		Edit privileges Export
Sad	localhost	Yes	USAGE	No		Edit privileges Export
john	localhost	Yes	SELECT, INSERT	Yes		Edit privileges Export
pma	localhost	No	USAGE	No		Edit privileges Export
root	127.0.0.1	No	ALL PRIVILEGES	Yes		Edit privileges Export
root	:::1	No	ALL PRIVILEGES	Yes		Edit privileges Export
root	localhost	No	ALL PRIVILEGES	Yes		Edit privileges Export

The bottom screenshot shows the phpMyAdmin interface at localhost/127.0.0.1. The 'Table: instructor' view is displayed, showing a list of instructor records. The table has columns: ID, name, dept\_name, and salary. The table lists 24 rows of instructor data, including names like Lembr, Bawa, Yazdi, Wisland, DAgostino, Liley, Kean, Atanassov, Green, Moreira, Gustafsson, Bourrier, Bondi, Solasalm-Soininen, Morris, Arias, Murata, Tung, Luo, Vicentino, Romero, Lent, Sarkar, Shuming, and Konstantinides.

ID	name	dept_name	salary
14365	Lembr	Accounting	32241.56
15347	Bawa	Athletics	72140.88
16807	Yazdi	Athletics	98333.65
19368	Wisland	Pol. Sci.	124651.41
22591	DAgostino	Psychology	59706.49
25945	Liley	Languages	96091.69
28097	Kean	English	35523.18
28400	Atanassov	Statistics	84582.92
30765	Green	Music	NULL
31955	Moreira	Accounting	71351.42
3199	Gustafsson	Elec. Eng.	82534.37
3335	Bourrier	Comp. Sci.	80797.83
34175	Bondi	Comp. Sci.	115469.11
35679	Solasalm-Soininen	Psychology	62579.61
36897	Morris	Marketing	43770.36
37687	Arias	Statistics	104563.38
4034	Murata	Athletics	61387.56
41930	Tung	Athletics	58482.03
4233	Luo	English	88791.45
42782	Vicentino	Elec. Eng.	34272.67
43779	Romero	Astronomy	79070.08
48507	Lent	Mech. Eng.	107978.47
48570	Sarkar	Pol. Sci.	87549.80
50330	Shuming	Physics	108011.61
50885	Konstantinides	Languages	32579.50

Chris Grimes  
CS 33007  
Homework 3

The image displays two screenshots of the phpMyAdmin web interface, showing database management operations.

**Top Screenshot:** The interface shows the 'university' database selected. The 'instructor' table is displayed, showing 51 rows. The table structure is as follows:

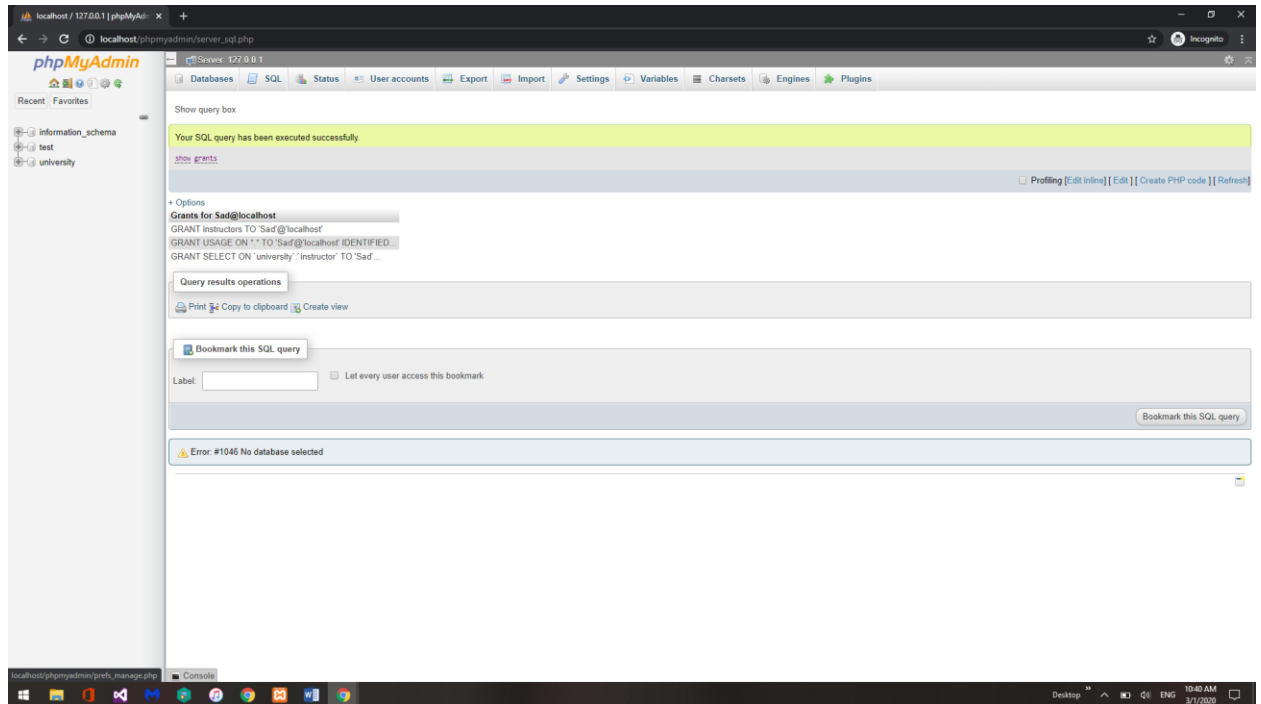
ID	name	dept_name	salary
14366	Lembr	Accounting	32241.56
15347	Bawa	Athletics	72140.88
16807	Yazdi	Athletics	98333.65
19368	Wieland	Pol. Sci.	124651.41
22591	DiAgostino	Psychology	59706.49
25945	Liley	Languages	90091.69
28097	Kean	English	35023.18
28400	Atanasov	Statistics	84582.92
30765	Green	Music	NULL
31955	Moreira	Accounting	71351.42
3199	Gustafsson	Elec. Eng.	82534.37
3335	Bourrier	Comp. Sci.	80797.83
34175	Bondi	Comp. Sci.	115469.11
35579	Solomon-Soininen	Psychology	62579.61
36897	Morris	Marketing	43770.36
37687	Arias	Statistics	104563.38
4034	Murata	Athletics	61387.56
41930	Tung	Athletics	50482.03
4233	Luo	English	88791.45
42782	Vicentino	Elec. Eng.	34272.67
43779	Romero	Astronomy	79070.08
48507	Lent	Mech. Eng.	107978.47
48570	Sarkar	Pol. Sci.	87549.80
50330	Shuming	Physics	108011.81
50885	Konstantinides	Languages	32578.50

**Bottom Screenshot:** The interface shows the 'university' database selected. The 'instructor' table is displayed. The SQL query results are shown, indicating that the query was executed successfully. The query results are as follows:

```
GRANTS for Matt@localhost
GRANT USAGE ON *.* TO Matt@localhost IDENTIFIED BY ''
GRANT SELECT ON `university`.`instructor` TO Matt
```

The bottom screenshot also shows the 'Query results operations' section, including options to print, copy to clipboard, and create view. There is also a 'Bookmark this SQL query' section with a label field and a checkbox to let every user access this bookmark. An error message is displayed at the bottom: 'Error: #1046 No database selected'.

Chris Grimes  
CS 33007  
Homework 3



Part 3:

1. Assume you are given two relations, student(name, rollno) and marks(rollno, exam, mark)  
Show names of all students who have got marks in at least two exams.

select name from student natural join marks where (select (rollno) from marks where mark is not null);

2. Create a view CSinstructors, showing all information about instructors from the Comp. Sci. department.

create view CSinstructors as select \* from instructor where dept\_name = 'Comp. Sci.';

3. Create a transaction query that the increase the salary of Biology instructors by 3% and Physics instructors by 5%.

START TRANSACTION;

UPDATE instructor SET salary = salary\* 1.03 where dept\_name = 'Biology';

UPDATE instructor SET salary = salary\* 1.05 where dept\_name = 'Physics';

COMMIT;

Chris Grimes

CS 33007

### Homework 3

4. Grant permission to one of your friends to view all data in your student relation. Also make sure that you are granting your friend to pass the permission to others.

GRANT SELECT on university.student TO [cgrimes91@gmail.com](mailto:cgrimes91@gmail.com) with GRANT option

5. (i) create a role CS\_staff

create role CS\_staff;

- (ii) Allow CS\_staff to only see/read data from the view CSInstructors created in exercise 2.

GRANT SELECT on university.CSInstructors to CS\_staff;

- (iii) Add user Matt (you created before in part 2) as CS\_staff.

GRANT CS\_staff to Matt@localhost;

- (iv) Login as Matt and show he can only execute select queries on CSInstructors

