

WEEK 2 ASSIGNMENT

Modern Application Development (Java Spring Boot)

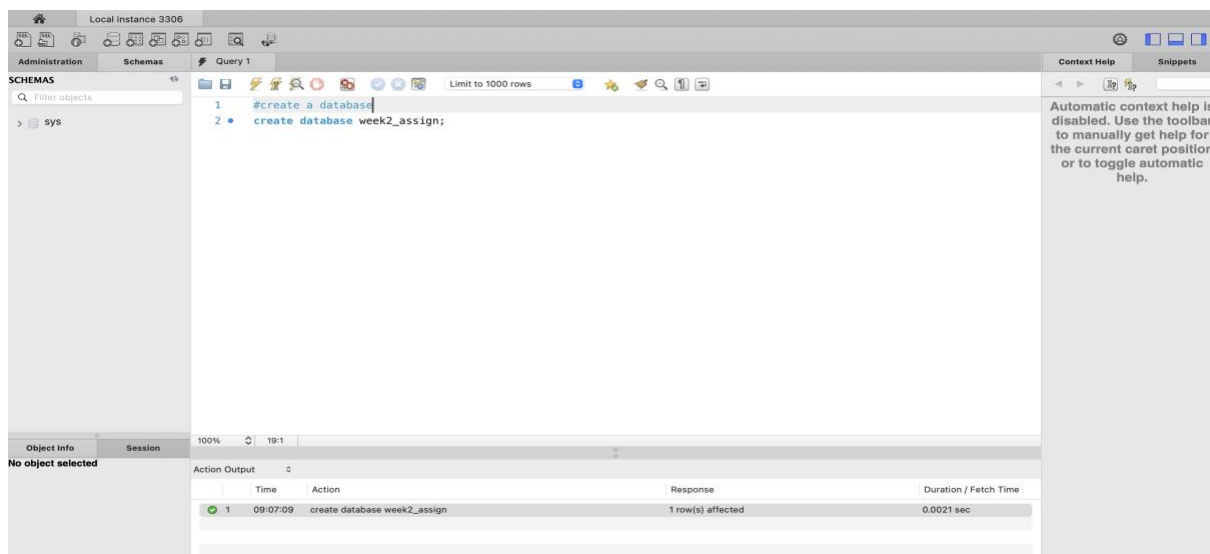
Name : Polagani Venkat Goud

Reg.No.:20BCE0084

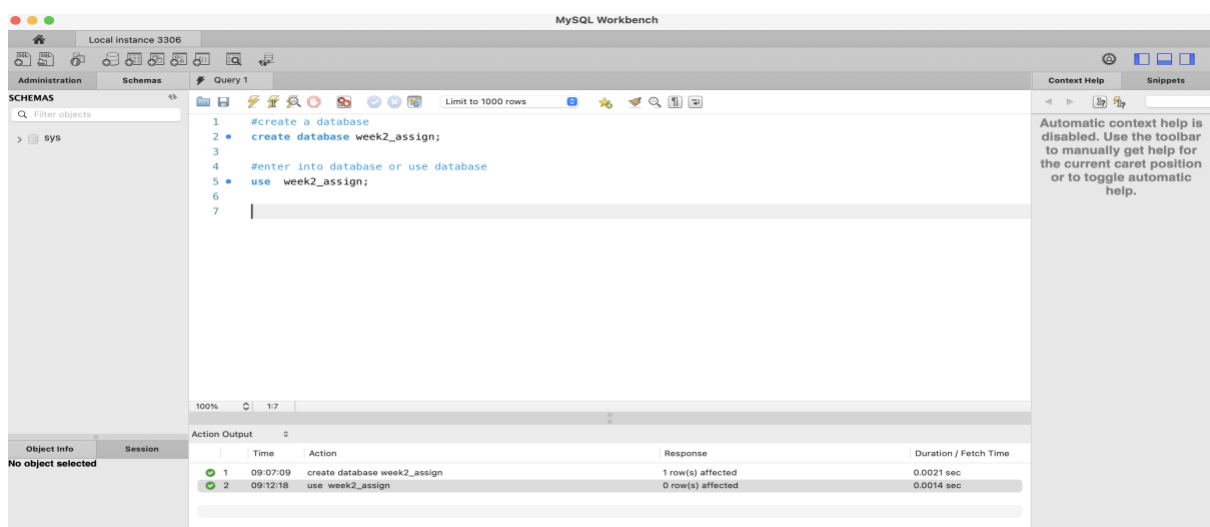
SQL:

CREATE UPDATE DELETE COMMANDS IN SQL

Create Database;



Use the database that is created



Example of numeric function

The screenshot shows a SQL script editor with the following code:

```
6
7 #numeric function
8
9 • select abs(-22);
```

Below the script editor, the 'Result Grid' is displayed, showing the output of the query:

abs(-22)
22

The interface includes a search bar, an 'Export' button, and a 'Filter Rows' section.

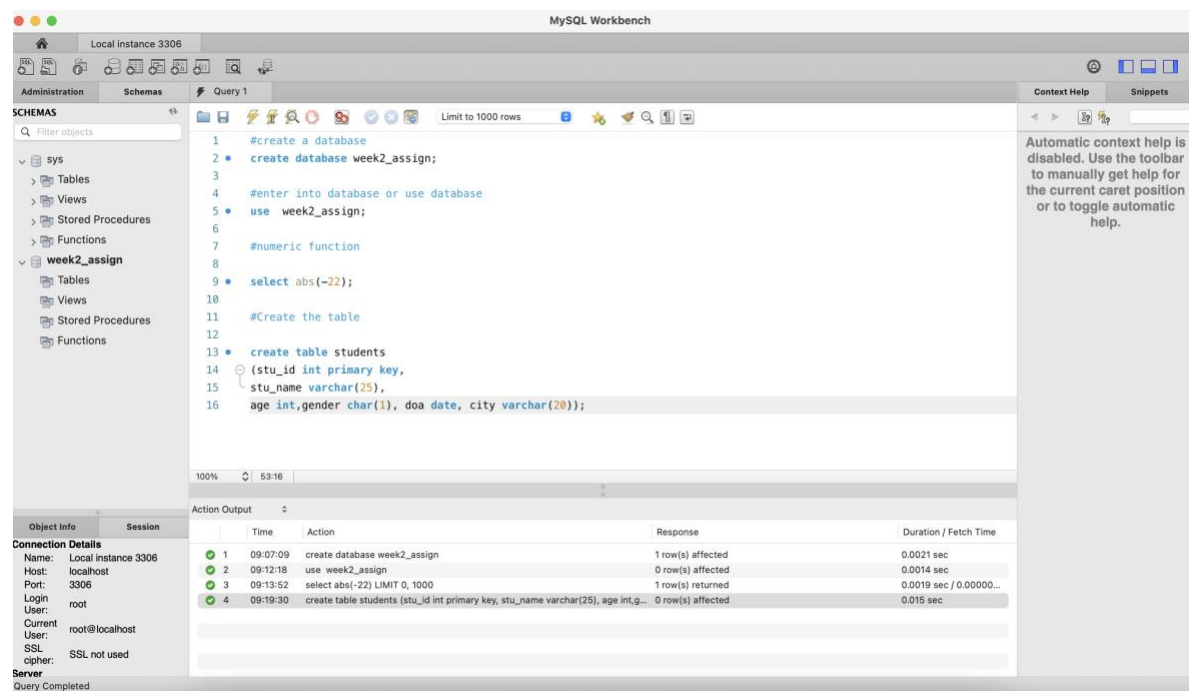
Find the created tables on the left side

The screenshot shows the 'SCHEMAS' pane on the left, which is expanded to show the 'week2_assign' database. The 'Tables' folder under 'week2_assign' is highlighted with a red circle. The main editor shows the following SQL script:

```
1 #create a database
2 • create database week2_assign;
3
4 #enter into database or use database
5 • use week2_assign;
6
7 #numeric function
8
9 • select abs(-22);
10
11 #Create the table
12
13 • create table students
14 (stu_id int primary key,
15  stu_name varchar(25),
16  age int,gender char(1), doa date, city varchar(20));
```

The interface includes a search bar, a 'Limit to 1000 rows' dropdown, and an 'Action Output' pane at the bottom.

Table creation:



The screenshot shows the MySQL Workbench interface. The 'SCHEMAS' panel on the left shows the 'week2_assign' database selected. The 'Query' editor in the center contains the following SQL code:

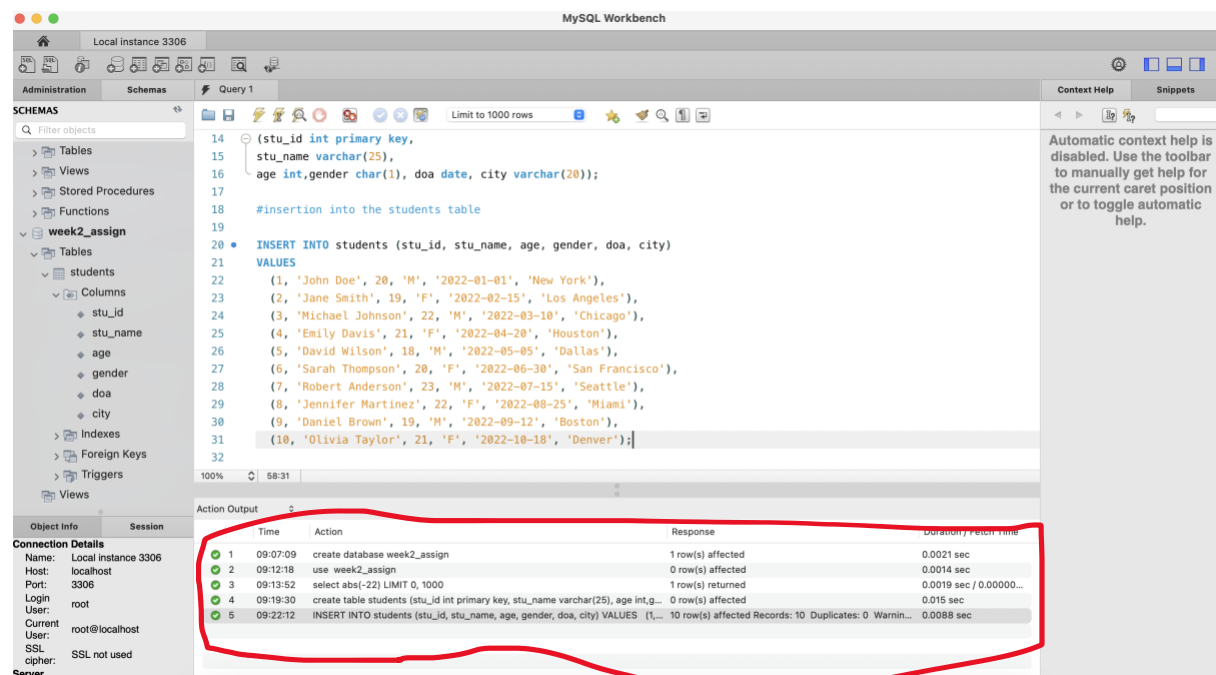
```
1 #create a database
2 • create database week2_assign;
3
4 #enter into database or use database
5 • use week2_assign;
6
7 #numeric function
8
9 • select abs(-22);
10
11 #Create the table
12
13 • create table students
14 (stu_id int primary key,
15  stu_name varchar(25),
16  age int,gender char(1), doa date, city varchar(20));
```

The 'Action Output' panel at the bottom shows the execution results:

	Time	Action	Response	Duration / Fetch Time
1	09:07:09	create database week2_assign	1 row(s) affected	0.0021 sec
2	09:12:18	use week2_assign	0 row(s) affected	0.0014 sec
3	09:13:52	select abs(-22) LIMIT 0, 1000	1 row(s) returned	0.0019 sec / 0.00000...
4	09:19:30	create table students (stu_id int primary key, stu_name varchar(25), age int,gender char(1), doa date, city varchar(20));	0 row(s) affected	0.015 sec

The 'Connection Details' panel on the left shows the connection to 'Local instance 3306' on 'localhost' port '3306' using the 'root' user.

Insert Into student's table



The screenshot shows the MySQL Workbench interface. The 'Query' editor in the center contains the following SQL code:

```
14 (stu_id int primary key,
15  stu_name varchar(25),
16  age int,gender char(1), doa date, city varchar(20));
17
18 #insertion into the students table
19
20 • INSERT INTO students (stu_id, stu_name, age, gender, doa, city)
21  VALUES
22  (1, 'John Doe', 20, 'M', '2022-01-01', 'New York'),
23  (2, 'Jane Smith', 19, 'F', '2022-02-15', 'Los Angeles'),
24  (3, 'Michael Johnson', 22, 'M', '2022-03-10', 'Chicago'),
25  (4, 'Emily Davis', 21, 'F', '2022-04-20', 'Houston'),
26  (5, 'David Wilson', 18, 'M', '2022-05-05', 'Dallas'),
27  (6, 'Sarah Thompson', 20, 'F', '2022-06-30', 'San Francisco'),
28  (7, 'Robert Anderson', 23, 'M', '2022-07-15', 'Seattle'),
29  (8, 'Jennifer Martinez', 22, 'F', '2022-08-25', 'Miami'),
30  (9, 'Daniel Brown', 19, 'M', '2022-09-12', 'Boston'),
31  (10, 'Olivia Taylor', 21, 'F', '2022-10-18', 'Denver');
```

The 'Action Output' panel at the bottom shows the execution results:

	Time	Action	Response	Duration / Fetch Time
1	09:07:09	create database week2_assign	1 row(s) affected	0.0021 sec
2	09:12:18	use week2_assign	0 row(s) affected	0.0014 sec
3	09:13:52	select abs(-22) LIMIT 0, 1000	1 row(s) returned	0.0019 sec / 0.00000...
4	09:19:30	create table students (stu_id int primary key, stu_name varchar(25), age int,gender char(1), doa date, city varchar(20));	0 row(s) affected	0.015 sec
5	09:22:12	INSERT INTO students (stu_id, stu_name, age, gender, doa, city) VALUES (1,... 10 row(s) affected Records: 10 Duplicates: 0 Warnin...	10 row(s) affected	0.0088 sec

The 'Connection Details' panel on the left shows the connection to 'Local instance 3306' on 'localhost' port '3306' using the 'root' user.

MySQL Workbench

Local instance 3306

SCHEMAS

Filter objects

- Stored Procedures
- Functions
- week2_assign
 - Tables
 - students
 - Columns
 - stu_id
 - stu_name
 - age
 - gender
 - doa
 - city
 - Indexes
 - Foreign Keys
 - Triggers
 - Views
 - Stored Procedures
 - Functions

Query 1

```

23 (2, 'Jane Smith', 19, 'F', '2022-02-15', 'Los Angeles'),
24 (3, 'Michael Johnson', 22, 'M', '2022-03-10', 'Chicago'),
25 (4, 'Emily Davis', 21, 'F', '2022-04-20', 'Houston'),
26 (5, 'David Wilson', 18, 'M', '2022-05-05', 'Dallas'),
27 (6, 'Sarah Thompson', 20, 'F', '2022-06-30', 'San Francisco'),

```

Result Grid

stu_id	stu_name	age	gender	doa	city
1	John Doe	20	M	2022-01-01	New York
2	Jane Smith	19	F	2022-02-15	Los Angeles
3	Michael Johnson	22	M	2022-03-10	Chicago
4	Emily Davis	21	F	2022-04-20	Houston
5	David Wilson	18	M	2022-05-05	Dallas
6	Sarah Thompson	20	F	2022-06-30	San Francisco
7	Robert Anderson	23	M	2022-07-15	Seattle
8	Jennifer Martinez	22	F	2022-08-25	Miami
9	Daniel Brown	19	M	2022-09-12	Boston
10	Olivia Taylor	21	F	2022-10-18	Denver

Action Output

Time	Action	Response	Duration / Fetch Time
09:07:09	create database week2_assign	1 row(s) affected	0.0021 sec
09:12:18	use week2_assign	0 row(s) affected	0.0014 sec
09:13:52	select abs(-22) LIMIT 0, 1000	1 row(s) returned	0.0019 sec / 0.00000...
09:19:30	create table students (stu_id int primary key, stu_name varchar(25), age int, g...	0 row(s) affected	0.015 sec
09:22:12	INSERT INTO students (stu_id, stu_name, age, gender, doa, city) VALUES (1,...	10 row(s) affected Records: 10 Duplicates: 0 Warnin...	0.0088 sec
09:24:53	select * from students LIMIT 0, 1000	10 row(s) returned	0.00059 sec / 0.000...

Connection Details

Name: Local instance 3306
Host: localhost
Port: 3306
Login: root
User: root
Current User: root@localhost
SSL: SSL not used
Cipher: cipher:

Server

Query Completed

Automatic context help is disabled. Use the toolbar to manually get help for the current caret position or to toggle automatic help.

UPDATION

MySQL Workbench

Local instance 3306

SCHEMAS

Filter objects

- Stored Procedures
- Functions
- week2_assign
 - Tables
 - students
 - Columns
 - stu_id
 - stu_name
 - age
 - gender
 - doa
 - city
 - Indexes
 - Foreign Keys
 - Triggers
 - Views
 - Stored Procedures
 - Functions

Query 1

```

32
33 #UPDATION
34
35 • update students set age = 22 where stu_id = 10;
36
37 • select * from students;
38

```

Result Grid

stu_id	stu_name	age	gender	doa	city
1	John Doe	20	M	2022-01-01	New York
2	Jane Smith	19	F	2022-02-15	Los Angeles
3	Michael Johnson	22	M	2022-03-10	Chicago
4	Emily Davis	21	F	2022-04-20	Houston
5	David Wilson	18	M	2022-05-05	Dallas
6	Sarah Thompson	20	F	2022-06-30	San Francisco
7	Robert Anderson	23	M	2022-07-15	Seattle
8	Jennifer Martinez	22	F	2022-08-25	Miami
9	Daniel Brown	22	M	2022-09-12	Boston
10	Olivia Taylor	22	F	2022-10-18	Denver

Action Output

Time	Action	Response	Duration / Fetch Time
09:07:09	create database week2_assign	1 row(s) affected	0.0021 sec
09:12:18	use week2_assign	0 row(s) affected	0.0014 sec
09:13:52	select abs(-22) LIMIT 0, 1000	1 row(s) returned	0.0019 sec / 0.00000...
09:19:30	create table students (stu_id int primary key, stu_name varchar(25), age int, g...	0 row(s) affected	0.015 sec
09:22:12	INSERT INTO students (stu_id, stu_name, age, gender, doa, city) VALUES (1,...	10 row(s) affected Records: 10 Duplicates: 0 Warnin...	0.0088 sec
09:24:53	select * from students LIMIT 0, 1000	10 row(s) returned	0.00059 sec / 0.000...
09:26:13	update students set age = 22 where stu_id = 10	1 row(s) affected Rows matched: 1 Changed: 1 Warni...	0.0028 sec
09:26:29	select * from students LIMIT 0, 1000	10 row(s) returned	0.00053 sec / 0.0000...

Connection Details

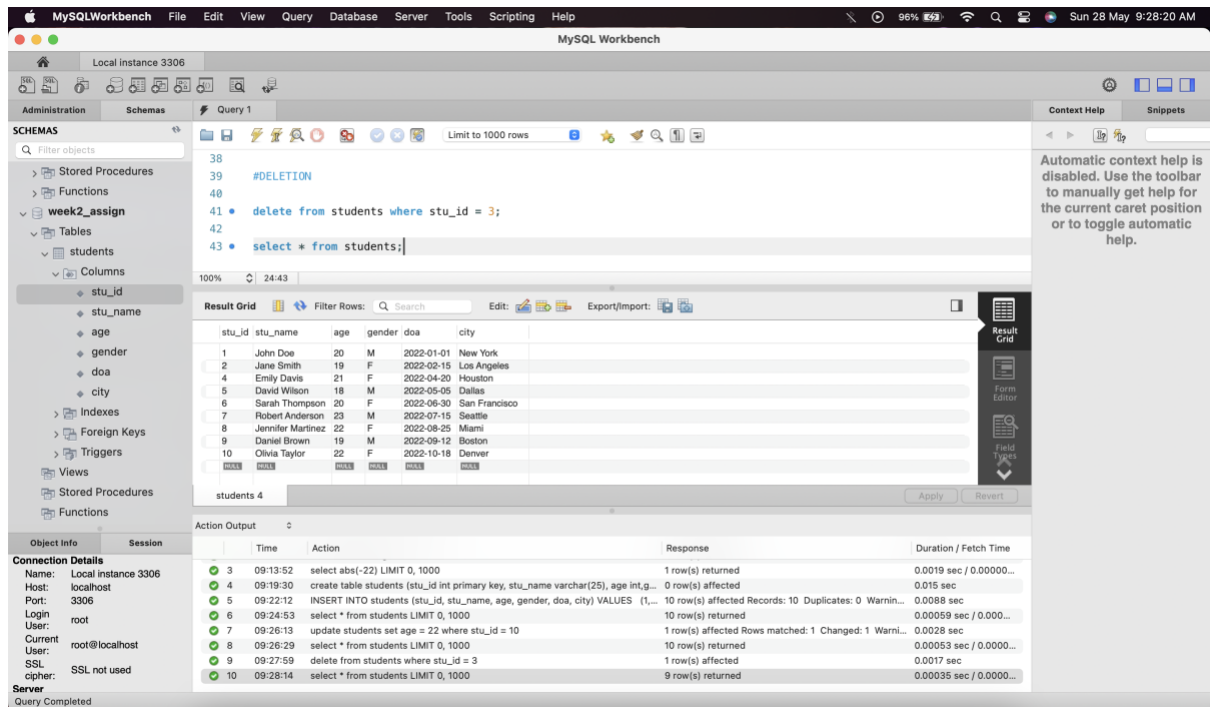
Name: Local instance 3306
Host: localhost
Port: 3306
Login: root
User: root
Current User: root@localhost
SSL: SSL not used
Cipher: cipher:

Server

Query Completed

Automatic context help is disabled. Use the toolbar to manually get help for the current caret position or to toggle automatic help.

DELETION - Deleted the stu_id 3 records success



PERFORMED QUERIES:

```
#create a database
create database week2_assign;
```

```
#enter into database or use database
use week2_assign;
```

```
#numeric function
```

```
select abs(-22);
```

```
#Create the table
```

```
create table students
(stu_id int primary key,
stu_name varchar(25),
age int,gender char(1), doa date, city varchar(20));
```

```
#insertion into the students table
```

```
INSERT INTO students (stu_id, stu_name, age, gender, doa, city)
VALUES
(1, 'John Doe', 20, 'M', '2022-01-01', 'New York'),
(2, 'Jane Smith', 19, 'F', '2022-02-15', 'Los Angeles'),
(3, 'Michael Johnson', 22, 'M', '2022-03-10', 'Chicago'),
(4, 'Emily Davis', 21, 'F', '2022-04-20', 'Houston'),
(5, 'David Wilson', 18, 'M', '2022-05-05', 'Dallas'),
(6, 'Sarah Thompson', 20, 'F', '2022-06-30', 'San Francisco'),
```

```
(7, 'Robert Anderson', 23, 'M', '2022-07-15', 'Seattle'),  
(8, 'Jennifer Martinez', 22, 'F', '2022-08-25', 'Miami'),  
(9, 'Daniel Brown', 19, 'M', '2022-09-12', 'Boston'),  
(10, 'Olivia Taylor', 21, 'F', '2022-10-18', 'Denver');
```

#UPDATION

```
update students set age = 22 where stu_id = 10;
```

```
select * from students;
```

#DELETION

```
delete from students where stu_id = 3;
```

```
select * from students;
```

CREATE TABLES AND PERFORM JOINS

Creation and insertion of table and values

The screenshot shows the MySQL Workbench interface. The left sidebar displays the 'SCHEMAS' tree with 'week2_assign' selected. The main editor shows a SQL query to create a table 'courses' and insert data into it. The 'Action Output' pane at the bottom shows the execution results.

```
CREATE TABLE courses (  
  course_id int PRIMARY KEY,  
  course_name varchar(50),  
  instructor varchar(50),  
  department varchar(50)  
);  
  
#insert into courses  
INSERT INTO courses (course_id, course_name, instructor, department)  
VALUES  
(1, 'Mathematics', 'Professor Smith', 'Mathematics Department'),  
(2, 'Computer Science', 'Professor Johnson', 'Computer Science Department'),  
(3, 'Physics', 'Professor Davis', 'Physics Department'),  
(4, 'History', 'Professor Wilson', 'History Department'),  
(5, 'Chemistry', 'Professor Thompson', 'Chemistry Department'),  
(6, 'English Literature', 'Professor Anderson', 'English Department'),  
(7, 'Biology', 'Professor Martinez', 'Biology Department');
```

Time	Action	Response	Duration / Fetch Time
09:27:59	delete from students where stu_id = 3	1 row(s) affected	0.0017 sec
09:28:14	select * from students LIMIT 0, 1000	9 row(s) returned	0.00035 sec / 0.0000...
09:33:10	INSERT INTO students (stu_id, stu_name, age, gender, doa, city) VALUES (3, ...	1 row(s) affected	0.0021 sec
09:33:19	select * from students LIMIT 0, 1000	10 row(s) returned	0.00063 sec / 0.000...
09:34:53	CREATE TABLE courses (course_id int PRIMARY KEY, course_name varcha...	0 row(s) affected	0.014 sec
09:36:12	INSERT INTO courses (course_id, course_name, instructor, department) VAL...	10 row(s) affected Records: 10 Duplicates: 0 Warnin...	0.0037 sec
09:36:25	select * from the courses LIMIT 0, 1000	Error Code: 1146. Table 'week2_assign.the' doesn't e...	0.0026 sec
09:37:28	select * from courses LIMIT 0, 1000	10 row(s) returned	0.00046 sec / 0.000...

The screenshot shows the MySQL Workbench interface with the 'courses' table selected in the 'SCHEMAS' tree. The 'Result Grid' pane displays the data inserted into the table. The 'Action Output' pane at the bottom shows the execution results.

course_id	course_name	instructor	department
1	Mathematics	Professor Smith	Mathematics Department
2	Computer Science	Professor Johnson	Computer Science Department
3	Physics	Professor Davis	Physics Department
4	History	Professor Wilson	History Department
5	Chemistry	Professor Thompson	Chemistry Department
6	English Literature	Professor Anderson	English Department
7	Biology	Professor Martinez	Biology Department
8	Economics	Professor Brown	Economics Department
9	Art History	Professor Taylor	Art History Department
10	Psychology	Professor White	Psychology Department

INNER JOIN

The screenshot shows the MySQL Workbench interface. The 'Schemas' pane on the left shows the 'week2_assign' database selected. The 'Query' editor in the center contains the following SQL query:

```
26
27 * SELECT students.stu_name, courses.course_name
28 FROM students
29 INNER JOIN courses ON students.stu_id = courses.course_id;
30
```

The 'Result Grid' shows the results of the query:

stu_name	course_name
John Doe	Mathematics
Jane Smith	Computer Science
John Smith	Physics
Emily Davis	History
David Wilson	Chemistry
Sarah Thompson	English Literature
Robert Anderson	Biology
Jennifer Martinez	Economics
Daniel Brown	Art History
Olivia Taylor	Psychology

The 'Action Output' pane at the bottom shows the execution log, including the query execution and the results returned.

LEFT JOIN

The screenshot shows the MySQL Workbench interface. The 'Schemas' pane on the left shows the 'week2_assign' database selected. The 'Query' editor in the center contains the following SQL query:

```
29
30 * SELECT students.stu_name, courses.course_name
31 FROM students
32 LEFT JOIN courses ON students.stu_id = courses.course_id;
33
```

The 'Result Grid' shows the results of the query:

stu_name	course_name
John Doe	Mathematics
Jane Smith	Computer Science
John Smith	Physics
Emily Davis	History
David Wilson	Chemistry
Sarah Thompson	English Literature
Robert Anderson	Biology
Jennifer Martinez	Economics
Daniel Brown	Art History
Olivia Taylor	Psychology

The 'Action Output' pane at the bottom shows the execution log, including the query execution and the results returned.

RIGHT JOIN

The screenshot shows the MySQL Workbench interface. The query editor contains the following SQL code:

```
34 #RIGHT JOIN
35 * SELECT students.stu_name, courses.course_name
36 FROM students
37 RIGHT JOIN courses ON students.stu_id = courses.course_id;
38
```

The result grid shows the following data:

stu_name	course_name
John Doe	Mathematics
Jane Smith	Computer Science
John Smith	Physics
Emily Davis	History
David Wilson	Chemistry
Sarah Thompson	English Literature
Robert Anderson	Biology
Jennifer Martinez	Economics
Daniel Brown	Art History
Olivia Taylor	Psychology

The action output shows the following log:

Time	Action	Response	Duration / Fetch Time
09:34:53	CREATE TABLE courses (course_id int PRIMARY KEY, course_name varchar...	0 row(s) affected	0.014 sec
09:36:12	INSERT INTO courses (course_id, course_name, instructor, department) VAL...	10 row(s) affected Records: 10 Duplicates: 0 Warnin...	0.0037 sec
09:36:25	select * from the courses LIMIT 0, 1000	Error Code: 1146. Table 'week2_assign.the' doesn't e...	0.0026 sec
09:37:28	select * from courses LIMIT 0, 1000	10 row(s) returned	0.00046 sec / 0.000...
09:39:21	SELECT students.stu_name, courses.course_name FROM students INNER JOI...	10 row(s) returned	0.00069 sec / 0.000...
09:40:43	INSERT INTO courses (course_id, course_name, instructor, department) VAL...	Error Code: 1062. Duplicate entry '1' for key 'courses...	0.00045 sec
09:41:32	SELECT students.stu_name, courses.course_name FROM students LEFT JOI...	10 row(s) returned	0.00061 sec / 0.000...
09:42:32	SELECT students.stu_name, courses.course_name FROM students RIGHT JOI...	10 row(s) returned	0.00054 sec / 0.000...

There are 5 records that doesn't match id so the value returned is null in student id.

The screenshot shows the MySQL Workbench interface. The query editor contains the following SQL code:

```
47 (15, 'Music Appreciation', 'Professor Anderson', 'Music Department');
48
49 #perform one join again
50
51 * SELECT students.stu_name, courses.course_name
```

The result grid shows the following data:

stu_name	course_name
John Doe	Mathematics
Jane Smith	Computer Science
John Smith	Physics
Emily Davis	History
David Wilson	Chemistry
Sarah Thompson	English Literature
Robert Anderson	Biology
Jennifer Martinez	Economics
Daniel Brown	Art History
Olivia Taylor	Psychology
Olivia Taylor	Computer Netwo...
Olivia Taylor	Statistics
Olivia Taylor	Sociology
Olivia Taylor	Marketing

The action output shows the following log:

Time	Action	Response	Duration / Fetch Time
09:36:25	select * from the courses LIMIT 0, 1000	Error Code: 1146. Table 'week2_assign.the' doesn't e...	0.0026 sec
09:37:28	select * from courses LIMIT 0, 1000	10 row(s) returned	0.00046 sec / 0.000...
09:39:21	SELECT students.stu_name, courses.course_name FROM students INNER JOI...	10 row(s) returned	0.00069 sec / 0.000...
09:40:43	INSERT INTO courses (course_id, course_name, instructor, department) VAL...	Error Code: 1062. Duplicate entry '1' for key 'courses...	0.00045 sec
09:41:32	SELECT students.stu_name, courses.course_name FROM students LEFT JOI...	10 row(s) returned	0.00061 sec / 0.000...
09:42:32	SELECT students.stu_name, courses.course_name FROM students RIGHT JOI...	10 row(s) returned	0.00054 sec / 0.000...
09:44:05	INSERT INTO courses (course_id, course_name, instructor, department) VAL...	5 row(s) affected Records: 5 Duplicates: 0 Warnings...	0.0020 sec
09:44:40	SELECT students.stu_name, courses.course_name FROM students RIGHT JOI...	15 row(s) returned	0.00057 sec / 0.000...

PERFORMED QUERIES:

#create another table

```
CREATE TABLE courses (  
    course_id int PRIMARY KEY,  
    course_name varchar(50),  
    instructor varchar(50),  
    department varchar(50)  
);
```

#insert into courses

```
INSERT INTO courses (course_id, course_name, instructor,  
department)  
VALUES  
    (1, 'Mathematics', 'Professor Smith', 'Mathematics  
Department'),  
    (2, 'Computer Science', 'Professor Johnson', 'Computer  
Science Department'),  
    (3, 'Physics', 'Professor Davis', 'Physics Department'),  
    (4, 'History', 'Professor Wilson', 'History  
Department'),  
    (5, 'Chemistry', 'Professor Thompson', 'Chemistry  
Department');
```

#INNER JOIN

```
SELECT students.stu_name, courses.course_name  
FROM students  
INNER JOIN courses ON students.stu_id = courses.course_id;
```

#LEFT JOIN

```
SELECT students.stu_name, courses.course_name  
FROM students  
LEFT JOIN courses ON students.stu_id = courses.course_id;
```

#RIGHT JOIN

```
SELECT students.stu_name, courses.course_name
FROM students
RIGHT JOIN courses ON students.stu_id = courses.course_id;
```

#inserted more records

```
INSERT INTO courses (course_id, course_name, instructor,
department)
VALUES
  (11, 'Computer Networks', 'Professor Thompson',
'Computer Science Department'),
  (12, 'Statistics', 'Professor Wilson', 'Mathematics
Department'),
  (13, 'Sociology', 'Professor Davis', 'Social Sciences
Department'),
  (14, 'Marketing', 'Professor Johnson', 'Business
Department'),
  (15, 'Music Appreciation', 'Professor Anderson', 'Music
Department');
```

#perform one join again

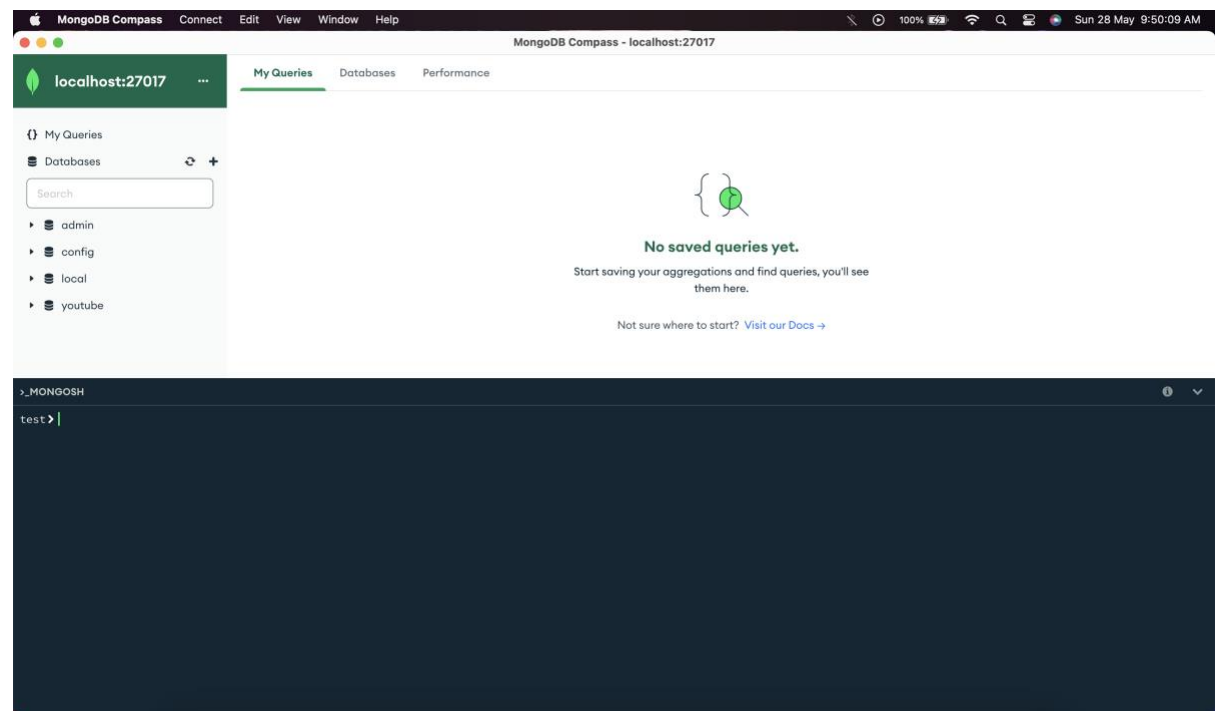
```
SELECT students.stu_name, courses.course_name
FROM students
RIGHT JOIN courses ON students.stu_id = courses.course_id;
```

CREATE UPDATE DELETE COMMANDS IN MONGODB

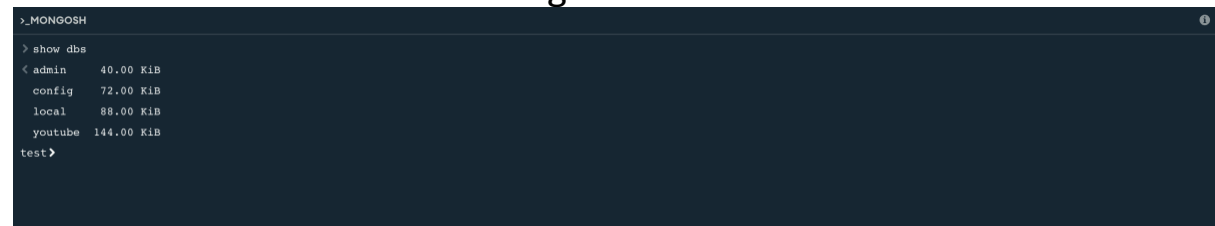
Start **mongoDB**

```
venkatgoud — -zsh — 80x24
Last login: Sat May 27 22:05:35 on ttys000
venkatgoud@Venkats-MacBook-Air ~ % brew services start mongodb-community@6.0
==> Successfully started `mongodb-community` (label: homebrew.mxcl.mongodb-commu
venkatgoud@Venkats-MacBook-Air ~ %
```

MongoDB compass



Available databases in mongoDB



Create a new database week2

```
> show dbs
< admin      40.00 KiB
  config     72.00 KiB
  local      88.00 KiB
  youtube    144.00 KiB
> use week2
< switched to db week2
> show dbs
< admin      40.00 KiB
  config     96.00 KiB
  local      88.00 KiB
  youtube    144.00 KiB
> db
< week2
week2>|
```

Create a collection student

The screenshot shows the MongoDB Compass web interface. At the top, there's a header with 'localhost:27017' and tabs for 'My Queries', 'Databases', and 'Performance'. Below the header, there's a sidebar with 'My Queries' and 'Databases'. The main area shows a terminal window with the following commands and output:

```
> .MONGOSH
> show dbs
< admin      40.00 KiB
  config     72.00 KiB
  local      88.00 KiB
  youtube    144.00 KiB
> use week2
< switched to db week2
> show dbs
< admin      40.00 KiB
  config     96.00 KiB
  local      88.00 KiB
  youtube    144.00 KiB
> db
< week2
> db.createCollection('student')
< { ok: 1 }
week2>
```

INSERTION

The screenshot shows the MongoDB CLI interface with the following commands and output:

```
> show dbs
admin      40.00 KiB
config     72.00 KiB
local      88.00 KiB
youtube    144.00 KiB

> use week2
switched to db week2

> show dbs
admin      40.00 KiB
config     96.00 KiB
local      88.00 KiB
youtube    144.00 KiB

> db
week2

> db.createCollection('student')
{ ok: 1 }

> db.student.insertOne({'name':'venkat','regno':1,'phno':'1234512341'})
{ acknowledged: true, insertedId: ObjectId("6472da571eb2c6af1ac50581") }

week2>
```

The second screenshot shows the continuation of the commands:

```
> db.student.insertOne({'name':'jhon','regno':2})
{ acknowledged: true, insertedId: ObjectId("6472da911eb2c6af1ac50582") }

week2>
```

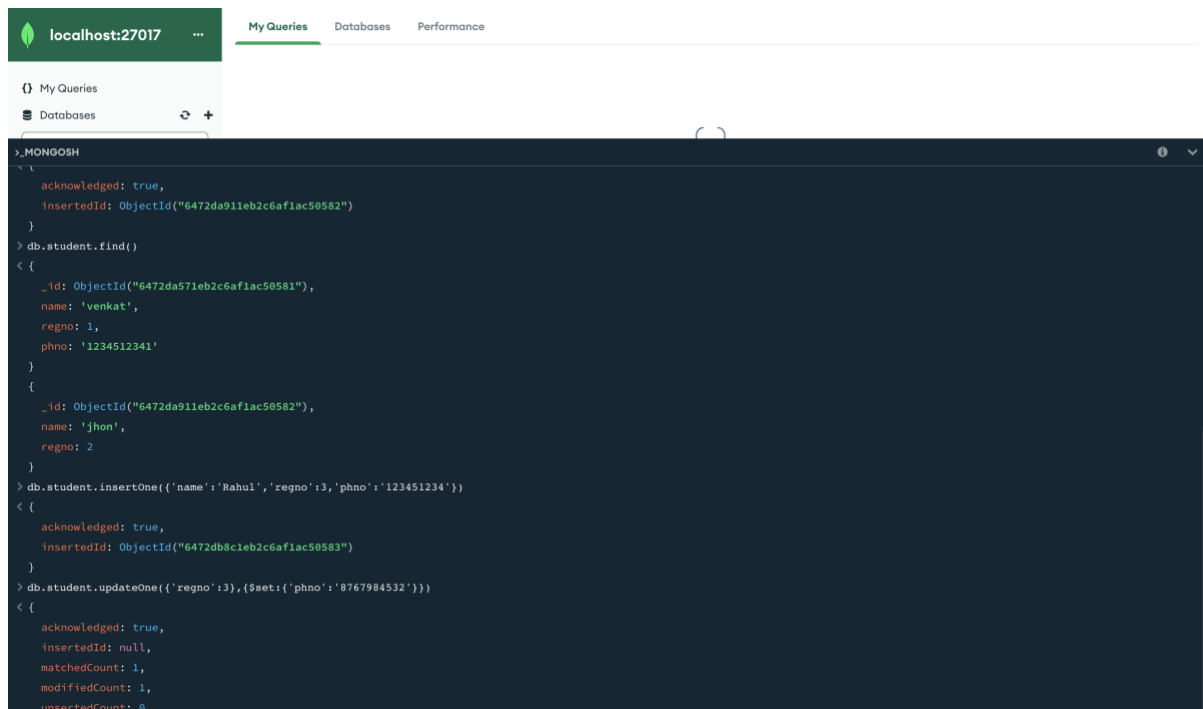
Find the inserted records

The screenshot shows the MongoDB CLI interface with the following command and output:

```
> db.student.find()
{
  _id: ObjectId("6472da571eb2c6af1ac50581"),
  name: 'venkat',
  regno: 1,
  phno: '1234512341'
}
{
  _id: ObjectId("6472da911eb2c6af1ac50582"),
  name: 'jhon',
  regno: 2
}
```

UPDATION

Updating the phone number of regno 3



The screenshot shows the MongoDB Shell interface with the following commands and results:

```
> use mydb
> db.student.insertOne({name:'Rahul', 'regno':3, 'phno':'123451234'})
{acknowledged: true, insertedId: ObjectId("6472da91eb2c6af1ac50582")}
> db.student.find()
< {
  _id: ObjectId("6472da571eb2c6af1ac50581"),
  name: 'venkat',
  regno: 1,
  phno: '1234512341'
}
{
  _id: ObjectId("6472da91eb2c6af1ac50582"),
  name: 'jhon',
  regno: 2
}
> db.student.insertOne({name:'Rahul', 'regno':3, 'phno':'123451234'})
{acknowledged: true, insertedId: ObjectId("6472db8c1eb2c6af1ac50583")}
> db.student.updateOne({'regno':3}, {$set: {'phno': '8767984532'}})
< {
  acknowledged: true,
  insertedId: null,
  matchedCount: 1,
  modifiedCount: 1,
  upsertedCount: 0
}
```

Result.



The screenshot shows the result of the update operation:

```
> db.student.updateOne({'regno':3}, {$set: {'phno': '8767984532'}})
< {
  acknowledged: true,
  insertedId: null,
  matchedCount: 1,
  modifiedCount: 1,
  upsertedCount: 0
}
```

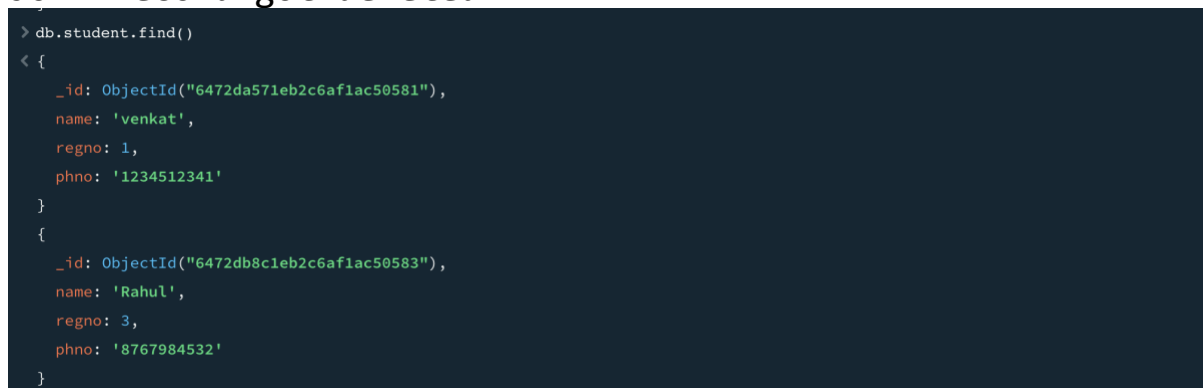
DELETION



The screenshot shows the result of a delete operation:

```
> db.student.deleteOne({'name': 'jhon'})
< {
  acknowledged: true,
  deletedCount: 1
}
```

John record got deleted



The screenshot shows the final state of the student collection after deletion:

```
> db.student.find()
< {
  _id: ObjectId("6472da571eb2c6af1ac50581"),
  name: 'venkat',
  regno: 1,
  phno: '1234512341'
}
{
  _id: ObjectId("6472db8c1eb2c6af1ac50583"),
  name: 'Rahul',
  regno: 3,
  phno: '8767984532'
}
```

Refresh the compass databases on the left pane you can find the database and records

The screenshot shows the MongoDB Compass interface. On the left, the 'Databases' pane lists 'student' as the selected database. The main area displays the 'week2.student' database with 2 documents and 1 index. The 'Documents' tab is active, showing two documents in a list view. Each document contains fields: '_id' (ObjectId), 'name', 'regno', and 'phno'.

Document 1	Document 2
<pre>{ "_id": ObjectId("6472da571eb2c6af1ac50581"), "name": "venkat", "regno": 1, "phno": "1234512341" }</pre>	<pre>{ "_id": ObjectId("6472db8c1eb2c6af1ac50583"), "name": "Rahul", "regno": 3, "phno": "8767984532" }</pre>