

## WEEK 2 ASSESSMENT

### Modern Application Development (Java Spring Boot)

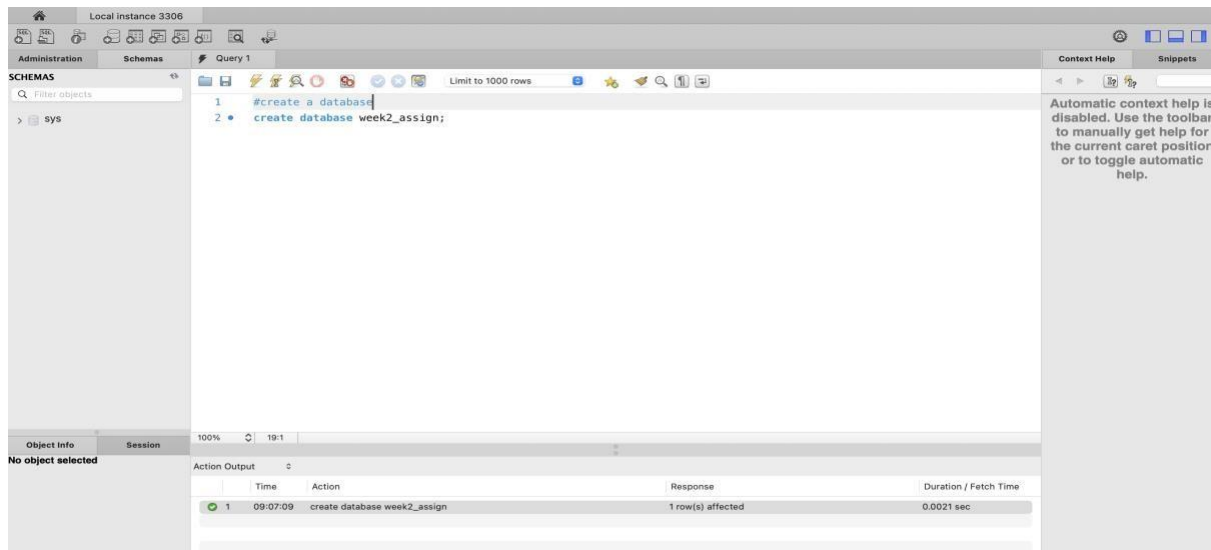
Name: LADE VIVEK

Reg.no: 20BCE0432

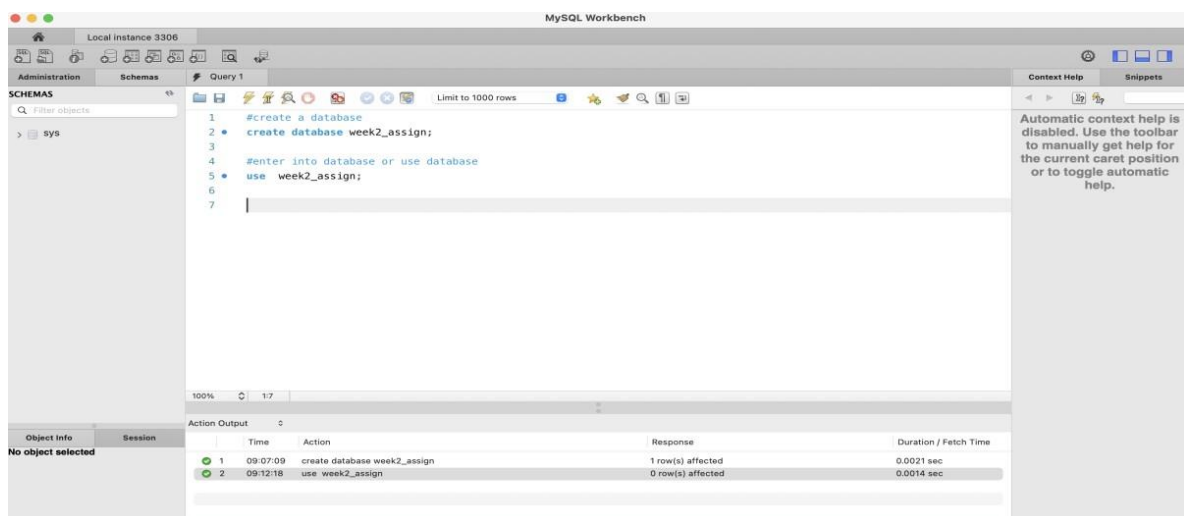
Mail: [lade.vivek2020@vitstudent.ac.in](mailto:lade.vivek2020@vitstudent.ac.in)

# SQL:

## Create Database;



## Use the database that is created



## Example of numeric function

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

100% 17:9

**Result Grid** Filter Rows: Search Export:

abs(-22)
22

## Table creation:

The screenshot shows the MySQL Workbench interface for 'Local Instance 3306'. The 'SCHEMAS' sidebar on the left shows the 'week2\_assign' database selected. The main editor contains 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 'Action Output' pane 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

Connection Details:

- Name: Local Instance 3306
- Host: localhost
- Port: 3306
- Login User: root
- Current User: root@localhost
- SSL: SSL not used
- Server: cphpr

Query Completed

## Insert Into student's table

The screenshot shows the MySQL Workbench interface for 'Local Instance 3306'. The 'SCHEMAS' sidebar on the left shows the 'week2\_assign' database selected, and the 'students' table is highlighted under the 'Tables' folder. The main editor contains the following SQL script:

```
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');
```

MySQL Workbench

Local instance 3306

Administration Schemas Query 1

Limit to 1000 rows

Context Help Snippets

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

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

Object Info Session

Connection Details

Name: Local instance 3306

Host: localhost

Port: 3306

Login: root

User: root

Current User: root@localhost

SSL: SSL not used

Server

Query Completed

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

students 2

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...

## UPDATION

MySQL Workbench

Local instance 3306

Administration Schemas Query 1

Limit to 1000 rows

Context Help Snippets

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

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

Object Info Session

Connection Details

Name: Local instance 3306

Host: localhost

Port: 3306

Login: root

User: root

Current User: root@localhost

SSL: SSL not used

Server

Query Completed

Result Grid

```

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

```

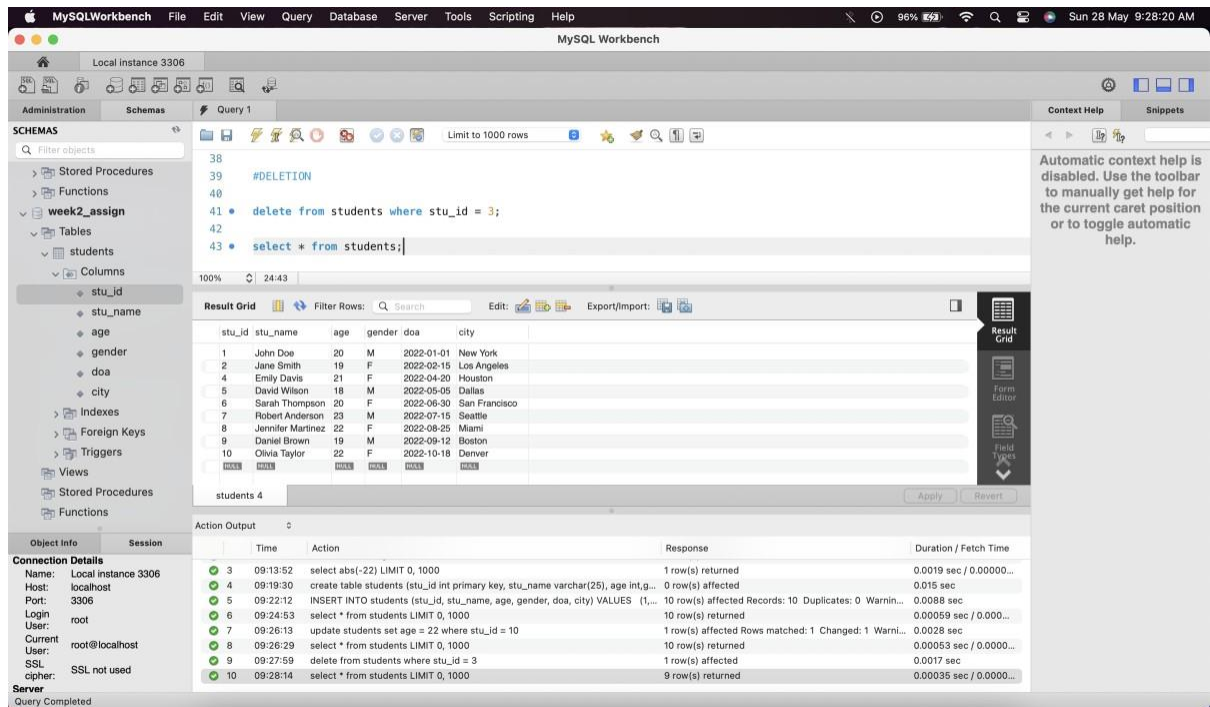
stu_id	stu_name	age	gender	doa	city
1	John Doe	20	M	2022-01-01	New York
2	Jane Smith	19	F	2022-02-15	Los Angeles
3	Michael Johnson	22	M	2022-03-10	Chicago
4	Emily Davis	21	F	2022-04-20	Houston
5	David Wilson	18	M	2022-05-05	Dallas
6	Sarah Thompson	20	F	2022-06-30	San Francisco
7	Robert Anderson	23	M	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

students 3

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 / 0.000...

DELETION – Deleted the stu\_id 3 records success



## PERFORMED QUERIES:

#create a database

create database week2\_assign;

#enter into database or use databaseuse  
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 'SCHEMAS' panel on the left shows the 'week2\_assign' database selected. The 'Query' editor in the center contains the following SQL code:

```
2
3 CREATE TABLE courses (
4   course_id int PRIMARY KEY,
5   course_name varchar(50),
6   instructor varchar(50),
7   department varchar(50)
8 );
9
10
11 #insert into courses
12 INSERT INTO courses (course_id, course_name, instructor, department)
13 VALUES
14 (1, 'Mathematics', 'Professor Smith', 'Mathematics Department'),
15 (2, 'Computer Science', 'Professor Johnson', 'Computer Science Department'),
16 (3, 'Physics', 'Professor Davis', 'Physics Department'),
17 (4, 'History', 'Professor Wilson', 'History Department'),
18 (5, 'Chemistry', 'Professor Thompson', 'Chemistry Department'),
19 (6, 'English Literature', 'Professor Anderson', 'English Department'),
20 (7, 'Biology', 'Professor Martinez', 'Biology Department');
```

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

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 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 'Result Grid' panel at the bottom. The 'Result Grid' displays the data inserted into the 'courses' table:

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

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

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 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...



# INNER JOIN

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

```
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 displays 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 pane shows the following log entries:

Time	Action	Response	Duration / Fetch Time
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 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...

# LEFT JOIN

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

```
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 displays 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 pane shows the following log entries:

Time	Action	Response	Duration / Fetch Time
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 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 / 0.0000...

# RIGHT JOIN

MySQL Workbench

Local instance 3306

AdministrationSchemasQuery 1Context HelpSnippets

SCHEMAS

Filter objects

sys

Tables

Views

Stored Procedures

Functions

week2\_assign

Tables

courses

students

Views

Stored Procedures

Functions

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

100%

59:37

Result Grid

Filter Rows: SearchExport

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

Result 9

Read Only

Object InfoSession

Connection Details

Name: Local instance 3306

Host: localhost

Port: 3306

Login User: root

Current User: root@localhost

SSL cipher: SSL not used

Server

Query Completed

	Time	Action	Response	Duration / Fetch Time
13	09:34:53	CREATE TABLE courses ( course_id int PRIMARY KEY, course_name varcha...	0 row(s) affected	0.014 sec
14	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
15	09:36:25	select * from the courses LIMIT 0, 1000	Error Code: 1146. Table 'week2_assign.the' doesn't e...	0.0026 sec
16	09:37:28	select * from courses LIMIT 0, 1000	10 row(s) returned	0.00046 sec / 0.000...
17	09:39:21	SELECT students.stu_name, courses.course_name FROM students INNER JOI...	10 row(s) returned	0.00069 sec / 0.000...
18	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
19	09:41:32	SELECT students.stu_name, courses.course_name FROM students LEFT JOI...	10 row(s) returned	0.00061 sec / 0.0000...
20	09:42:32	SELECT students.stu_name, courses.course_name FROM students RIGHT JOI...	10 row(s) returned	0.00054 sec / 0.000...

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

## 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', 'MathematicsDepartment'),
    (2, 'Computer Science', 'Professor Johnson', 'ComputerScience Department'),
    (3, 'Physics', 'Professor Davis', 'Physics Department'), (4, 'History', 'Professor
    Wilson', 'History
    Department'),
    (5, 'Chemistry', 'Professor Thompson', 'ChemistryDepartment');
```

#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_nameFROM
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', 'MathematicsDepartment'),
  (13, 'Sociology', 'Professor Davis', 'Social SciencesDepartment'),
  (14, 'Marketing', 'Professor Johnson', 'BusinessDepartment'),
  (15, 'Music Appreciation', 'Professor Anderson', 'MusicDepartment');
```

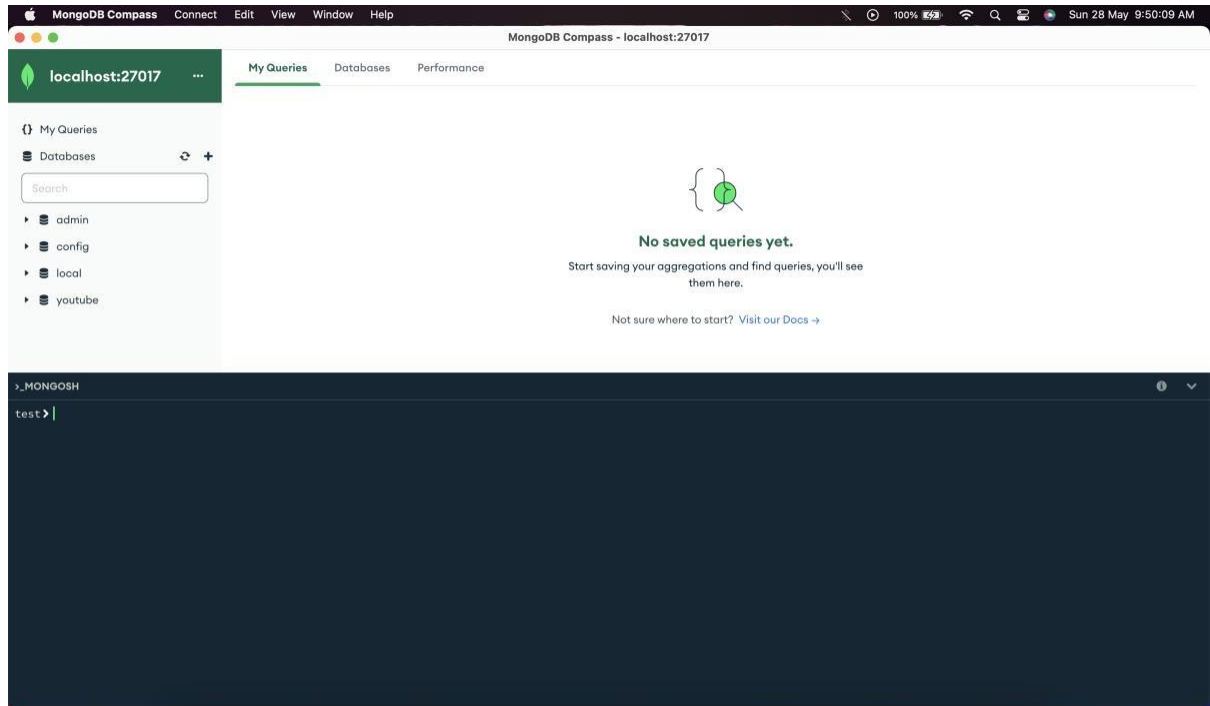
#perform one join again

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

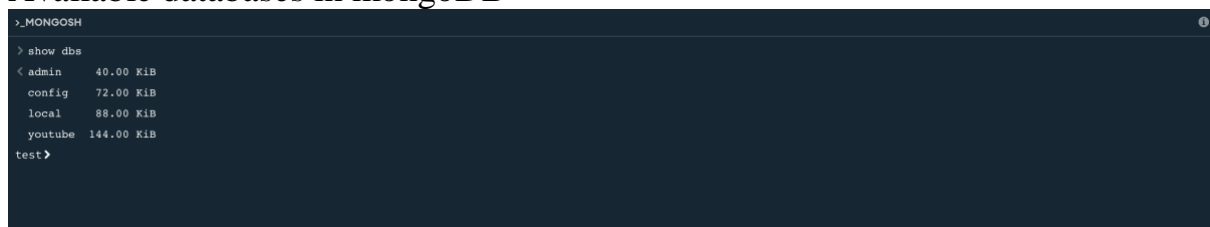
# CREATE UPDATE DELETE COMMANDS IN MONGODB

Start mongoDB

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 the 'week2' database selected. In the terminal window, the following commands are executed:

```
> 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 command-line interface (CLI) running on localhost:27017. The interface has a top bar with 'localhost:27017' and a menu icon, and tabs for 'My Queries', 'Databases', and 'Performance'. Below the top bar, there are sections for 'My Queries' and 'Databases'. The main area is 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 }
> db.student.insertOne({'name':'venkat','regno':1,'phno':'1234512341'})
< {
  acknowledged: true,
  insertedId: ObjectId("6472da571eb2c6af1ac50581")
}
week2>
```

The second screenshot shows the same CLI with additional 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 }
> db.student.insertOne({'name':'venkat','regno':1,'phno':'1234512341'})
< {
  acknowledged: true,
  insertedId: ObjectId("6472da571eb2c6af1ac50581")
}
> db.student.insertOne({'name':'jhon','regno':2})
< {
  acknowledged: true,
  insertedId: ObjectId("6472da911eb2c6af1ac50582")
}
week2>
```

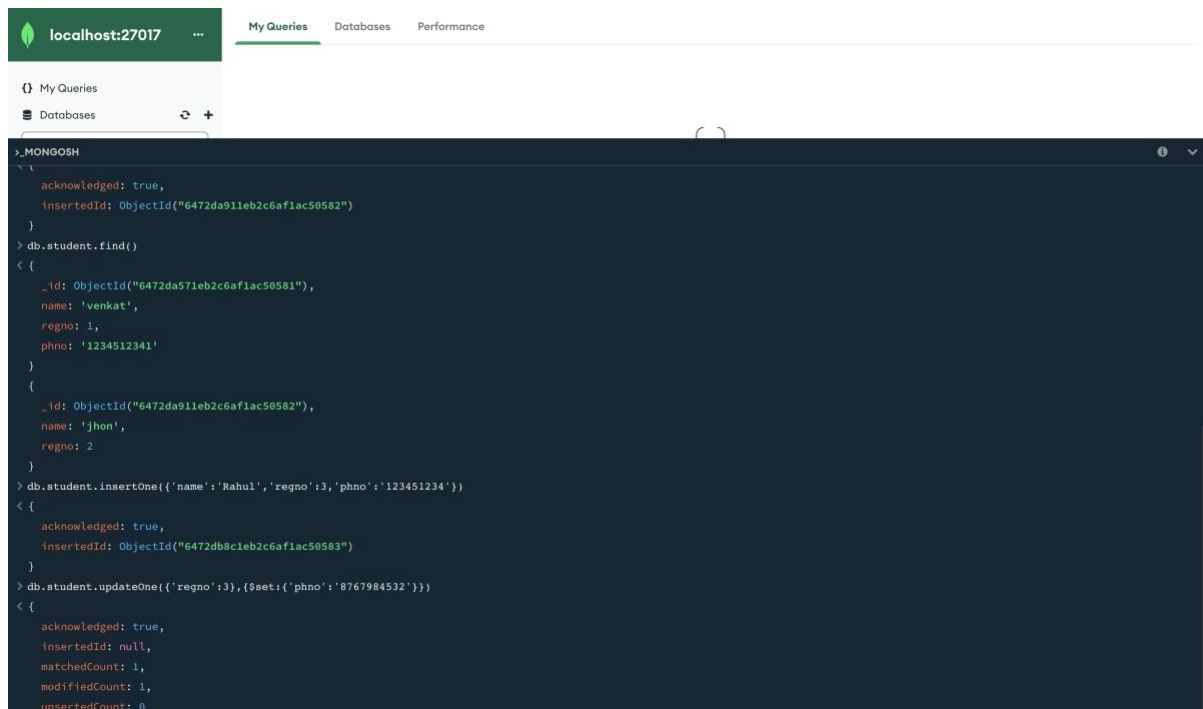
Find the inserted records

The screenshot shows the MongoDB command-line interface (CLI) running on localhost:27017. The interface has a top bar with 'localhost:27017' and a menu icon, and tabs for 'My Queries', 'Databases', and 'Performance'. Below the top bar, there are sections for 'My Queries' and 'Databases'. The main area is a terminal window with the following commands and output:

```
> MONGOSH
> 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 Compass interface. The top bar indicates the connection to 'localhost:27017'. Below the top bar, there are tabs for 'My Queries', 'Databases', and 'Performance'. The 'My Queries' tab is active, showing a list of queries. Below the list, there is a 'Databases' section with a '+' icon. The main area displays the MongoDB shell output for a series of commands. The commands include finding all documents in the 'student' collection, inserting a new document for 'Rahul', and updating the phone number of the document with 'regno: 3' to '8767984532'. The output shows the updated document and the result of the update operation, which is a success.

```
> _MONGOSH
> use student
> db.student.find()
< {
  _id: ObjectId("6472da571eb2c6af1ac50581"),
  name: 'venkat',
  regno: 1,
  phno: '1234512341'
}
{
  _id: ObjectId("6472da911eb2c6af1ac50582"),
  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 MongoDB shell output for the update operation. The command is 'db.student.updateOne({'regno': 3}, {\$set: {'phno': '8767984532'}})'. The output is a JSON object indicating the update was successful, with 'acknowledged: true', 'insertedId: null', 'matchedCount: 1', 'modifiedCount: 1', and 'upsertedCount: 0'.

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

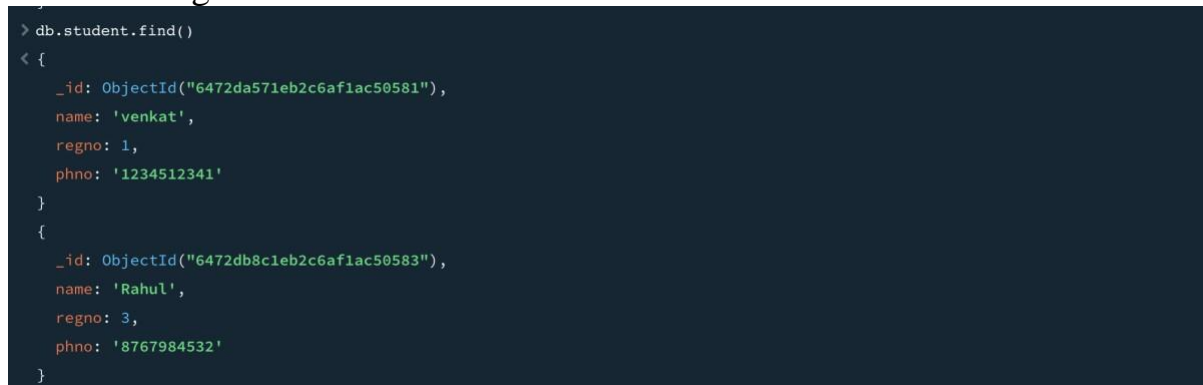
## DELETION



The screenshot shows the MongoDB shell output for a delete operation. The command is 'db.student.deleteOne({'name': 'jhon'})'. The output is a JSON object indicating the deletion was successful, with 'acknowledged: true' and 'deletedCount: 1'.

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

John record got deleted



The screenshot shows the MongoDB shell output for a find operation. The command is 'db.student.find()'. The output is a JSON array containing two documents. The first document is for 'venkat' with 'regno: 1' and 'phno: '1234512341''. The second document is for 'Rahul' with 'regno: 3' and 'phno: '8767984532''. The document for 'jhon' has been successfully deleted.

```
> 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 web interface. On the left sidebar, under 'Databases', the 'week2' database is expanded, and the 'student' collection is selected. The main panel displays the 'week2.student' collection with 2 documents and 1 index. The 'Documents' tab is active, showing a list of two documents. Each document is a JSON object with fields: '\_id', 'name', 'regno', and 'phno'.

Document	_id	name	regno	phno
1	ObjectId('6472da571eb2c6af1ac50581')	"venkat"	1	"1234512341"
2	ObjectId('6472db8c1eb2c6af1ac50583')	"Rahul"	3	"8767984532"