

MODERN APPLICATION DEVELOPMENT

JAVA SPRING BOOT

WEEKLY ASSIGNMET:2

NAME:ASHWIN SIVASANKAR V

REGNO:20BCE1804

QUESTION:

```
create ,update,delete commands in mysql  
create tables and perform joins in mysql  
create ,update,delete commands in mongo
```

MYSQL:

CREATE COMMANDS

The screenshot displays a MySQL IDE interface. The main editor window shows a SQL query to create a table named 'college' in the 'college' database. The query is as follows:

```
1 CREATE TABLE college.college (  
2     college_id INT PRIMARY KEY,  
3     college_name VARCHAR(50),  
4     location VARCHAR(50)  
5 );  
6
```

Below the editor, the 'Output' tab is selected, showing a log of database actions. The log includes the following entries:

#	Time	Action	Message	Duration / Fetch
33	12:10:35	SELECT employee.id, employee.name, employee.age, employee.department, comp...	3 row(s) returned	0.000 sec / 0.000 sec
34	12:17:56	Apply changes to companies	Changes applied	
35	12:18:15	CREATE TABLE companies.company (company_id INT PRIMARY KEY, com...	0 row(s) affected	0.015 sec
36	12:19:49	DROP DATABASE 'companies'	1 row(s) affected	0.031 sec
37	12:20:00	Apply changes to college	Changes applied	
38	12:20:15	CREATE TABLE college.college (college_id INT PRIMARY KEY, college_na...	0 row(s) affected	0.016 sec

INSERT COMMANDS

The screenshot displays the SQL Developer environment. The main window shows a query editor with the following SQL code:

```
1 • INSERT INTO college (college_id, college_name, location)
2 VALUES
3     (1, 'VIT', 'Vellore'),
4     (2, 'VIT', 'Chennai'),
5     (3, 'VIT', 'Bhopal');
6
```

The toolbar at the top includes icons for file operations, editing, and execution. A dropdown menu indicates "Limit to 1000 rows". On the right side, there is a sidebar titled "SQLAdditions" containing a message about automatic context help being disabled.

Below the query editor, there is a section labeled "Output" with tabs for "Context Help" and "Snippets". The "Action Output" tab is active, displaying a log of database actions:

#	Time	Action	Message	Duration / Fetch
✓ 33	12:10:35	SELECT employee.id, employee.name, employee.age, employee.department, comp...	3 row(s) returned	0.000 sec / 0.000 sec
✓ 34	12:17:56	Apply changes to companies	Changes applied	
✓ 35	12:18:15	CREATE TABLE companies.company (company_id INT PRIMARY KEY, com...	0 row(s) affected	0.015 sec
✓ 36	12:19:49	DROP DATABASE 'companies'	1 row(s) affected	0.031 sec
✓ 37	12:20:00	Apply changes to college	Changes applied	
✓ 38	12:20:15	CREATE TABLE college.college (college_id INT PRIMARY KEY, college_na...	0 row(s) affected	0.016 sec

Query 1 x

Limit to 1000 rows

```

1 • USE college;
2 • Select * from college;

```

Result Grid

college_id	college_name	location
1	VIT	Vellore
2	VIT	Madras
3	VIT	Bhopal
NULL	NULL	NULL

Result Grid

UPDATE COMMANDS:

Local instance MySQL80 x

File Edit View Query Database Server Tools Scripting Help

Navigator

SCHEMAS

- college
 - Tables
 - Views
 - Stored Procedures
 - Functions
- employees
 - Tables
 - Views
 - Stored Procedures
 - Functions
- sakila
- sys
- world

Query 1 x

Limit to 1000 rows

```

1 • UPDATE college
2 • SET college_name = 'VIT', location = 'Chennai'
3 • WHERE college_id = 2;

```

Result Grid

college_id	college_name	location
1	VIT	Vellore
2	VIT	Chennai
3	VIT	Bhopal
NULL	NULL	NULL

SQLAdditions

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

Administration Schemas

Information

No object selected

Object Info Session

college 11 x

Apply Revert Context Help Snippets

Output

#	Time	Action	Message	Duration / Fetch
44	12:28:31	INSERT INTO college (college_id, college_name, location) VALUES (1, 'VIT', V...	3 row(s) affected Records: 3 Duplicates: 0 Warnings: 0	0.016 sec
45	12:28:57	USE college	0 row(s) affected	0.000 sec
46	12:28:57	Select * from college LIMIT 0, 1000	3 row(s) returned	0.000 sec / 0.000 sec
47	12:30:06	UPDATE college SET college_name = 'VIT', location = 'Chennai' WHERE college_id = 2;	1 row(s) affected Rows matched: 1 Changed: 1 Warnings: 0	0.015 sec
48	12:31:17	USE college	0 row(s) affected	0.000 sec
49	12:31:17	Select * from college LIMIT 0, 1000	3 row(s) returned	0.000 sec / 0.000 sec

Query 1 x

Limit to 1000 rows

```

1 • USE college;
2 • Select * from college;

```

Result Grid

college_id	college_name	location
1	VIT	Vellore
2	VIT	Chennai
3	VIT	Bhopal
NULL	NULL	NULL

Result Grid

JOIN COMMANDS:

Query 1

Limit to 1000 rows

```

1 CREATE TABLE college.students (
2     college_id INT PRIMARY KEY,
3     student_name VARCHAR(50)
4 );

```

SQLAdditions

Jump to

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

Context Help Snippets

Output

Action Output

#	Time	Action	Message	Duration / Fetch
49	12:31:17	Select * from college LIMIT 0, 1000	3 row(s) returned	0.000 sec / 0.000 sec
50	12:34:06	CREATE TABLE student (college_id INT PRIMARY KEY, student_name VAR...	0 row(s) affected	0.015 sec
51	12:35:38	INSERT INTO student (college_id, student_name) VALUES (1, 'Will'), (2, 'Ash...	3 row(s) affected Records: 3 Duplicates: 0 Warnings: 0	0.000 sec
52	12:37:27	USE student	Error Code: 1049. Unknown database 'student'	0.000 sec
53	12:37:51	CREATE TABLE college.student (college_id INT PRIMARY KEY, student_na...	Error Code: 1050. Table 'student' already exists	0.000 sec
54	12:38:24	CREATE TABLE college.students (college_id INT PRIMARY KEY, student_n...	0 row(s) affected	0.032 sec

Query 1

Limit to 1000 rows

```

1 INSERT INTO students (college_id, student_name)
2 VALUES
3     (1, 'John'),
4     (2, 'Ashwin'),
5     (3, 'Will');

```

SQLAdditions

Jump to

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

Context Help Snippets

Output

Action Output

#	Time	Action	Message	Duration / Fetch
50	12:34:06	CREATE TABLE student (college_id INT PRIMARY KEY, student_name VAR...	0 row(s) affected	0.015 sec
51	12:35:38	INSERT INTO student (college_id, student_name) VALUES (1, 'Will'), (2, 'Ash...	3 row(s) affected Records: 3 Duplicates: 0 Warnings: 0	0.000 sec
52	12:37:27	USE student	Error Code: 1049. Unknown database 'student'	0.000 sec
53	12:37:51	CREATE TABLE college.student (college_id INT PRIMARY KEY, student_na...	Error Code: 1050. Table 'student' already exists	0.000 sec
54	12:38:24	CREATE TABLE college.students (college_id INT PRIMARY KEY, student_n...	0 row(s) affected	0.032 sec
55	12:40:33	INSERT INTO students (college_id, student_name) VALUES (1, 'John'), (2, 'As...	3 row(s) affected Records: 3 Duplicates: 0 Warnings: 0	0.016 sec

Query 1

Limit to 1000 rows

1 • USE college;
2 • Select * from students;

SQL Additions:

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

Result Grid

	college_id	student_name
1		John
2		Ashwin
3		Will

students 12

Output

Action Output

#	Time	Action	Message	Duration / Fetch
53	12:37:51	CREATE TABLE college.student (college_id INT PRIMARY KEY, student_na...	Error Code: 1050. Table 'student' already exists	0.000 sec
54	12:38:24	CREATE TABLE college.students (college_id INT PRIMARY KEY, student_n...	0 row(s) affected	0.032 sec
55	12:40:33	INSERT INTO students (college_id, student_name) VALUES (1, 'John'), (2, 'As...	3 row(s) affected Records: 3 Duplicates: 0 Warnings: 0	0.016 sec
56	12:41:19	USE students	Error Code: 1049. Unknown database 'students'	0.000 sec
57	12:41:50	USE college	0 row(s) affected	0.000 sec
58	12:41:50	Select * from students LIMIT 0, 1000	3 row(s) returned	0.000 sec / 0.000 sec

Query 1

Limit to 1000 rows

1 • SELECT student.student_name, college.college_name, college.location
2 FROM student
3 JOIN college ON student.college_id = college.college_id;

SQL Additions:

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

Result Grid

	student_name	college_name	location
1	Will	VIT	Vellore
2	Ashwin	VIT	Chennai
3	John	VIT	Bhopal

Result 13

Output

Action Output

DELETE COMMANDS:

Local instance MySQL80 x

File Edit View Query Database Server Tools Scripting Help

Navigator: SCHEMAS

Filter objects

- Views
- Stored Procedures
- Functions
- employees
 - Tables
 - company
 - Columns
 - Indexes
 - Foreign Keys
 - Triggers
 - employee
- Views
- Stored Procedures
- Functions

Administration Schemas

Information: Table: employee

Columns:

- id int PK
- name varchar(50)
- age int
- department varchar(50)

Object Info Session

Query 1 x

Limit to 1000 rows

SQLAdditions: Jump to

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

1 • DELETE FROM employee WHERE id = 1;

2

Result Grid

	id	name	age	department
2	William	20	IT	
3	Sony	40	IT	
4	Sam	50	IT	

employee 1 x

Apply Revert Context Help Snippets

Output

Action Output

#	Time	Action	Message	Duration / Fetch
5	16:44:05	DELETE FROM employee WHERE id = 1	Error Code: 1046. No database selected Select th...	0.016 sec
6	16:44:43	DELETE FROM employee WHERE id = 1	Error Code: 1046. No database selected Select th...	0.000 sec
7	16:45:53	DELETE FROM employee WHERE id = 1	Error Code: 1046. No database selected Select th...	0.000 sec
8	16:47:16	DELETE FROM employee WHERE id = 1	1 row(s) affected	0.031 sec
9	16:47:50	USE employees	0 row(s) affected	0.000 sec
10	16:47:50	Select * from employee LIMIT 0, 1000	3 row(s) returned	0.000 sec / 0.000 sec

Local instance MySQL80 x

File Edit View Query Database Server Tools Scripting Help

Navigator: SCHEMAS

Filter objects

- college
 - Tables
 - college
 - Columns
 - college_id
 - college_name
 - location
 - Indexes
 - Foreign Keys
 - Triggers
 - student
 - students
 - Views

Administration Schemas

Information: Schema: college

Query 1 x

Limit to 1000 rows

SQLAdditions: Jump to

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

1 • USE college;

2 • select * from college;

3

Result Grid

	college_id	college_name	location
2	VIT	Chennai	
3	VIT	Bhopal	

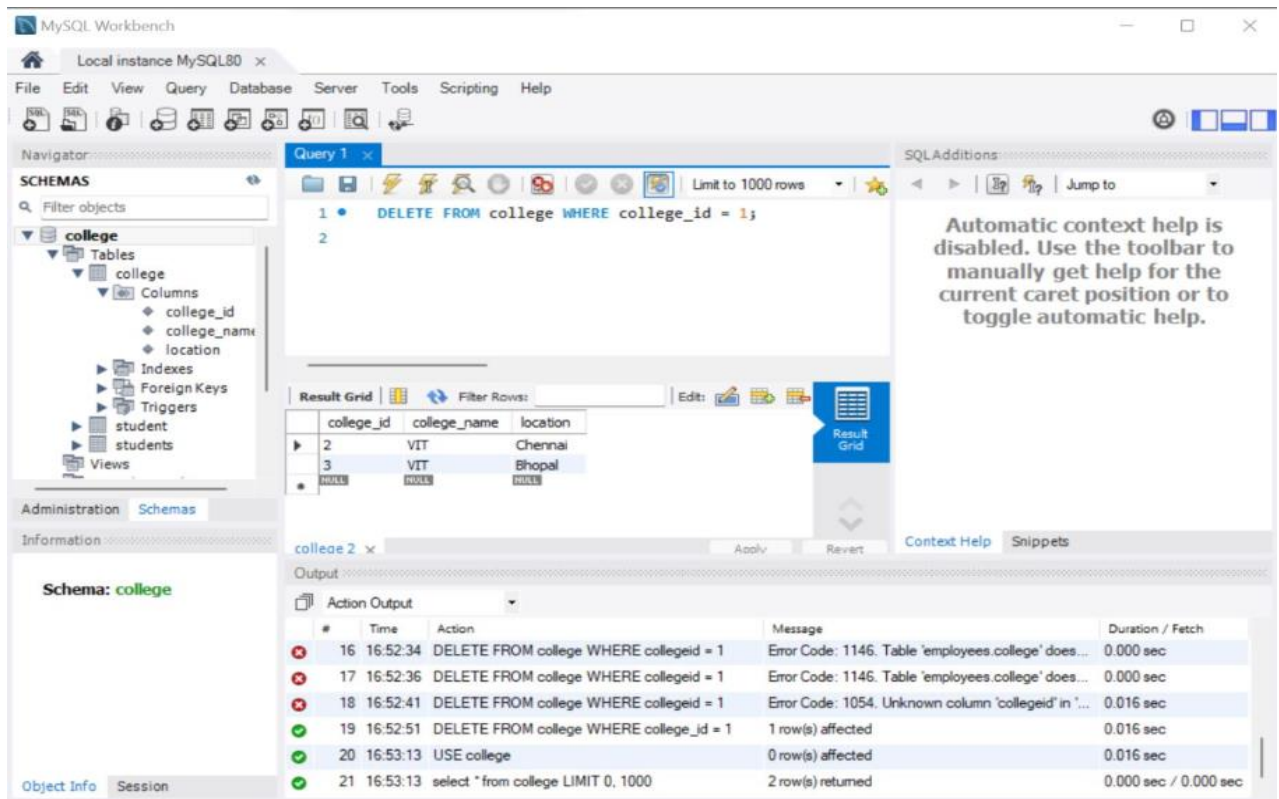
college 2 x

Apply Revert Context Help Snippets

Output

Action Output

#	Time	Action	Message	Duration / Fetch
16	16:52:34	DELETE FROM college WHERE collegeid = 1	Error Code: 1146. Table 'employees.college' does...	0.000 sec
17	16:52:36	DELETE FROM college WHERE collegeid = 1	Error Code: 1146. Table 'employees.college' does...	0.000 sec
18	16:52:41	DELETE FROM college WHERE collegeid = 1	Error Code: 1054. Unknown column 'collegeid' in '...	0.016 sec
19	16:52:51	DELETE FROM college WHERE college_id = 1	1 row(s) affected	0.016 sec
20	16:53:13	USE college	0 row(s) affected	0.016 sec
21	16:53:13	select * from college LIMIT 0, 1000	2 row(s) returned	0.000 sec / 0.000 sec



MONGODB:

```
> use dbs
< switched to db dbs
> use cscorner
< switched to db cscorner
> db.employee.insertOne({ id: 1,name:"hari",age:25,status:"employed",dept:"Mech"});
< {
  acknowledged: true,
  insertedId: ObjectId("6473384c34056c98ecee70ba")
}
> db.employee.insertMany([
  { id: 5,name:"ricky",age:21,status:"employed",dept:"Mech"},
  { id: 6,name:"johnny",age:25,status:"unemployed",dept:"CSE" },
]);
< {
  acknowledged: true,
  insertedIds: {
    '0': ObjectId("647338a334056c98ecee70bb"),
    '1': ObjectId("647338a334056c98ecee70bc")
  }
}
```

Screenshots of mongo shell:

```

Please enter a MongoDB connection string (Default: mongodb://localhost/):

Current Mongosh Log ID: 64732298f462768aae9bb8e8
Connecting to:      mongodb://127.0.0.1:27017/?directConnection=true&serverSelectionTimeoutMS=2000&appName=mongosh+1.9.1
Using MongoDB:      6.0.6
Using Mongosh:      1.9.1

For mongosh info see: https://docs.mongodb.com/mongodb-shell/

To help improve our products, anonymous usage data is collected and sent to MongoDB periodically (https://www.mongodb.com/legal/privacy-policy).
You can opt-out by running the disableTelemetry() command.

-----
  The server generated these startup warnings when booting
  2023-05-28T14:56:45.567+05:30: Access control is not enabled for the database. Read and write access to data and configuration is unrestricted
-----

test> show dbs
admin    40.00 KiB
config   72.00 KiB
local    72.00 KiB
student  8.00 KiB
test> use cscorner
switched to db cscorner
cscorner> show dbs
admin    40.00 KiB
config   72.00 KiB
local    72.00 KiB
student  8.00 KiB
cscorner>
cscorner> db.createCollection("employees")
MongoServerError: Collection cscorner.employees already exists.
cscorner> show collections
employees
cscorner> use employees
switched to db employees
cscorner> db.createCollection("employees")
MongoServerError: Collection cscorner.employees already exists.
cscorner> show collections
employees
cscorner> use employees

```

INSERTING COMMANDS:

```

employees> db.employees.insertOne({id:1,name:"Pat",age:180,status:"single",dept:"CSE"})
{
  acknowledged: true,
  insertedId: ObjectId("64732606f462768aae9bb8e9")
}
employees> db.employees.insertMany([{id:2,name:"adv",age:20,status:"American",dept:"CSE"},{id:3,name:"ash",age:20,status:"Single",dept:"Mech"}])
{
  acknowledged: true,
  insertedIds: {
    '0': ObjectId("647326acf462768aae9bb8ea"),
    '1': ObjectId("647326acf462768aae9bb8eb")
  }
}

```

UPDATE COMMANDS:


```
employees> db.employees.updateOne({name:"Pat"},{$set:{name:"yeshu"}});
{
  acknowledged: true,
  insertedId: null,
  matchedCount: 1,
  modifiedCount: 1,
  upsertedCount: 0
}
employees> db.employees.find().pretty()
[
  {
    _id: ObjectId("64732606f462768aae9bb8e9"),
    id: 1,
    name: 'yeshu',
    age: 180,
    status: 'single',
    dept: 'CSE'
  },
  {
    _id: ObjectId("647326acf462768aae9bb8ea"),
    id: 2,
    name: 'adv',
    age: 20,
    status: 'American',
    dept: 'CSE'
  },
  {
    _id: ObjectId("647326acf462768aae9bb8eb"),
    id: 3,
    name: 'ash',
    age: 20,
    status: 'Single',
    dept: 'Mech'
  }
]
```

```

employees> db.employees.updateMany({age:20},{ $set :{name:"sai"}});
{
  acknowledged: true,
  insertedId: null,
  matchedCount: 2,
  modifiedCount: 2,
  upsertedCount: 0
}
employees> db.employees.find().pretty()
[
  {
    _id: ObjectId("64732606f462768aae9bb8e9"),
    id: 1,
    name: 'yeshu',
    age: 180,
    status: 'single',
    dept: 'CSE'
  },
  {
    _id: ObjectId("647326acf462768aae9bb8ea"),
    id: 2,
    name: 'sai',
    age: 20,
    status: 'American',
    dept: 'CSE'
  },
  {
    _id: ObjectId("647326acf462768aae9bb8eb"),
    id: 3,
    name: 'sai',
    age: 20,
    status: 'Single',
    dept: 'Mech'
  }
]

```

DELETE COMMANDS:

```
employees> db.employees.find().pretty()
[
  {
    _id: ObjectId("64732606f462768aae9bb8e9"),
    id: 1,
    name: 'yeshu',
    age: 180,
    status: 'single',
    dept: 'CSE'
  },
  {
    _id: ObjectId("647326acf462768aae9bb8ea"),
    id: 2,
    name: 'sai',
    age: 20,
    status: 'American',
    dept: 'CSE'
  },
  {
    _id: ObjectId("647326acf462768aae9bb8eb"),
    id: 3,
    name: 'sai',
    age: 20,
    status: 'Single',
    dept: 'Mech'
  }
]
```

```

employees> db.employees.deleteOne({ id: 3 });
{ acknowledged: true, deletedCount: 1 }
employees> db.employees.find().pretty()
[
  {
    _id: ObjectId("64732606f462768aae9bb8e9"),
    id: 1,
    name: 'yeshu',
    age: 180,
    status: 'single',
    dept: 'CSE'
  },
  {
    _id: ObjectId("647326acf462768aae9bb8ea"),
    id: 2,
    name: 'sai',
    age: 20,
    status: 'American',
    dept: 'CSE'
  }
]
employees> db.employees.insertMany([{id:2,name:"yash",age:40,status:"korean",dept:"CSE"},{id:4,name:"mapla",age:50,status:"Single",dept:"ECE"}])
{
  acknowledged: true,
  insertedIds: {
    '0': ObjectId("64732918f462768aae9bb8ec"),
    '1': ObjectId("64732918f462768aae9bb8ed")
  }
}
employees> db.employees.find().pretty()
[
  {
    _id: ObjectId("64732606f462768aae9bb8e9"),
    id: 1,
    name: 'yeshu',
    age: 180,
    status: 'single',
    dept: 'CSE'
  },
  {
    _id: ObjectId("64732918f462768aae9bb8ec"),
    id: 2,
    name: 'yash',
    age: 40,
    status: 'korean',
    dept: 'CSE'
  },
  {
    _id: ObjectId("64732918f462768aae9bb8ed"),
    id: 4,
    name: 'mapla',
    age: 50,
    status: 'Single',
    dept: 'ECE'
  }
]

```

```

employees> db.employees.deleteMany({ id: 2 },{id:1});
{ acknowledged: true, deletedCount: 1 }
employees> db.employees.find().pretty()
[
  {
    _id: ObjectId("64732606f462768aae9bb8e9"),
    id: 1,
    name: 'yeshu',
    age: 180,
    status: 'single',
    dept: 'CSE'
  },
  {
    _id: ObjectId("64732918f462768aae9bb8ed"),
    id: 4,
    name: 'mapla',
    age: 50,
    status: 'Single',
    dept: 'ECE'
  }
]
employees>

```

Mongocompass :

The screenshot shows the MongoDB Compass web interface. The top bar indicates the connection to 'localhost:27017' and the selected database 'employees'. The left sidebar shows a tree view of databases and collections, with 'employees' selected. The main panel displays the 'employees' collection with two documents. The first document has id: 1, name: 'yeshu', age: 180, status: 'single', and dept: 'CSE'. The second document has id: 4, name: 'mapla', age: 50, status: 'Single', and dept: 'ECE'. The interface includes a search bar, a filter section, and buttons for adding and exporting data.

localhost:27017 ... Documents employees.emplo...

My Queries Databases Search

employees

employees.employees 2 DOCUMENTS 1 INDEXES

Documents Aggregations Schema Explain Plan Indexes Validation

Filter Type a query: { field: 'value' } Reset Find More Options

ADD DATA EXPORT DATA 1 - 2 of 2

```

_id: ObjectId('64732606f462768aae9bb8e9')
id: 1
name: "yeshu"
age: 180
status: "single"
dept: "CSE"

```

```

_id: ObjectId('64732918f462768aae9bb8ed')
id: 4
name: "mapla"
age: 50
status: "Single"
dept: "ECE"

```