

Name : S. Sai Prahas

Reg.no: 20MIS0162

Campus: Vellore

---

## **Assignment - 2**

### **Question:**

1. Create, update, delete commands in mysql
2. Create tables, and perform joins in mysql
3. Create, update, delete commands in mongo

### **MySQL commands:**

#### **1. Create, update, delete commands in MySQL**

```
create database mydb1;
use mydb1;
create table students_detail(id int primary key,
stud_name varchar(45) not null,
branch varchar(45) not null,
age int not null
);

insert into students_detail(id, stud_name, branch,age)
values(1, 'Sudharshan', 'IT', 21),
(2, 'Kiran', 'CSE', 22),
(3, 'Chandu', 'MIS', 23),
(4, 'Mukesh', 'ECE', 24)
;

update students_detail set stud_name = "Vishnu" where id = 1;

delete from students_detail where id = 2;
```

MySQL Workbench

MyConnection x

File Edit View Query Database Server Tools Scripting Help

Navigator

MANAGEMENT

- Server Status
- Client Connections
- Users and Privileges
- Status and System Variables
- Data Export
- Data Import/Restore

INSTANCE

- Startup / Shutdown
- Server Logs
- Options File

PERFORMANCE

- Dashboard
- Performance Reports
- Performance Schema Setup

Administration Schemas

Information

No object selected

Query 1

```
1 create database mydb1;
2 use mydb1;
3 create table students_detail(id int primary key,
4 stud_name varchar(45) not null,
5 branch varchar(45) not null,
6 age int not null
7 );
8
9 insert into students_detail(id, stud_name, branch, age)
10 values(1, 'Sudharshan', 'IT', 21),
11 (2, 'Kiran', 'CSE', 22),
12 (3, 'Chandu', 'MIS', 23),
13 (4, 'Nukesh', 'ECE', 24)
14 ;
15
16 update students_detail set stud_name = 'Vishnu' where id = 1;
17
18 delete from students_detail where id = 2;
```

Output

#	Time	Action	Message	Duration / Fetch
2	09:40:10	create database mydb1	1 row(s) affected	0.109 sec
3	09:40:10	use mydb1	0 row(s) affected	0.000 sec
4	09:40:10	create table students_detail(id int primary key, stud_name varchar(45) not null, branch varchar(45) not null, age int not null)	0 row(s) affected	0.172 sec
5	09:40:10	insert into students_detail(id, stud_name, branch, age) values(1, 'Sudharshan', 'IT', 21), (2, 'Kiran', 'CSE', 22), (3, 'Chandu', 'MIS', 23), (4, 'Nukesh', 'ECE', 24)	4 row(s) affected Records: 4 Duplicates: 0 Warnings: 0	0.032 sec
6	09:40:10	update students_detail set stud_name = 'Vishnu' where id = 1	1 row(s) affected Rows matched: 1 Changed: 1 Warnings: 0	0.031 sec
7	09:43:11	delete from students_detail where id = 2	1 row(s) affected	0.109 sec

Object Info Session

Query Completed

37°C Sunny

Search

ENG IN 10:14 28-05-2023

MySQL Workbench

MyConnection x

File Edit View Query Database Server Tools Scripting Help

Navigator

SCHEMAS

- Filter objects
- detail1
- mydb1
- sys

Query 1

```
11 (2, 'Kiran', 'CSE', 22),
12 (3, 'Chandu', 'MIS', 23),
13 (4, 'Nukesh', 'ECE', 24)
14 ;
15
16 update students_detail set stud_name = 'Vishnu' where id = 1;
17
18 delete from students_detail where id = 1;
19
20 select * from students_detail;
```

Result Grid

id	stud_name	branch	age
2	Kiran	CSE	22
3	Chandu	MIS	23
4	Nukesh	ECE	24

students\_detail 3 x

Output

#	Time	Action	Message	Duration / Fetch
3	16:13:02	use mydb1	0 row(s) affected	0.000 sec
4	16:13:02	create table students_detail(id int primary key, stud_name varchar(45) not null, branch varchar(45) not null, age int not null)	0 row(s) affected	1.469 sec
5	16:13:03	insert into students_detail(id, stud_name, branch, age) values(1, 'Sudharshan', 'IT', 21), (2, 'Kiran', 'CSE', 22), (3, 'Chandu', 'MIS', 23), (4, 'Nukesh', 'ECE', 24)	4 row(s) affected Records: 4 Duplicates: 0 Warnings: 0	0.125 sec
6	16:13:04	update students_detail set stud_name = 'Vishnu' where id = 1	1 row(s) affected Rows matched: 1 Changed: 1 Warnings: 0	0.078 sec
7	16:13:04	delete from students_detail where id = 1	1 row(s) affected	0.063 sec
8	16:15:21	show tables	1 row(s) returned	0.047 sec / 0.000 sec
9	16:16:22	describe students_detail	4 row(s) returned	0.078 sec / 0.000 sec
10	16:16:47	select * from students_detail LIMIT 0, 1000	3 row(s) returned	0.031 sec / 0.000 sec

Object Info Session

37°C Hot weather

Search

ENG IN 16:17 28-05-2023

## 2. Create tables, and perform join operations in MySQL

```
create database mydb;
use mydb;
create table customers(id int primary key,
cust_name varchar(45) not null,
occupation varchar(45) not null,
income int not null,
qualification varchar(45) not null
);
```

```
insert into customers
(id, cust_name, occupation, income, qualification)
values (1, 'Sudharshan', 'Developer', 20000, 'Btech'),
(2, 'Kusuma', 'Engineer', 40000, 'Btech'),
(3, 'Mukesh', 'Scientists', 60000, 'MSc'),
(4, 'Navya', 'Businessman', 100000, 'MBA'),
(5, 'Chandu', 'Manager', 50000, 'MBA');
```

```
create table orders(order_id int primary key,
date date not null,
customer_id int not null,
price int not null
);
```

```
insert into orders
(order_id, date, customer_id, price)
values (1001, '2020-03-20', 2, 300),
(1002, '2020-02-15', 4, 2500),
(1003, '2020-01-31', 5, 5000),
(1004, '2020-09-23', 2, 1500),
(1005, '2020-09-30', 1, 4500);
```

```
select customers.id, customers.cust_name, customers.income,
orders.price
from customers
```

inner join orders

on customers.id = orders.customer\_id;

select customers.id, customers.cust\_name, customers.income,  
orders.price

from customers

left join orders

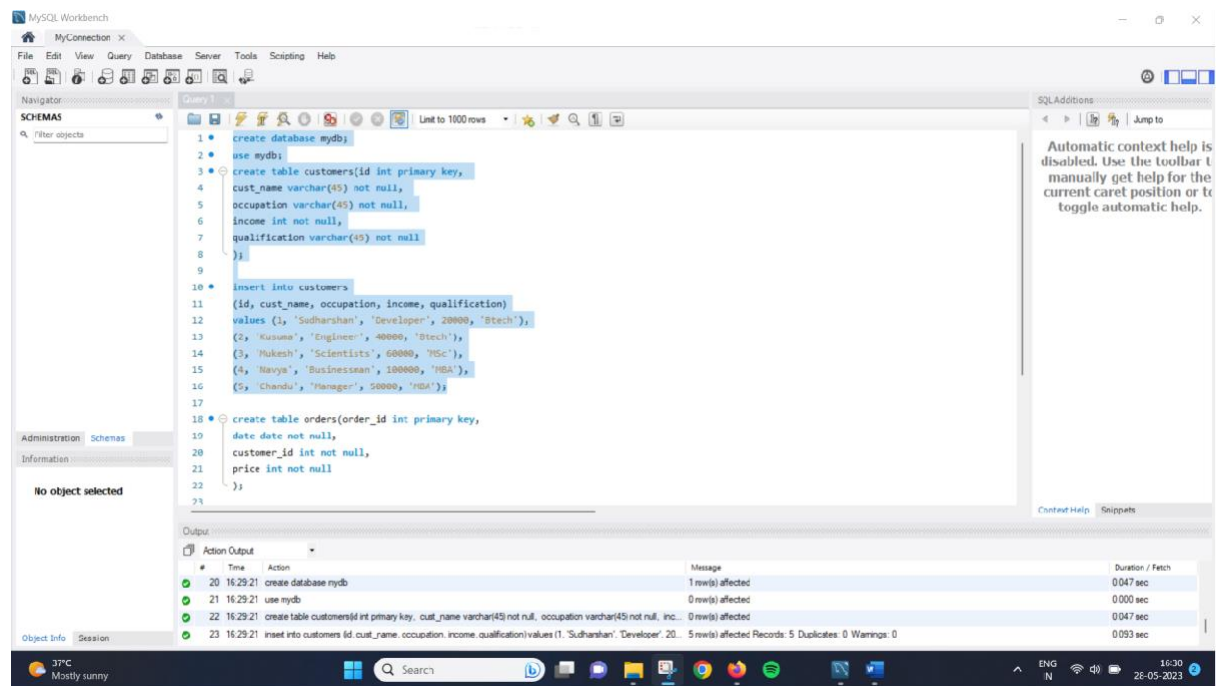
on customers.id = orders.customer\_id;

select customers.id, customers.cust\_name, customers.income,  
orders.price

from customers

right join orders

on customers.id = orders.customer\_id;



MySQL Workbench

MyConnection x

File Edit View Query Database Server Tools Scripting Help

Navigator

SCHEMAS

Filter objects

mydb

Tables

customers

Columns

Indexes

Foreign Keys

Triggers

Views

Stored Procedures

Functions

Query 1

Limit to 1000 rows

```

17
18 create table orders(order_id int primary key,
19 date date not null,
20 customer_id int not null,
21 price int not null
22 );
23
24 insert into orders
25 (order_id, date, customer_id, price)
26 values (1001, '2020-03-20', 2, 300),
27 (1002, '2020-02-15', 4, 2500),
28 (1003, '2020-01-31', 5, 5000),
29 (1004, '2020-09-23', 2, 1500),
30 (1005, '2020-09-30', 1, 4500);
31
32
33

```

SQLAdditions

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

Administration Schemas

Information

No object selected

Object Info Session

Output

Action Output

#	Time	Action	Message	Duration / Fech
1	16:41:37	create table orders(order_id int primary key, date date not null, customer_id int not null, price int not null)	0 row(s) affected	0.187 sec
2	16:41:37	insert into orders (order_id, date, customer_id, price) values (1001, '2020-03-20', 2, 300), (1002, '2020-02-15', 4, 2500), (1003, '2020-01-31', 5, 5000), (1004, '2020-09-23', 2, 1500), (1005, '2020-09-30', 1, 4500);	5 row(s) affected Records: 5 Duplicates: 0 Warnings: 0	0.015 sec

37°C Mostly sunny

Search

ENG N

16:41 26-05-2023

MySQL Workbench

MyConnection x

File Edit View Query Database Server Tools Scripting Help

Navigator

SCHEMAS

Filter objects

mydb

Tables

customers

Columns

Indexes

Foreign Keys

Triggers

Views

Stored Procedures

Functions

Query 1

Limit to 1000 rows

```

29
30 (1004, '2020-09-23', 2, 1500),
31 (1005, '2020-09-30', 1, 4500);
32
33 select customers.id, customers.cust_name, customers.income, orders.price
34 from customers
35 inner join orders
36 on customers.id = orders.customer_id;
37

```

SQLAdditions

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

Administration Schemas

Information

No object selected

Object Info Session

Output

Action Output

#	Time	Action	Message	Duration / Fech
1	16:47:40	select customers.id, customers.cust_name, customers.income, orders.price from customers inner join orders on c...	0 row(s) returned	0.015 sec / 0.000 sec
2	16:48:10	select customers.id, customers.cust_name, customers.income, orders.price from customers inner join orders on c...	4 row(s) returned	0.003 sec / 0.000 sec
3	16:48:56	DROP TABLE `mydb`.`orders`	0 row(s) affected	0.032 sec
4	16:49:03	create table orders(order_id int primary key, date date not null, customer_id int not null, price int not null)	0 row(s) affected	0.140 sec
5	16:49:03	insert into orders (order_id, date, customer_id, price) values (1001, '2020-03-20', 2, 300), (1002, '2020-02-15', 4, 2500), (1003, '2020-01-31', 5, 5000), (1004, '2020-09-23', 2, 1500), (1005, '2020-09-30', 1, 4500);	5 row(s) affected Records: 5 Duplicates: 0 Warnings: 0	0.031 sec
6	16:49:05	select customers.id, customers.cust_name, customers.income, orders.price from customers inner join orders on c...	5 row(s) returned	0.003 sec / 0.000 sec

Result Grid

id	cust_name	income	price
2	Kusuma	40000	300
4	Navya	100000	2500
5	Chandu	50000	5000
2	Kusuma	40000	1500
1	Sudharshan	20000	4500

37°C Mostly sunny

Search

ENG N

16:49 26-05-2023

MySQL Workbench

MyConnection x

File Edit View Query Database Server Tools Scripting Help

Navigator

SCHEMAS

Filter objects

mydb

Tables

customers

Columns

Indexes

Foreign Keys

Triggers

Views

Stored Procedures

Functions

Query 1

Limit to 1000 rows

SQLAdditions

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

```
36 on customers.id = orders.customer_id;
37
38
39 select customers.id, customers.cust_name, customers.income, orders.price
40 from customers
41 left join orders
42 on customers.id = orders.customer_id;
43
44
45 select customers.id, customers.cust_name, customers.income, orders.price
```

Result Grid

#	id	cust_name	income	price
1	Sudharshan	20000	4500	
2	Kusuma	40000	1500	
2	Kusuma	40000	300	
3	Mukesh	60000		
4	Navya	100000	2500	

Result 9 x

Output

Information

No object selected

Object Info Session

37°C Hot weather

Search

ENG N

16:51 26-05-2023

MySQL Workbench

MyConnection x

File Edit View Query Database Server Tools Scripting Help

Navigator

SCHEMAS

Filter objects

mydb

Tables

customers

Columns

Indexes

Foreign Keys

Triggers

Views

Stored Procedures

Functions

Query 1

Limit to 1000 rows

SQLAdditions

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

```
41 left join orders
42 on customers.id = orders.customer_id;
43
44
45 select customers.id, customers.cust_name, customers.income, orders.price
46 from customers
47 right join orders
48 on customers.id = orders.customer_id;
49
50
```

Result Grid

#	id	cust_name	income	price
2	Kusuma	40000	300	
4	Navya	100000	2500	
5	Chandu	50000	5000	
2	Kusuma	40000	1500	
1	Sudharshan	20000	4500	

Result 10 x

Output

Information

No object selected

Object Info Session

37°C Hot weather

Search

ENG N

16:51 26-05-2023

### 3. Create, update, delete commands in MongoDB.

#### Create:

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

> db.student.insert([ {name:"Sudharshan", email:"sudharshan@gmail.com", branch:"CSE", age:21}, {name:"Chandu", email:"chandu@gmail.com", branch:"MIS", age:23}, {name:"Vishnu", email:"vishnu@gmail.com", age:24} ])
< DeprecationWarning: Collection.insert() is deprecated. Use insertOne, insertMany, or bulkWrite.
< {
  acknowledged: true,
  insertedIds: {
    '0': ObjectId("64733f16e94a66239e8467e8"),
    '1': ObjectId("64733f16e94a66239e8467e9"),
    '2': ObjectId("64733f16e94a66239e8467ea")
  }
}

> db.student.find()
< {
  _id: ObjectId("64733f16e94a66239e8467e8"),
  name: 'Sudharshan',
  email: 'sudharshan@gmail.com',
  branch: 'CSE',
  age: 21
}
{
  _id: ObjectId("64733f16e94a66239e8467e9"),
  name: 'Chandu',
  email: 'chandu@gmail.com',
  branch: 'MIS',
  age: 23
}
{
  _id: ObjectId("64733f16e94a66239e8467ea"),
  name: 'Vishnu',
  email: 'vishnu@gmail.com',
  branch: 'MIS',
  age: 24
}
```

#### Update:

```
> db.student.update({age:23},{ $set:{name:"Kiran"}})
< {
  acknowledged: true,
  insertedId: null,
  matchedCount: 1,
  modifiedCount: 1,
  upsertedCount: 0
}

> db.student.find()
< {
  _id: ObjectId("64733f16e94a66239e8467e9"),
  name: 'Kiran',
  email: 'chandu@gmail.com',
  branch: 'MIS',
  age: 23
}
```

## Delete:

```
> db.student.remove({name:"Sudharshan"})
< DeprecationWarning: Collection.remove() is deprecated. Use deleteOne, deleteMany, findOneAndDelete, or bulkWrite.
< {
  acknowledged: true,
  deletedCount: 1
}
> db.student.find()
< {
  _id: ObjectId("64733f16e94a66239e8467e9"),
  name: 'Chandu',
  email: 'chandu@gmail.com',
  branch: 'MIS',
  age: 23
}
{
  _id: ObjectId("64733f16e94a66239e8467ea"),
  name: 'Vishnu',
  email: 'vishnu@gmail.com',
  branch: 'MIS',
  age: 24
}
```

