Modern Application Development

Java Spring Boot

Assignment -2

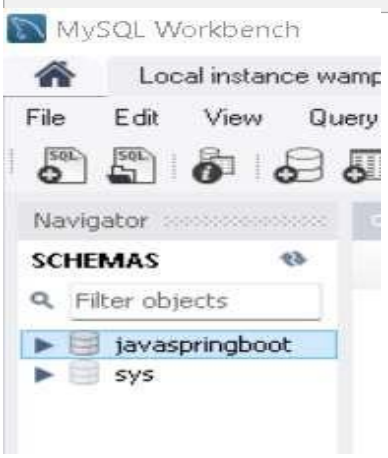
MySQL

1. Create Command

Create Database:

Create database (database_name);

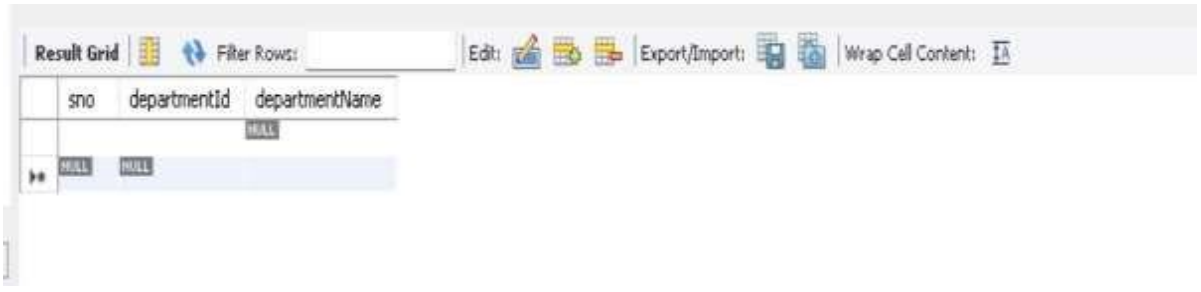
```
CREATE database javaspringboot;
```



Create Table:

Create table table_name (columns_name with datatype)

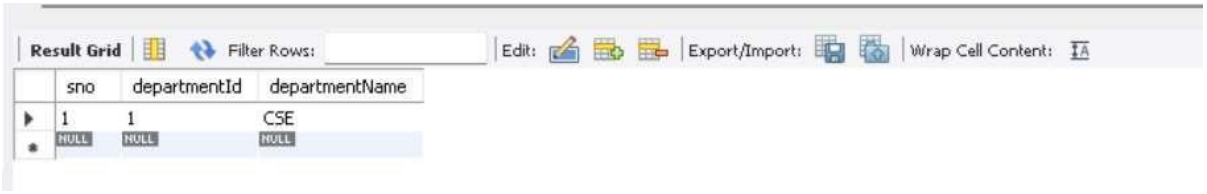
```
create table javaSpringBoot.department

(sno INT NOT NULL,

departmentId INT NOT NULL,

departmentName VARCHAR(20) NOT NULL,

PRIMARY KEY(departmentId));
```



2. Insert Command

Insert into database.table_name values( )

```
INSERT INTO javaSpringBoot.DEPARTMENT VALUES(1,1,'CSE');
```



3. Delete Command

Delete from table_name **where** column_name = 'any data';

```
Delete from javaspringBoot.department where sno=1;
```

4. Create tables and perform joins

Create tables:

```
create table javaSpringBoot.department(sno INT NOT NULL, departmentId INT NOT NULL,
departmentName VARCHAR(20) NOT NULL, PRIMARY KEY(departmentId));

Create table javaSpringBoot.faculty(SNO INT,departmentId INT,facultyId INT,facuLtyName
varchar(2), PRIMARY KEY(SNO), FOREIGN KEY(departmentId) REFERENCES
department(departmentId));

create table javaSpringBoot.Student(sno INT,departmentId INT,studentId INT, StudentName
varchar(20), primary key(sno), foreign key(departmentId) references department(departmentId));
```

Inserting values:

Department Table

```
INSERT INTO javaSpringBoot.DEPARTMENT VALUES(1,1,'CSE')
INSERT INTO javaSpringBoot.DEPARTMENT VALUES(2,2,'MAT );
INSERT INTO javaSpringBoot.DEPARTMENT VALUES(3,3,'IT');
INSERT INTO javaSpringBoot.DEPARTMENT VALUES(4,4,'BCA');
```

Faculty Table

```
INSERT INTO javaSpringBoot.FACULTY VALUES(1,1,1,'AB');
INSERT INTO javaSpringBoot.FACULTY VALUES(2,1,1,'CD');
INSERT INTO javaSpringBoot.FACULTY VALUES(3,2,5,'EF')
INSERT INTO javaSpringBoot.FACULTY VALUES(4,3,9,'GH');
```

Department Table

```
INSERT INTO javaSpringBoot.STUDENT VALUES(1,1,1,'XY');
INSERT INTO javaSpringBoot.STUDENT VALUES(2,4,1,'WW');
INSERT INTO javaSpringBoot.STUDENT VALUES(4,3,1,'GG');
INSERT INTO javaSpringBoot.STUDENT VALUES(9,2,1,'HH');
```

Join comments:

1. INNER JOIN

```
SELECT * FROM javaSpringBoot.DEPARTMENT JOIN javaSpringBoot.FACULTY ON
DEPARTMENT.DEPARTMENTID=FACULTY.DEPARTMENTID;
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

sno	departmentId	departmentName	SNO	departmentId	facultyId	faculTyName
1	1	CSE	1	1	1	AB
1	1	CSE	2	1	1	CD
2	2	MAT	3	2	5	EF
3	3	IT	4	3	9	GH

2. LEFT (OUTER JOIN)

```
SELECT * FROM javaSpringBoot.DEPARTMENT LEFT JOIN javaSpringBoot.FACULTY ON
DEPARTMENT.DEPARTMENTID=FACULTY.DEPARTMENTID;
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

sno	departmentId	departmentName	SNO	departmentId	facultyId	faculTyName
1	1	CSE	1	1	1	AB
1	1	CSE	2	1	1	CD
2	2	MAT	3	2	5	EF
3	3	IT	4	3	9	GH
4	4	BCA	NULL	NULL	NULL	NULL

3. RIGHT (OUTER JOIN)

```
SELECT * FROM javaSpringBoot.DEPARTMENT RIGHT JOIN javaSpringBoot.FACULTY ON
DEPARTMENT.DEPARTMENTID=FACULTY.DEPARTMENTID;
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

sno	departmentId	departmentName	SNO	departmentId	facultyId	faculTyName
1	1	CSE	1	1	1	AB
1	1	CSE	2	1	1	CD
2	2	MAT	3	2	5	EF
3	3	IT	4	3	9	GH

4. FULL OUTER JOIN

SELECT * **FROM** javaSpringBoot.DEPARTMENT **FULL JOIN** javaSpringBoot.FACULTY;

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

sno	departmentId	departmentName	SNO	departmentId	facultyId	faculTyName
4	4	BCA	1	1	1	AB
3	3	IT	1	1	1	AB
2	2	MAT	1	1	1	AB
1	1	CSE	1	1	1	AB
4	4	BCA	2	1	1	CD
3	3	IT	2	1	1	CD
2	2	MAT	2	1	1	CD
1	1	CSE	2	1	1	CD
4	4	BCA	3	2	5	EF
3	3	IT	3	2	5	EF
2	2	MAT	3	2	5	EF
1	1	CSE	3	2	5	EF
4	4	BCA	4	3	9	GH
3	3	IT	4	3	9	GH
2	2	MAT	4	3	9	GH
1	1	CSE	4	3	9	GH

Mongo

1. Create Command

Insert into mongo collection:

db.student.insertOne({'studentName':'abc'})
db.student.insertOne({name:"Andrew",age:45})

```
> db.student.insertOne({'studentName':'abc'})
< {
    acknowledged: true,
    insertedId: ObjectId("647360cd066f8da5f9f25712")
  }
> db.student.insertOne({name:"Andrew",age:45})
< {
    acknowledged: true,
    insertedId: ObjectId("647363d8066f8da5f9f25713")
  }
```

2. Update Command

db.student.updateOne({name:'Andrew'},{$set: {age:20} })

```
> db.student.updateOne({name:'Andrew'},{$set: {age:20} })
< {
    acknowledged: true,
    insertedId: null,
    matchedCount: 1,
    modifiedCount: 1,
    upsertedCount: 0
  }
> db.student.find()
< {
    _id: ObjectId("647360cd066f8da5f9f25712"),
    studentName: 'abc'
  }
  {
    _id: ObjectId("647363d8066f8da5f9f25713"),
    name: 'Andrew',
    age: 20
  }
```

3. Delete Command

db.student.deleteOne({name:"Andrew"})

```
> db.student.deleteOne({name:"Andrew"})
< {
    acknowledged: true,
    deletedCount: 1
  }
> db.student.find()
< {
    _id: ObjectId("647360cd066f8da5f9f25712"),
    studentName: 'abc'
  }
```