PRESENTATION
ON
EXAM PORTAL

UNDER THE GUIDENCE OF

RAHUL GABA

TECHNICAL TRAINER

EDUBRIDGE (JAVA FULL STACK DEVELOPER)

SUBMITTED BY

SNEHAL TALEKAR

Batch: 2022-7590

Center: Mumbai

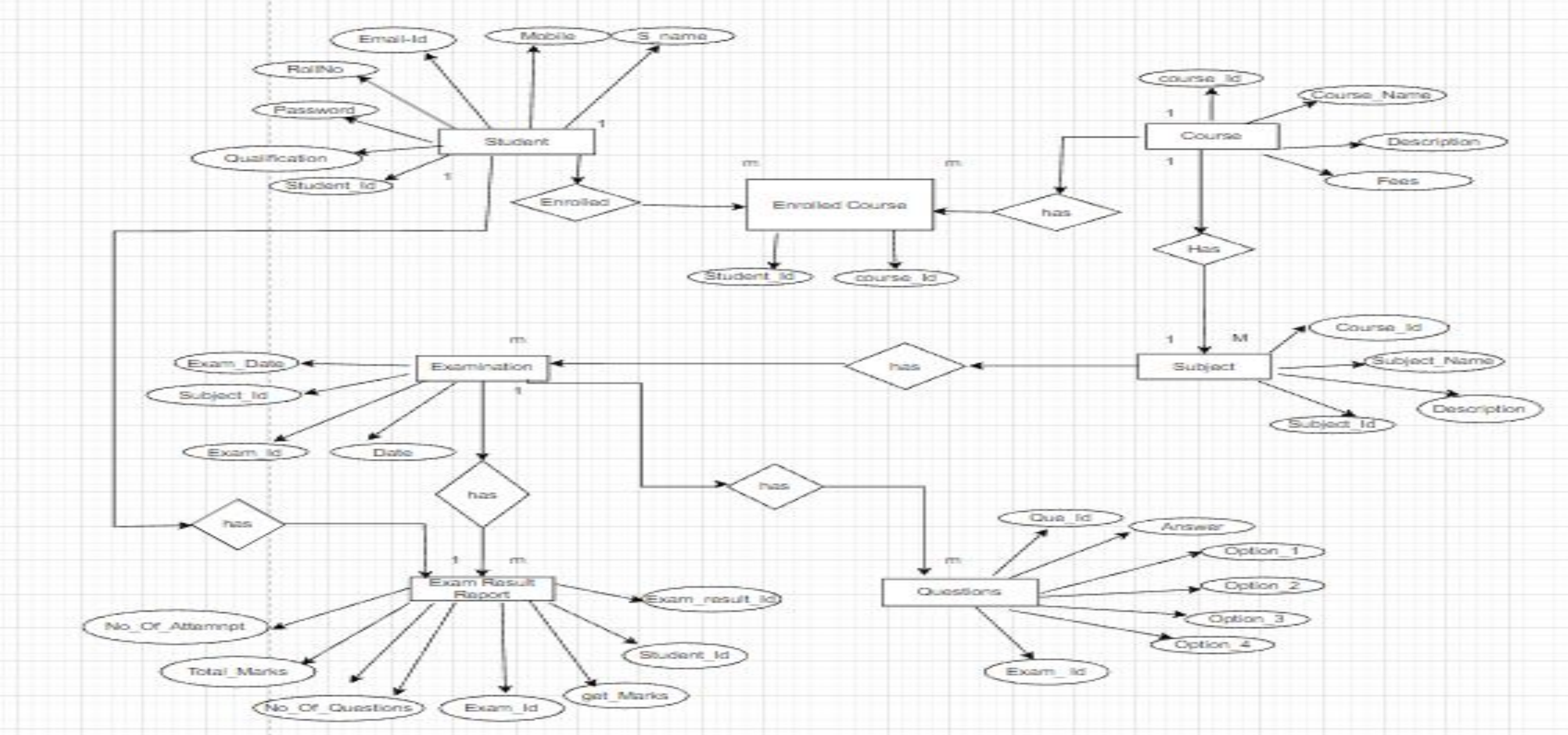
ABSTRACT

Exam Portal, the main objective of Exam Portal is to develop software which manages Student details and Exam Portal can manage the details of all students , admin can manage all the courses. The reason behind for creating this project to help an organization to maintain the student records in a proper manner.

INTRODUCTION

- It provide a platform where it allows the admin to view the students who are currently in organization and also provides information on the courses and the exam of the students so that the organization may track and assess their student records.
- It helps the organization to find/track the student records according to their courses and exam.

ER DIAGRAM


Email-Id
Mobile
S_name
RollNo
Password
Qualification
Student_Id
Student
1
1
Enrolled
m
Enrolled Course
Student_Id
course_Id
m
has
course_Id
Course_Name
1
Course
Description
1
Fees
Has
1
M
Course_Id
Subject
Subject_Name
Description
Subject_Id
has
m
Exam_Date
Examination
Subject_Id
1
Exam_Id
Date
has
has
has
1
m
Exam Result Report
Exam_result_Id
No_Of_Attemmpt
Total_Marks
No_Of_Questions
Exam_Id
get_Marks
Student_Id
m
Questions
Que_Id
Answer
Option_1
Option_2
Option_3
Option_4
Exam_Id

SYSTEM REQUIREMENTS

HARDWARE REQUIREMENT:-

- i3 or i5 Processor
- 4 GB RAM
- 100 GB Hard Disk
- Windows 10 OS

SOFTWARE REQUIREMENT:-

- **Back End**: Spring Boot and JPA Hibernate
- **Front End** : Angular
- **DB Server**: MySQL Server 8.0.CE
- **Web Server**: Apache Tomcat
- **IDE**: Spring Tool Suite(STS), Visual Studio Code
- **Tool**: Postman
- **Testing**: Junit

EXAM PORTAL MODULES

- Student Module
- Course Module
- Subject Module
- Exam Module
- Question Module
- ExamResultReport

STUDENT MODULE

- In Student Module we can save the data by using PostMan tool.
- In Student Module can add the student details by using Post Mapping and using Get Mapping we can get list of student details , and using Put Mapping we can update student details , and using Delete Mapping We can delete the student data in database .
- We can use JUnit test method for Student Module .

COURSE MODULE

- After completing the student Module , We can move into the Course Module .
- In Course Module we can save the data by using PostMan tool.
- In Course Module can add the course details by using Post Mapping and using Get Mapping we can get list of course details , and using Put Mapping we can update course details , and using Delete Mapping We can delete the course data in database .
- We can use JUnit test method for Course Module .
- After adding the Course details we can move to add the subject details to the database.

SUBJECT MODULE

- After completing the subject Module , We can move into the Subject Module .
- In Subject Module we can save the data by using PostMan tool.
- In Subject Module can add the subject details by using Post Mapping and using Get Mapping we can get list of subject details , and using Put Mapping we can update subject details , and using Delete Mapping We can delete the subject data in database .
- We can use JUnit test method for Subject Module .
- After adding the Subject details we can move to add the exam details to the database.

EXAMINATION MODULE

- After completing the subject Module , We can move into the Examination Module .
- In Examination Module we can save the data by using Postman tool.
- In Examination Module can add the examination details by using Post Mapping and using Get Mapping we can get list of examination details , and using Put Mapping we can update examination details , and using Delete Mapping We can delete the examination data in database.
- We can use JUnit test method for Examination Module .
- After adding the Examination details we can move to add the question details to the database.

QUESTION MODULE

- After completing the Examination Module , We can move into the question Module .
- In Question Module we can save the data by using PostMan tool.
- In Question Module can add the question details by using Post Mapping and using Get Mapping we can get list of question details , and using Put Mapping we can update question details , and using Delete Mapping We can delete the question data in database.
- We can use JUnit test method for Question Module .

EXAMRESULTREPORT MODULE

- After completing the Examination Module , We can move into the ExamResultReport Module .
- In ExamResultReport module we can save the data by using PostMan tool.
- In ExamResultReport Module can add the ExamReport.

PROJECT IMPLEMENTATION

- The Project is Developed in Spring boot technology with rest API. We used Spring Tool Suite IDE for Design and develop project coding part. To store project data I used MySQL Server 8.0 in that I created tables or entities, use to data related each module.
- First we created a Spring Starter Project, in that we add dependencies like Lombok, Model Mapper, Spring-Devtools, Spring Web, MySql Connector, Spring-test,Spring-Validation,Spring-Jpa in the Pom(Project Object Model).
- We added Spring framework Starter as Parent in the Pom File.

- Created the layers Entity, payload(DTO-Data To Object), Service, Service Implementation, Repository , Exception handling and Controller layer.

- Created the files in the respected layer and implemented according the modules and attributes.

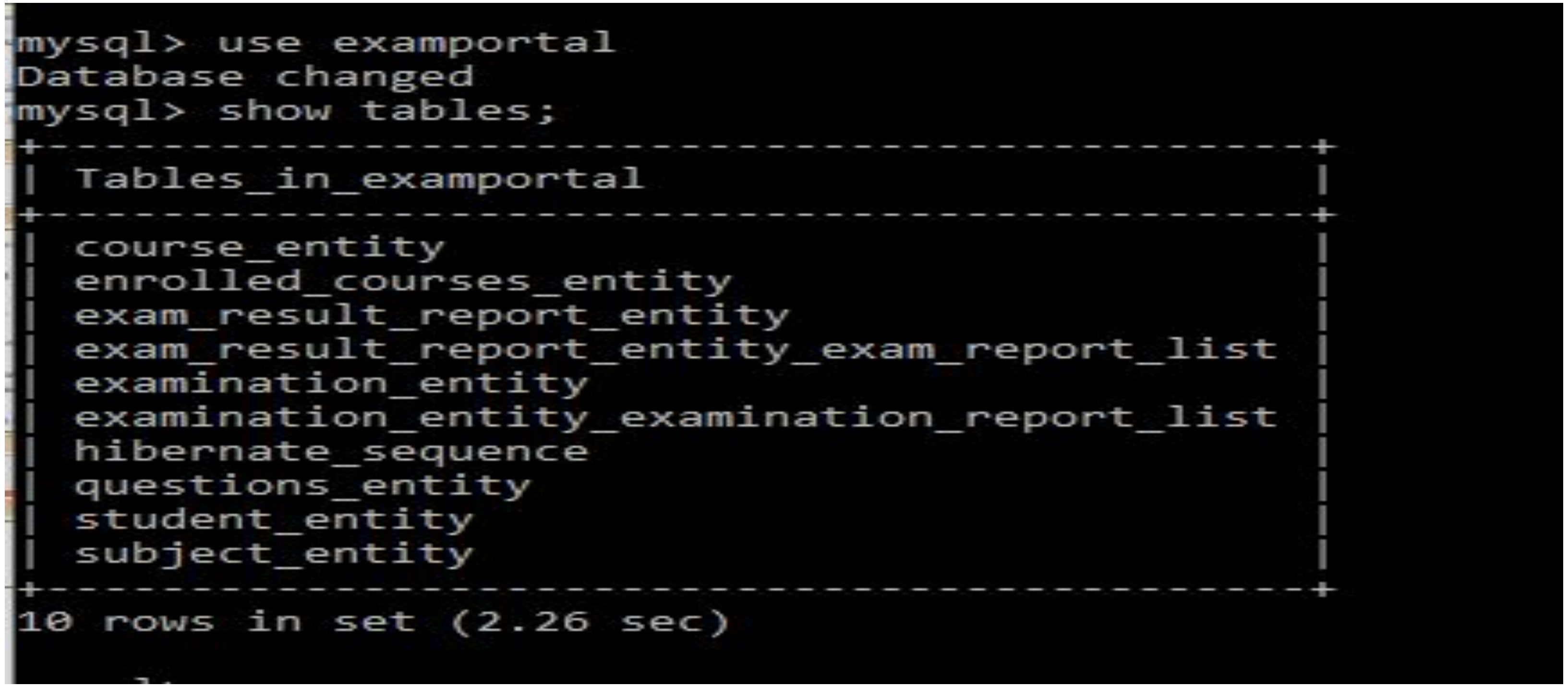
- Connection to the Database via application Properities file, their we need to mention to Driver Class Name(com.mysql.cj.jdbc.Driver) and URL

- (jdbc:mysql://localhost:3306/examportal), Database UserName, password, Dialect, Server port Number:8080

- We can give a input via Postman Tool ,their we need to create request for HTTP Methods .Post(Creating records), Get(Fetching records), Put(Updating records), Delete(Deleting records) .

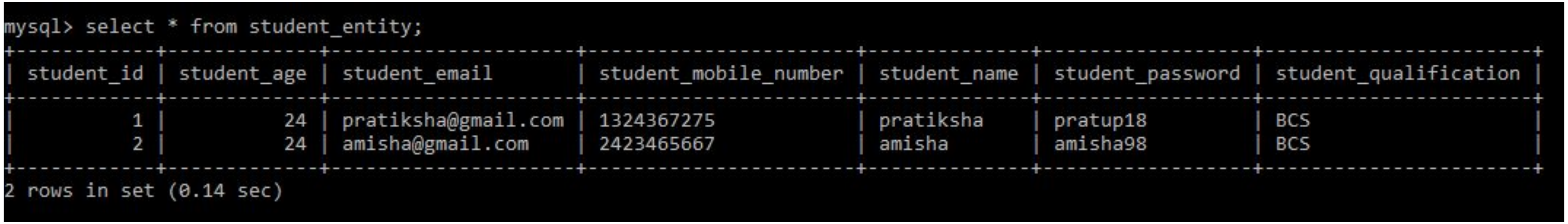
- We need to mention proper URI pattern same as the end point in the Controller.

DATA DICTIONARY



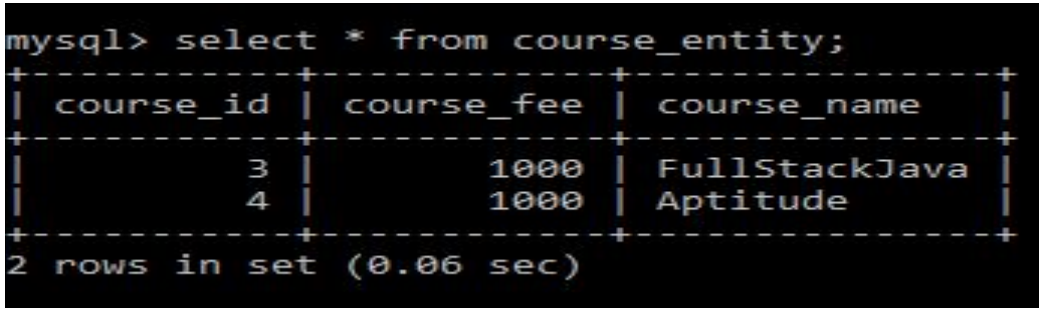
```
mysql> use examportal
Database changed
mysql> show tables;
+--------------------------------------------+
| Tables_in_examportal                       |
+--------------------------------------------+
| course_entity                              |
| enrolled_courses_entity                    |
| exam_result_report_entity                  |
| exam_result_report_entity_exam_report_list |
| examination_entity                         |
| examination_entity_examination_report_list |
| hibernate_sequence                         |
| questions_entity                           |
| student_entity                             |
| subject_entity                             |
+--------------------------------------------+
10 rows in set (2.26 sec)
```

STUDENT TABLE



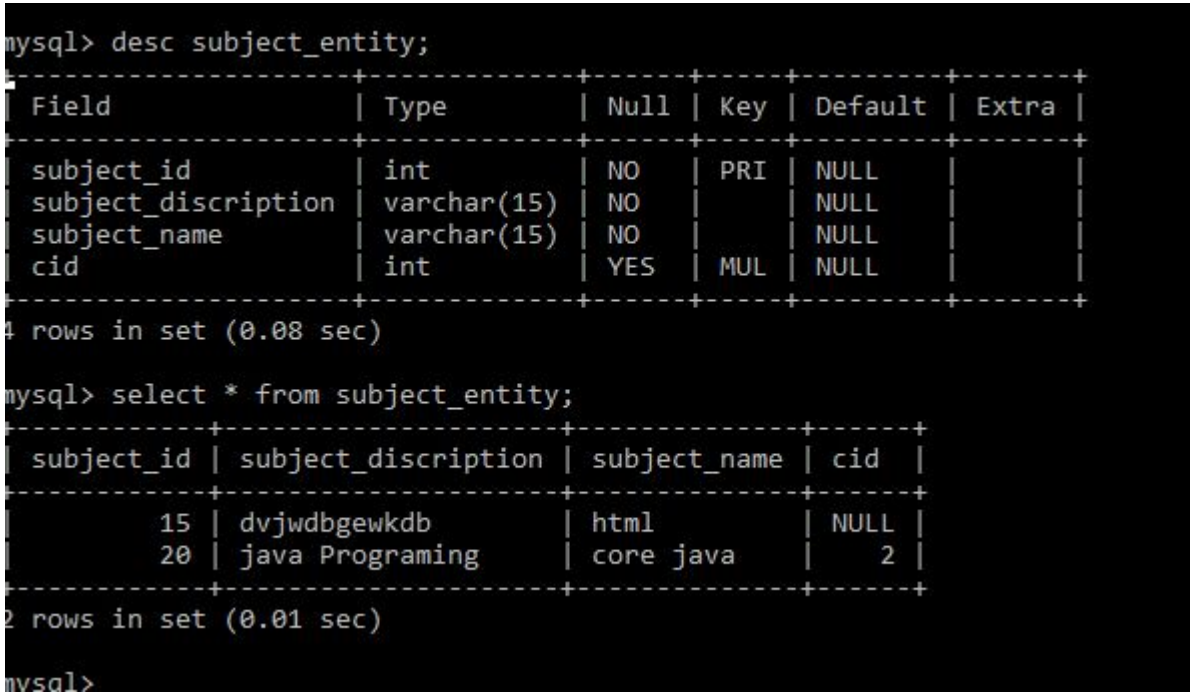
```
mysql> select * from student_entity;
+------------+-------------+---------------------+-----------------------+--------------+------------------+-----------------------+
| student_id | student_age | student_email       | student_mobile_number | student_name | student_password | student_qualification |
+------------+-------------+---------------------+-----------------------+--------------+------------------+-----------------------+
|          1 |          24 | pratiksha@gmail.com | 1324367275            | pratiksha    | pratup18         | BCS                   |
|          2 |          24 | amisha@gmail.com    | 2423465667            | amisha       | amisha98         | BCS                   |
+------------+-------------+---------------------+-----------------------+--------------+------------------+-----------------------+
2 rows in set (0.14 sec)
```

COURSE TABLE



```
mysql> select * from course_entity;
+-----------+------------+---------------+
| course_id | course_fee | course_name   |
+-----------+------------+---------------+
|         3 |       1000 | FullStackJava |
|         4 |       1000 | Aptitude      |
+-----------+------------+---------------+
2 rows in set (0.06 sec)
```

SUBJECT TABLE

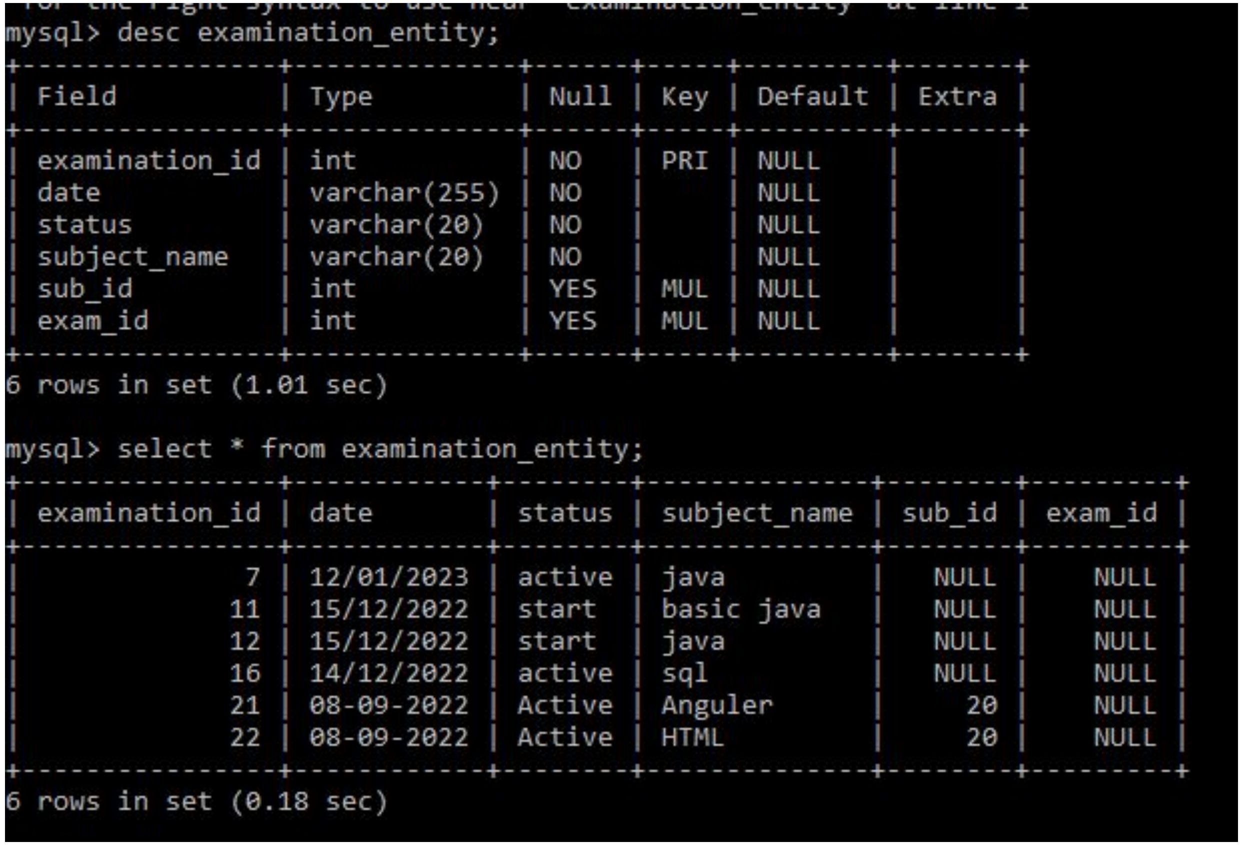


```
mysql> desc subject_entity;
+---------------------+-------------+------+-----+---------+-------+
| Field               | Type        | Null | Key | Default | Extra |
+---------------------+-------------+------+-----+---------+-------+
| subject_id          | int         | NO   | PRI | NULL    |       |
| subject_discription | varchar(15) | NO   |     | NULL    |       |
| subject_name        | varchar(15) | NO   |     | NULL    |       |
| cid                 | int         | YES  | MUL | NULL    |       |
+---------------------+-------------+------+-----+---------+-------+
4 rows in set (0.08 sec)

mysql> select * from subject_entity;
+------------+---------------------+--------------+------+
| subject_id | subject_discription | subject_name | cid  |
+------------+---------------------+--------------+------+
|         15 | dvjwdbgewkdb        | html         | NULL |
|         20 | java Programing     | core java    |    2 |
+------------+---------------------+--------------+------+
2 rows in set (0.01 sec)

mysql>
```

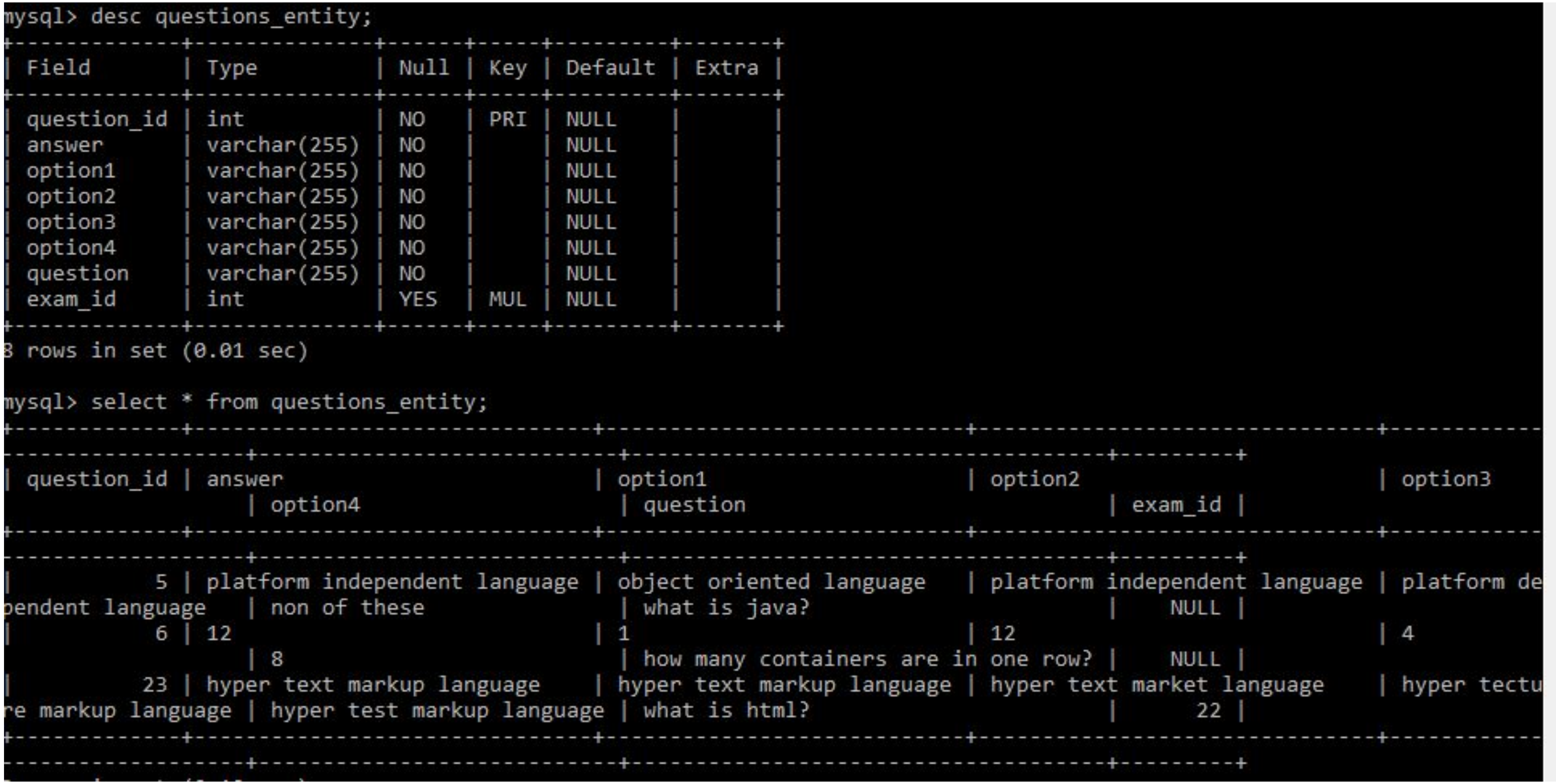
EXAM TABLE



```
for the right syntax to use near 'examination_entity' at line 1
mysql> desc examination_entity;
+----------------+--------------+------+-----+---------+-------+
| Field          | Type         | Null | Key | Default | Extra |
+----------------+--------------+------+-----+---------+-------+
| examination_id | int          | NO   | PRI | NULL    |       |
| date           | varchar(255) | NO   |     | NULL    |       |
| status         | varchar(20)  | NO   |     | NULL    |       |
| subject_name   | varchar(20)  | NO   |     | NULL    |       |
| sub_id         | int          | YES  | MUL | NULL    |       |
| exam_id        | int          | YES  | MUL | NULL    |       |
+----------------+--------------+------+-----+---------+-------+
6 rows in set (1.01 sec)

mysql> select * from examination_entity;
+----------------+------------+--------+--------------+--------+---------+
| examination_id | date       | status | subject_name | sub_id | exam_id |
+----------------+------------+--------+--------------+--------+---------+
|              7 | 12/01/2023 | active | java         |   NULL |    NULL |
|             11 | 15/12/2022 | start  | basic java   |   NULL |    NULL |
|             12 | 15/12/2022 | start  | java         |   NULL |    NULL |
|             16 | 14/12/2022 | active | sql          |   NULL |    NULL |
|             21 | 08-09-2022 | Active | Anguler      |     20 |    NULL |
|             22 | 08-09-2022 | Active | HTML         |     20 |    NULL |
+----------------+------------+--------+--------------+--------+---------+
6 rows in set (0.18 sec)
```

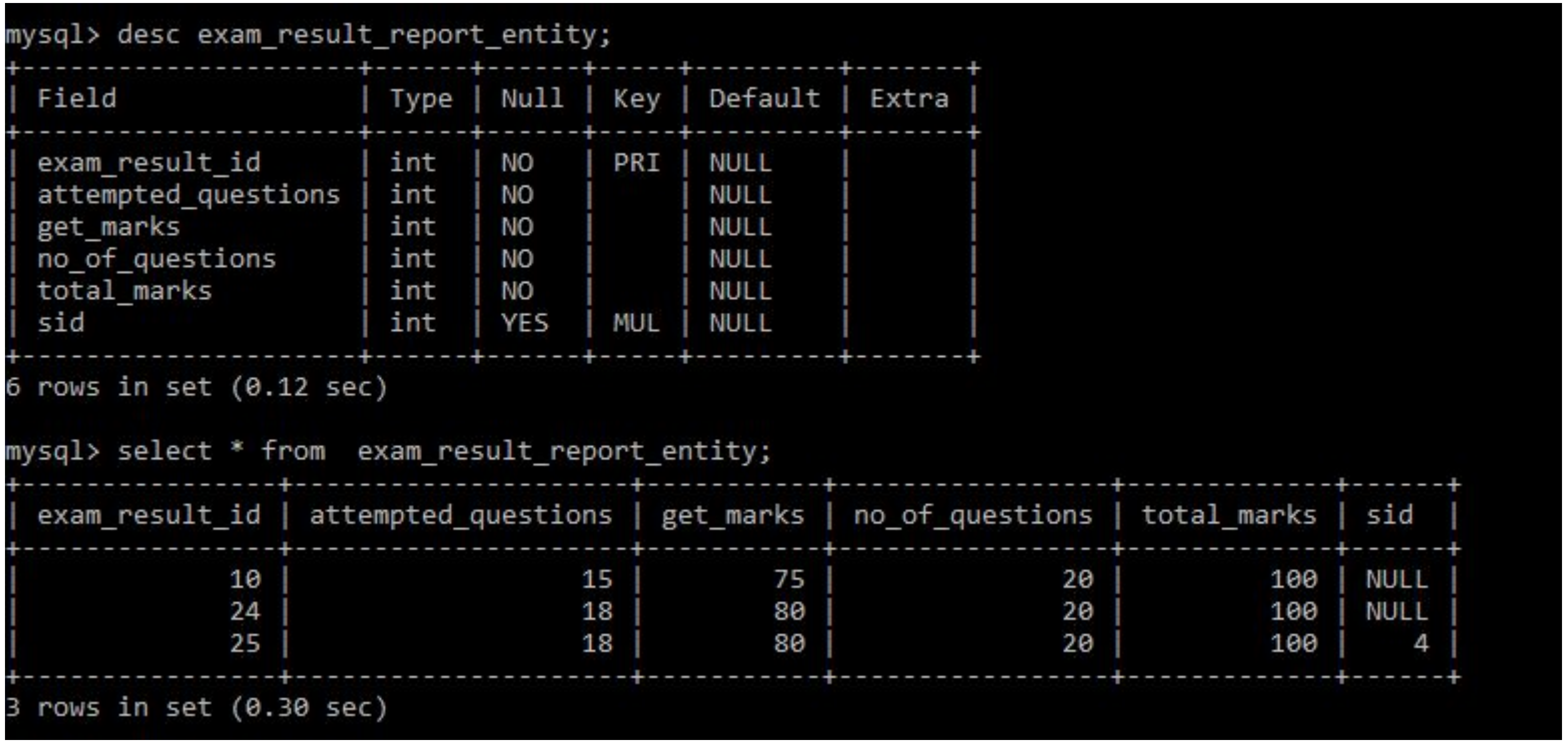
QUESTION TABLE



```
mysql> desc questions_entity;
+-------------+--------------+------+-----+---------+-------+
| Field       | Type         | Null | Key | Default | Extra |
+-------------+--------------+------+-----+---------+-------+
| question_id | int          | NO   | PRI | NULL    |       |
| answer      | varchar(255) | NO   |     | NULL    |       |
| option1     | varchar(255) | NO   |     | NULL    |       |
| option2     | varchar(255) | NO   |     | NULL    |       |
| option3     | varchar(255) | NO   |     | NULL    |       |
| option4     | varchar(255) | NO   |     | NULL    |       |
| question    | varchar(255) | NO   |     | NULL    |       |
| exam_id     | int          | YES  | MUL | NULL    |       |
+-------------+--------------+------+-----+---------+-------+
8 rows in set (0.01 sec)

mysql> select * from questions_entity;
+-------------+-------------------------------+----------------------------+-------------------------------+------------
-------------------+----------------------------+-------------------------------------+---------+
| question_id | answer                        | option1                    | option2                       | option3
                   | option4                    | question                            | exam_id |
+-------------+-------------------------------+----------------------------+-------------------------------+------------
-------------------+----------------------------+-------------------------------------+---------+
|           5 | platform independent language | object oriented language   | platform independent language | platform de
pendent language   | non of these               | what is java?                       |    NULL |
|           6 | 12                            | 1                          | 12                            | 4
                   | 8                          | how many containers are in one row? |    NULL |
|          23 | hyper text markup language    | hyper text markup language | hyper text market language    | hyper tectu
re markup language | hyper test markup language | what is html?                       |      22 |
+-------------+-------------------------------+----------------------------+-------------------------------+------------
-------------------+----------------------------+-------------------------------------+---------+
```

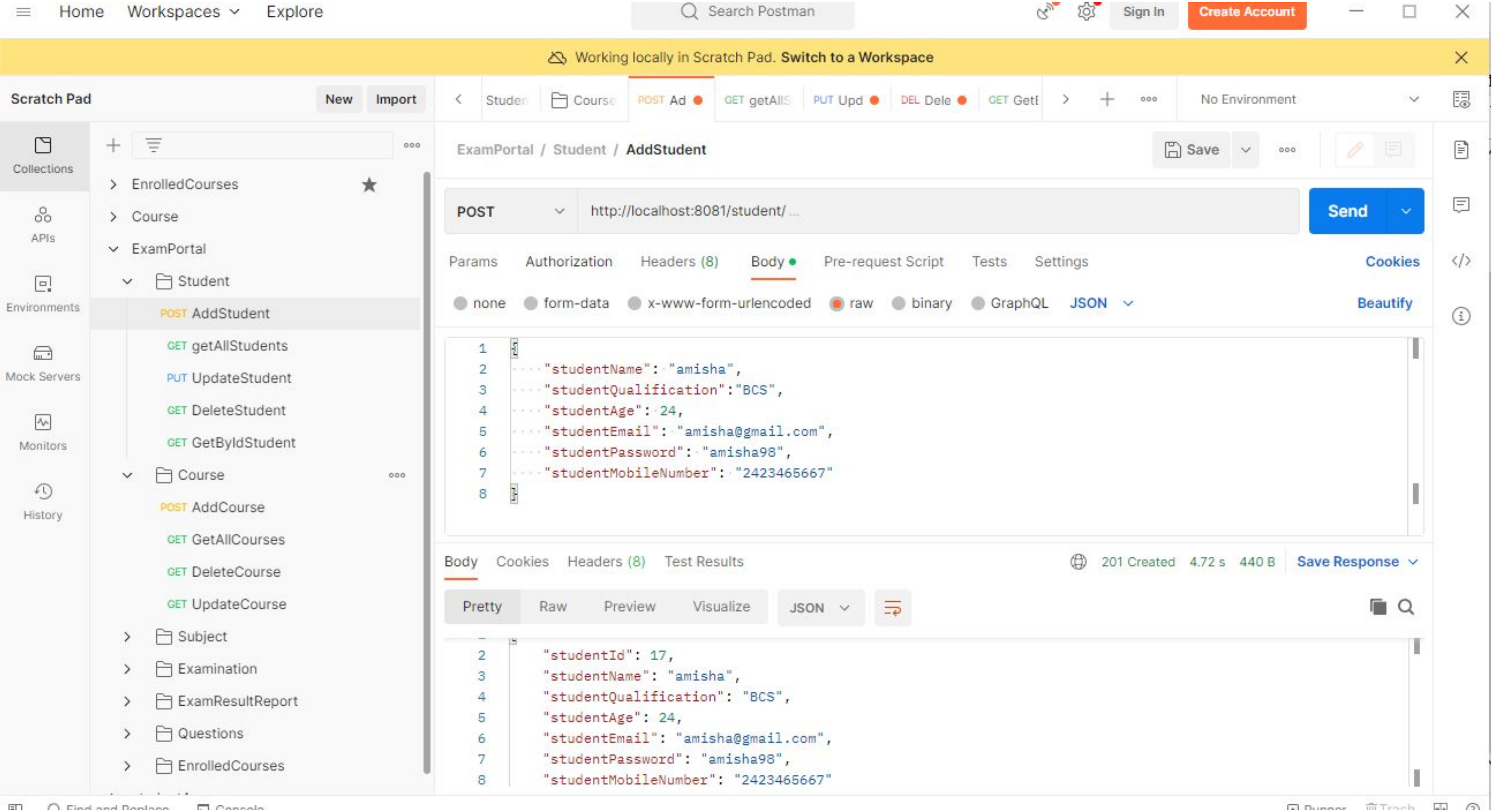
Exam Result Report



```
mysql> desc exam_result_report_entity;
+---------------------+------+------+-----+---------+-------+
| Field               | Type | Null | Key | Default | Extra |
+---------------------+------+------+-----+---------+-------+
| exam_result_id      | int  | NO   | PRI | NULL    |       |
| attempted_questions | int  | NO   |     | NULL    |       |
| get_marks           | int  | NO   |     | NULL    |       |
| no_of_questions     | int  | NO   |     | NULL    |       |
| total_marks         | int  | NO   |     | NULL    |       |
| sid                 | int  | YES  | MUL | NULL    |       |
+---------------------+------+------+-----+---------+-------+
6 rows in set (0.12 sec)

mysql> select * from  exam_result_report_entity;
+----------------+---------------------+-----------+-----------------+-------------+------+
| exam_result_id | attempted_questions | get_marks | no_of_questions | total_marks | sid  |
+----------------+---------------------+-----------+-----------------+-------------+------+
|             10 |                  15 |        75 |              20 |         100 | NULL |
|             24 |                  18 |        80 |              20 |         100 | NULL |
|             25 |                  18 |        80 |              20 |         100 |    4 |
+----------------+---------------------+-----------+-----------------+-------------+------+
3 rows in set (0.30 sec)
```

SCREENSHOTS FROM POSTMAN



POST http://localhost:8081/student/

Body ● raw JSON

```
{
    "studentName": "amisha",
    "studentQualification":"BCS",
    "studentAge": 24,
    "studentEmail": "amisha@gmail.com",
    "studentPassword": "amisha98",
    "studentMobileNumber": "2423465667"
}
```

201 Created 4.72 s 440 B

```
    "studentId": 17,
    "studentName": "amisha",
    "studentQualification": "BCS",
    "studentAge": 24,
    "studentEmail": "amisha@gmail.com",
    "studentPassword": "amisha98",
    "studentMobileNumber": "2423465667"
```

TESTING SCREENSHOTS

STUDENT REPOSITORY

COURSE REPOSITORY

CONCLUSION

In this project presentation, we explained the system design, requirements and implementation and the modules we have done till now. In accessing the outcome of this system, we have come to the final conclusion that Exam Portal System is more convenient for Exam Process .

REFERENCES

- www.tutorialspoint.com
- www.javatpoint.com
- www.wikipedia.com

THANK YOU