Designing and implementing a Relational DBMS on

"Student Result System Database"

In

Information Technology

by

Aditya Narendra Bachal

December 2019

CONTENTS

1. ABSTRACT

In this project I created one application which is easy to access and user friendly. For this application I used the backend as MySQL to store the data which is used in the application and for the user interface, we can use PHP and HTML but here I have created a database only. Student Result System Database can be used by education institutes to maintain the records of students easily. Achieving this objective is difficult using a manual system as the information is scattered, can be redundant and collecting relevant information may be very time consuming. All these problems are solved using this project.

2. INTRODUCTION

Student result database system is a system designed and engineered for colleges that need to manage results across multiple branches and students that need to track, manage and report results. This application can run on any kind of operating system. At a time, we can see all the years result in a single sheet and we can see the individual candidate's results separately. A database containing all the information such as DOB, Email-id, Class, Subjects, etc. can be very helpful. Suppose a student's academic record for the past year is to be checked. Then in such case, all one has to do is to enter the appropriate command for the database to extract the data of that particular student and display it. In these a proper database can be maintained and information can be extracted any time. Such systems have decreased the paper work and human effort required to maintain a piece of information.

1.1 Problem Statement.

To develop a database using MySQL which contains the data of students like Name, Student Id, etc. This system is capable of displaying the results of a particular student.

1.2 Motivation

DBMS provides a fair advantage over file systems in case of data redundancy, inconsistency, data sharing, data concurrency, data searching, data integrity and system crashing. Therefore, DBMS was chosen to create the student result system so that it can be easily accessible.

1.3 Objectives

To Develop a system that will manage:

- Information about various users
- Information about subjects offered in various semesters
- Marks obtained by students in different subjects
- Generation of reports

3. Data Types

- Integer: one optional sign character (+ or -) followed by at least one digit (0-9). Leading and trailing blanks are ignored. No other character is allowed.

- Varchar: it is used to store alpha numeric characters. In this data type we can set the maximum number of characters up to 8000 ranges by the default SQL server will set the size to 50 characters range.

- Date: the DATE data type accepts date values. No parameters are required when declaring a DATE data type. Date values should be supported in the form YYYY-MM-DD. However, point base will also accept single digit entries for month and day values.

- Time: the TIME data type accepts time values. No parameters are required while declaring a TIME data type. Data values should be specified in the format HH:MM:SS. An additional fractional value can be added to represent nanoseconds.

- Binary: it has a maximum length of 8000 bytes. It contains fixed length binary data.

- Var binary: it has a maximum length of 8000 bytes. It contains variable length binary data.

- Float: It is used to specify a floating-point value.

- Real: it specifies a single precision floating point number.

- Bit: It has the number of bits to store.

- Decimal: It specifies a numeric value that can have a decimal number.

- Numeric: it is used to specify a numeric value.

- Char: It contains fixed length non Unicode characters.

4. Data Modeling using ER Model:

3.1 REQUIREMENTS COLLECTION AND ANALYSIS

We list the data requirements for the database project here, and then create its conceptual schema step-by-step as we introduce the modeling concepts of the ER model. The Student result database keeps track of student's name, Id, roll Id, email, subjects, results in those particular subjects. Suppose that after the requirements collection and analysis phase, the database designers provide the following description of the mini world—the details of that student will be represented in the database.

The student result database is organized according to the attributes of students and their results in a particular subject . Each student has a unique Student Id and a unique roll number. We keep track of the results of a particular student i.e pass/fail, marks obtained, etc.

■ A student has some selected subjects each of which has a unique subject code, a subject name and the semester in which the examinations for that particular subject will take place. The result is finalized considering these subjects. For a student to display the result he/she must enter a unique student Id. After that the result will be displayed successfully.

■ We track a particular student's record by the parameters like Marks Obtained and Total marks for that subject.

3.1.1Entity Types, Entity Sets, Attributes, and Keys

1. An entity type Student with attributes Name, Roll no, Date of Birth, email_id, Gender and Student_id. Name, Gender and Date of Birth are the attributes. We can specify that both Student_id is a (separate) key attribute because it was specified to be unique.

2. An entity type Result with attributes id, Class_id, Student_id, Subject_id and Marks. Both Student_id and Subject_id are foreign key attributes.

3. An entity type Subject with attributes Subject_name, Subject_code and Subject_id. Subject_id is a (separate) Key attribute.

4. An entity type Class with attributes Class_id, Section and class_name. Class_id is a separate key attribute.

5. E-R Diagram

An entity–relationship model (ER model) describes inter-related things of interest in a specific domain of knowledge. An ER model is composed of entity types (which classify the things of interest) and specifies relationships that can exist between instances of those entity types. In software engineering an ER model is commonly formed to represent things that a business needs to remember in order to perform business processes. Consequently, the ER model becomes an abstract data model that defines a data or information structure that can be implemented in a database, typically a relational database. Entity–relationship modeling was developed for database design by Peter Chen and published in a 1976 paper. However, variants of the idea existed previously, some ER modelers show super and subtype entities connected by generalization-specialization relationships, and an ER model can be used also in the specification of domain-specific ontology. An ER model is typically implemented as a database. In a simple relational database implementation, each row of a table represents one instance of an entity type, and each field in a table represents an attribute type. In a relational database a relationship between entities is implemented by storing the primary key of one entity as a pointer or "foreign key" in the table of another entity There is a tradition for ER/data models to be built at two or three levels of abstraction. Note that the conceptual-logical-physical hierarchy below is used in other kinds of specification, and is different from the three schema approach to software engineering.

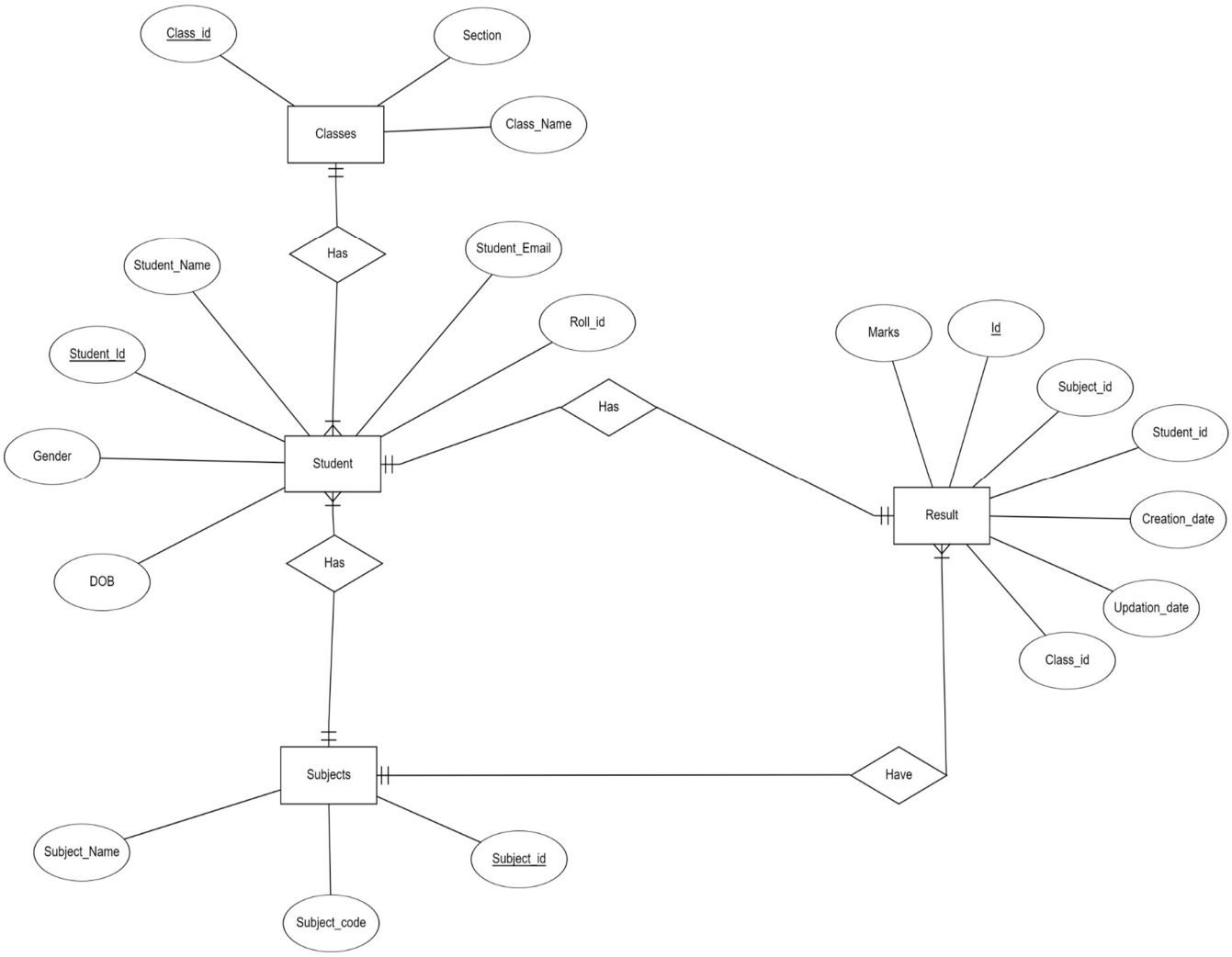

Fig 3.1 The ER conceptual schema diagram for the student result database.

6. Relational Database Design Using ER-to-Relational Mapping

The Student Er diagram which is fig 3.1 shows the relationship between entities and relationships.

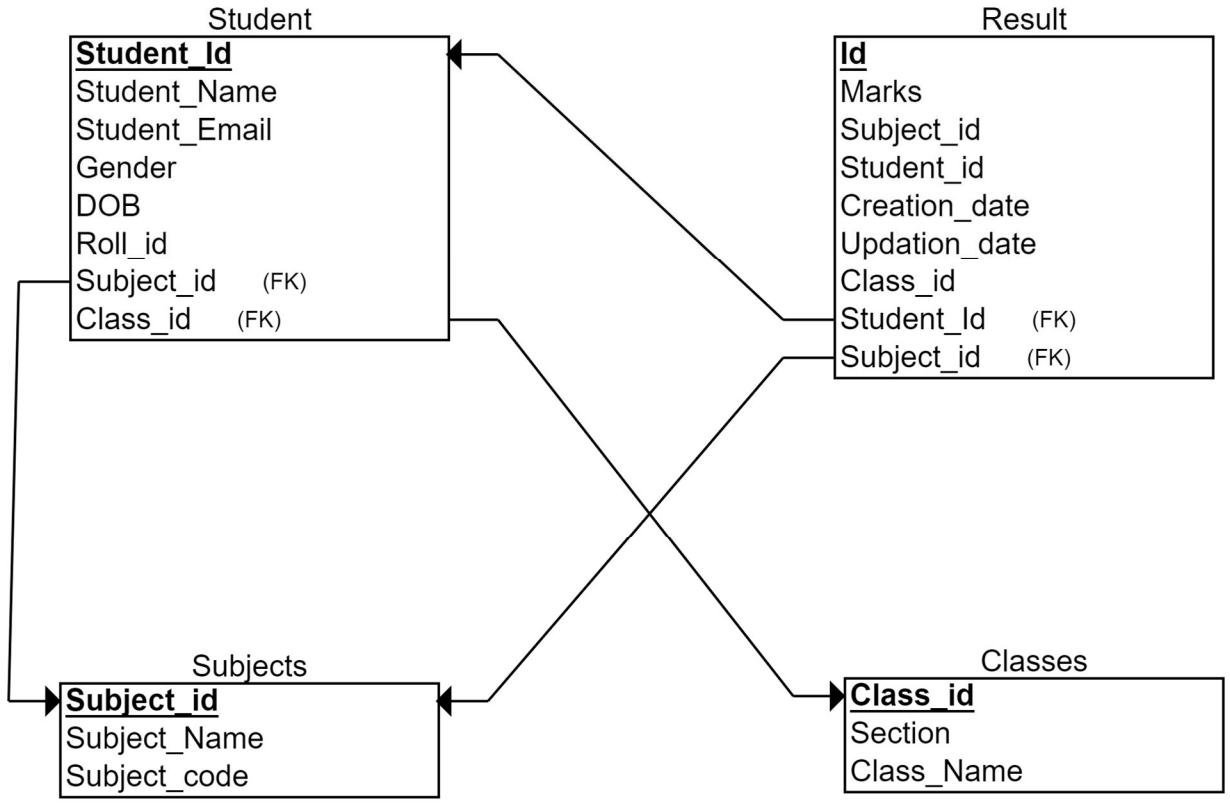

Fig: 3.2

7. ER-to-Relational Mapping Algorithm

The relational model constraints, which include primary keys, unique keys (if any), and referential integrity constraints on the relations, will also be specified in the mapping results.

Step 1: Mapping of Regular Entity Types:

For mapping the the regular entity types, we need to create a relation which will link the entity and its simple attributes. The relations HAS AND HAVE are created to correspond to the regular entity types which are STUDENT, CLASSES, RESULT and SUBJECTS. The foreign key and relationship attributes, if any, are not included yet; they will be added during subsequent steps. In our project we choose id, class_id, student_id and subject_id as our primary keys for RESULT, CLASS, STUDENT and SUBJECT.

Step 2: Mapping of Weak Entity Types:

There are no weak entities here as all the entities included in the database are strong entities and can be identified uniquely.

Step 3: Mapping of Binary 1:1 Relationship Types:

In our project we have used the following approach for mapping binary 1:1 relationship type:

Foreign key approach:

In our project we map 1:1 relationship type 'HAS' from Fig 3.1 by choosing the participating entity as 'STUDENT' to serve in the role of the relationship because its participation in the 'HAS' relationship type is total (each student has RESULT). We include the primary key of the STUDENT relation as foreign key in the RESULT relation and rename it as ID.

Step 4: Mapping of Binary 1:N Relationship Types.

In our project, we identify two 1:N relationships- 'STUDENT has SUBJECTS' and 'STUDENT has CLASSES'. In the first relationship, we have considered that a particular student has many subjects. Hence there exists a 1:N relationship between STUDENT and SUBJECTS. Subject_id is a foreign key in the STUDENT attribute. In the second relationship, we have considered that a student has many classes. Thus there exists a 1:N relationship between STUDENT and CLASSES. Class_id is a foreign key in STUDENT.

Step 5: Mapping of Binary M:N Relationship Types:

In our project, there are no M:N relationship types as most of the relationships defined in fig 3.1, are 1:1 and 1:N type relationships.

Step 6: Mapping of Multivalued Attributes:

In our project, we have mostly used simple attributes and there are no multivalued attributes as per the ER diagram which is fig 3.1.

8. CREATING DATABASE USING MYSQL

A schema for the student result management system was created. The elements include tables, constraints, views, domains and other structures that describe the schema. Here is the example of the statements used to create a schema and perform operations on it.

1. admin table

```
CREATE TABLE `admin` (
  `id` int(11) NOT NULL,
  `UserName` varchar(100) NOT NULL,
  `Password` varchar(100) NOT NULL,
)
```

insert

```
INSERT INTO `admin` (`id`, `UserName`, `Password`) VALUES
(1, 'admin'),
(2, 'saurabhL');
```

2. class table

```
CREATE TABLE `classes` (
  `id` int(11) NOT NULL,
  `ClassName` varchar(80) DEFAULT NULL,
  `ClassNameNumeric` int(4) NOT NULL,
  `Section` varchar(5) NOT NULL,
)
```

insert

```
INSERT INTO `classes` (`id`, `ClassName`, `ClassNameNumeric`, `Section`) VALUES
(1, 'First', 1, 'C'),
(2, 'Second', 2, 'A'),
(4, 'Fourth', 4, 'C'),
(5, 'Sixth', 6, 'A'),
(6, 'Sixth', 6, 'B'),
(7, 'Seventh', 7, 'B''),
(8, 'Eight', 8, 'A');
```

3. result table

```
CREATE TABLE `result` (
  `id` int(11) NOT NULL,
  `StudentId` int(11) DEFAULT NULL,
  `ClassId` int(11) DEFAULT NULL,
  `SubjectId` int(11) DEFAULT NULL,
  `marks` int(11) DEFAULT NULL,
)
```

insert

```
INSERT INTO `result` (`id`, `StudentId`, `ClassId`, `SubjectId`, `marks`) VALUES
(2, 1, 1, 2, 100),
(3, 1, 1, 1, 80),
(4, 1, 1, 5, 78),
(5, 1, 1, 4, 60),
(6, 2, 4, 2, 90),
(7, 2, 4, 1, 75),
(8, 2, 4, 5, 56),
(9, 2, 4, 4, 80),
(10, 4, 7, 2, 54),
(11, 4, 7, 1, 85),
(12, 4, 7, 5, 55),
(13, 4, 7, 7, 65),
(14, 5, 8, 2, 75),
(15, 5, 8, 1, 56),
(16, 5, 8, 5, 52),
(17, 5, 8, 4, 80);
```

4. student table

```
CREATE TABLE `students` (
  `StudentId` int(11) NOT NULL,
  `StudentName` varchar(100) NOT NULL,
  `RollId` varchar(100) NOT NULL,
  `StudentEmail` varchar(100) NOT NULL,
  `Gender` varchar(10) NOT NULL,
  `DOB` varchar(100) NOT NULL,
  `ClassId` int(11) NOT NULL,
```

)

insert

```
INSERT INTO `students` (`StudentId`, `StudentName`, `RollId`, `StudentEmail`,
`Gender`, `DOB`, `ClassId`) VALUES
(1, 'Abhishek', '46456', 'abhishek@gmail.com', 'Male', '1997-07-09', 1),
(2, 'Akshat', '10861', 'aks@gmail.com', 'Male', '1997-06-11', 4),
(3, 'Mohit', '2626', 'mohit@gmail.com', 'Male', '2014-08-06', 6),
(4, 'Divyanshu', '990', 'divyanshu@gmail.com', 'Male', '1997-02-03', 7),
(5, 'Deepankar', '122', 'eepankar@gmail.com', 'Male', '1997-02-03', 8);
```

5. combination table

```
CREATE TABLE `subjectcombination` (
  `id` int(11) NOT NULL,
  `ClassId` int(11) NOT NULL,
  `SubjectId` int(11) NOT NULL,
)
```

insert

```
INSERT INTO `subjectcombination` (`id`, `ClassId`, `SubjectId`) VALUES
(3, 2, 5),
(4, 1, 2),
(5, 1, 4),
(6, 1, 5),
(8, 4, 4),
(10, 4, 1),
(12, 4, 2),
(13, 4, 5),
(14, 6, 1),
(15, 6, 2),
(16, 6, 4),
(17, 6, 6),
(18, 7, 1),
(19, 7, 7),
(20, 7, 2),
(21, 7, 6),
(22, 7, 5),
(23, 8, 1),
```

```
(24, 8, 2),
(25, 8, 4),
(26, 8, 6),
(27, 8, 5);
```

6. subjects table

```
CREATE TABLE `subjects` (
  `id` int(11) NOT NULL,
  `SubjectName` varchar(100) NOT NULL,
  `SubjectCode` varchar(100) NOT NULL,
)
```

insert

```
INSERT INTO `subjects` (`id`, `SubjectName`, `SubjectCode`) VALUES
(1, 'Maths', 'MTH01'),
(2, 'English', 'ENG02'),
(4, 'Science', 'SC03'),
(5, 'Music', 'MS04'),
(6, 'Social Studies', 'SS05'),
(7, 'Physics', 'PH06''),
(8, 'Chemistry', 'CH07');
```

***** SET PRIMARY KEY**

```
ALTER TABLE `admin`
  ADD PRIMARY KEY (`id`);

ALTER TABLE `classes`
  ADD PRIMARY KEY (`id`);

ALTER TABLE `result`
  ADD PRIMARY KEY (`id`);

ALTER TABLE `students`
```

```
  ADD PRIMARY KEY (`StudentId`);


ALTER TABLE `subjectcombination`
  ADD PRIMARY KEY (`id`);


ALTER TABLE `subjects`
  ADD PRIMARY KEY (`id`);



*** AUTO_INCREMENT FOR TABLES


ALTER TABLE `admin`
  MODIFY `id` int(11) NOT NULL AUTO_INCREMENT, AUTO_INCREMENT=3;


ALTER TABLE `classes`
  MODIFY `id` int(11) NOT NULL AUTO_INCREMENT, AUTO_INCREMENT=9;


ALTER TABLE `result`
  MODIFY `id` int(11) NOT NULL AUTO_INCREMENT, AUTO_INCREMENT=18;


ALTER TABLE `students`
  MODIFY `StudentId` int(11) NOT NULL AUTO_INCREMENT,
AUTO_INCREMENT=6;


ALTER TABLE `subjectcombination`
  MODIFY `id` int(11) NOT NULL AUTO_INCREMENT, AUTO_INCREMENT=28;


ALTER TABLE `subjects`
  MODIFY `id` int(11) NOT NULL AUTO_INCREMENT, AUTO_INCREMENT=9;
COMMIT;
```

9. Test Queries (Minimum 25 Queries)

Here are 25 queries that were tested on this database:

A small description of the queries in mentioned above the snapshot of each query:

1. **What is the marks of Abhishek? /other students (provided name and class)**

```
MariaDB [student_result]> SELECT S.StudentId,S.StudentName,D.SubjectName,R.marks FROM students S INNER JOIN subjects
NNER JOIN result R ON S.StudentName =  'Abhishek' AND R.SubjectId = D.id AND R.ClassId = 1;
+-----------+-------------+-------------+-------+
| StudentId | StudentName | SubjectName | marks |
+-----------+-------------+-------------+-------+
|         1 | Abhishek    | English     |   100 |
|         1 | Abhishek    | Maths       |    80 |
|         1 | Abhishek    | Music       |    78 |
|         1 | Abhishek    | Science     |    60 |
+-----------+-------------+-------------+-------+
4 rows in set (0.001 sec)

MariaDB [student_result]>
```

2. **Write a Query to Display all the Entries in the Classes with B Section.**

```
MariaDB [student_result]> SELECT * FROM classes WHERE section = 'B';
+----+-----------+------------------+---------+---------------------+---------------------+
| id | ClassName | ClassNameNumeric | Section | CreationDate        | UpdationDate        |
+----+-----------+------------------+---------+---------------------+---------------------+
|  6 | Sixth     |                6 | B       | 2018-10-23 22:51:20 | 2018-10-23 22:51:20 |
|  7 | Seventh   |                7 | B       | 2018-10-23 22:51:20 | 2018-10-23 22:51:20 |
+----+-----------+------------------+---------+---------------------+---------------------+
2 rows in set (0.001 sec)

MariaDB [student_result]>
```

3. **Write a query to display all the students.**

```
MariaDB [student_result]> SELECT * FROM students;
+-----------+-------------+--------+----------------+--------+------------+---------+---------------------+----------
-------+--------+
| StudentId | StudentName | RollId | StudentEmail   | Gender | DOB        | ClassId | RegDate             | UpdationD
       | Status |
+-----------+-------------+--------+----------------+--------+------------+---------+---------------------+----------
-------+--------+
|         1 | Abhishek    | 46456  | abhi@gmail.com | Male   | 1997-07-09 |       1 | 2018-10-12 16:00:57 | 2018-10-2
0:14:37 |      1 |
|         2 | Aditya      | 10861  | adi@gmail.com  | Male   | 1997-06-11 |       4 | 2018-10-20 00:12:28 | 2018-10-2
0:16:16 |      0 |
|         3 | Kunal       | 2626   | kun@gmail.com  | Male   | 2014-08-06 |       6 | 2018-10-20 00:12:28 | 2018-10-2
0:18:34 |      1 |
|         4 | Ayush       | 990    | ayus@gmail.com | Male   | 1997-02-03 |       7 | 2018-10-20 00:12:28 | 2018-10-2
0:17:54 |      1 |
|         5 | Kaustubh    | 122    | kk@gmail.com   | Male   | 1997-02-03 |       8 | 2018-10-20 00:12:28 | 2018-10-2
0:17:21 |      1 |
+-----------+-------------+--------+----------------+--------+------------+---------+---------------------+----------
-------+--------+
5 rows in set (0.001 sec)

MariaDB [student_result]>
```

4. **Write a Query to Show all Subjects in Class 1**

```
MariaDB [student_result]> SELECT * FROM subjectcombination WHERE classid = '1';
+----+---------+-----------+--------+---------------------+---------------------+
| id | ClassId | SubjectId | status | CreationDate        | Updationdate        |
+----+---------+-----------+--------+---------------------+---------------------+
|  4 |       1 |         2 |      1 | 2017-06-12 12:16:32 | 2017-06-12 12:16:32 |
|  5 |       1 |         4 |      1 | 2017-06-12 12:16:35 | 2017-06-12 12:16:35 |
|  6 |       1 |         5 |      1 | 2017-06-12 12:16:40 | 2017-06-12 12:16:40 |
+----+---------+-----------+--------+---------------------+---------------------+
3 rows in set (0.008 sec)

MariaDB [student_result]>
```

5. **Write a query to show the student id and marks of students who scored below 50 for exams.**

```
MariaDB [student_result]> SELECT studentid, marks FROM result WHERE marks <= '50';
Empty set (0.001 sec)

MariaDB [student_result]>
```

6. **What is the Highest mark of Abhishek Choudhary compared to other students?**

```
MariaDB [student_result]> SELECT S.StudentId,S.StudentName,D.SubjectName,MAX(R.marks) AS marks FROM students S INNER JOI
N subjects D INNER JOIN result R ON S.StudentName = 'Abhishek Choudhary' AND R.SubjectId = D.id AND R.ClassId = 1;
+-----------+-------------+-------------+-------+
| StudentId | StudentName | SubjectName | marks |
+-----------+-------------+-------------+-------+
|      NULL | NULL        | NULL        |  NULL |
+-----------+-------------+-------------+-------+
1 row in set (0.015 sec)

MariaDB [student_result]>
```

7. **Write a Query to Display all the Entries in the result table**

```
MariaDB [student_result]>  SELECT * FROM result;
+----+-----------+---------+-----------+-------+---------------------+---------------------+
| id | StudentId | ClassId | SubjectId | marks | PostingDate         | UpdationDate        |
+----+-----------+---------+-----------+-------+---------------------+---------------------+
|  2 |         1 |       1 |         2 |   100 | 2017-08-24 23:24:09 | 2017-08-29 00:04:32 |
|  3 |         1 |       1 |         1 |    80 | 2017-08-24 23:24:09 | 2017-08-29 00:04:25 |
|  4 |         1 |       1 |         5 |    78 | 2017-08-24 23:24:09 | 2017-08-29 00:04:26 |
|  5 |         1 |       1 |         4 |    60 | 2017-08-24 23:24:09 | 2017-08-29 00:04:26 |
|  6 |         2 |       4 |         2 |    90 | 2017-08-29 00:08:18 | NULL                |
|  7 |         2 |       4 |         1 |    75 | 2017-08-29 00:08:18 | NULL                |
|  8 |         2 |       4 |         5 |    56 | 2017-08-29 00:08:18 | 2017-08-29 00:56:42 |
|  9 |         2 |       4 |         4 |    80 | 2017-08-29 00:08:18 | 2017-08-29 00:56:42 |
| 10 |         4 |       7 |         2 |    54 | 2017-08-29 00:26:21 | 2017-08-29 00:33:10 |
| 11 |         4 |       7 |         1 |    85 | 2017-08-29 00:26:21 | NULL                |
| 12 |         4 |       7 |         5 |    55 | 2017-08-29 00:26:21 | 2017-08-29 00:33:10 |
| 13 |         4 |       7 |         7 |    65 | 2017-08-29 00:26:21 | 2017-08-29 00:33:10 |
| 14 |         5 |       8 |         2 |    75 | 2017-08-29 00:55:07 | NULL                |
| 15 |         5 |       8 |         1 |    56 | 2017-08-29 00:55:07 | NULL                |
| 16 |         5 |       8 |         5 |    52 | 2017-08-29 00:55:07 | NULL                |
| 17 |         5 |       8 |         4 |    80 | 2017-08-29 00:55:07 | NULL                |
+----+-----------+---------+-----------+-------+---------------------+---------------------+
16 rows in set (0.000 sec)
```

8. **Write a Query to Show all Subjects in Class 2 along with the subject name and subject code**

```
MariaDB [student_result]> SELECT subjectname, subjectcode FROM subjects SB, subjectcombination SBC WHERE SBC.subjectid =
 SB.id AND SBC.classid = '2';
+-------------+-------------+
| subjectname | subjectcode |
+-------------+-------------+
| Music       | MS          |
+-------------+-------------+
1 row in set (0.014 sec)

MariaDB [student_result]>
```

9. **Write a Query to perform the Union operation to find out the total marks of all the students**

```
MariaDB [student_result]> SELECT StudentName FROM students UNION SELECT SUM(R.marks) AS marks FROM  result R;
+-------------+
| StudentName |
+-------------+
| Abhishek    |
| Aditya      |
| Kunal       |
| Ayush       |
| Kaustubh    |
| 1141        |
+-------------+
6 rows in set (0.002 sec)
```

10. Write a query to show the student id and marks of students who scored above 75 for exams

```
MariaDB [student_result]> SELECT studentid, marks FROM result WHERE marks >= '75';
+-----------+-------+
| studentid | marks |
+-----------+-------+
|         1 |   100 |
|         1 |    80 |
|         1 |    78 |
|         2 |    90 |
|         2 |    75 |
|         2 |    80 |
|         4 |    85 |
|         5 |    75 |
|         5 |    80 |
+-----------+-------+
9 rows in set (0.000 sec)

MariaDB [student_result]>
```

11. Write a Query to Display the total marks of Ayush

```
MariaDB [student_result]> use student_result;
Database changed
MariaDB [student_result]> SELECT S.StudentId,S.StudentName,D.SubjectName,SUM(R.marks) AS marks FROM students S INNER JOI
N subjects D INNER JOIN result R ON S.StudentName = 'Ayush' AND R.SubjectId = D.id AND R.ClassId = 1;
+-----------+-------------+-------------+-------+
| StudentId | StudentName | SubjectName | marks |
+-----------+-------------+-------------+-------+
|         4 | Ayush       | English     |   318 |
+-----------+-------------+-------------+-------+
1 row in set (0.019 sec)

MariaDB [student_result]>
```

12. Write a Query to Display the details of a the male students in class 2

```
MariaDB [student_result]> SELECT * FROM students where Classid = 2 AND gender = 'MALE';
Empty set (0.000 sec)

MariaDB [student_result]>
```

13. Write a Query to perform FULL OUTER JOIN

```
MariaDB [student_result]> SELECT * FROM students LEFT JOIN result ON students.StudentId= result.id UNION SELECT * FROM students RIGHT JOIN result ON students.StudentId=result.id;
+-----------+-------------+--------+---------------+--------+------------+---------+---------------------+---------------------+--------+------+-----------+---------+-----------+-------+---------------------+---
----------------+
| StudentId | StudentName | RollId | StudentEmail  | Gender | DOB        | ClassId | RegDate             | UpdationDate        | Status | id   | StudentId | ClassId | SubjectId | marks | PostingDate         | U
odationDate       |
+-----------+-------------+--------+---------------+--------+------------+---------+---------------------+---------------------+--------+------+-----------+---------+-----------+-------+---------------------+---
----------------+
|         1 | Abhishek    | 46456  | abhi@gmail.com | Male   | 1997-07-09 |       1 | 2018-10-12 16:00:57 | 2018-10-24 00:14:37 |      1 | NULL |      NULL |    NULL |      NULL |  NULL | NULL                | N
ULL               |
|         2 | Aditya      | 10861  | adi@gmail.com  | Male   | 1997-06-11 |       4 | 2018-10-20 00:12:28 | 2018-10-24 00:16:16 |      0 |    2 |         1 |       1 |         2 |   100 | 2017-08-24 23:24:09 | 2
017-08-29 00:04:32 |
|         3 | Kunal       | 2626   | kun@gmail.com  | Male   | 2014-08-06 |       6 | 2018-10-20 00:12:28 | 2018-10-24 00:18:34 |      1 |    3 |         1 |       1 |         1 |    80 | 2017-08-24 23:24:09 | 2
017-08-29 00:04:25 |
|         4 | Ayush       | 990    | ayus@gmail.com | Male   | 1997-02-03 |       7 | 2018-10-20 00:12:28 | 2018-10-24 00:17:54 |      1 |    4 |         1 |       1 |         5 |    78 | 2017-08-24 23:24:09 | 2
017-08-29 00:04:26 |
|         5 | Kaustubh    | 122    | kk@gmail.com   | Male   | 1997-02-03 |       8 | 2018-10-20 00:12:28 | 2018-10-24 00:17:21 |      1 |    5 |         1 |       1 |         4 |    60 | 2017-08-24 23:24:09 | 2
017-08-29 00:04:26 |
|      NULL | NULL        | NULL   | NULL          | NULL   | NULL       |    NULL | NULL                | NULL                |   NULL |    6 |         2 |       4 |         2 |    90 | 2017-08-29 00:08:18 | N
ULL               |
|      NULL | NULL        | NULL   | NULL          | NULL   | NULL       |    NULL | NULL                | NULL                |   NULL |    7 |         2 |       4 |         1 |    75 | 2017-08-29 00:08:18 | N
ULL               |
|      NULL | NULL        | NULL   | NULL          | NULL   | NULL       |    NULL | NULL                | NULL                |   NULL |    8 |         2 |       4 |         5 |    56 | 2017-08-29 00:08:18 | 2
017-08-29 00:56:42 |
|      NULL | NULL        | NULL   | NULL          | NULL   | NULL       |    NULL | NULL                | NULL                |   NULL |    9 |         2 |       4 |         4 |    80 | 2017-08-29 00:08:18 | 2
017-08-29 00:56:42 |
|      NULL | NULL        | NULL   | NULL          | NULL   | NULL       |    NULL | NULL                | NULL                |   NULL |   10 |         4 |       7 |         2 |    54 | 2017-08-29 00:26:21 | 2
017-08-29 00:33:10 |
|      NULL | NULL        | NULL   | NULL          | NULL   | NULL       |    NULL | NULL                | NULL                |   NULL |   11 |         4 |       7 |         1 |    85 | 2017-08-29 00:26:21 | N
ULL               |
|      NULL | NULL        | NULL   | NULL          | NULL   | NULL       |    NULL | NULL                | NULL                |   NULL |   12 |         4 |       7 |         5 |    55 | 2017-08-29 00:26:21 | 2
017-08-29 00:33:10 |
|      NULL | NULL        | NULL   | NULL          | NULL   | NULL       |    NULL | NULL                | NULL                |   NULL |   13 |         4 |       7 |         7 |    65 | 2017-08-29 00:26:21 | 2
017-08-29 00:33:10 |
|      NULL | NULL        | NULL   | NULL          | NULL   | NULL       |    NULL | NULL                | NULL                |   NULL |   14 |         5 |       8 |         2 |    75 | 2017-08-29 00:55:07 | N
ULL               |
|      NULL | NULL        | NULL   | NULL          | NULL   | NULL       |    NULL | NULL                | NULL                |   NULL |   15 |         5 |       8 |         1 |    56 | 2017-08-29 00:55:07 | N
ULL               |
|      NULL | NULL        | NULL   | NULL          | NULL   | NULL       |    NULL | NULL                | NULL                |   NULL |   16 |         5 |       8 |         5 |    52 | 2017-08-29 00:55:07 | N
ULL               |
|      NULL | NULL        | NULL   | NULL          | NULL   | NULL       |    NULL | NULL                | NULL                |   NULL |   17 |         5 |       8 |         4 |    80 | 2017-08-29 00:55:07 | N
ULL               |
+-----------+-------------+--------+---------------+--------+------------+---------+---------------------+---------------------+--------+------+-----------+---------+-----------+-------+---------------------+---
----------------+
17 rows in set (0.004 sec)
```

14. What is the Lowest mark of Ayush? other students (provided name and class)

```
MariaDB [student_result]> SELECT S.StudentName,S.StudentId, D.SubjectName,MIN(R.marks) AS marks FROM students S INN
ER JOIN subjects D INNER JOIN result R ON S.StudentName = 'Ayush' AND R.SubjectId= D.id AND R.ClassId = 1;
+-------------+-----------+-------------+-------+
| StudentName | StudentId | SubjectName | marks |
+-------------+-----------+-------------+-------+
| Ayush       |         4 | English     |    60 |
+-------------+-----------+-------------+-------+
1 row in set (0.001 sec)
```

15. Write a Query to perform RIGHT JOIN on the attributes StudentName and Section

```
MariaDB [student_result]> SELECT StudentName FROM students RIGHT JOIN classes ON students.StudentName=classes.Section;
+-------------+
| StudentName |
+-------------+
| NULL        |
| NULL        |
| NULL        |
| NULL        |
| NULL        |
| NULL        |
| NULL        |
+-------------+
7 rows in set (0.001 sec)
```

16. Write a query to display the name, DOB and email of the students who have opted the subject Science.

```
MariaDB [student_result]> SELECT S.StudentName, S.StudentEmail, S.DOB, D.SubjectName FROM students S INNER JOIN subjec
ts D ON D.SubjectName = 'Science';
+-------------+----------------+------------+-------------+
| StudentName | StudentEmail   | DOB        | SubjectName |
+-------------+----------------+------------+-------------+
| Abhishek    | abhi@gmail.com | 1997-07-09 | Science     |
| Aditya      | adi@gmail.com  | 1997-06-11 | Science     |
| Kunal       | kun@gmail.com  | 2014-08-06 | Science     |
| Ayush       | ayus@gmail.com | 1997-02-03 | Science     |
| Kaustubh    | kk@gmail.com   | 1997-02-03 | Science     |
+-------------+----------------+------------+-------------+
5 rows in set (0.000 sec)
```

17. Write a query using the GROUP BY to group StudentName.

```
MariaDB [student_result]> SELECT COUNT(StudentId), StudentName FROM students GROUP BY StudentName;
+------------------+-------------+
| COUNT(StudentId) | StudentName |
+------------------+-------------+
|                1 | Abhishek    |
|                1 | Aditya      |
|                1 | Ayush       |
|                1 | Kaustubh    |
|                1 | Kunal       |
+------------------+-------------+
5 rows in set (0.003 sec)
```

18. What is the Highest mark of Kaustubh? other students (provided name and class)

```
MariaDB [student_result]> SELECT S.StudentId,S.StudentName,D.SubjectName,MAX(R.marks) AS marks FROM students S INNE
R JOIN subjects D INNER JOIN result R ON S.StudentName = 'Kaustubh' AND R.SubjectId = D.id AND R.ClassId = 1;
+-----------+-------------+-------------+-------+
| StudentId | StudentName | SubjectName | marks |
+-----------+-------------+-------------+-------+
|         5 | Kaustubh    | English     |   100 |
+-----------+-------------+-------------+-------+
1 row in set (0.012 sec)
```

19. What is the marks of Ayush? /other students (provided name and class)

```
MariaDB [student_result]> SELECT S.StudentId,S.StudentName,D.SubjectName,R.marks FROM students S INNER JOIN subject
s D INNER JOIN result R ON S.StudentName =  'Ayush' AND R.SubjectId = D.id AND R.ClassId = 1;
+-----------+-------------+-------------+-------+
| StudentId | StudentName | SubjectName | marks |
+-----------+-------------+-------------+-------+
|         4 | Ayush       | English     |   100 |
|         4 | Ayush       | Maths       |    80 |
|         4 | Ayush       | Music       |    78 |
|         4 | Ayush       | Science     |    60 |
+-----------+-------------+-------------+-------+
4 rows in set (0.000 sec)
```

20. Write a query to perform LEFT JOIN on attributes StudentId and Marks

```
MariaDB [student_result]> SELECT marks FROM result LEFT JOIN students ON result.marks=students.StudentId;
+-------+
| marks |
+-------+
|   100 |
|    80 |
|    78 |
|    60 |
|    90 |
|    75 |
|    56 |
|    80 |
|    54 |
|    85 |
|    55 |
|    65 |
|    75 |
|    56 |
|    52 |
|    80 |
+-------+
16 rows in set (0.003 sec)
```

21. Write a query to display the average marks of Aditya

```
MariaDB [student_result]> SELECT S.StudentId,S.StudentName,D.SubjectName,AVG(R.marks) AS marks FROM students S INNER JOI
N subjects D INNER JOIN result R ON S.StudentName = 'Aditya' AND R.SubjectId = D.id AND R.ClassId = 1;
+-----------+-------------+-------------+---------+
| StudentId | StudentName | SubjectName | marks   |
+-----------+-------------+-------------+---------+
|         2 | Aditya      | English     | 79.5000 |
+-----------+-------------+-------------+---------+
1 row in set (0.001 sec)

MariaDB [student_result]>
```

22. Write a query to display the count of marks in English for Kaustubh.

```
MariaDB [student_result]> SELECT S.StudentId,S.StudentName,D.SubjectName,COUNT(R.marks) AS marks FROM students S INNER J
OIN subjects D INNER JOIN result R ON S.StudentName = 'Kaustubh' AND R.SubjectId = D.id AND R.ClassId = 1;
+-----------+-------------+-------------+-------+
| StudentId | StudentName | SubjectName | marks |
+-----------+-------------+-------------+-------+
|         5 | Kaustubh    | English     |     4 |
+-----------+-------------+-------------+-------+
1 row in set (0.001 sec)
```

23. Write a nested query to select the StudentName with ClassId=1

```
MariaDB [student_result]> SELECT StudentName FROM students WHERE StudentId IN (SELECT ClassId FROM classes WHERE classid
 =1);
+-------------+
| StudentName |
+-------------+
| Abhishek    |
+-------------+
1 row in set (0.002 sec)
```

24. Write a query to perform RIGHT JOIN on the attributes StudentName and Marks.

```
MariaDB [student_result]> SELECT StudentName FROM students RIGHT JOIN result ON students.StudentName = result.StudentId;

+-------------+
| StudentName |
+-------------+
| NULL        |
| NULL        |
| NULL        |
| NULL        |
| NULL        |
| NULL        |
| NULL        |
| NULL        |
| NULL        |
| NULL        |
| NULL        |
| NULL        |
| NULL        |
| NULL        |
| NULL        |
| NULL        |
+-------------+
16 rows in set, 80 warnings (0.000 sec)
```

25. Write a query to display the average marks of Aditya above 50

```
MariaDB [(none)]> use student_result;
Database changed
MariaDB [student_result]> SELECT S.StudentName, S.StudentId, AVG(R.marks) AS marks FROM students S INNER JOIN result R O
N S.StudentName = 'Aditya' WHERE marks>=50;
+-------------+-----------+---------+
| StudentName | StudentId | marks   |
+-------------+-----------+---------+
| Aditya      |         2 | 71.3125 |
+-------------+-----------+---------+
1 row in set (0.001 sec)
```

10. Conclusion:

A database for student results was created using MySQL. Some information was added in the database with the appropriate entities and attributes. Various operations were performed on the database. 25 different queries were tested. All the 25 queries have successfully displayed output as per the snapshots uploaded above this section. Hence a student result system was successfully created and tested using MySQL. This system can be used to organize data in a systematic way and interpret data as per the user's needs.

11. References.

https://db.grussell.org/section006.html

https://app.creately.com

https://www.javatpoint.com/sql-select-from-multiple-tables