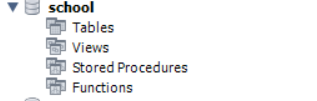
SQL LAB 2

Lab 1. Create a Database & Table Using MySQL Command-Line Client.

* Create a database with the name School.

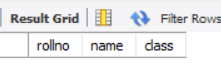
create database School;

use School;



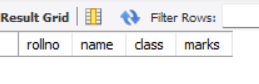
* Create a table with the name Student

create table student(rollno varchar(10), name varchar(5), class varchar(5));



* Alter table

alter table student add marks int;



* Insert data into table

INSERT INTO student (rollNo, name, class, marks) VALUES

(111, 'Ruchi', '10A', 85),

(112, 'Amit', '10B', 78),

(113, 'Neha', '10A', 92),

(114, 'Rohan', '10C', 67),

(115, 'Priya', '10B', 88),

(116, 'Suresh', '10A', 74),

(117, 'Meera', '10C', 90),

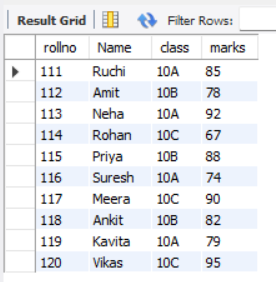
(118, 'Ankit', '10B', 82),

(119, 'Kavita', '10A', 79),

(120, 'Vikas', '10C', 95);

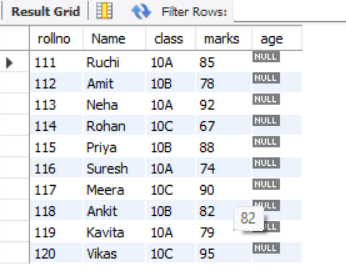
* Show table

Select \* from Student;



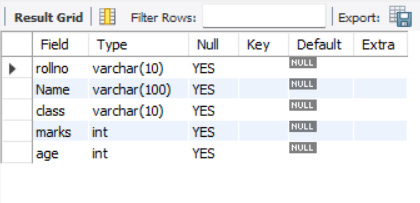
# 1. Add a new column age (integer) to the student table.

ALTER TABLE student ADD age INT;



# 2. Modify the class column in the student table to have a size of 10 characters.

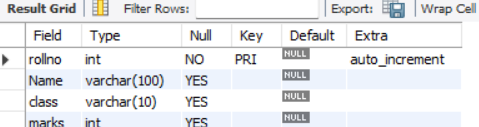
ALTER TABLE student MODIFY class VARCHAR(10);



# 4. Change the data type of rollno to INT and set it as an AUTO\_INCREMENT primary key.

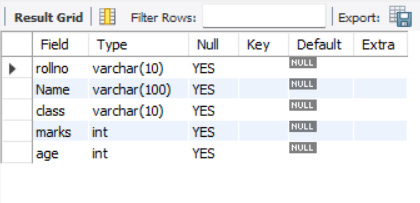
ALTER TABLE student modify rollno int AUTO\_INCREMENT;

ALTER TABLE student ADD PRIMARY KEY (rollno);



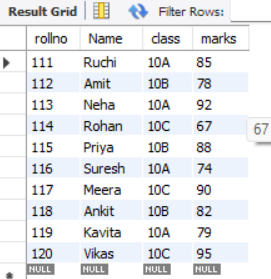
# 3. Remove the primary key constraint from the student table.

ALTER TABLE student DROP PRIMARY KEY;



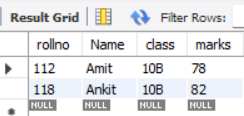
# 5. Drop the column age from the student table.

ALTER TABLE student DROP age;



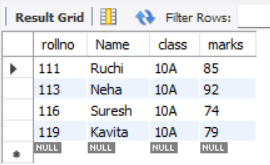
# 6. Retrieve all students whose names start with the letter 'A'.

SELECT \* FROM student WHERE name LIKE 'A%';



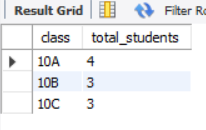
# 7. Display all students who belong to class '10A'.

SELECT \* FROM student WHERE class = '10A';



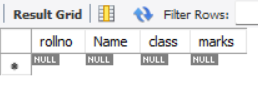
# 8. Fetch the total number of students in each class.

SELECT class, COUNT(\*) AS total\_students FROM student GROUP BY class;



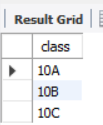
# 9. Select students whose roll number contains '123' anywhere in it.

SELECT \* FROM student WHERE rollno LIKE '%123%';



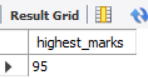
# 10. Retrieve distinct class names from the student table.

SELECT DISTINCT class FROM student;



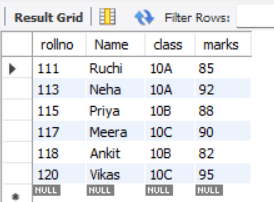
# 11. Retrieve the highest marks obtained by any student.

SELECT MAX(marks) AS highest\_marks FROM student;



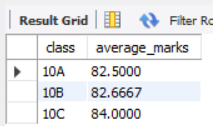
# 12. List the students who scored more than 80 marks.

SELECT \* FROM student WHERE marks > 80;



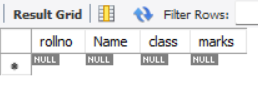
# 13. Find the average marks of students in each class.

SELECT class, AVG(marks) AS average\_marks FROM student GROUP BY class;



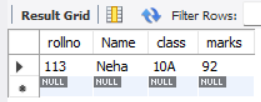
# 14. Retrieve the details of students who do not have marks recorded in the marks table.

SELECT \* FROM student WHERE marks IS NULL;



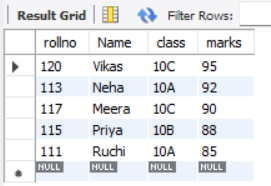
# 15. Find the students who have the second-highest marks.

SELECT \* FROM student WHERE marks = (SELECT MAX(marks) FROM student WHERE marks < (SELECT MAX(marks) FROM student));



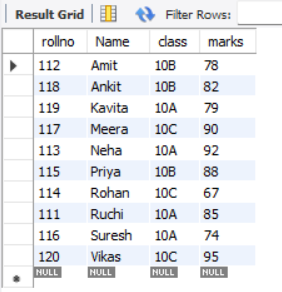
# 16. Retrieve the top 5 students based on their marks.

SELECT \* FROM student ORDER BY marks DESC LIMIT 5;



# 17. Display students sorted by their name in ascending order and class in descending order.

SELECT \* FROM student ORDER BY name ASC, class DESC;



commit;