**Presenting**

**Project**

**On**

**Foundation of SQL**

**By :**

**Shubhangi harichandra Salunkhe**

**Student at IT Vedant**

**(master’s in business data analytics)**

**Email:** [**shubhangisalunkhe32@gmail.com**](mailto:shubhangisalunkhe32@gmail.com)

**19 november 2022**

|  |  |
| --- | --- |
| **Sr.no** | **Topic** |
| **1** | **Abstract** |
| **2** | **Aim and objectives** |
| **3** | **Introduction** |
| **4** | **Difference between SQL and MYSQL** |
| **5** | **Main table (student and student\_result)** |
| **6** | **Queries** |
| **7** | **Conclusion** |

**Index**

**Abstract**

**In this project I have take two tables (student) in that i shown some students details like roll\_no, name, address, phone\_no, age.**

**Second table (student\_result) I have shown students result details like students\_id, subject, marks, and roll\_no.**

**Aim**

**To give you practical experience in how to create database, how to insert values, how to perform different queries, normalization and writing SQL statements to query a relational database**

**Objectives**

**The main goal of this project is to give the students a first hands experience on using DBMS to create and manipulate a small database. By completing this project, students should be able to create database systems, rite SQL queries, interact with a particular DBMS and manipulate the created database.**

**Introduction to SQL**

SQL is a standard language for accessing and manipulating databases.

QL (Structured Query Language) is used to perform operations on the records stored in the database, such as updating records, inserting records, deleting records, creating and modifying database tables, views, etc.

SQL is not a database system, but it is a query language.

Suppose you want to perform the queries of SQL language on the stored data in the database. You are required to install any database management system in your systems, for example, [Oracle](https://www.javatpoint.com/oracle-tutorial), [MySQL](https://www.javatpoint.com/mysql-tutorial), [MongoDB](https://www.javatpoint.com/mongodb-tutorial), [PostgreSQL](https://www.javatpoint.com/postgresql-tutorial), [SQL Server](https://www.javatpoint.com/sql-server-tutorial), [DB2](https://www.javatpoint.com/db2-tutorial), etc.

**What is SQL?**

* SQL stands for Structured Query Language
* SQL lets you access and manipulate databases
* SQL became a standard of the American National Standards Institute (ANSI) in 1986, and of the International Organization for Standardization (ISO) in 1987

## **What is SQL?**

SQL is a short-form of the structured query language, and it is pronounced as S-Q-L or sometimes as See-Quell.

This database language is mainly designed for maintaining the data in relational database management systems. It is a special tool used by data professionals for handling structured data (data which is stored in the form of tables). It is also designed for stream processing in RDSMS.

You can easily create and manipulate the database, access and modify the table rows and columns, etc. This query language became the standard of ANSI in the year of 1986 and ISO in the year of 1987.

If you want to get a job in the field of data science, then it is the most important query language to learn. Big enterprises like Facebook, Instagram, and LinkedIn, use SQL for storing the data in the back-end.

## **What Can SQL do?**

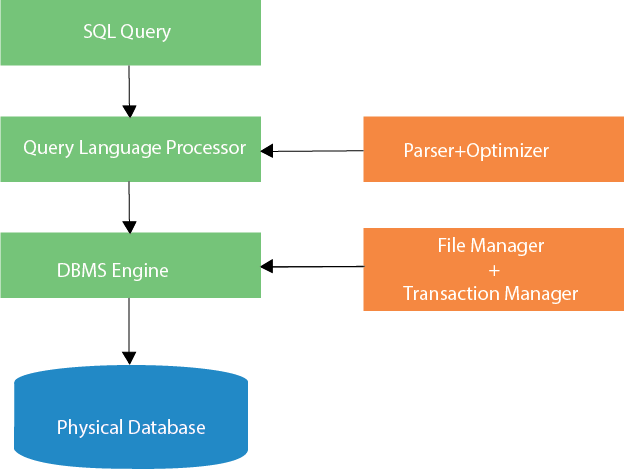
* SQL can execute queries against a database
* SQL can retrieve data from a database
* SQL can insert records in a database
* SQL can update records in a database
* SQL can delete records from a database
* SQL can create new databases
* SQL can create new tables in a database
* SQL can create stored procedures in a database
* SQL can create views in a database
* SQL can set permissions on tables, procedures, and views

## **What is Data?**

Data is a collection of a distinct small unit of information. It can be used in a variety of forms like text, numbers, media, bytes, etc. it can be stored in pieces of paper or electronic memory, etc.

## **What is Database?**

A **database** is an organized collection of data, so that it can be easily accessed and managed. You can organize data into tables, rows, columns, and index it to make it easier to find relevant information. The **main purpose** of the database is to operate a large amount of information by storing, retrieving, and managing data.



|  |  |  |
| --- | --- | --- |
| **Parameters** | **SQL** | **MySQL** |
| **Definition** | SQL is a domain-specific query language. | MySQL is an RDBMS that uses SQL to retrieve data from the database. |
| **Operations** | Helps retrieve and manipulate stored data in a database by performing different operations | Stores the existing data in separate tables in an organized manner |
| **Functions** | Used to write queries for databases | Facilitates data storing, modification, and management in a tabular format |
| **Format and Syntax** | Follows a standard format, and the basic syntax and commands remain the same | MySQL is updated frequently |
| **Support** | Apache Spark Connector for SQL does not come with any Microsoft support | Comes with MySQL workbench as an integrated tool that helps in designing and building databases |
| **Storage Engine** | Supports a single storage engine | Supports multiple and pluggable storage engines that make it more flexible |
| **Server and Database** | The server remains independent of the database in SQL, which means that you can perform other operations on the database during a data backup session. | The server blocks the database and you cannot perform other operations on the database during a data backup session, thus minimizing data corruption when switching from one MySQL version to another. |
| **Data Security** | External processes or third-party apps do have access to and cannot manipulate data directly | Database files can be easily manipulated or modified by using binaries during the run time |
| **Community Support** | Not being an open-source language, SQL does not come with community support | Offers rich and robust community support because it is an open-source platform |

**Difference between SQL and MYSQL**

Student

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Roll\_no** | **Name** | **Address** | **Phone\_no** | **Age** |
| 1 | Sanvi shinde | dombivali | 1024598765 | 20 |
| 2 | Suraj chikne | andheri | 3215694258 | 22 |
| 3 | Pooja patil | borivali | 2557846359 | 21 |
| 4 | Ovi jadhav | Grant road | 5487612005 | 19 |
| 5 | Sachin salunkhe | kalyan | 1024563089 | 20 |
| 6 | Pranav naik | vashi | 1024598765 | 20 |
| 7 | Tina dutta | kharghar | 1024578913 | 20 |
| 8 | Raj mahadik | andheri | 3654875589 | 21 |
| 9 | Nurvi raut | nerul | 3324587716 | 21 |
| 10 | Rudra singh | nerul | 2254887741 | 20 |
| 11 | Shvani shah | kharghar | 5455556981 | 18 |
| 12 | Janhavi joshi | thane | 2254896369 | 18 |

Student\_result

|  |  |  |  |
| --- | --- | --- | --- |
| **Student\_id** | **Subject** | **Marks** | **Roll\_no** |
| 1001 | maths | 45 | 1 |
| 1002 | hindi | 43 | 2 |
| 1003 | science | 30 | 3 |
| 1004 | marathi | 45 | 4 |
| 1005 | hindi | 49 | 5 |
| 1006 | Social science | 45 | 6 |
| 1007 | science | 43 | 7 |
| 1008 | english | 47 | 8 |
| 1009 | science | 40 | 9 |

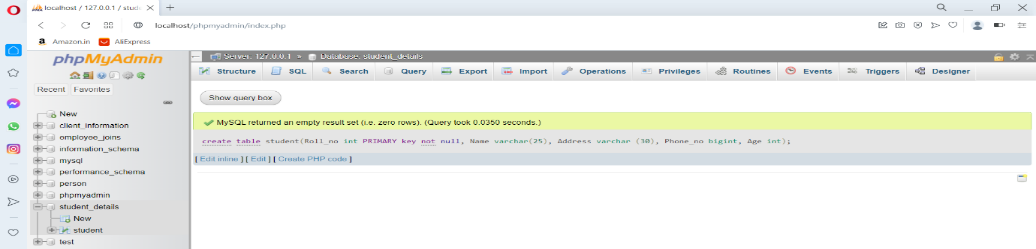
1. Create a new database

Syntax: create database student\_details;

1. Create a table

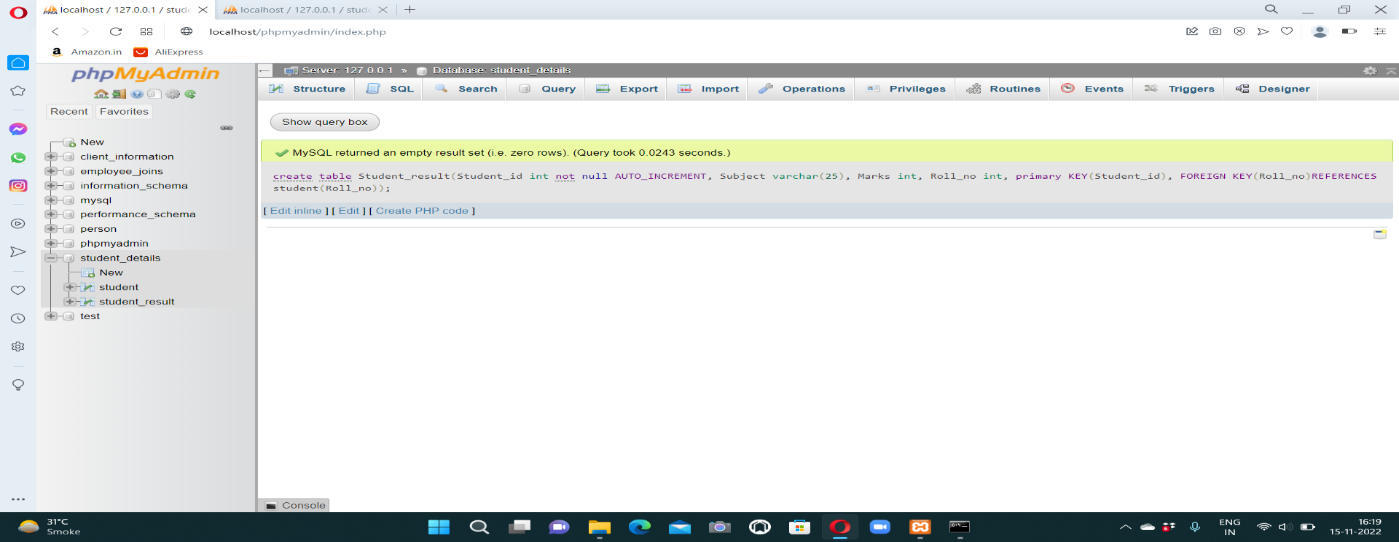
* Table 1 – student

Syntax: create table student(roll\_no int not null primary key, name varchar(25), address varchar(25), phone\_no bigint, age int);



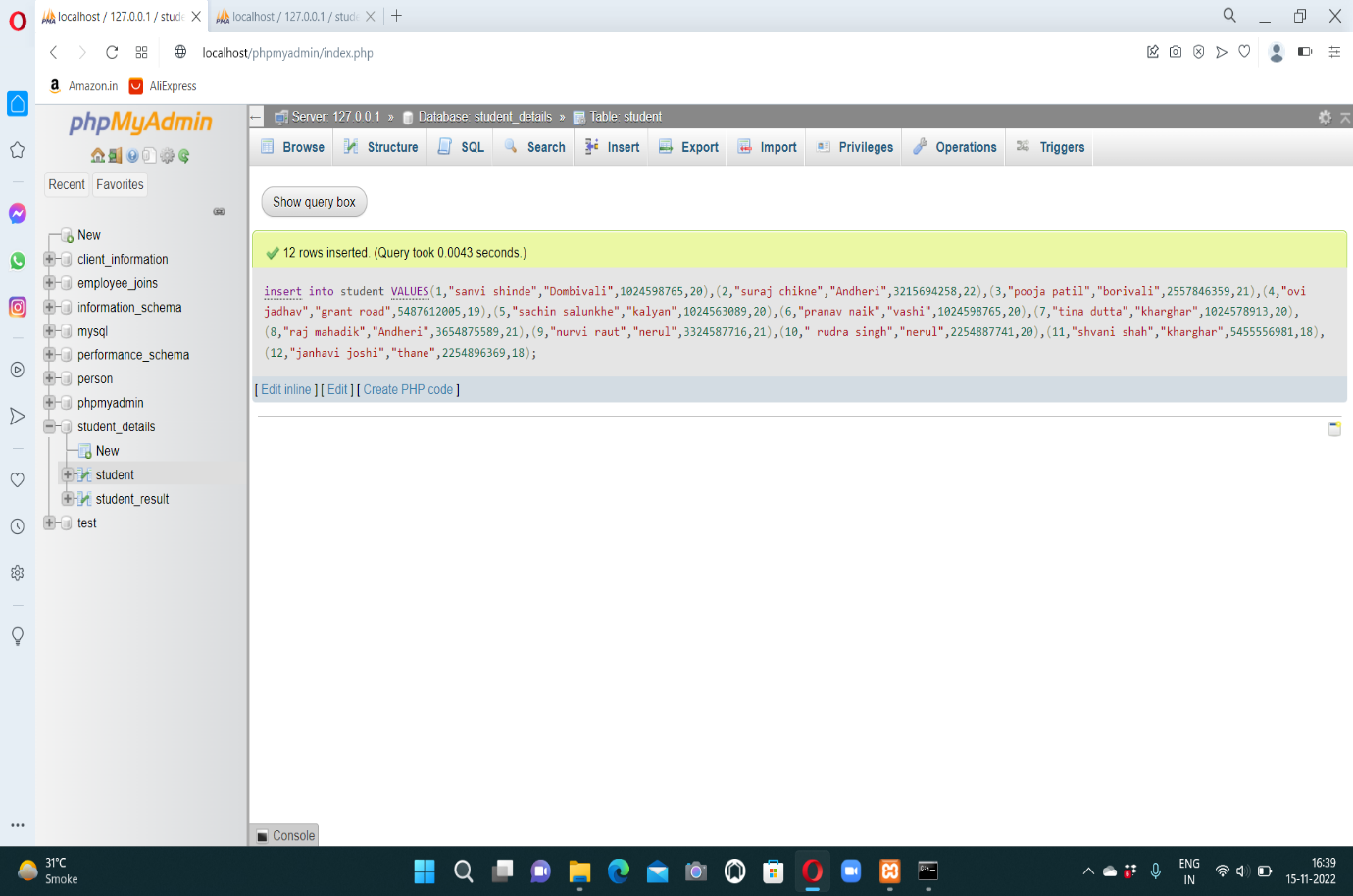
* Table 2- Student\_result

Syntax: create table student\_result(student\_id int not null auto\_increment, subject varchar(25), marks int primary key(student\_id), foreign key(roll\_no) refrences student(roll\_no));



1. Insert into

* Table 1- Syntax : insert into student values(value1,value2,………………… ………………………value10);



**Queries**

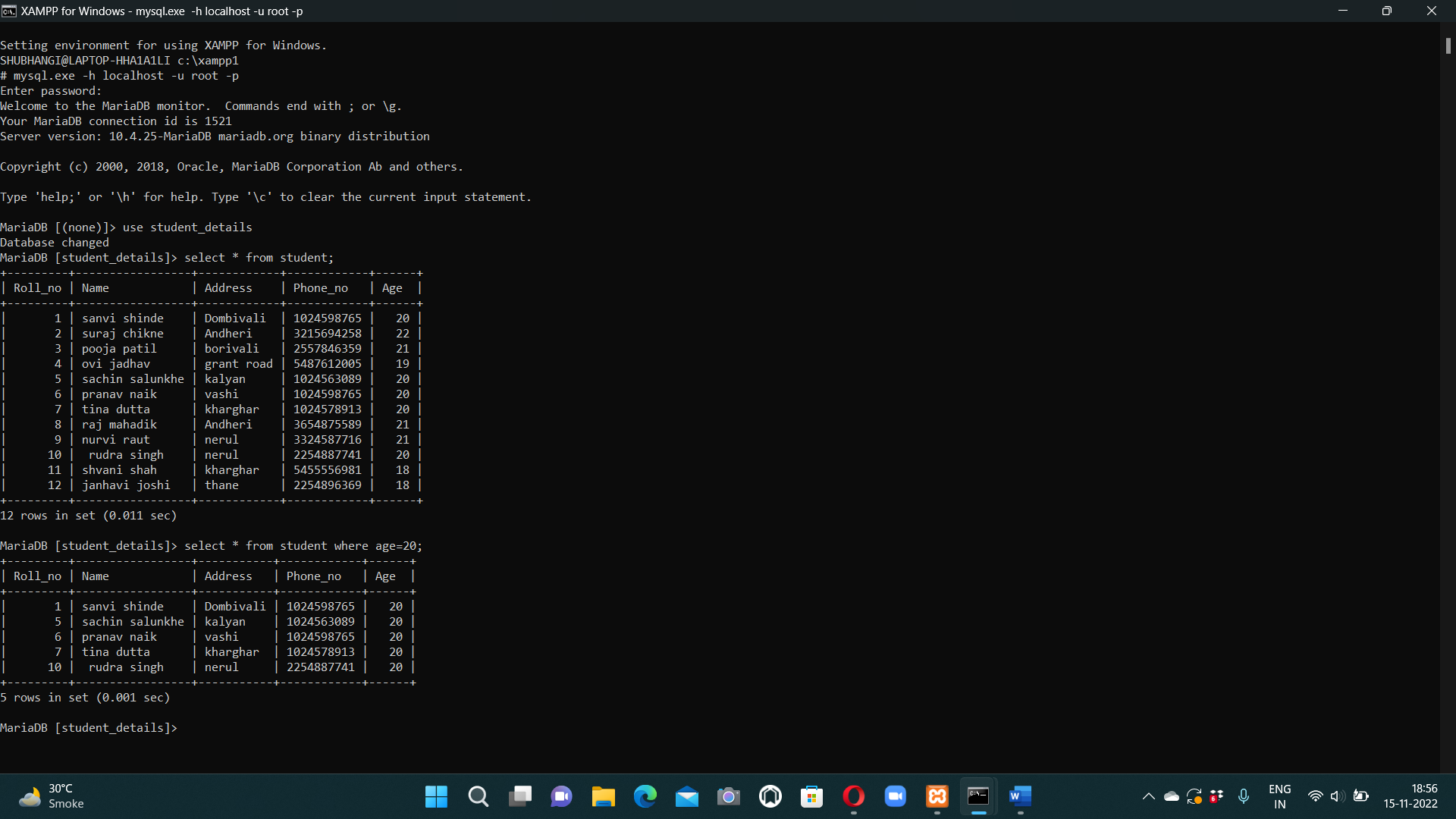
1. Fetch data from table

Syntax : select \* from student;



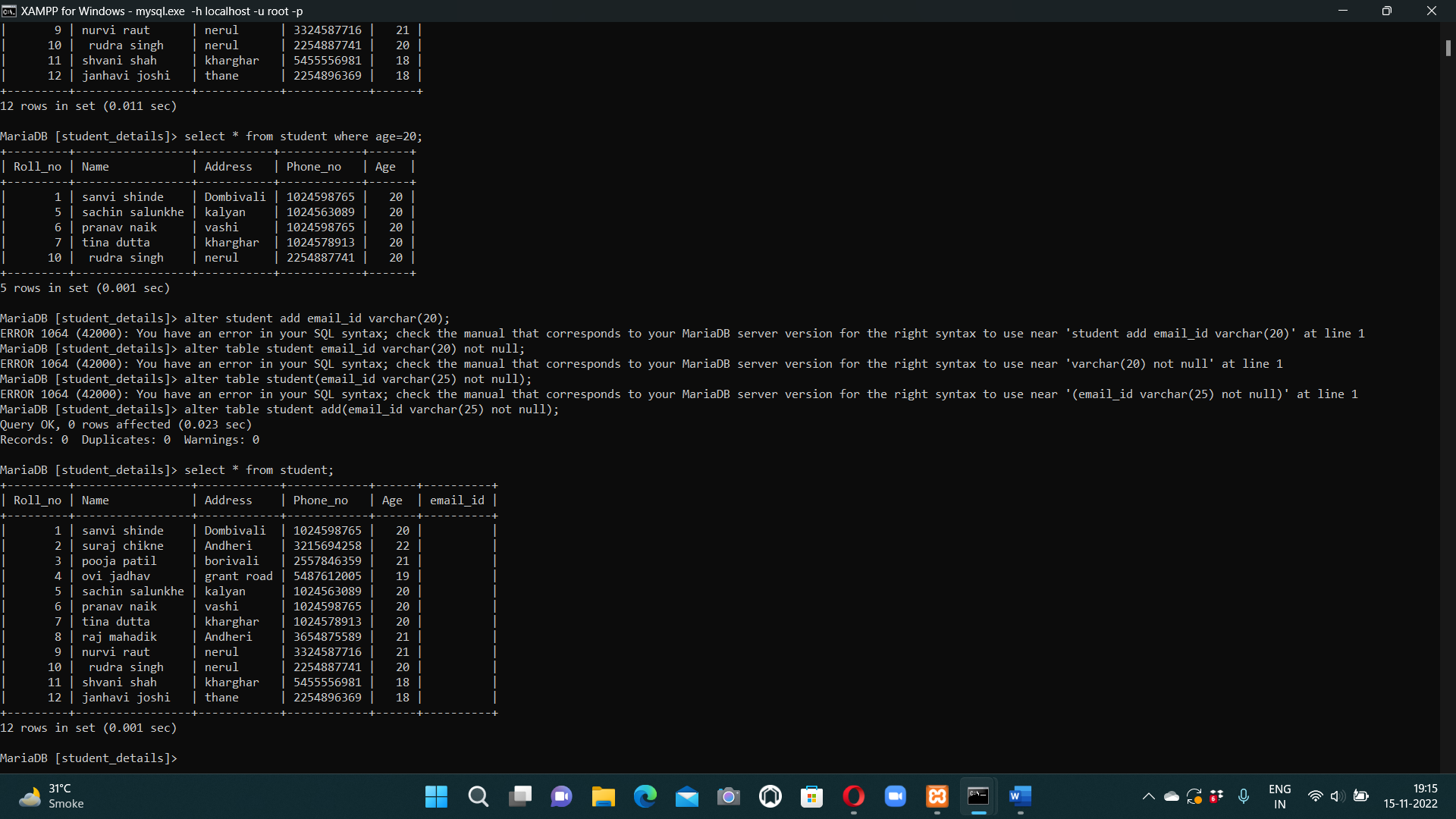
1. To see the record whose age is 20

Syntax: select \* from student where age=20;



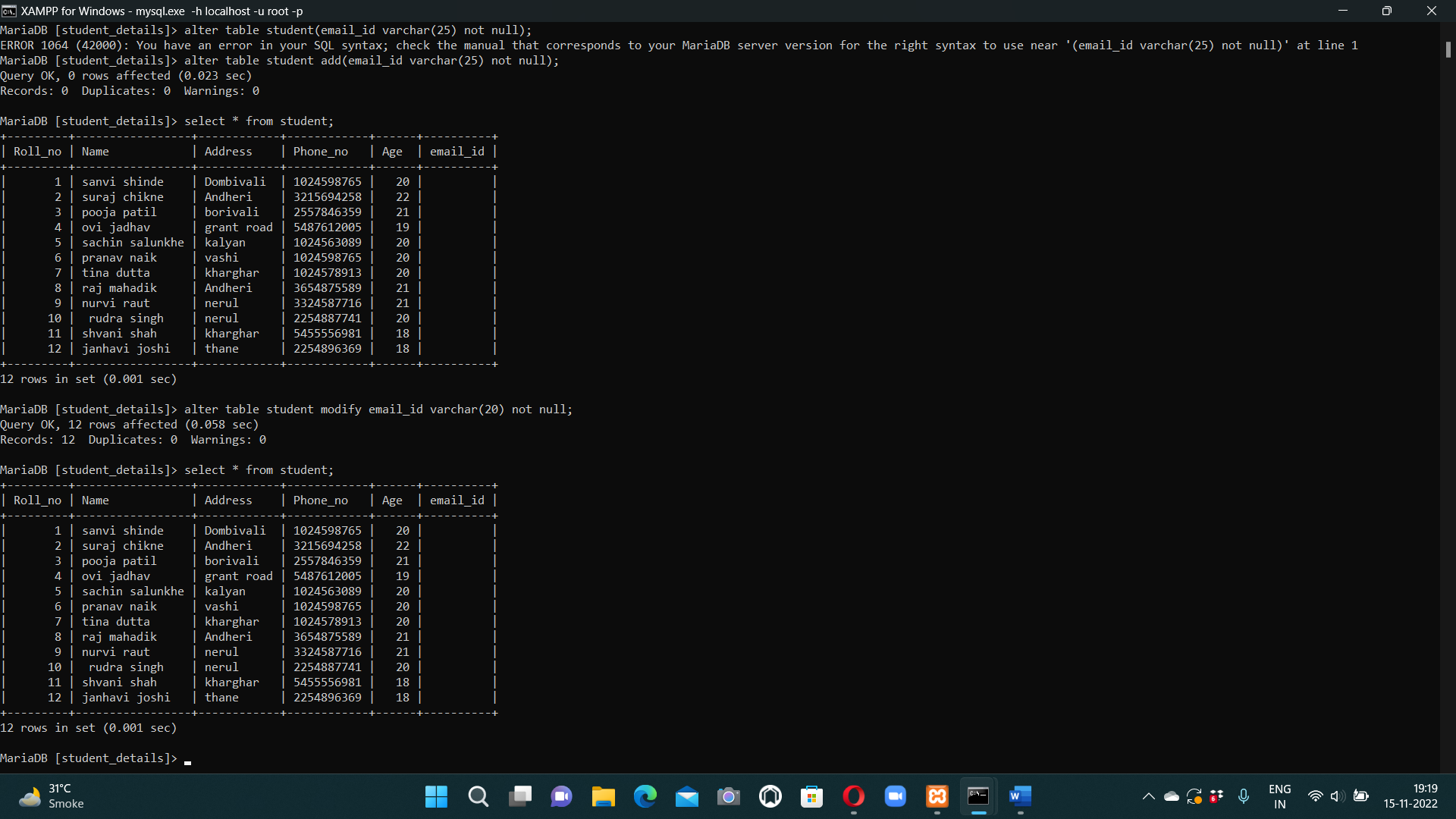
1. Add column email\_id

Syntax: alter table student add(email\_id varchar(25) not null);



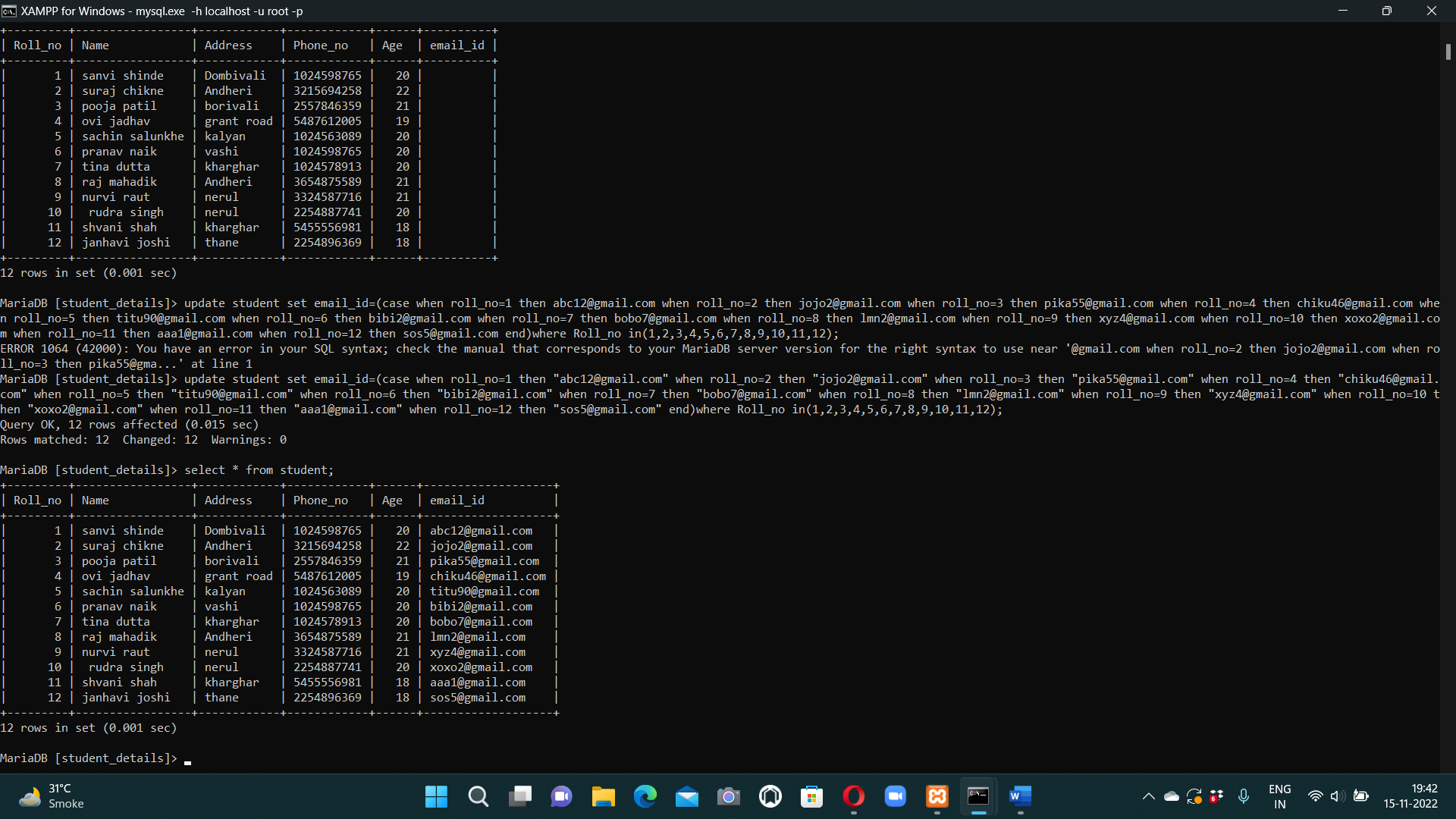
1. Change the datatype of email\_id column

Syntax : alter table student modify email\_id varchar(20) not null;



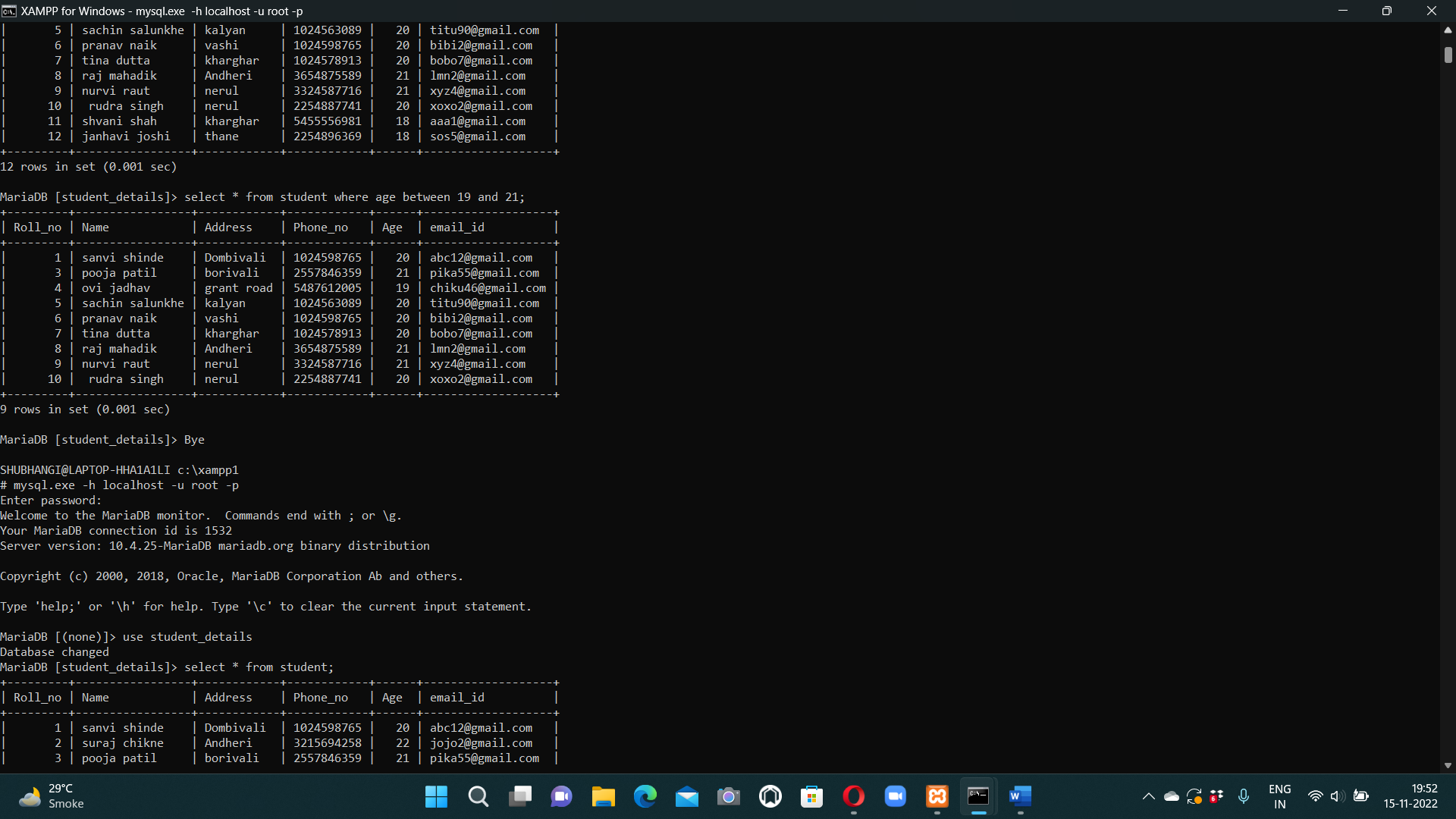
1. To adding the data into email\_id column

Syntax: update student set email\_id=(case when roll\_no=1 then "abc12@gmail.com" when roll\_no=2 then "jojo2@gmail.com" when roll\_no=3 then "pika55@gmail.com" when roll\_no=4 then "chiku46@gmail.com" when roll\_no=5 then "titu90@gmail.com" when roll\_no=6 then "bibi2@gmail.com" when roll\_no=7 then "bobo7@gmail.com" when roll\_no=8 then "lmn2@gmail.com" when roll\_no=9 then "xyz4@gmail.com" when roll\_no=10 then "xoxo2@gmail.com" when roll\_no=11 then "aaa1@gmail.com" when roll\_no=12 then "sos5@gmail.com" end)where Roll\_no in(1,2,3,4,5,6,7,8,9,10,11,12);



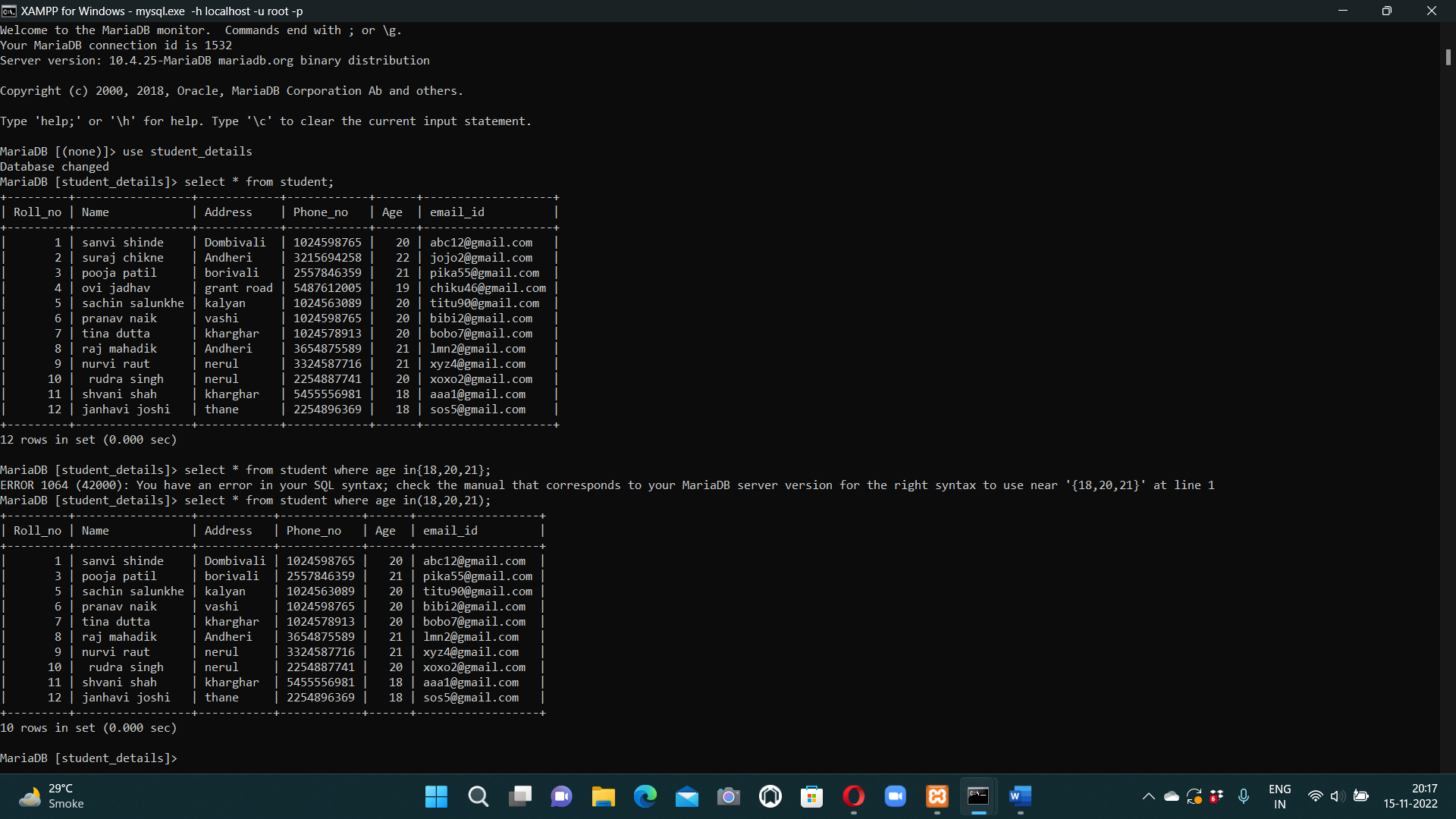
1. To get a students records where students age is between 19-22

Syntax: select \* from student where age between 19 and 21;



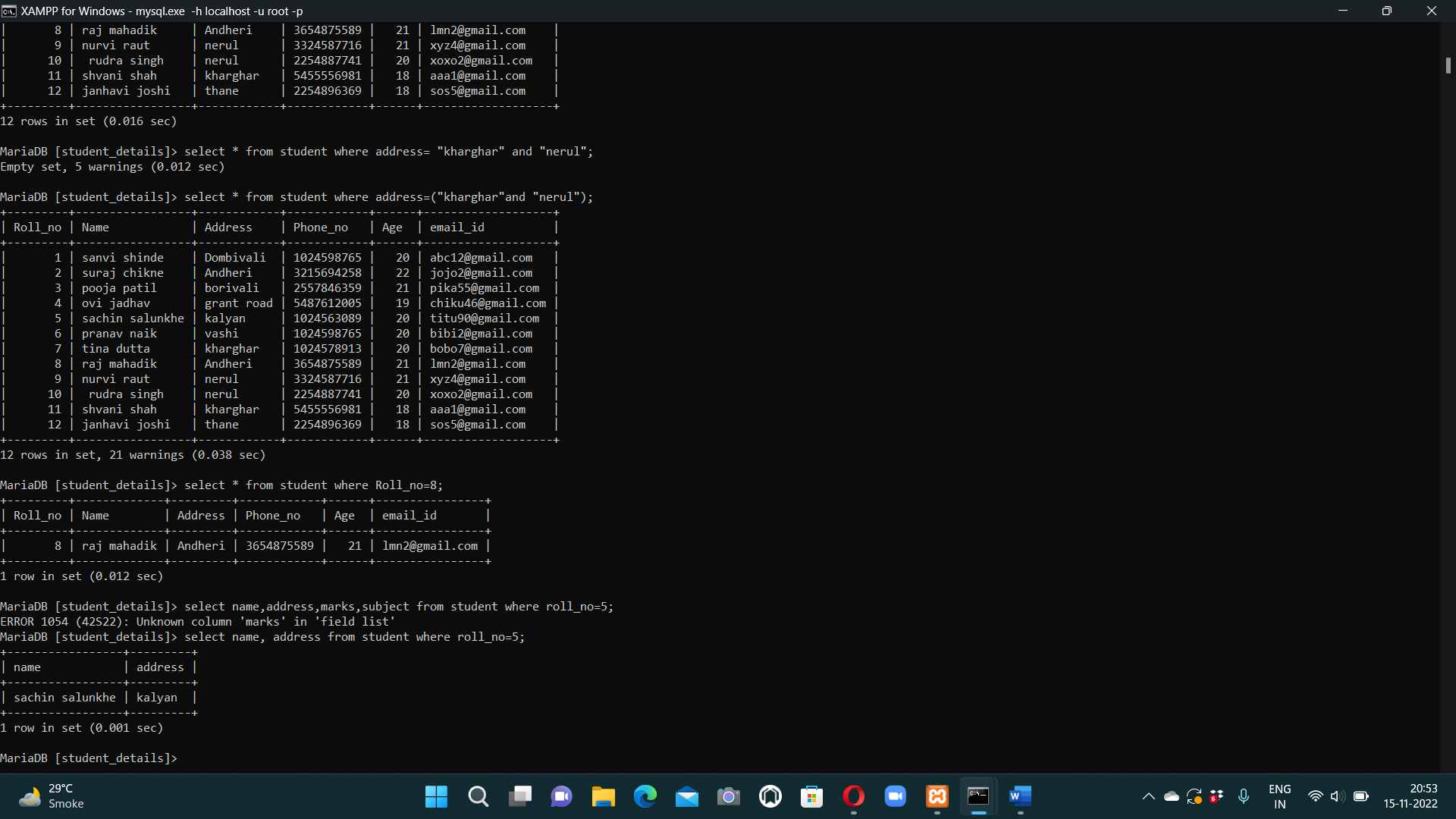
1. To get students record, where student’s age is in the set (18,20,21).

Syntax: select \* from student where age in(18,20,21);



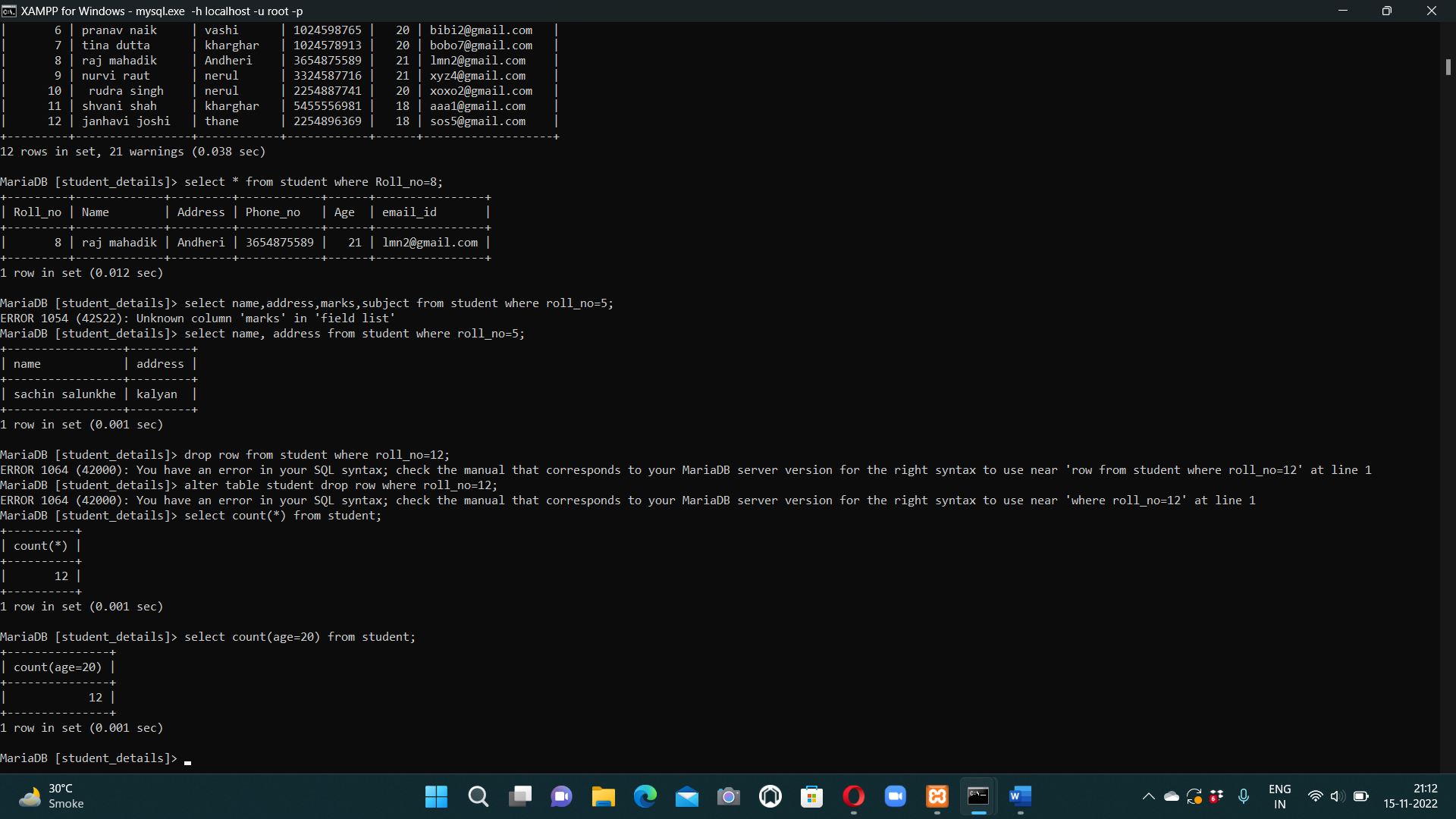
1. Display name, address for student whose rollno is 5.

Syntax : select name, address from student where roll\_no=5;



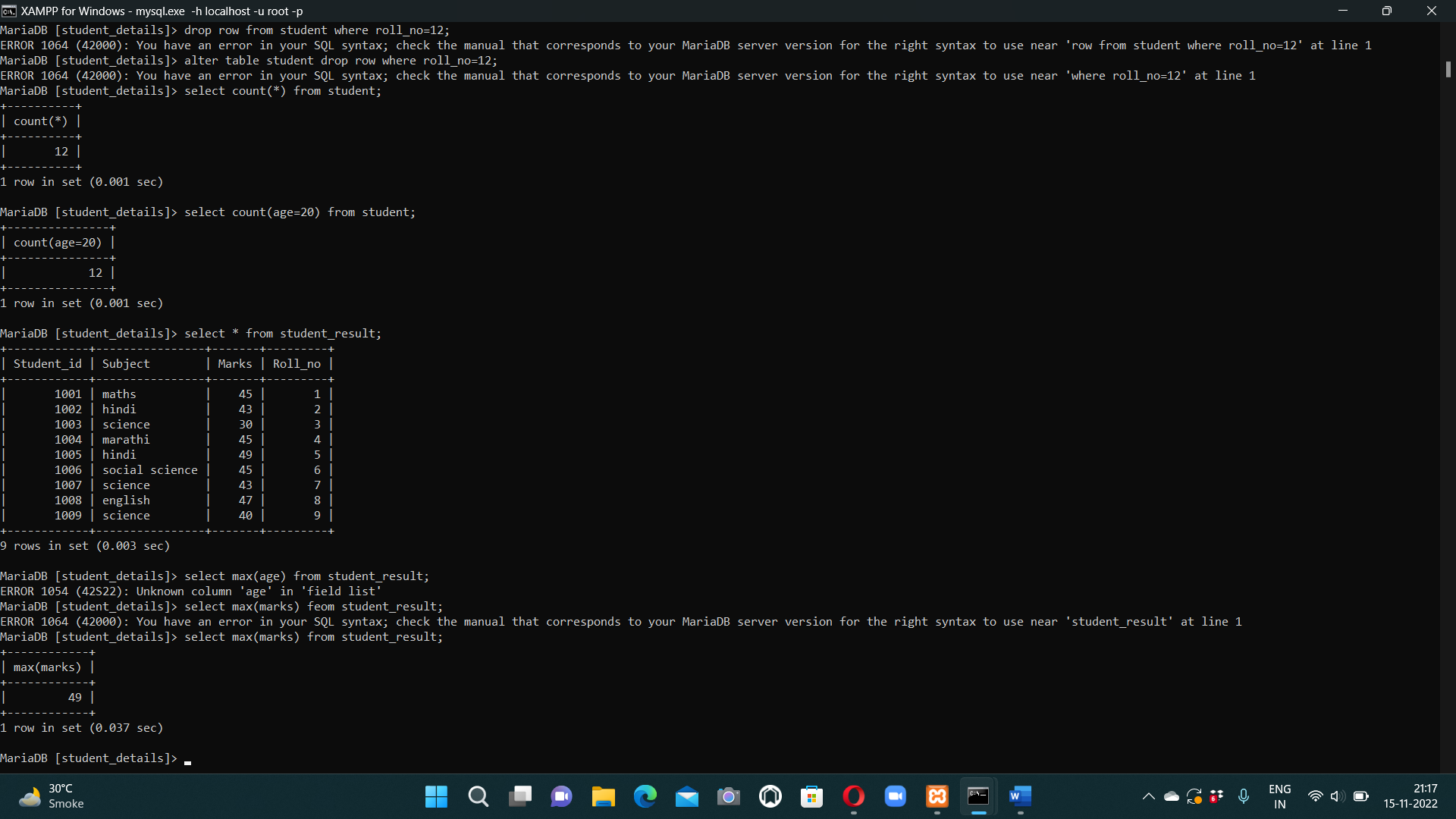
1. Fetch count of records

Syntax : select count(\*) from student;



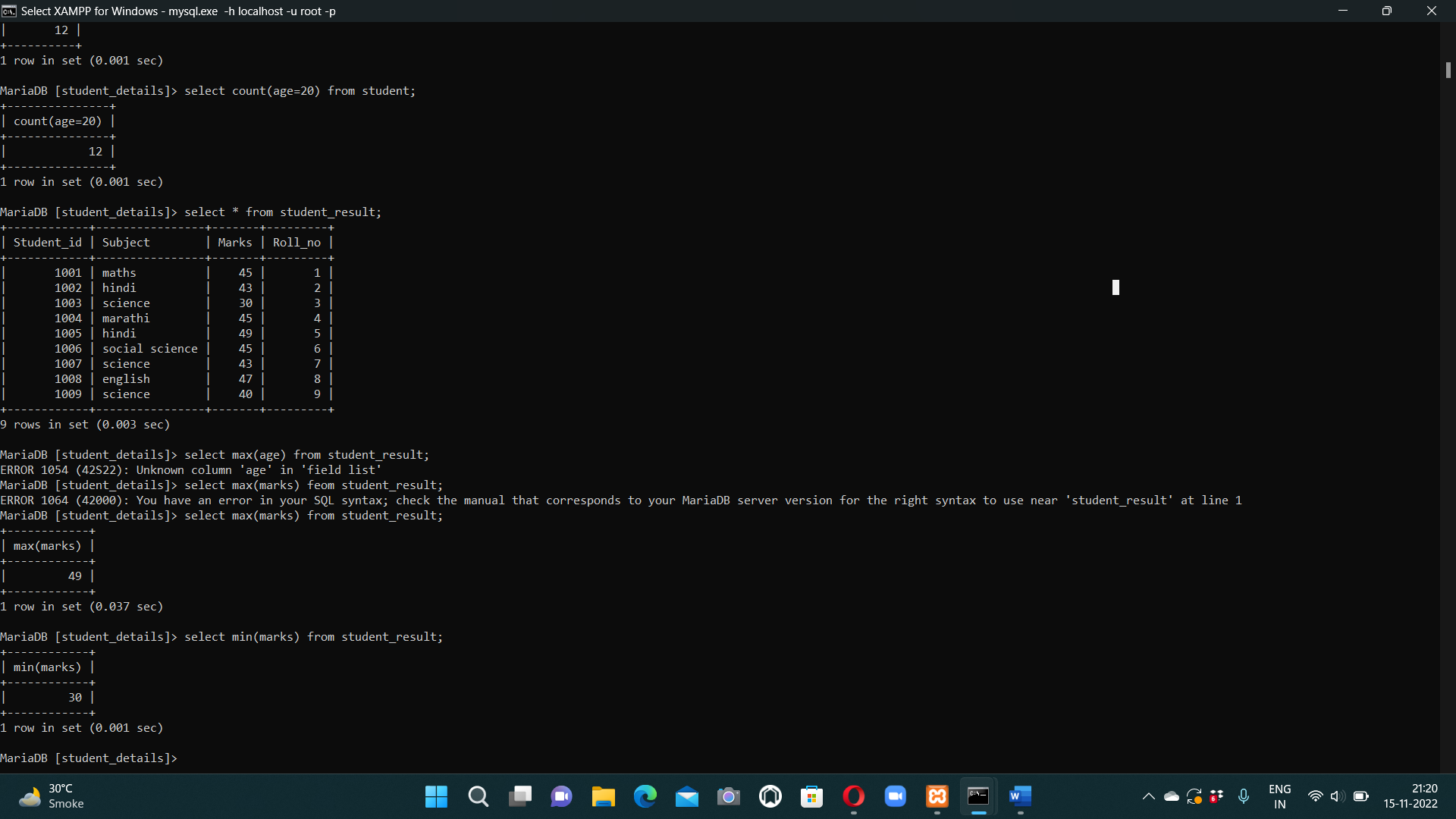
1. Fetch maximum marks from student\_result table.

Syntax: select max(marks) from student\_result;



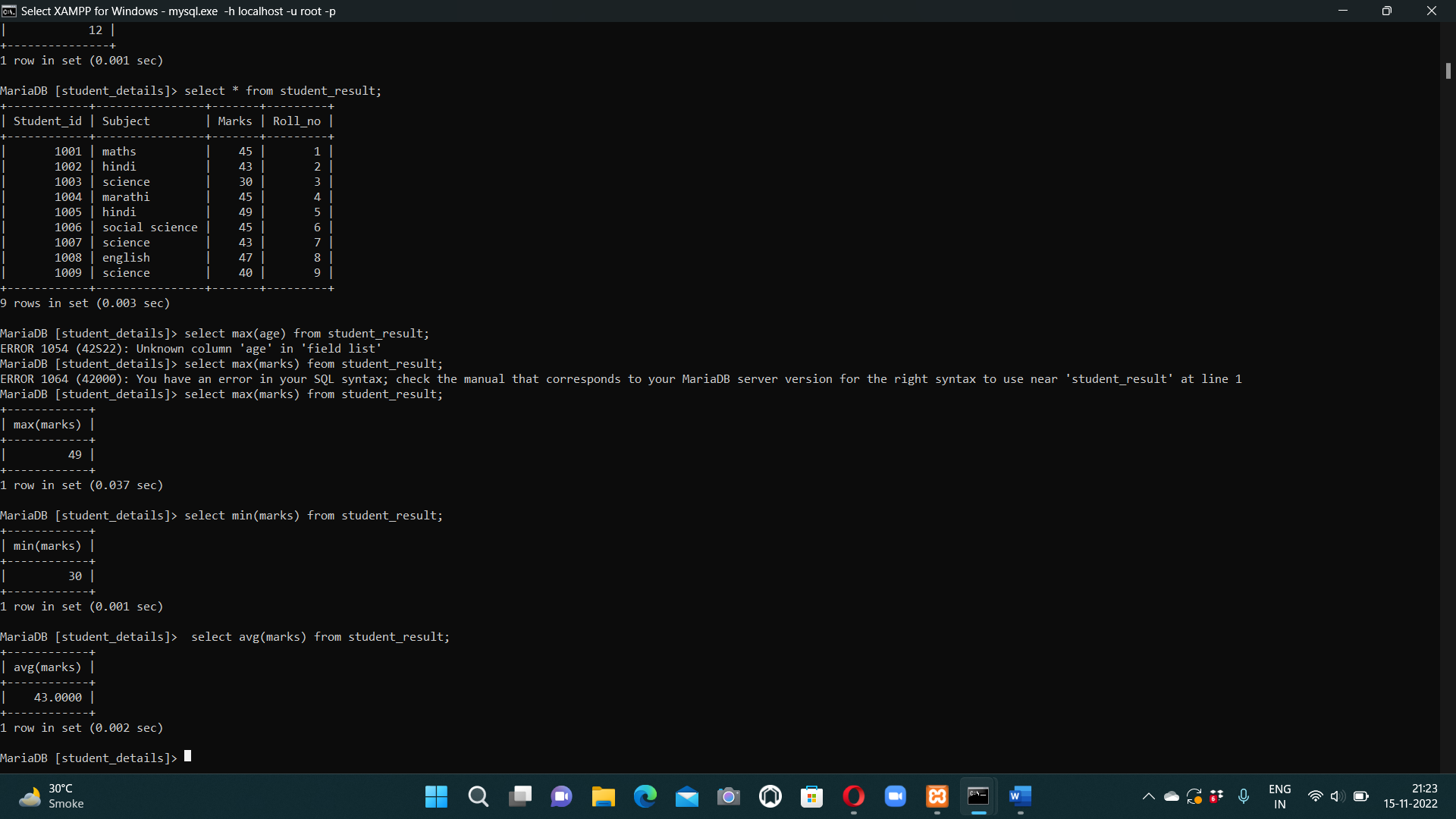
1. Fetch minimum marks from student\_result table.

Syntax : select min(marks) from student\_result;



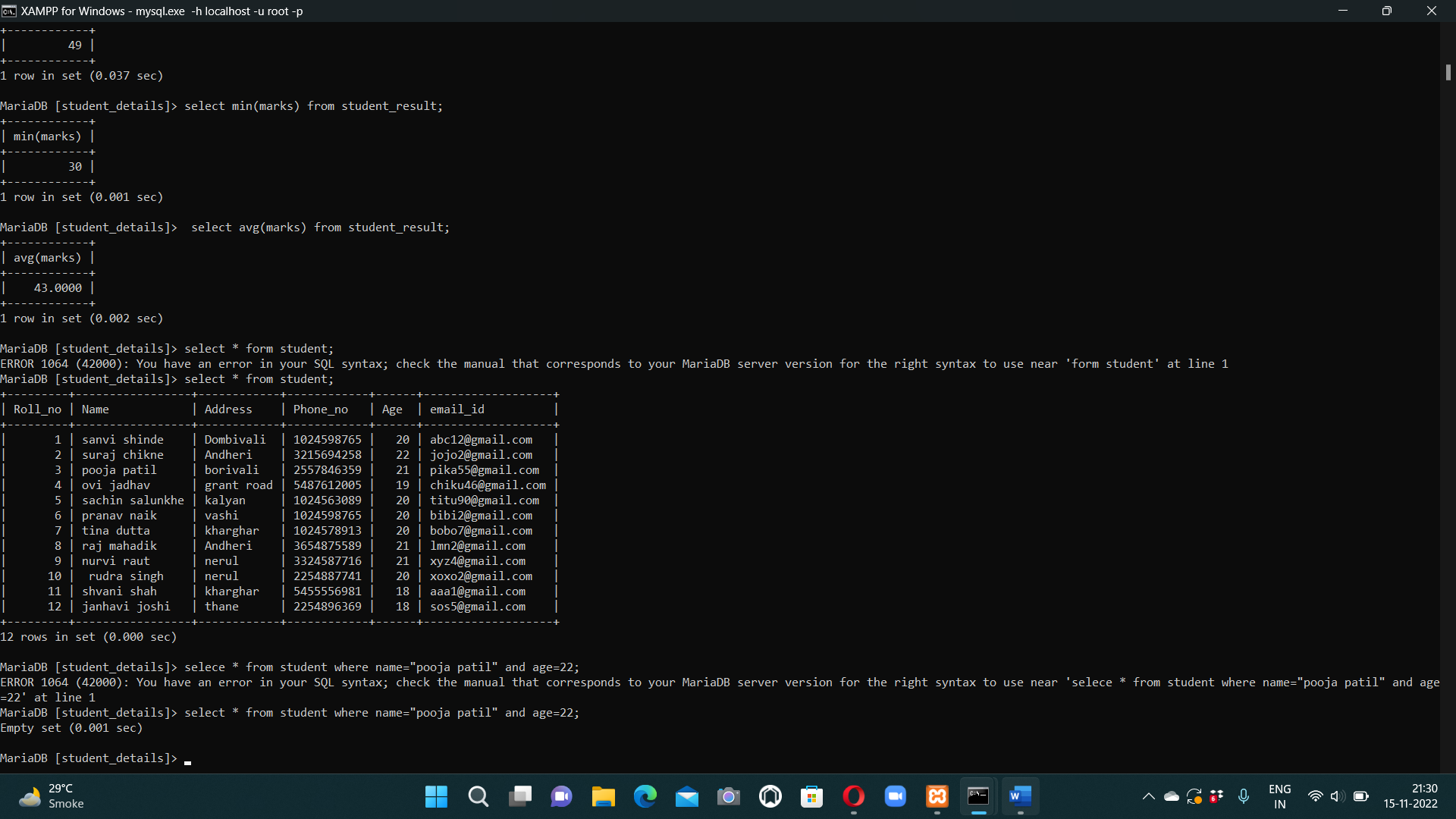
1. Fetch avg marks from student\_result.

Syntax : select avg(marks) from student\_result;



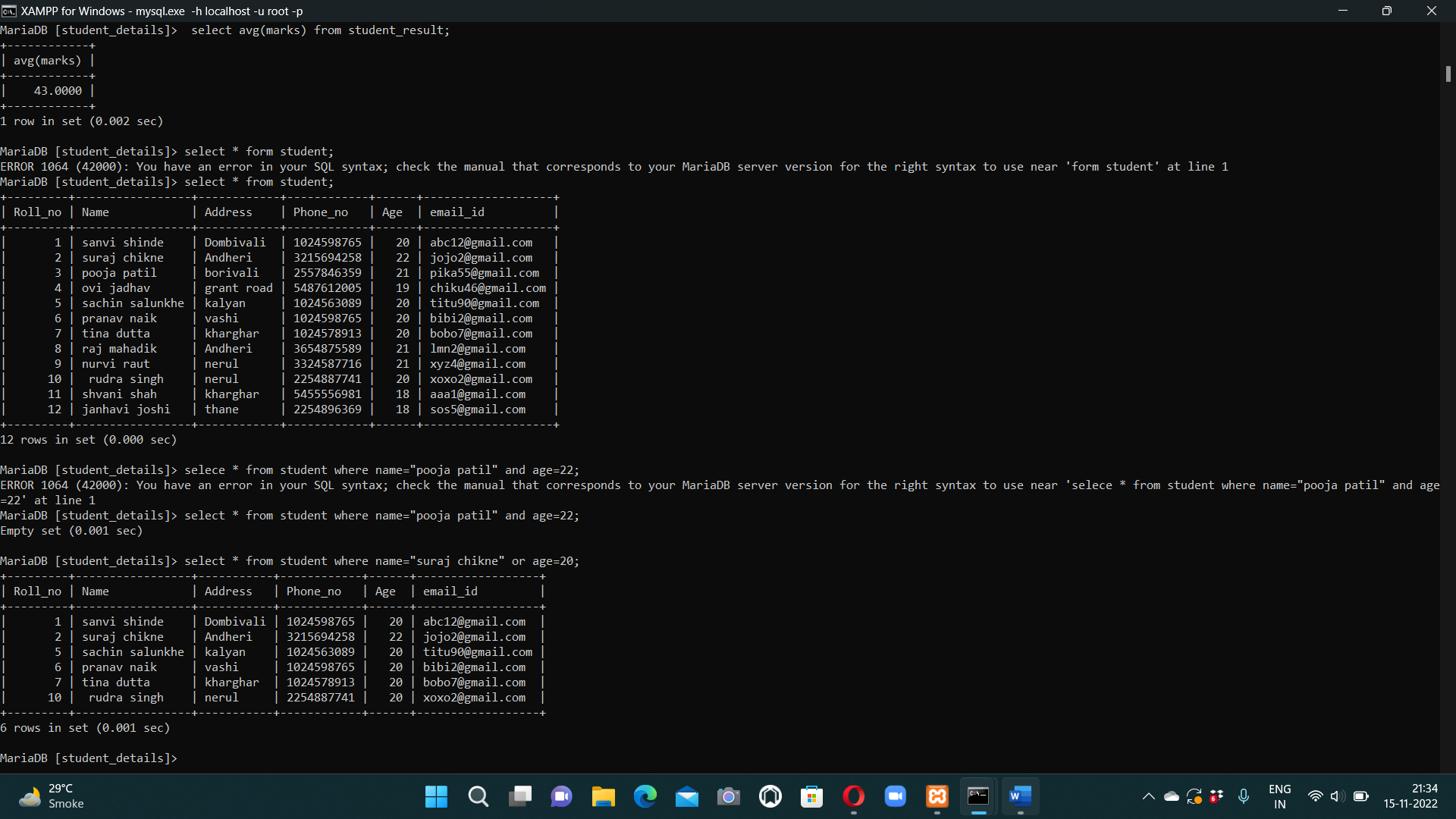
1. Display the details of student whose name is pooja patil and age is 22.

Syntax : select \* from student where name="pooja patil" and age=22;



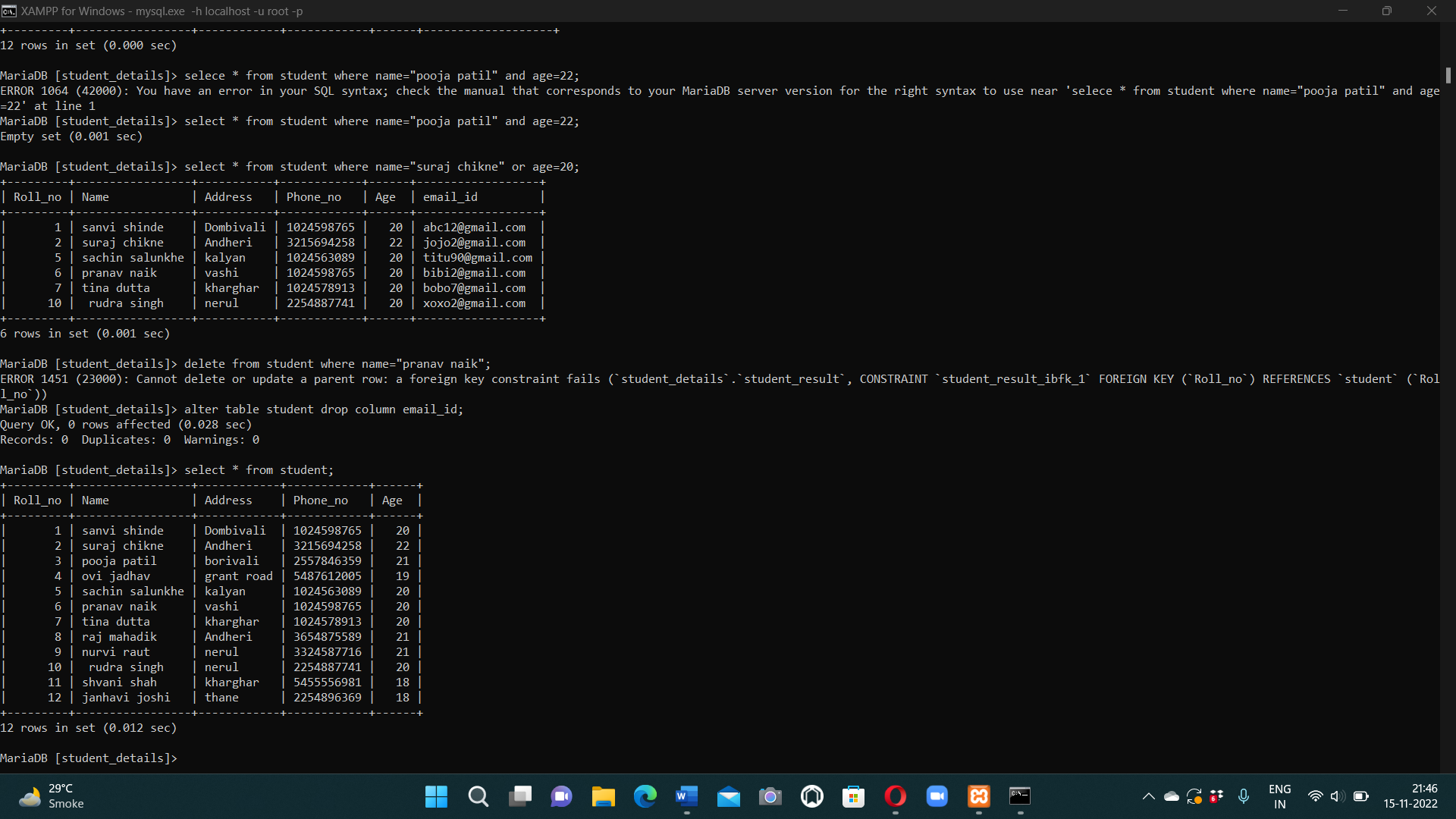
1. Display the details of student whose name is suraj chikne or age is 20.

Syntax : select \* from student where name=”suraj chikne” or age=20;



1. Delete the column email\_id.

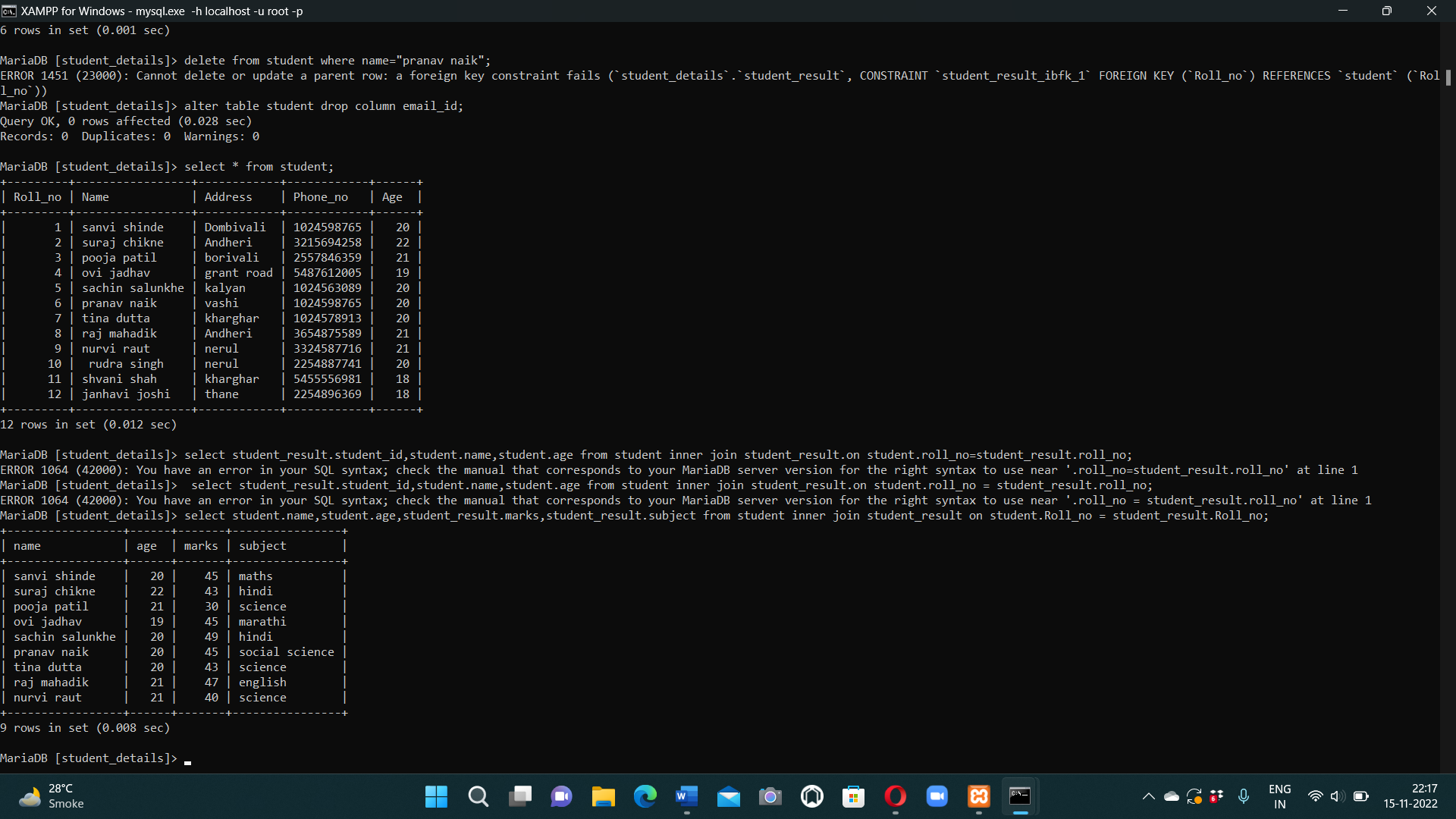
Syntax : alter table student drop column email\_id;



**joins**

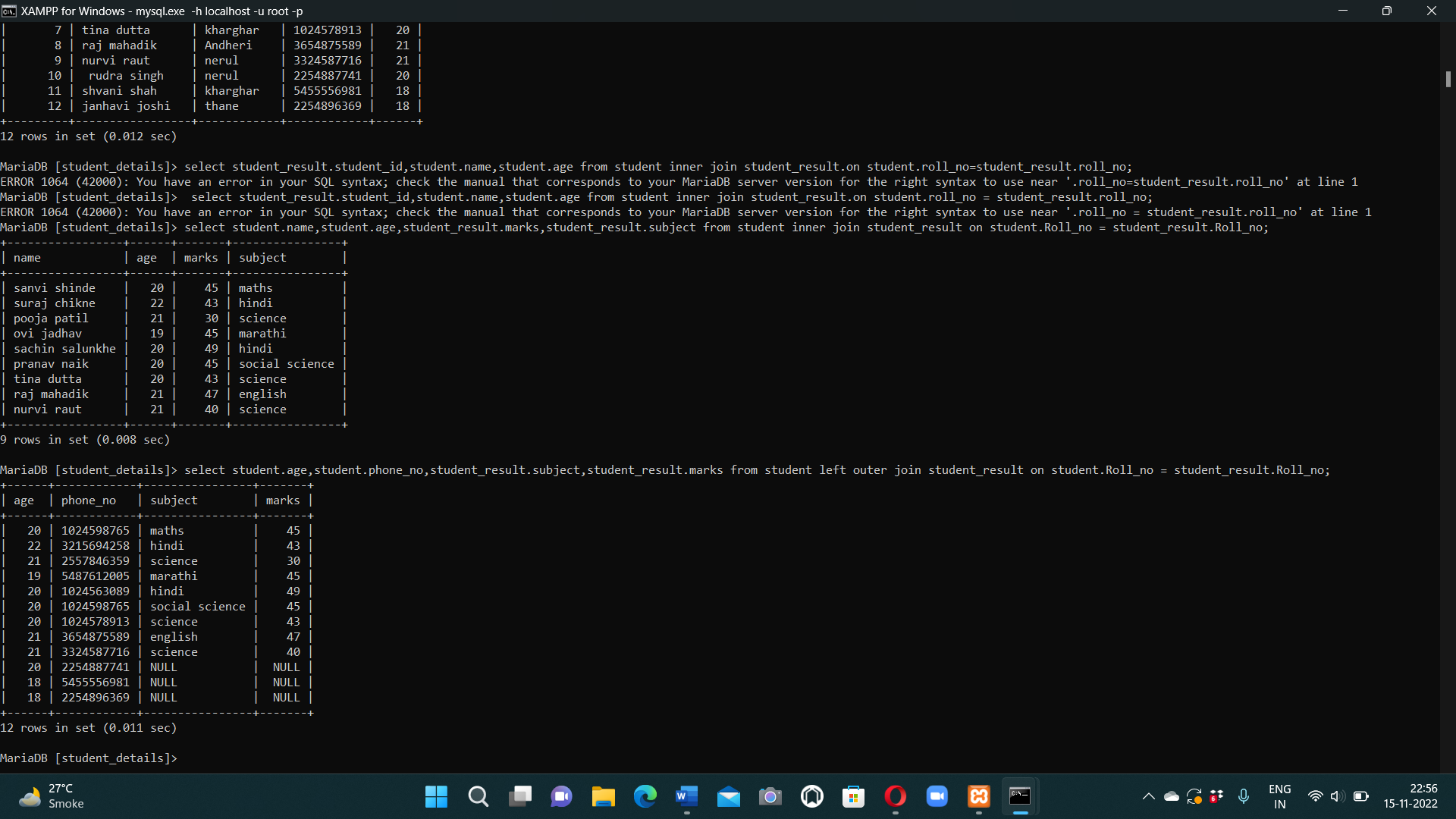
1. inner join

Syntax: select student.name,student.age,student\_result.marks,student\_result.subject from student inner join student\_result on student.Roll\_no = student\_result.Roll\_no;



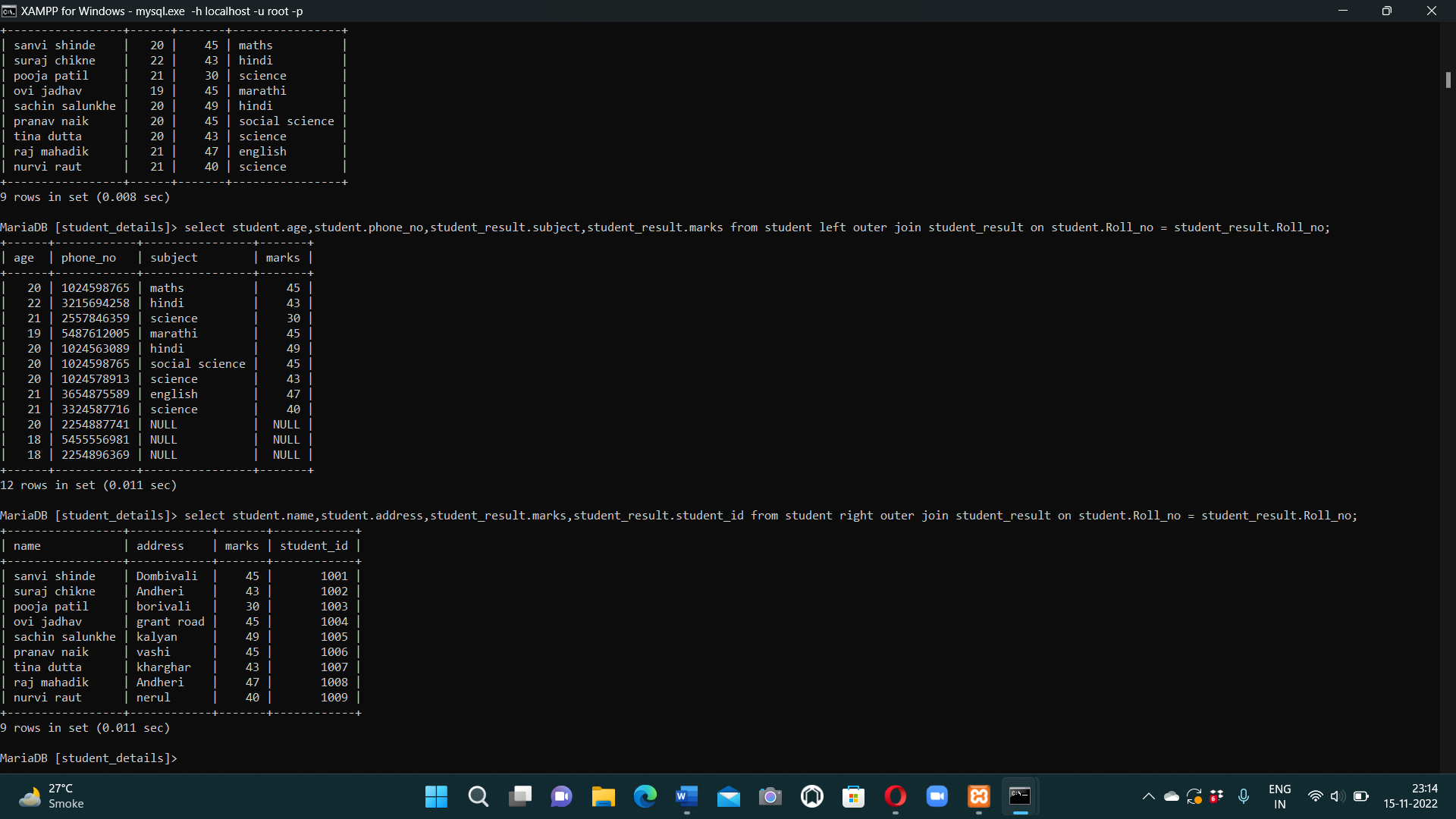
1. Left outer join

Syntax : select student.age,student.phone\_no,student\_result.subject,student\_result.marks from student left outer join student\_result on student.Roll\_no = student\_result.Roll\_no;



1. Right outer join

Syntax : select student.name,student.address,student\_result.marks,student\_result.student\_id from student right outer join student\_result on student.Roll\_no = student\_result.Roll\_no;

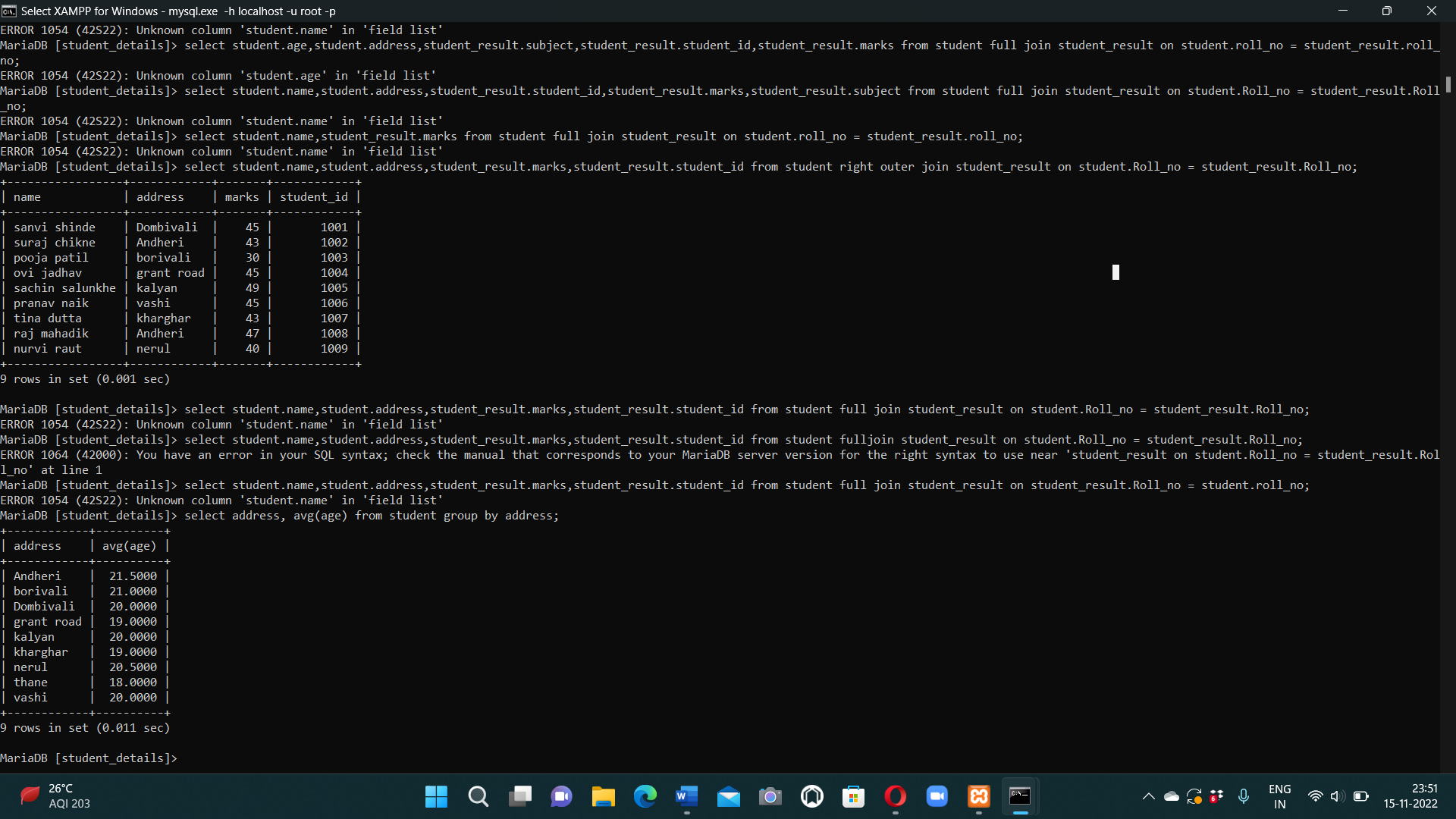


1. Full join

Syntax : SELECT column\_name(s)  
FROM table1  
FULL OUTER JOIN table2ON table1.column\_name = table2.column\_nameWHERE condition;

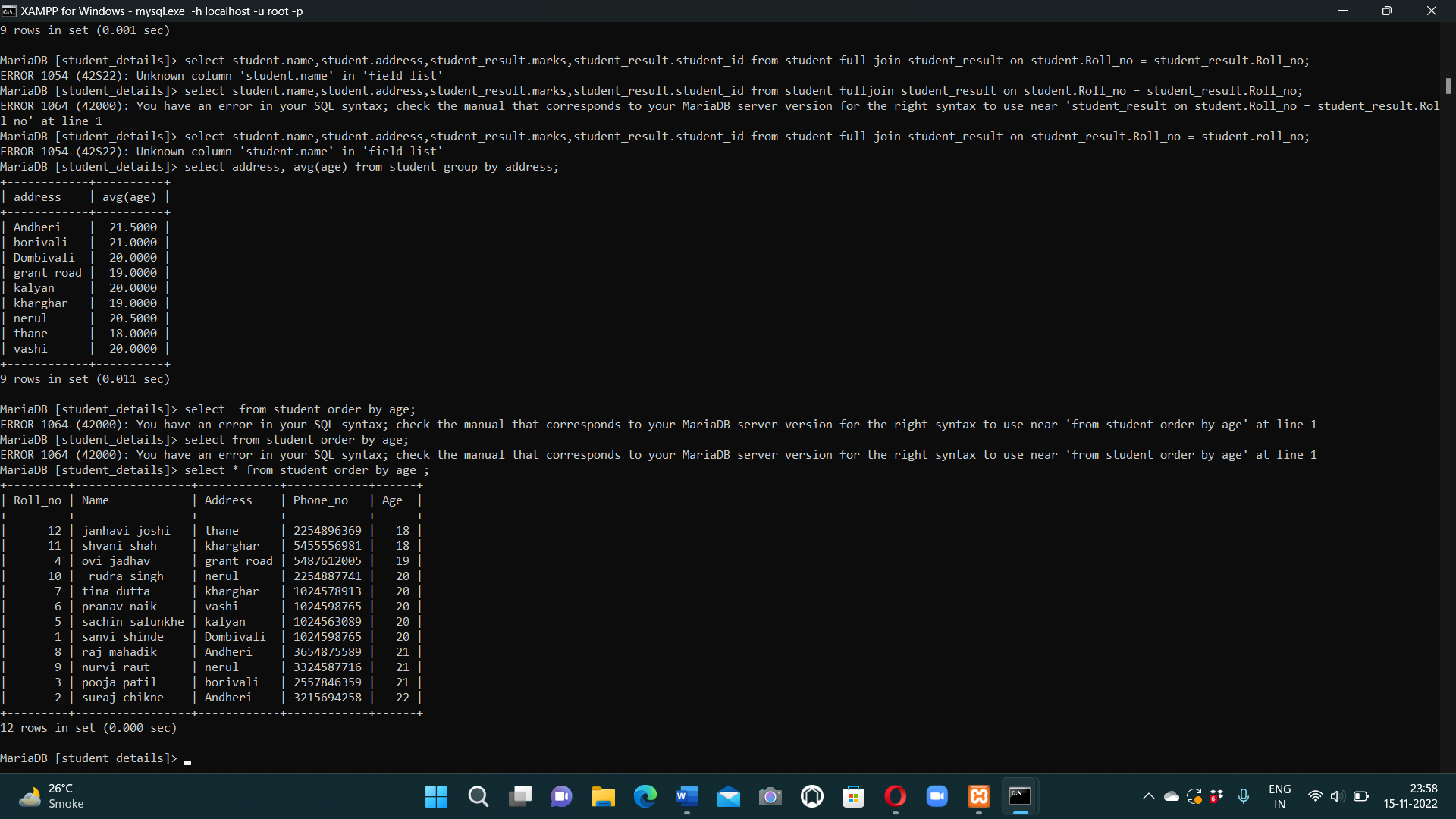
1. Group by – demonstrate group by clause.

Syntax : select address, avg(age) from student group by address;



1. Order by- display records in order of ascending ages.

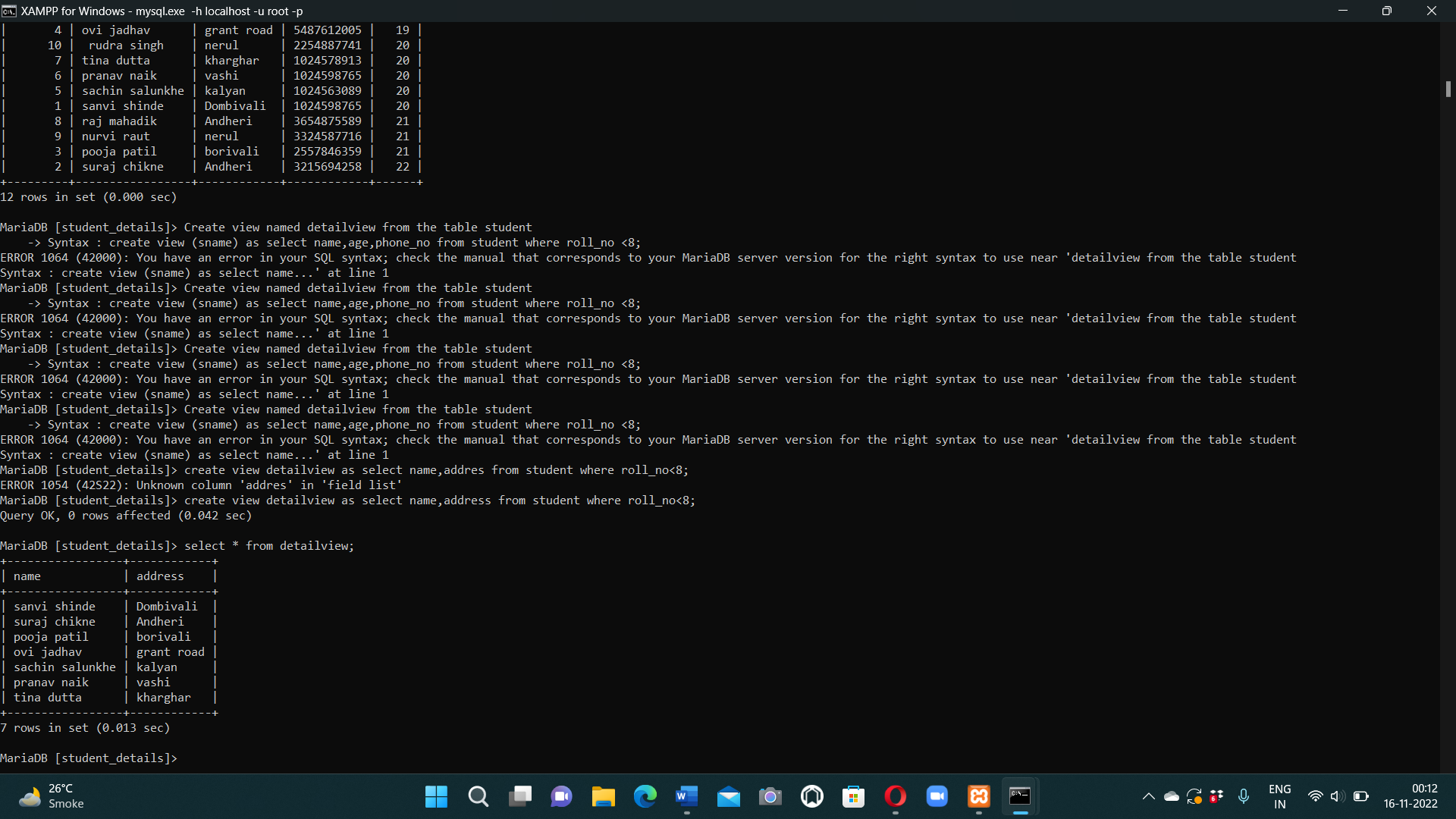
Syntax : select \* from student order by age ;



**views**

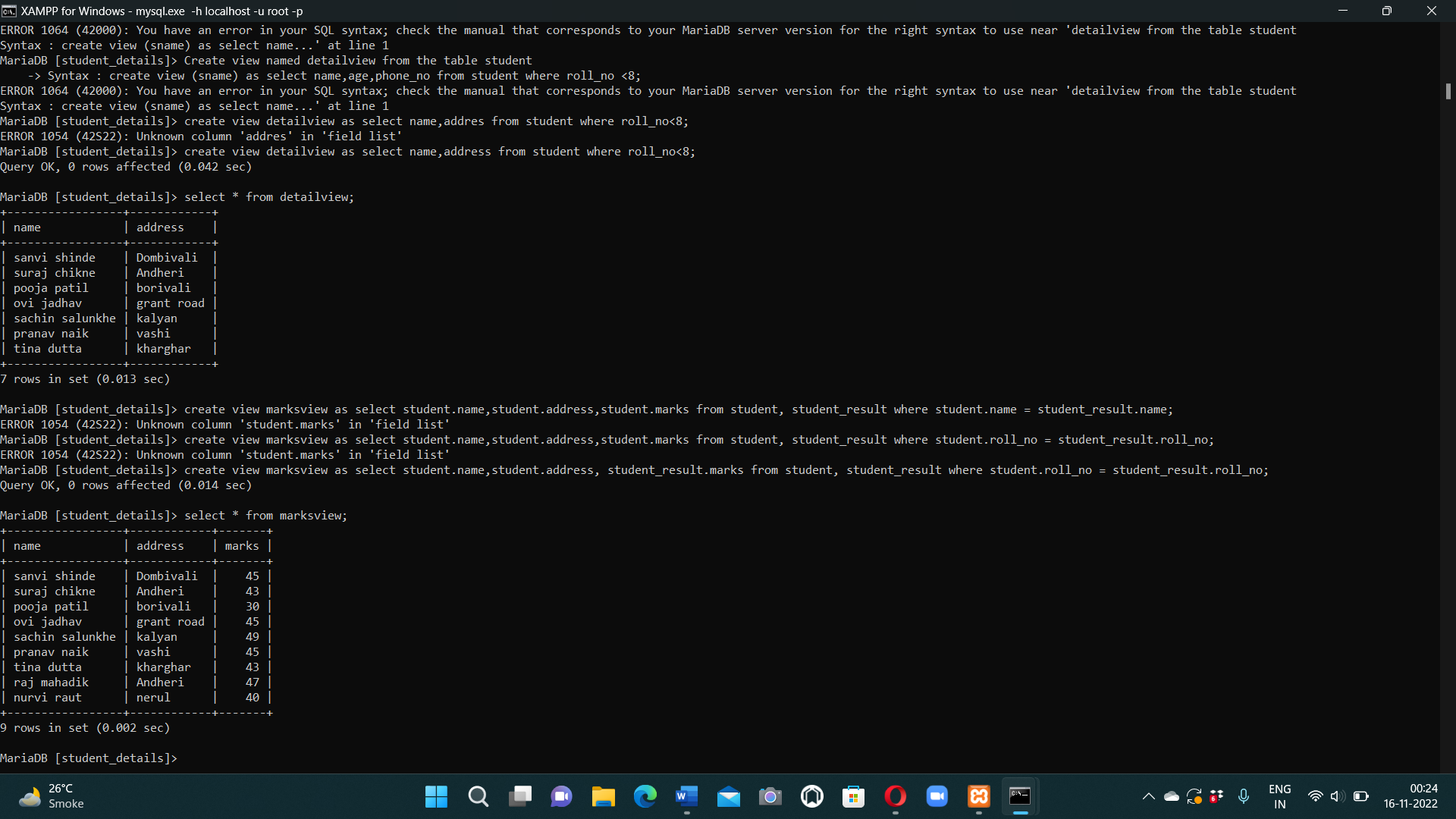
1. Views- single table

Syntax : create view detailview as select name,address from student where roll\_no<8;



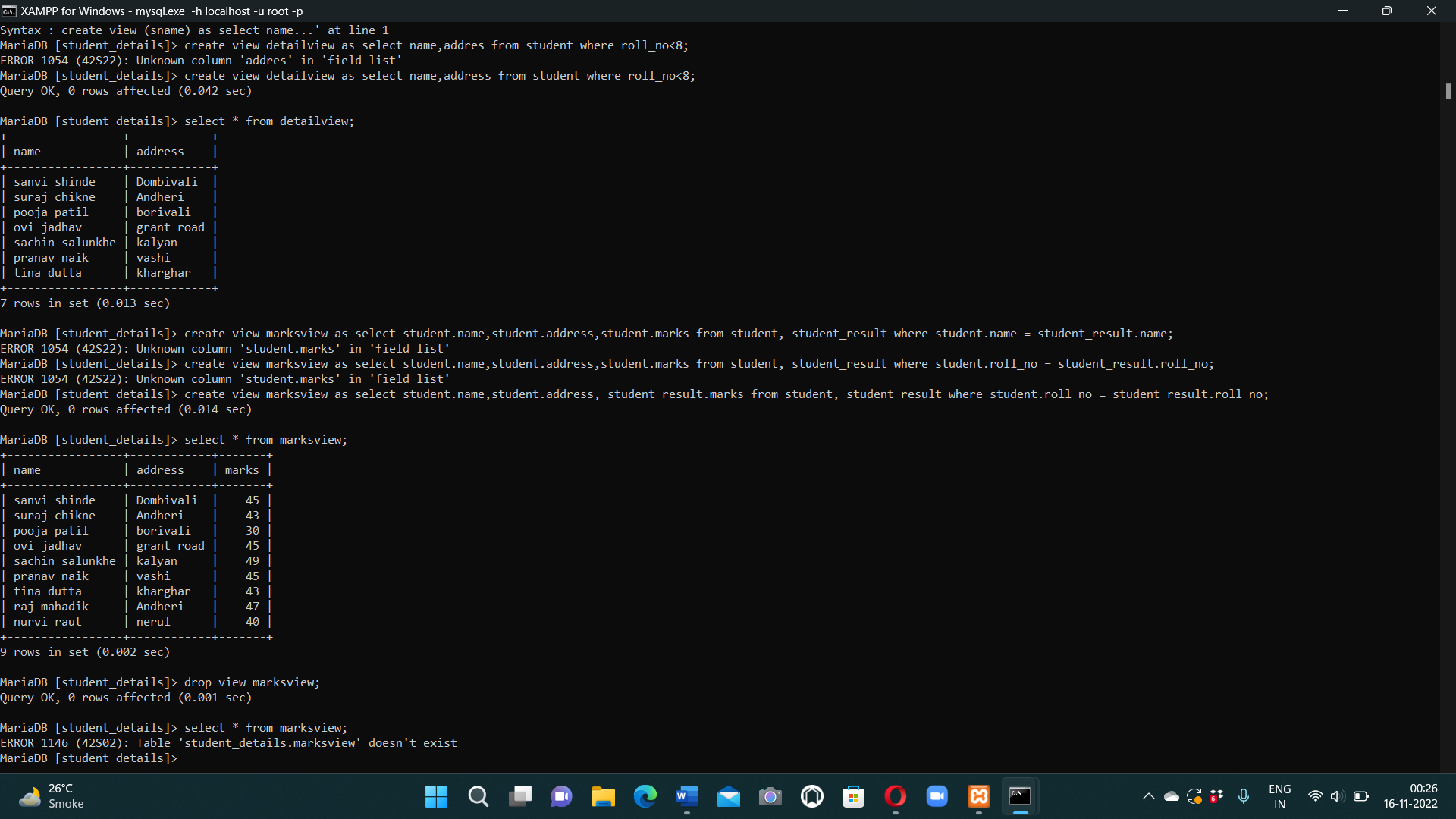
1. View- multiple table

Syntax : create view marksview as select student.name,student.address, student\_result.marks from student, student\_result where student.roll\_no = student\_result.roll\_no;



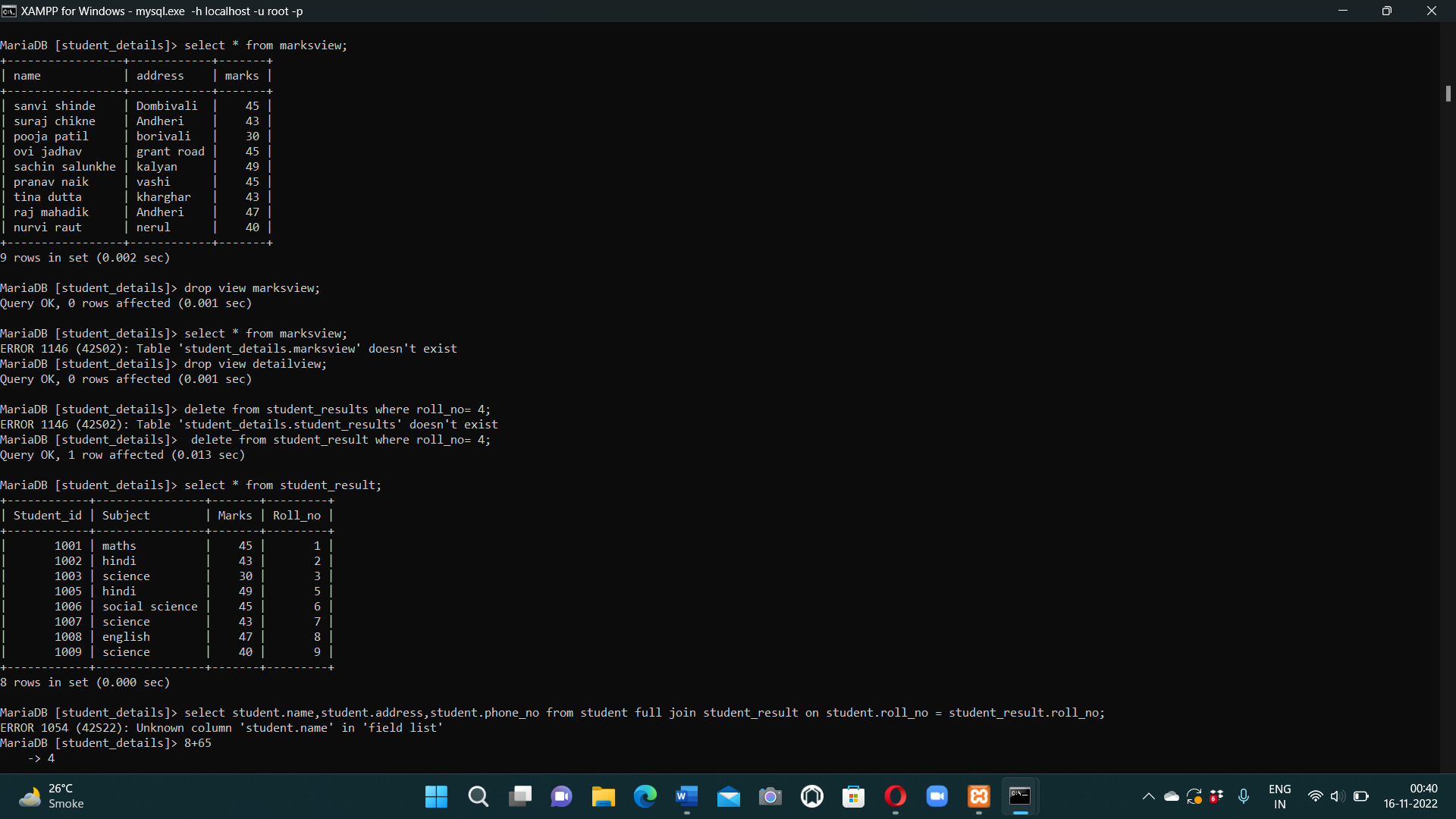
1. Drop view

Syntax : drop view marksview;



1. Delete data whose roll no is 4 .

Syntax : delete from student\_result where roll\_no=4;



**Conclusion**

**I hereby conclude that I have completed my project and achieved my main aim is to easily get a information of individual student separately by performing queries. Also understand that this concept is time saving, manages information.**

**By performing all the SQL queries to best of my knowledge.**

**Thank you…**