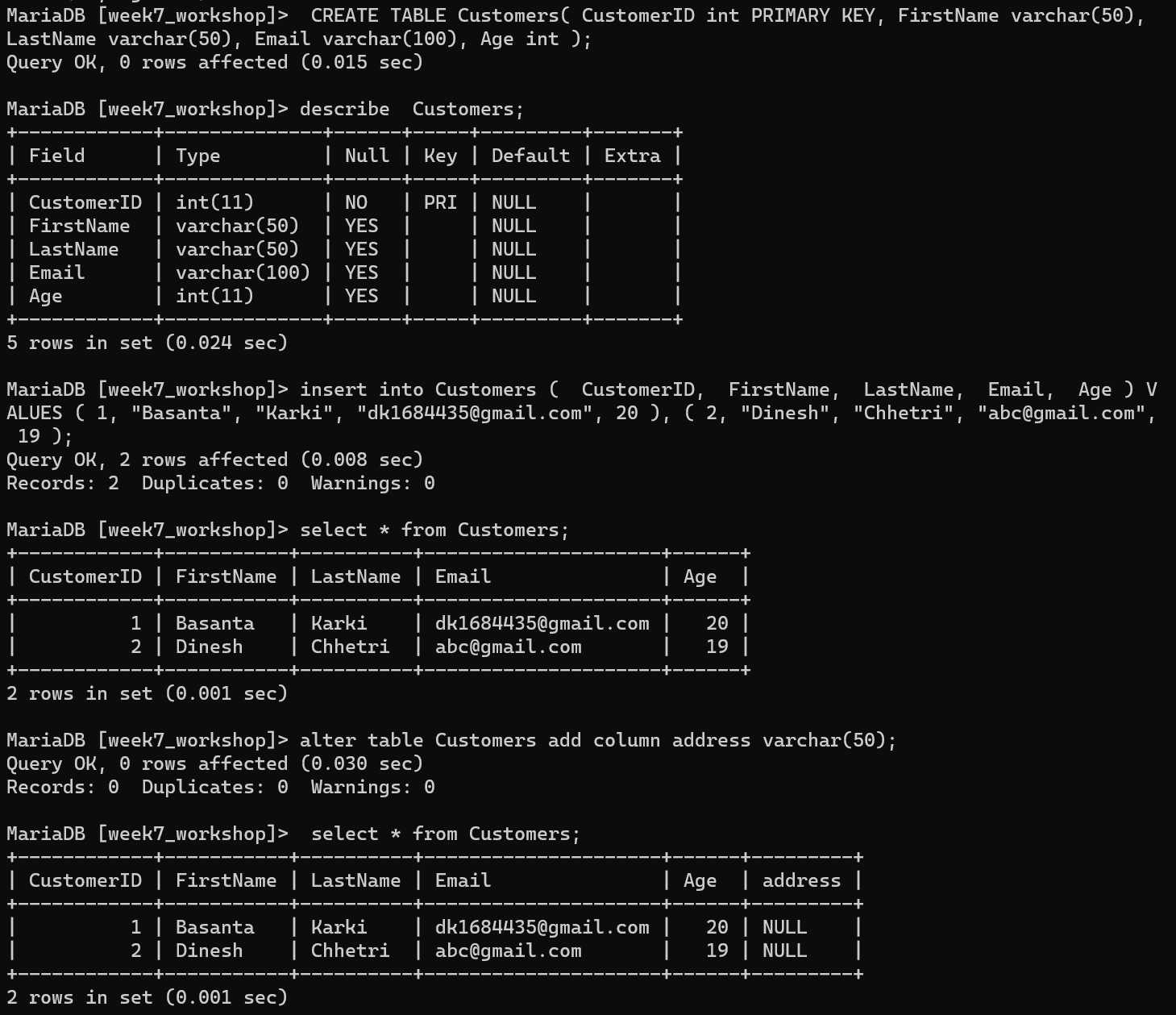
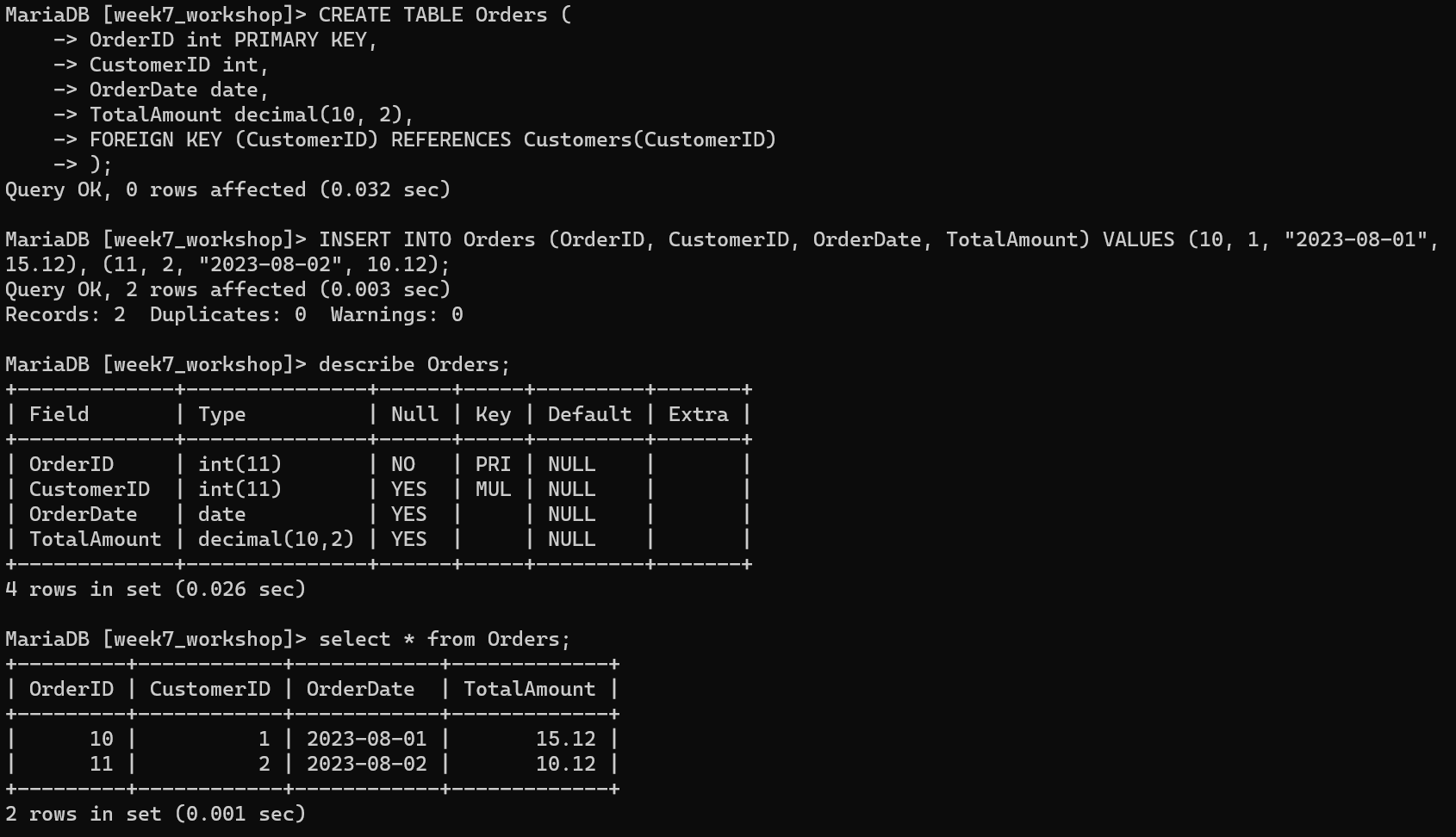
**Task 1**

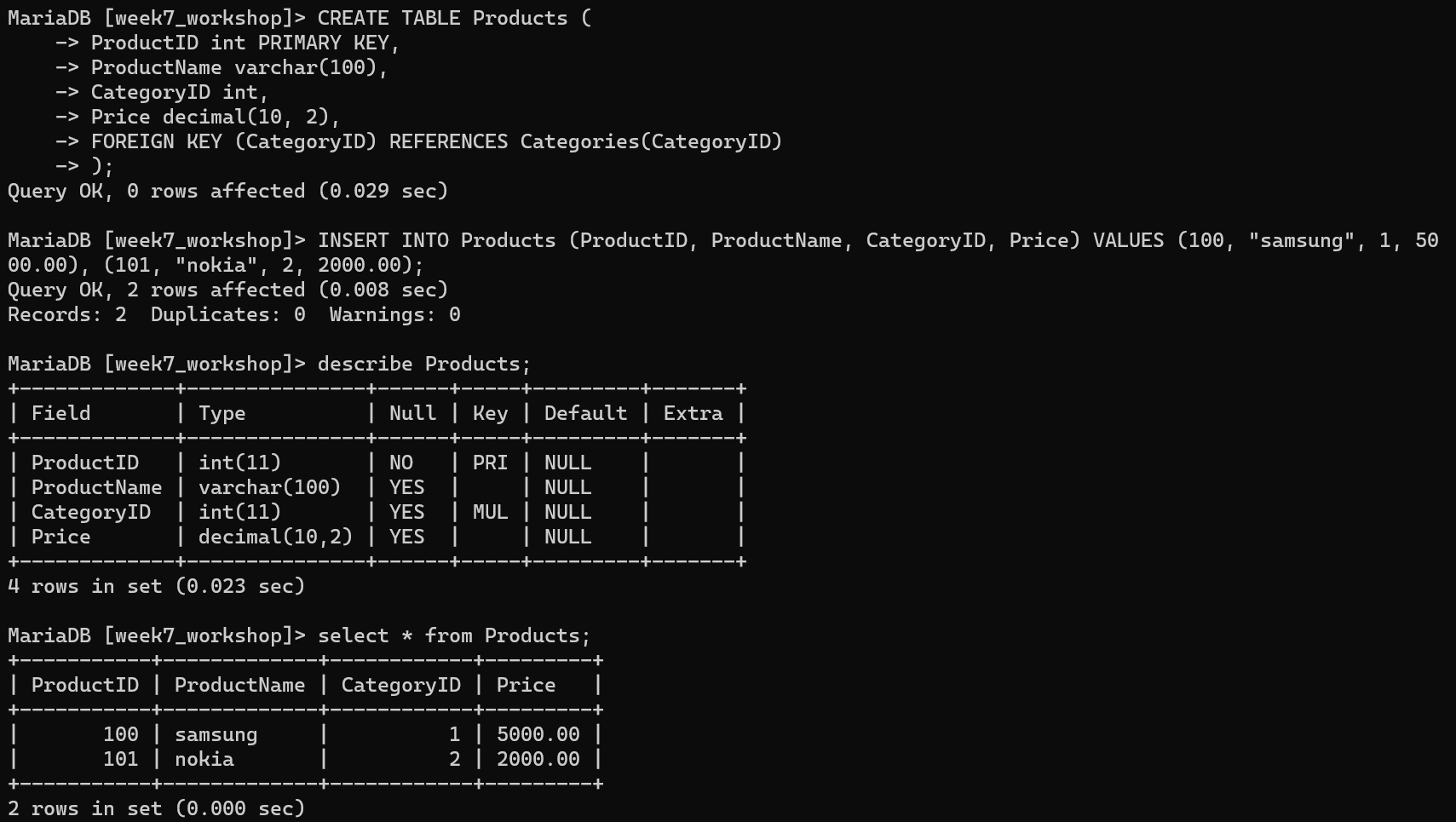
* Create a table named "Customers" with the following columns:
  + CustomerID (integer, primary key)
  + FirstName (varchar(50))
  + LastName (varchar(50))
  + Email (varchar(100))
  + Age (integer)
* Add a column named "Address" to the "Customers" table with the data type varchar(200).



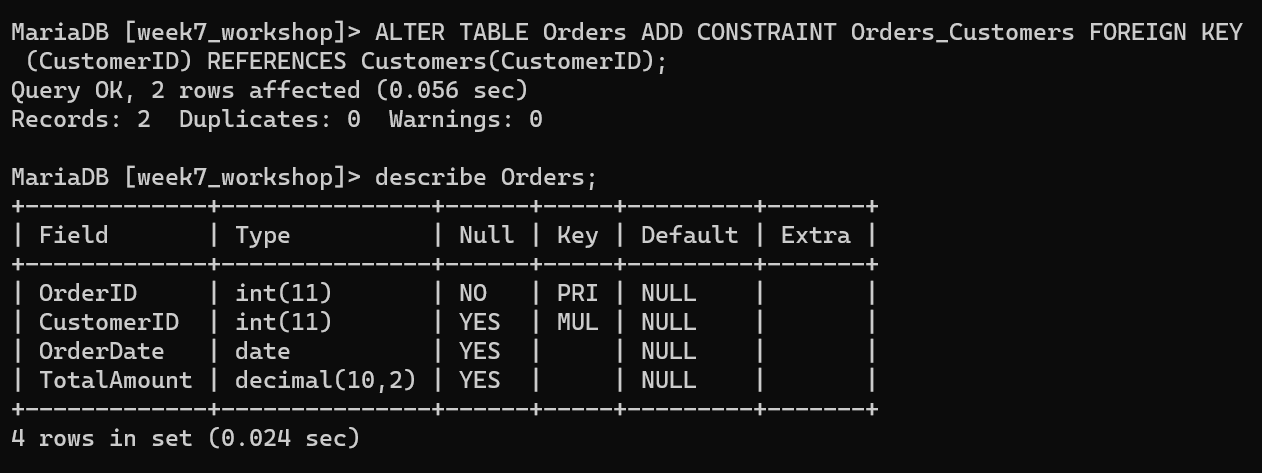
* Create a table named "Orders" with the following columns:
  + OrderID (integer, primary key)
  + CustomerID (integer, foreign key referencing CustomerID in the Customers table)
  + OrderDate (date)
  + TotalAmount (decimal(10,2))



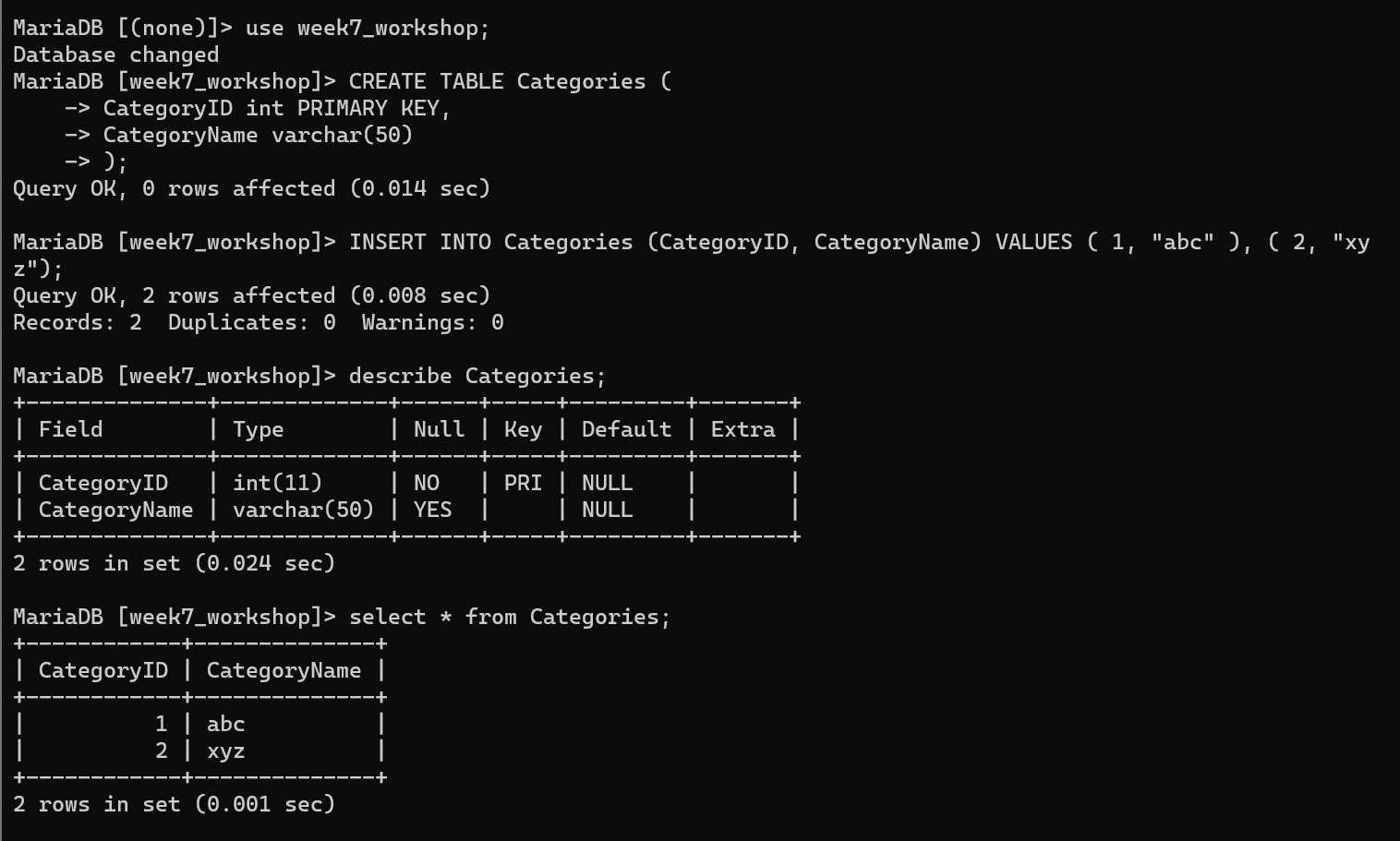
* Create a table named "Products" with the following columns:
  + ProductID (integer, primary key)
  + ProductName (varchar(100))
  + CategoryID (integer, foreign key referencing CategoryID in the Categories table)
  + Price (decimal(10,2))



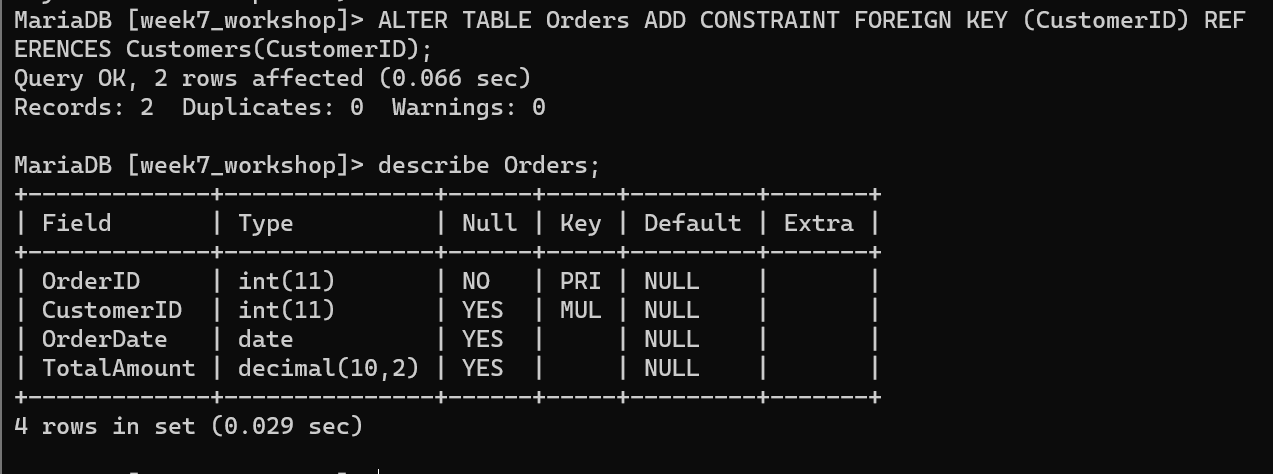
* Add a foreign key constraint to the "Orders" table, referencing the "CustomerID" column to the "Customers" table.



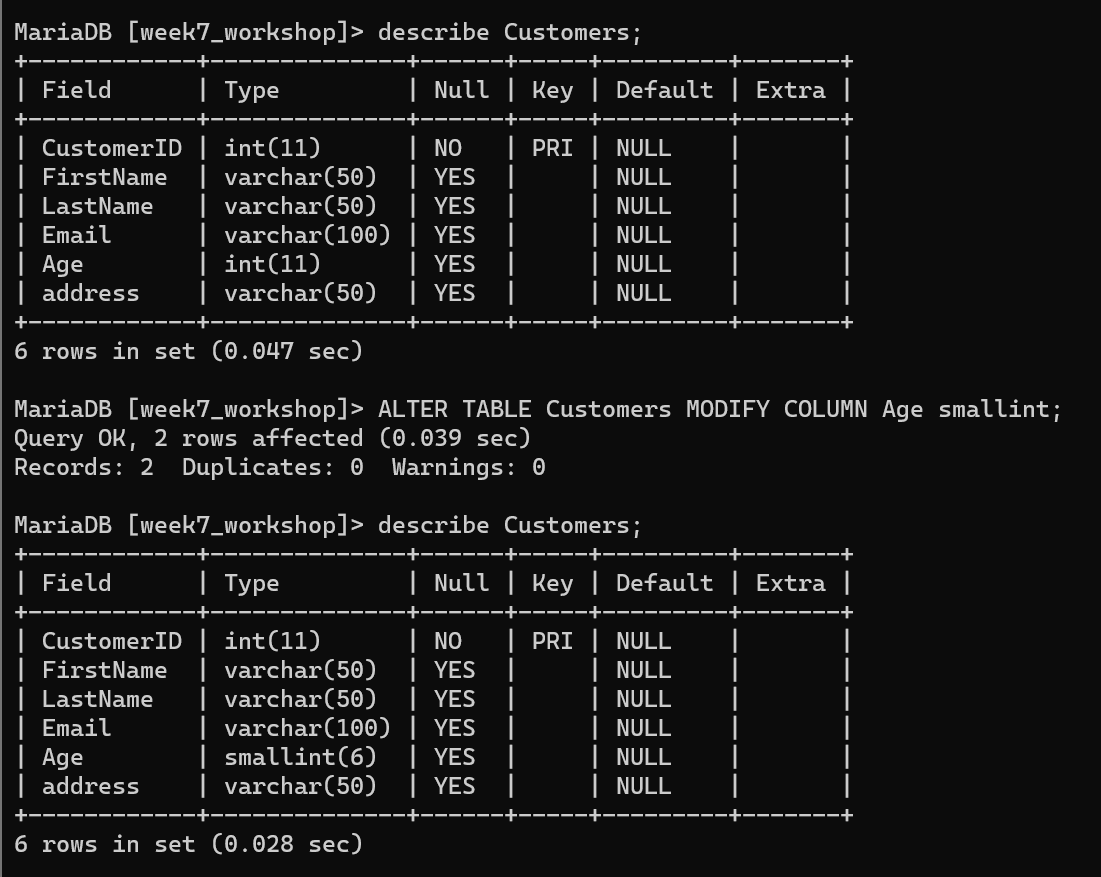
* Create a table named "Categories" with the following columns:
  + CategoryID (integer, primary key)
  + CategoryName (varchar(50))



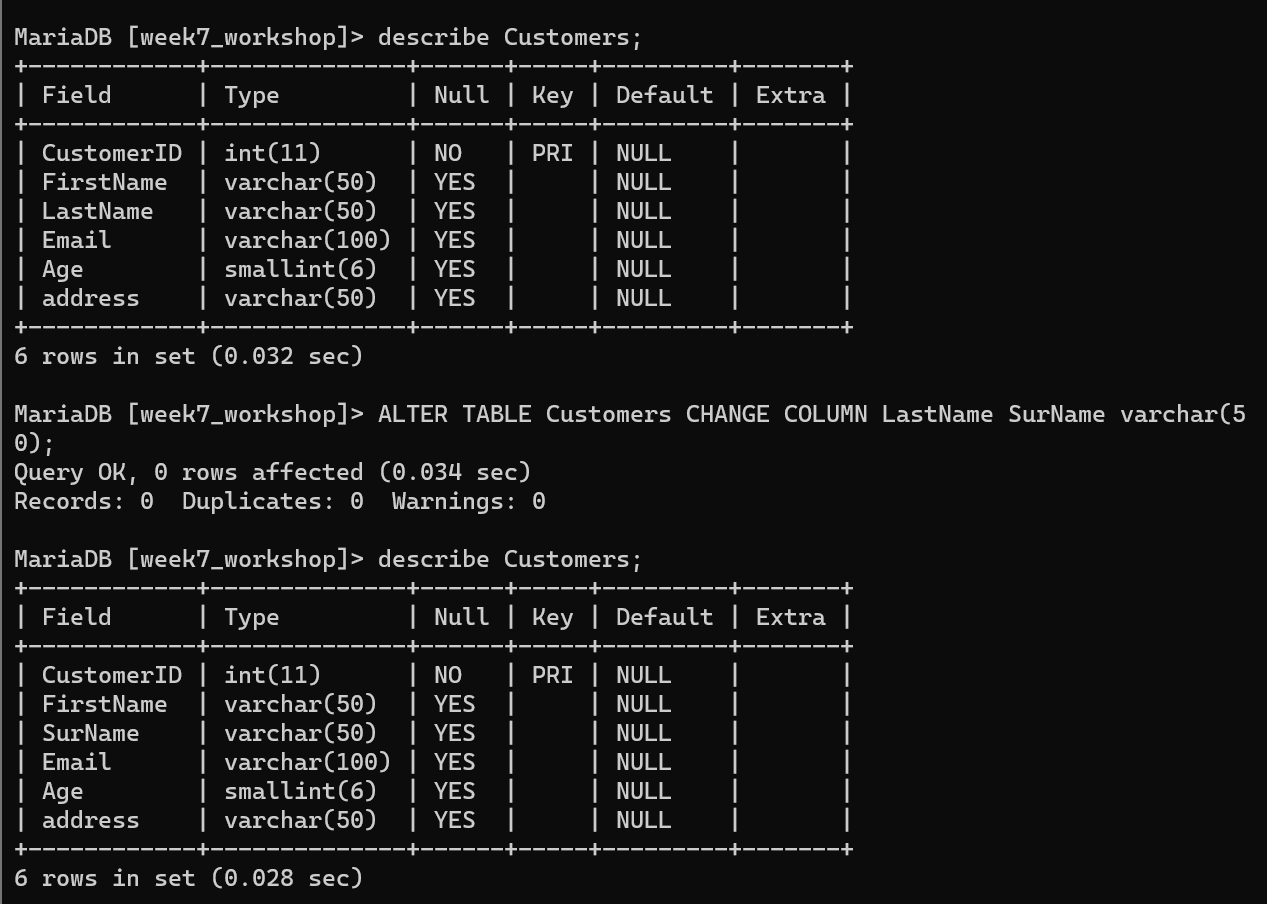
* Create an index on the "ProductName" column in the "Products" table.



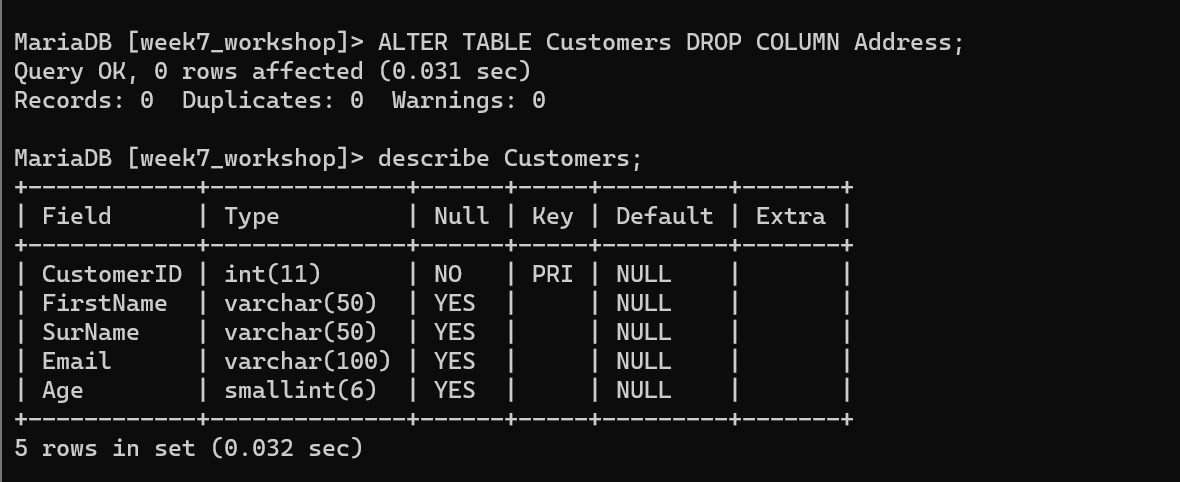
* Modify the data type of the "Age" column in the "Customers" table to smallint.



* Rename the "LastName" column in the "Customers" table to "Surname".

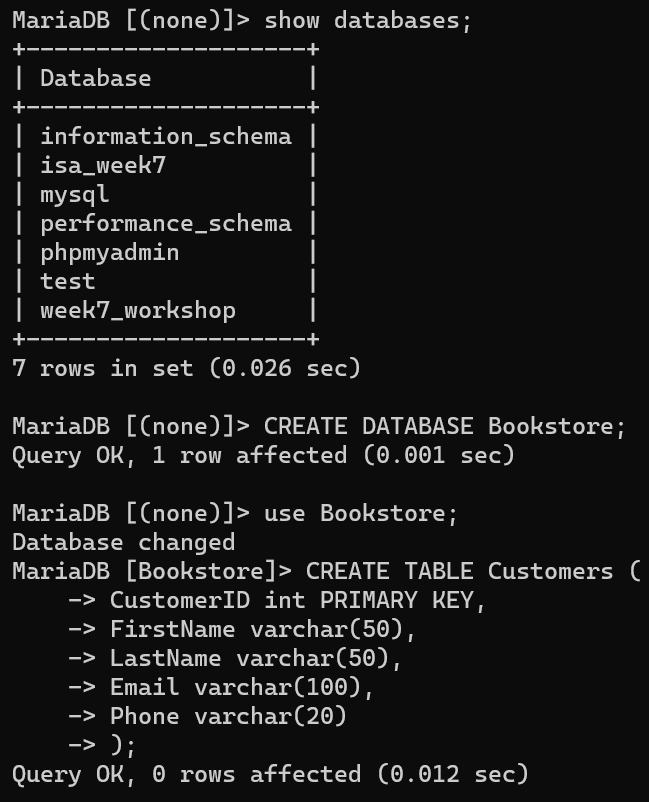


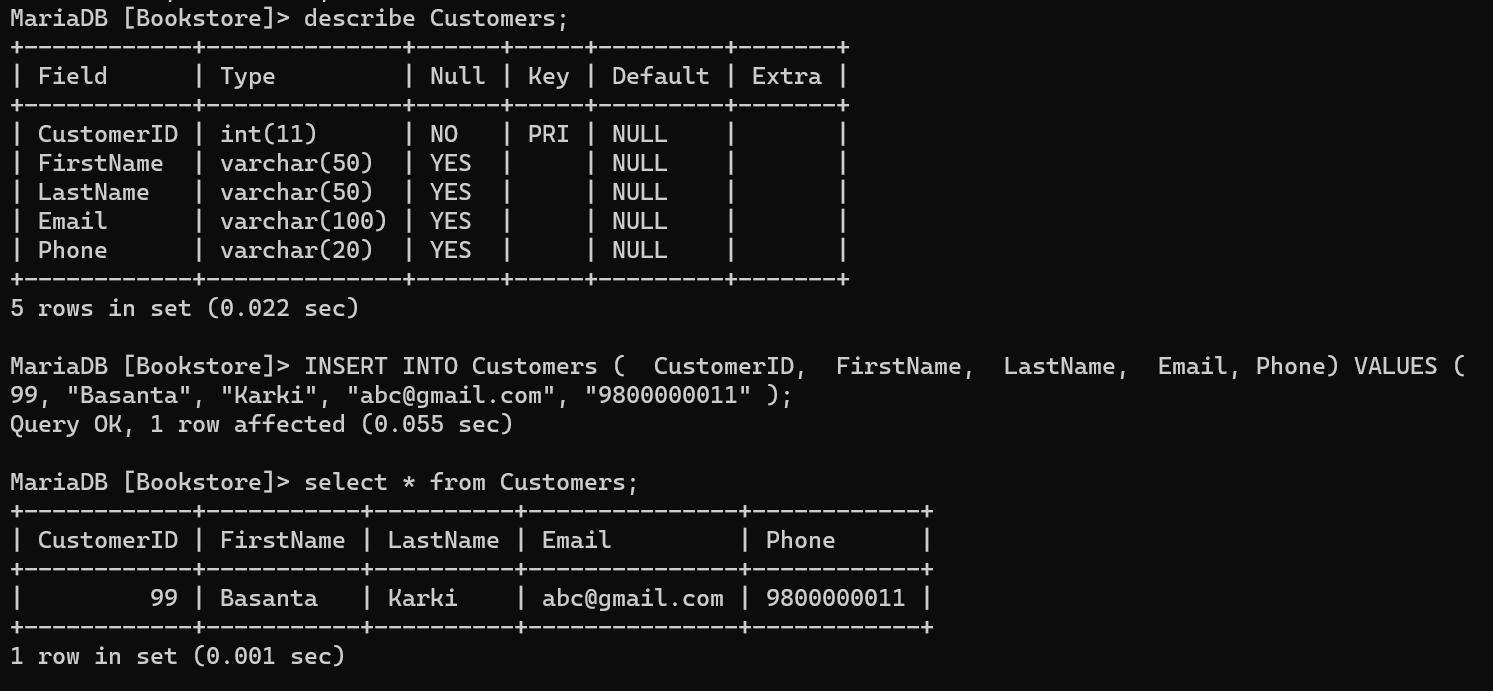
* Remove the "Address" column from the "Customers" table.



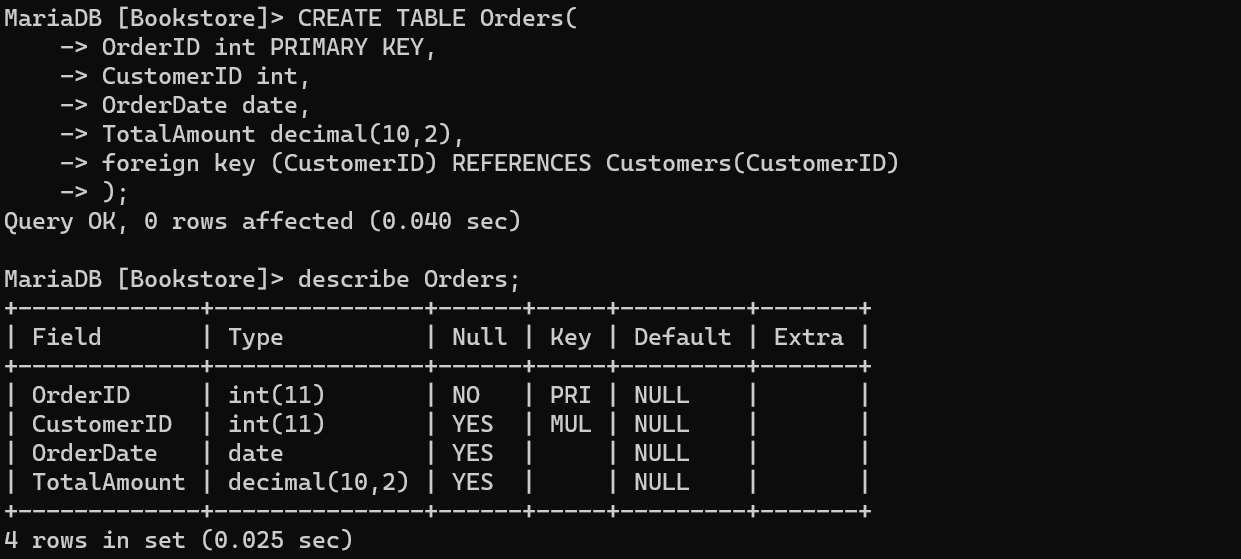
**Task 2**

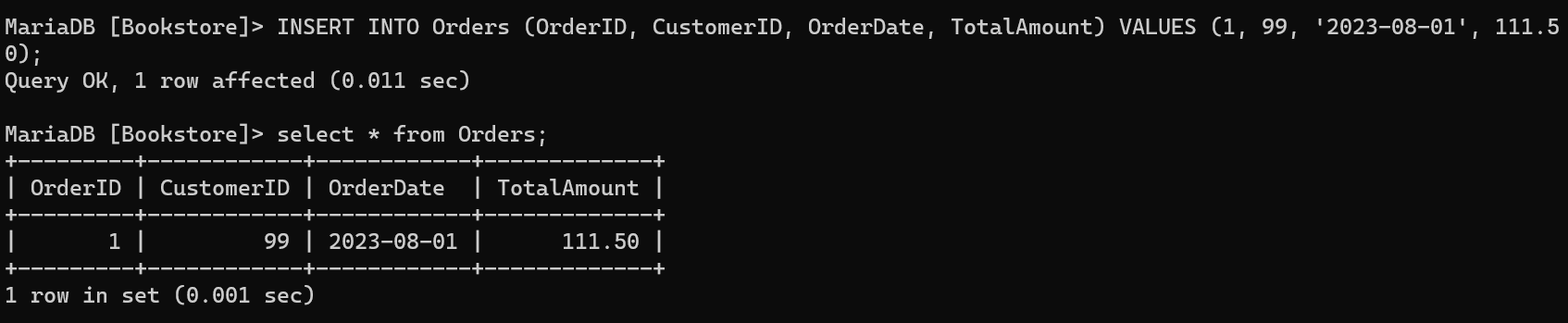
* Create a database named "Bookstore" with the following tables:
  + Customers table with columns:
    - CustomerID (integer, primary key)
    - FirstName (varchar(50))
    - LastName (varchar(50))
    - Email (varchar(100))
    - Phone (varchar(20))



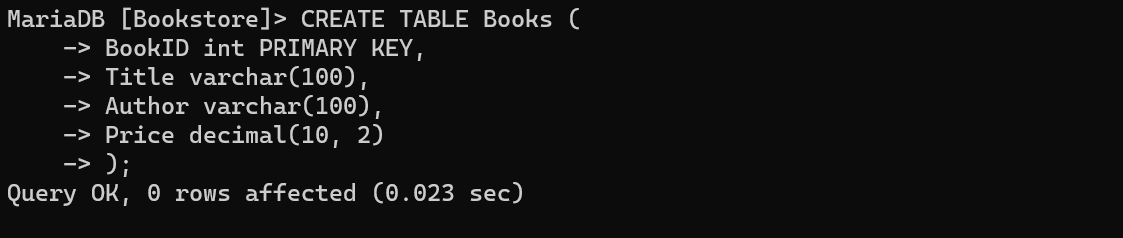


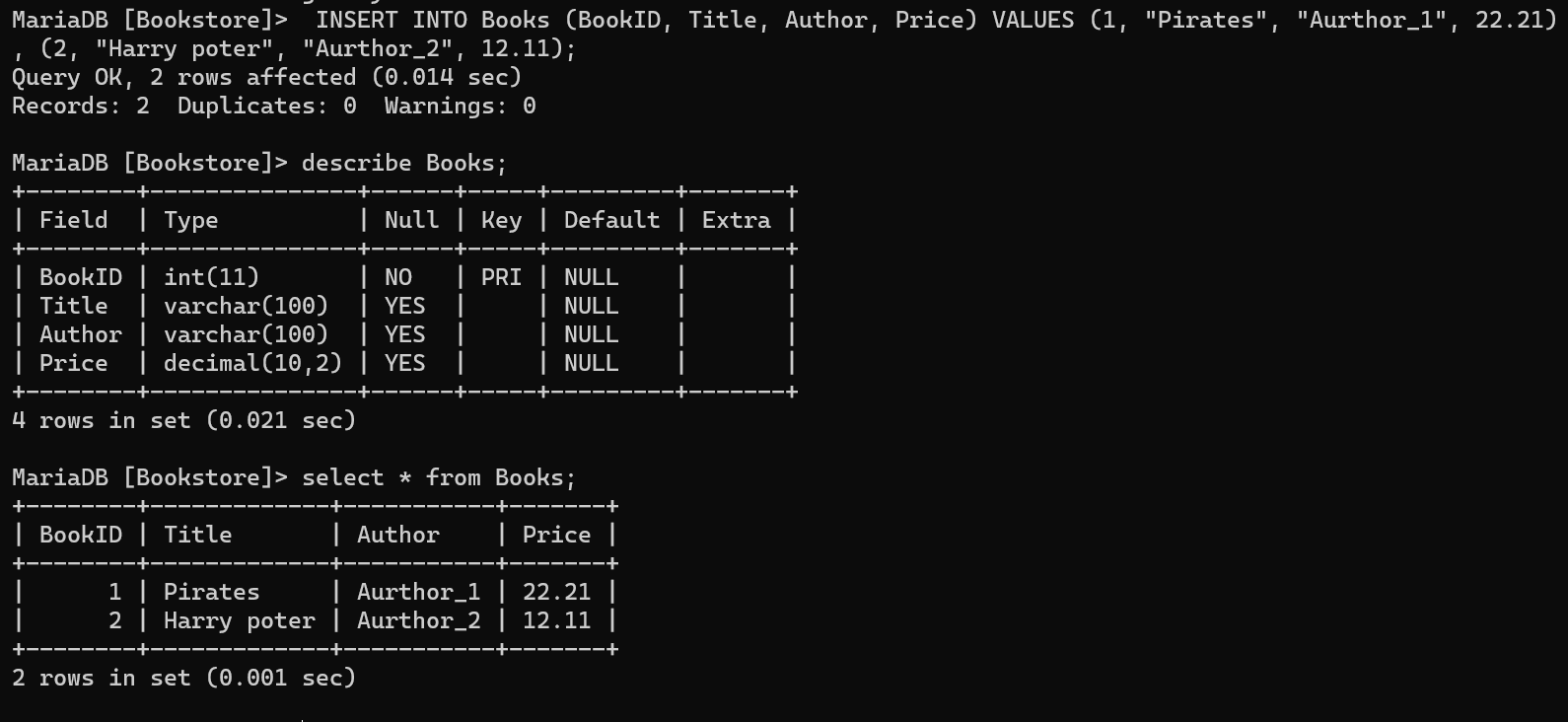
* + Orders table with columns:
    - OrderID (integer, primary key)
    - CustomerID (integer, foreign key referencing CustomerID in the Customers table)
    - OrderDate (date)
    - TotalAmount (decimal(10,2))



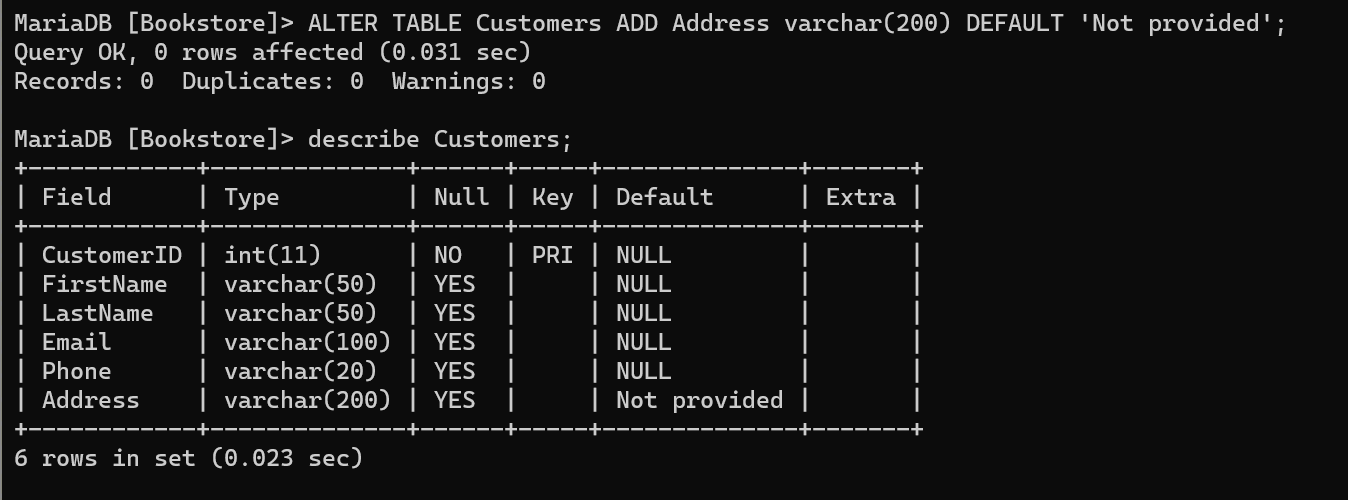


* + Books table with columns:
    - BookID (integer, primary key)
    - Title (varchar(100))
    - Author (varchar(100))
    - Price (decimal(10,2))

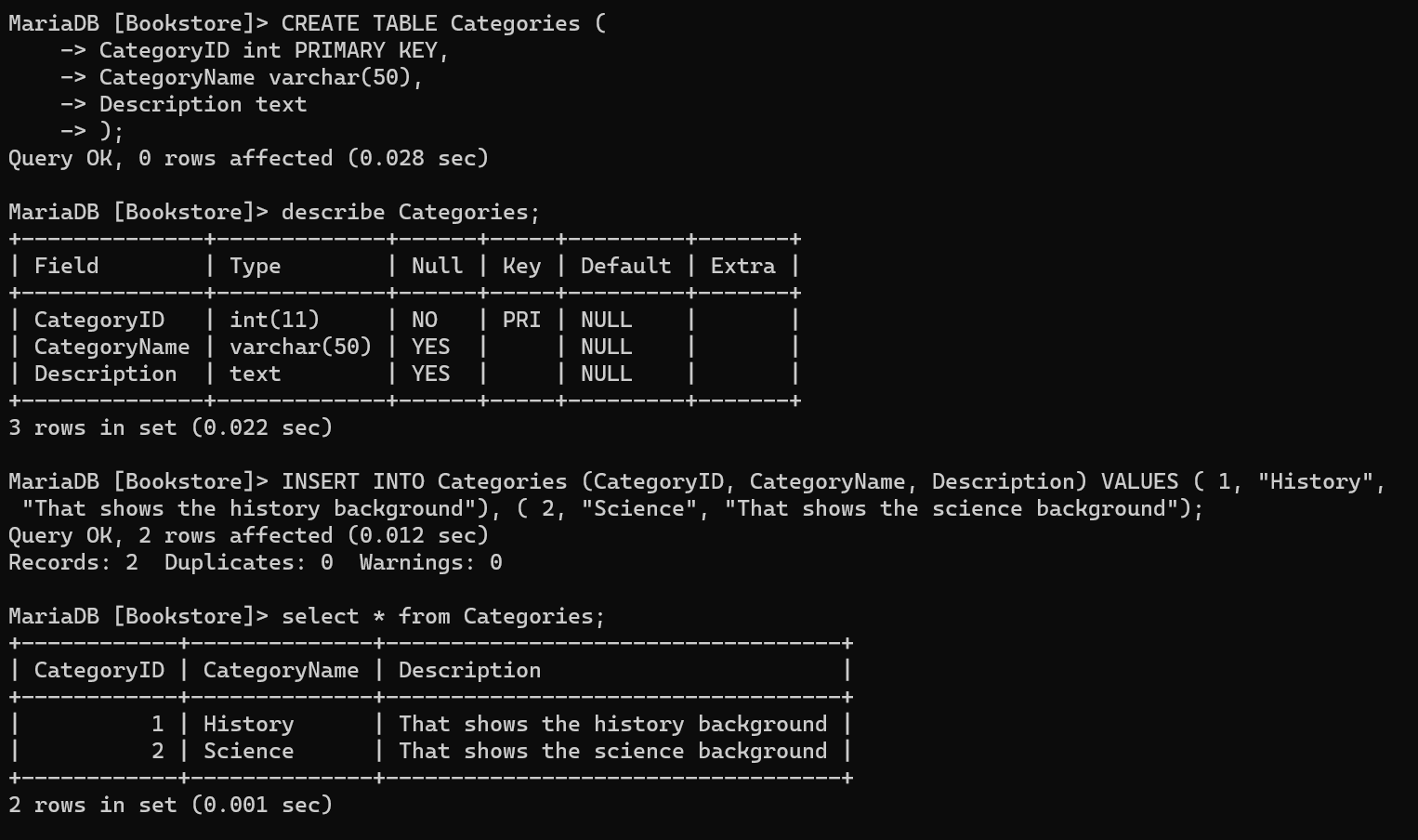




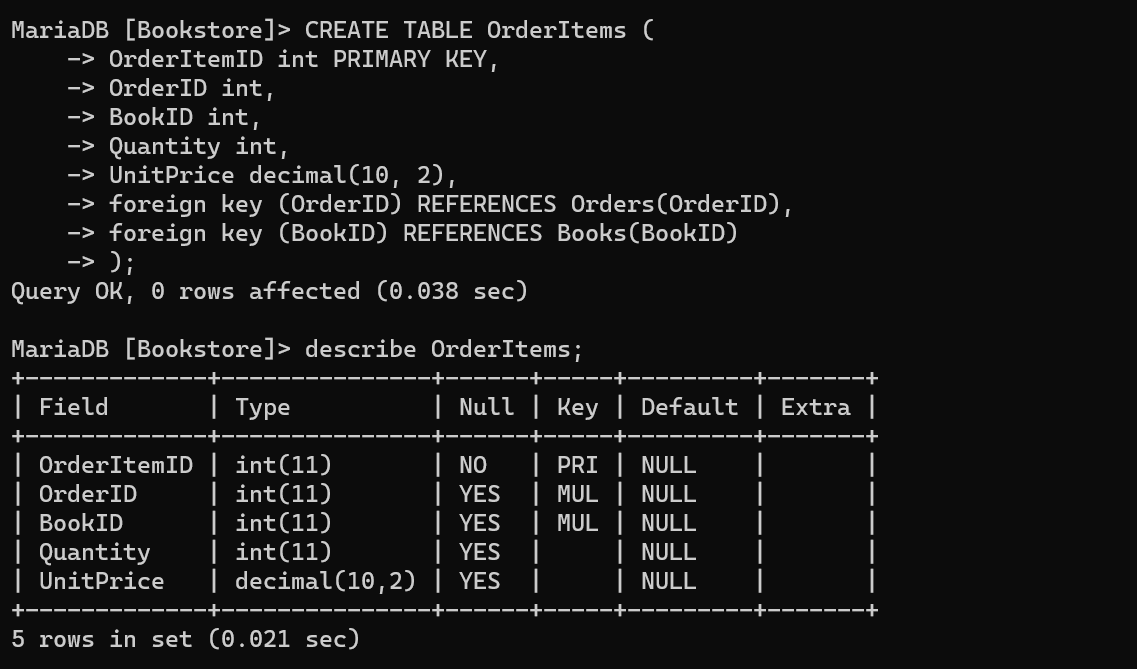
* Add a column named "Address" to the "Customers" table with the data type varchar(200) and set a default value of 'Not provided'.

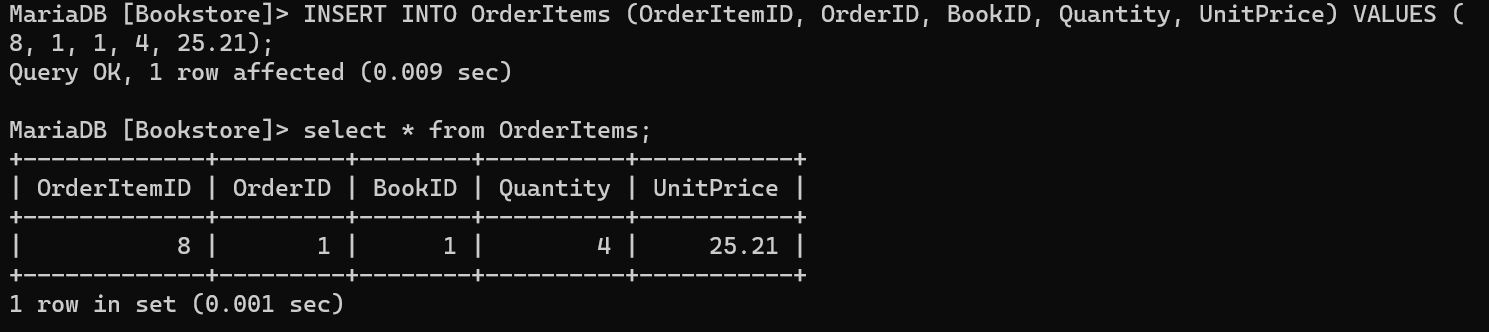


* Create a table named "Categories" with the following columns:
  + CategoryID (integer, primary key)
  + CategoryName (varchar(50))
  + Description (text)

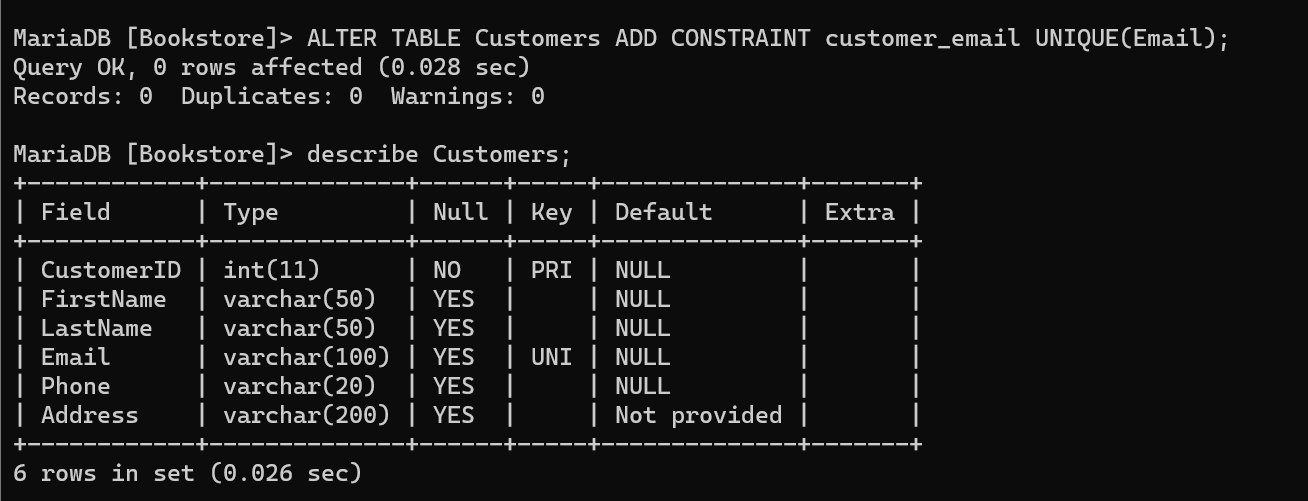


* Create a table named "OrderItems" with the following columns:
  + OrderItemID (integer, primary key)
  + OrderID (integer, foreign key referencing OrderID in the Orders table)
  + BookID (integer, foreign key referencing BookID in the Books table)
  + Quantity (integer)
  + UnitPrice (decimal(10,2))

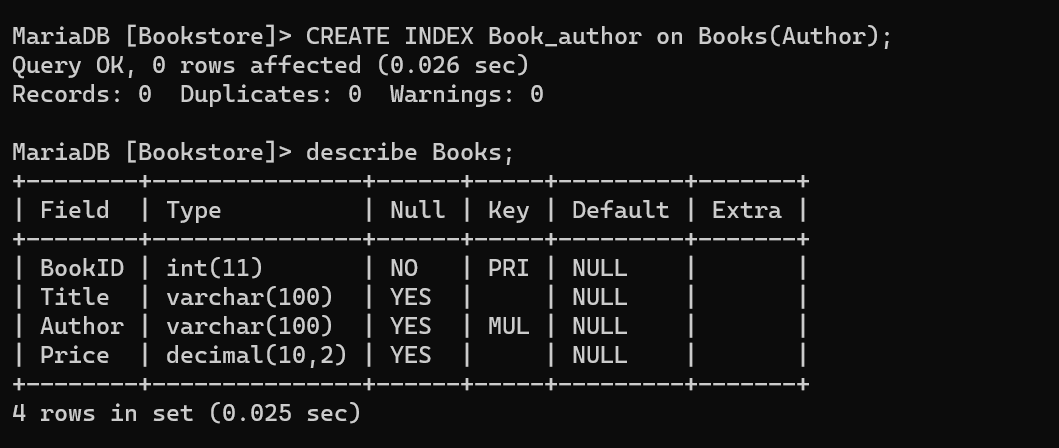




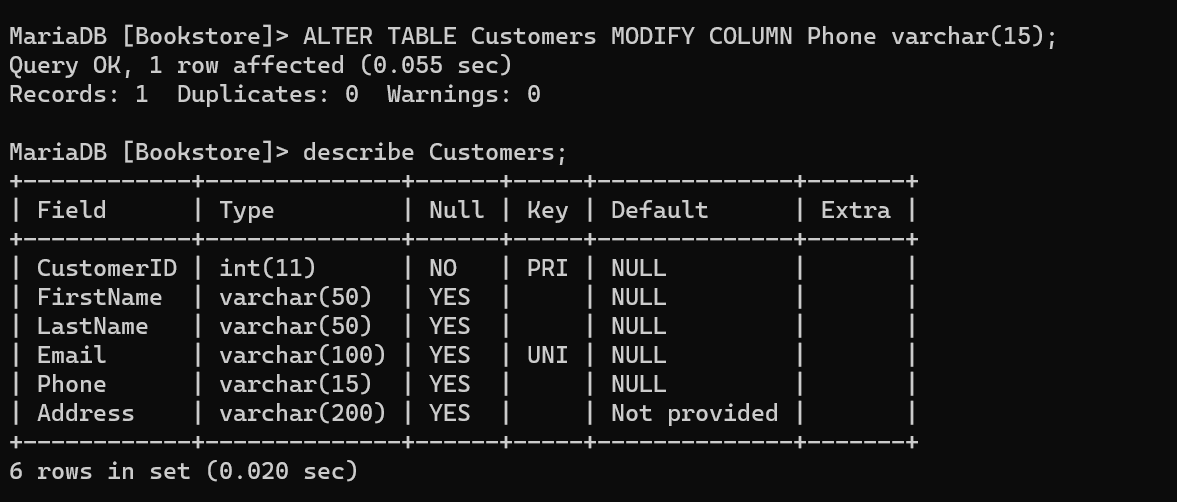
* Add a unique constraint to the "Email" column in the "Customers" table to ensure that each email address is unique.



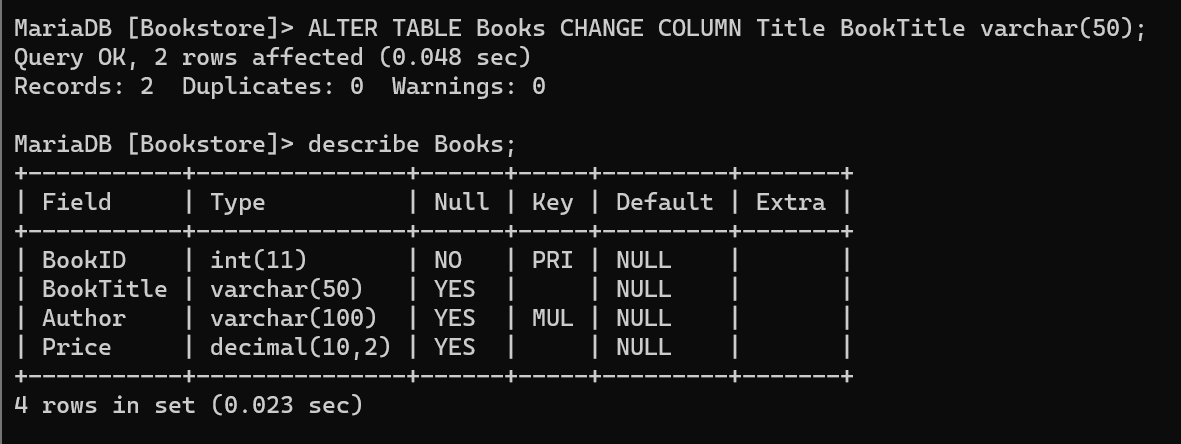
* Create an index on the "Author" column in the "Books" table.



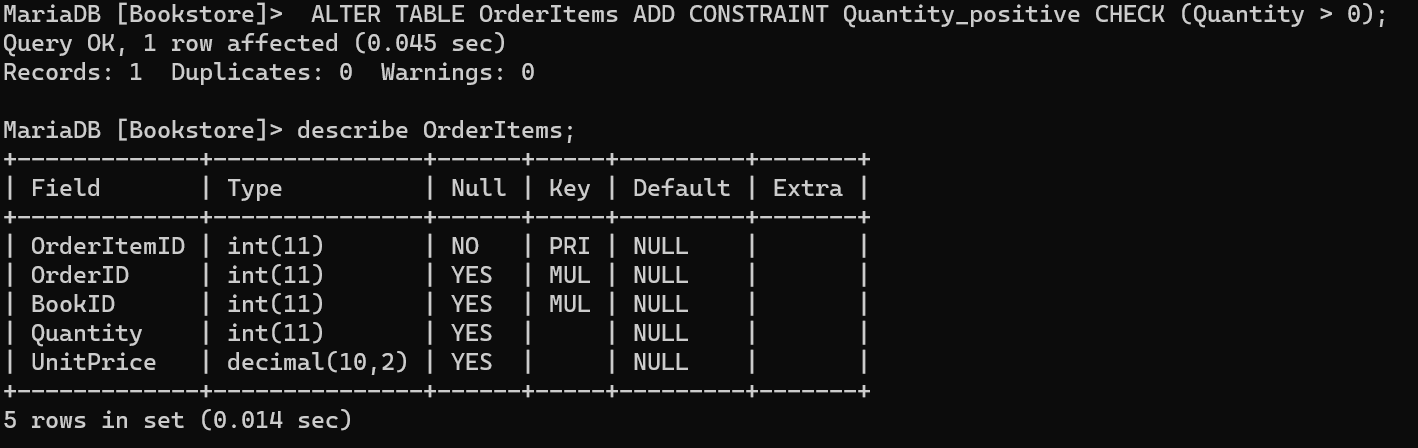
* Modify the data type of the "Phone" column in the "Customers" table to varchar(15) to accommodate international phone numbers.



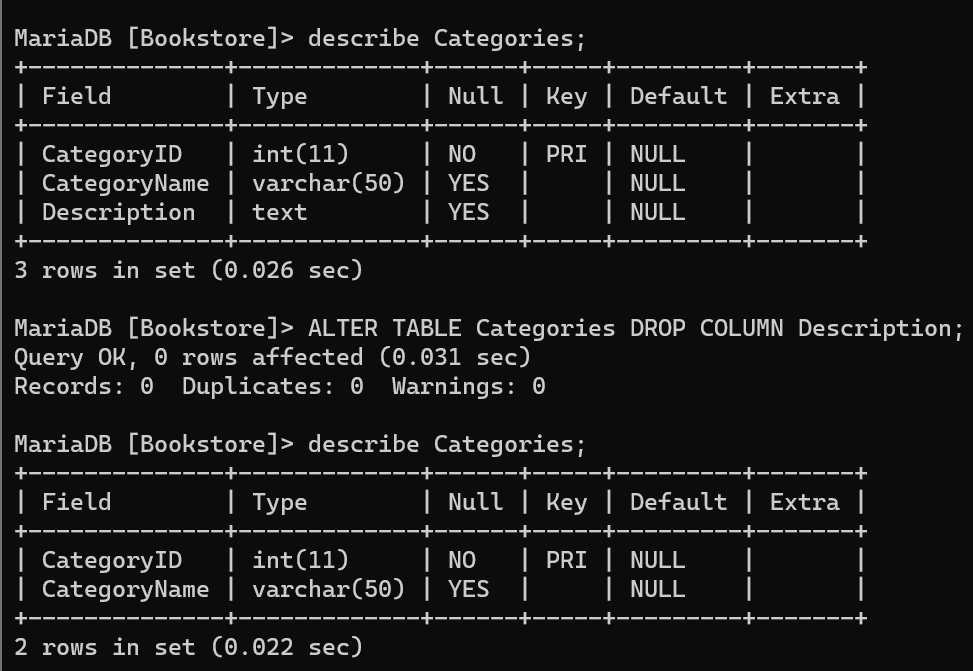
* Rename the "Title" column in the "Books" table to "BookTitle".



* Add a check constraint to the "Quantity" column in the "OrderItems" table to ensure that the quantity is greater than zero.

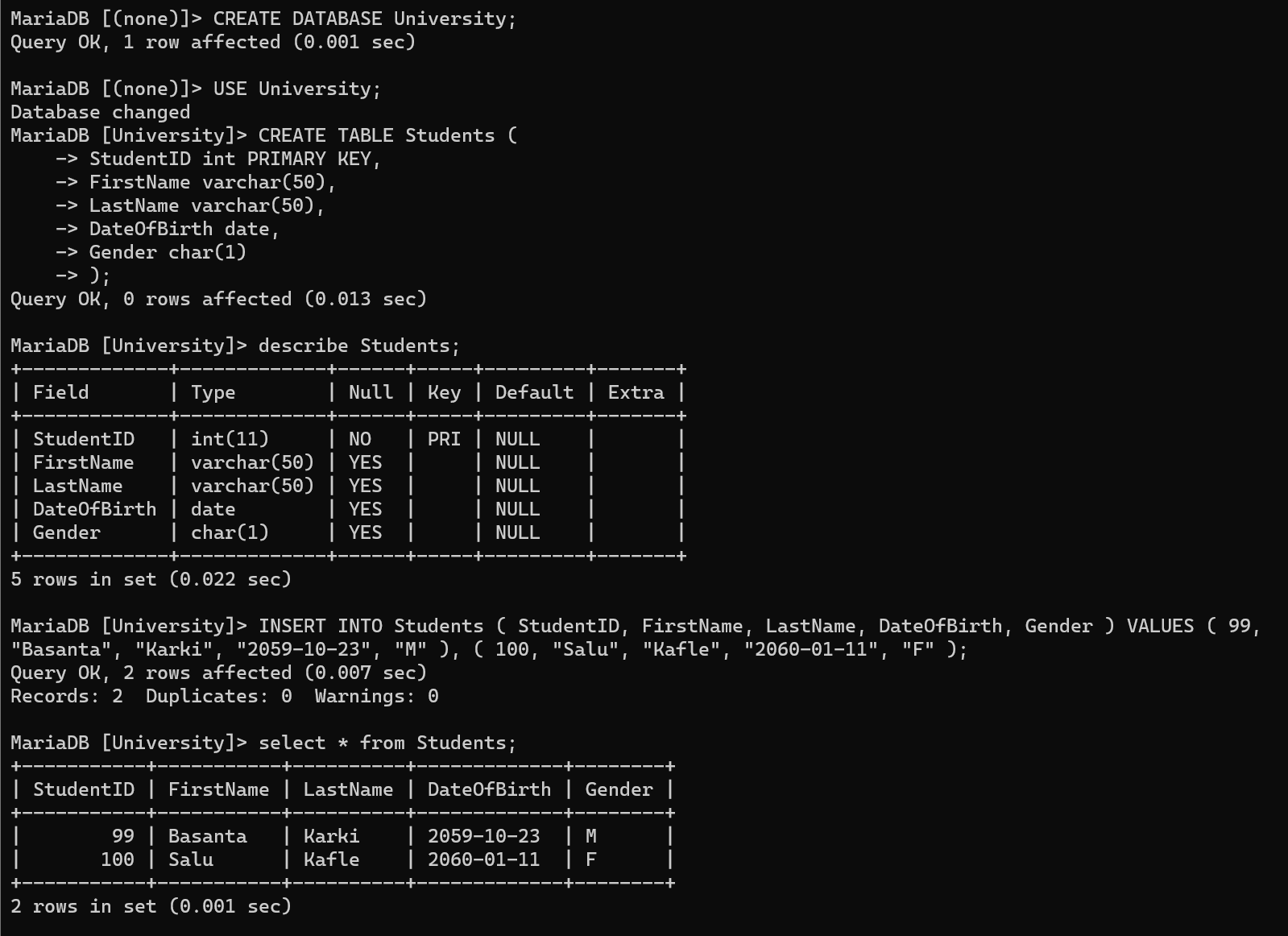


* Remove the "Description" column from the "Categories" table.

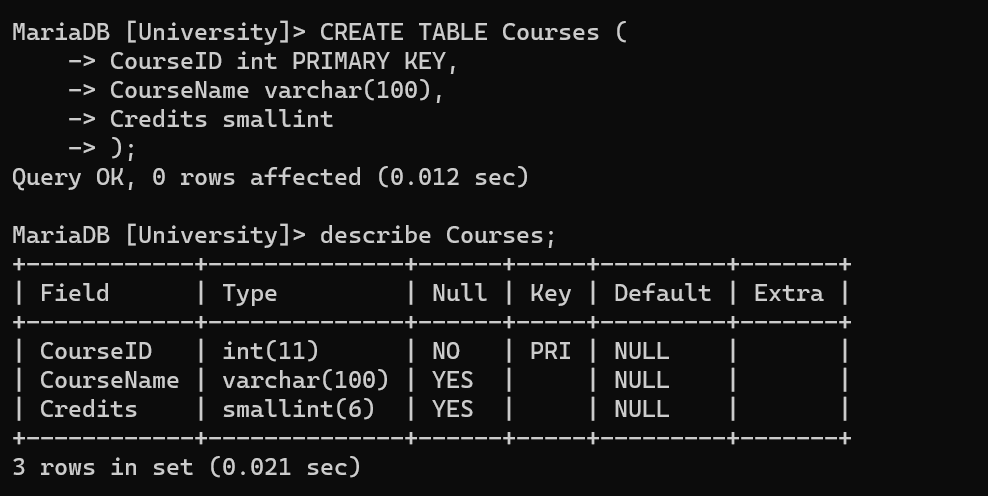


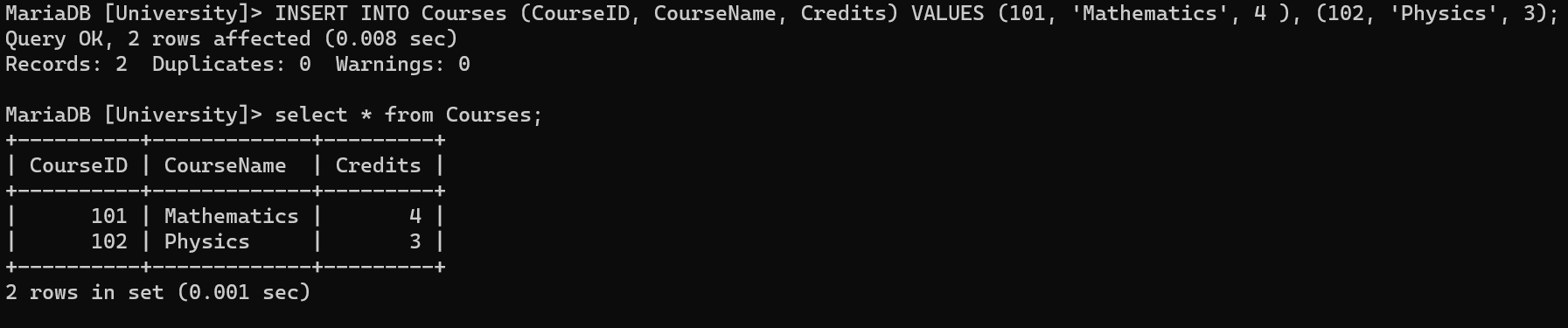
**Task 3**

* Create a database named "University" with the following tables:
  + Students table with columns:
    - StudentID (integer, primary key)
    - FirstName (varchar(50))
    - LastName (varchar(50))
    - DateOfBirth (date)
    - Gender (char(1)) [M for male, F for female]

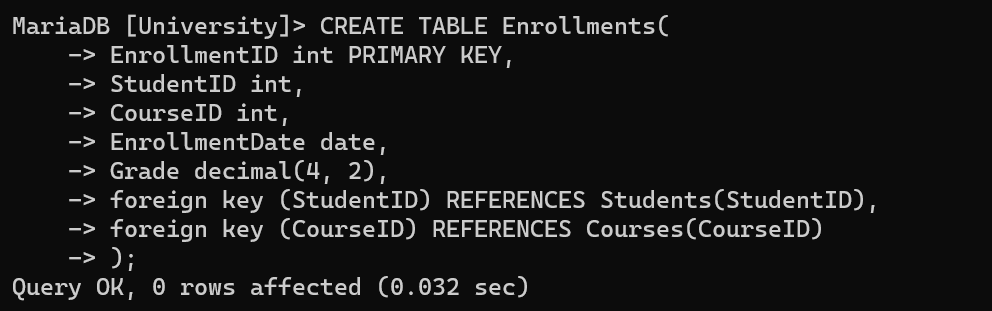


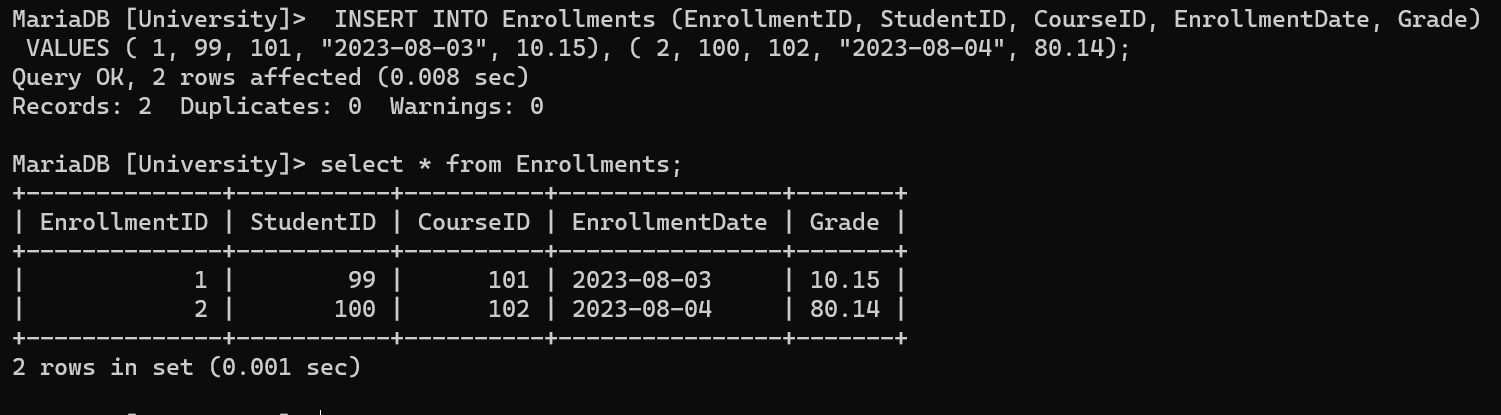
* + Courses table with columns:
    - CourseID (integer, primary key)
    - CourseName (varchar(100))
    - Credits (smallint)



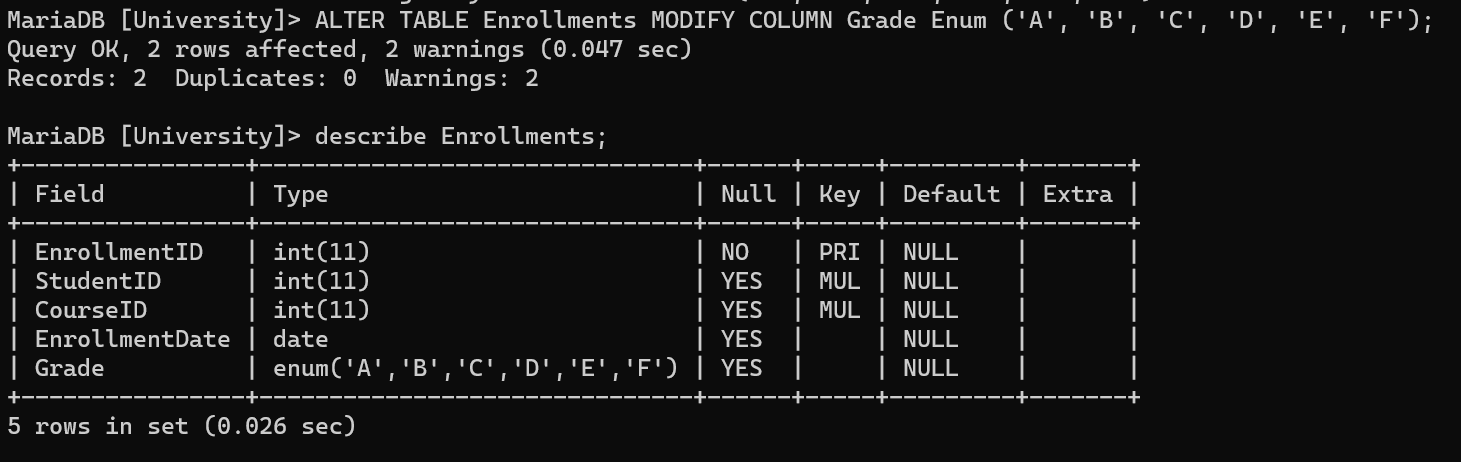


* + Enrollments table with columns:
    - EnrollmentID (integer, primary key)
    - StudentID (integer, foreign key referencing StudentID in the Students table)
    - CourseID (integer, foreign key referencing CourseID in the Courses table)
    - EnrollmentDate (date)
    - Grade (decimal(4, 2))

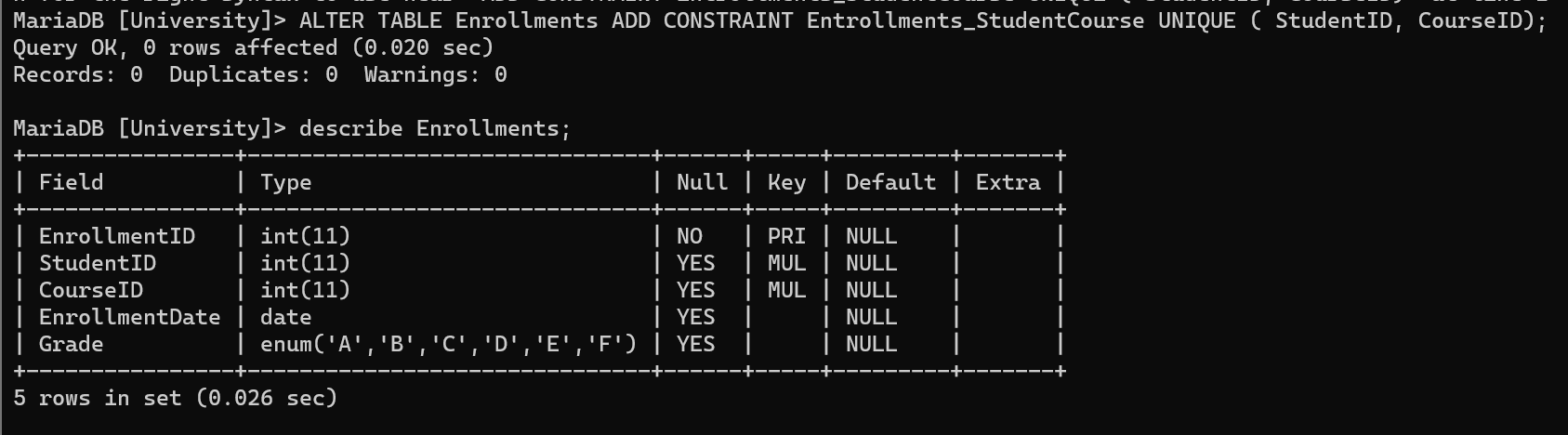




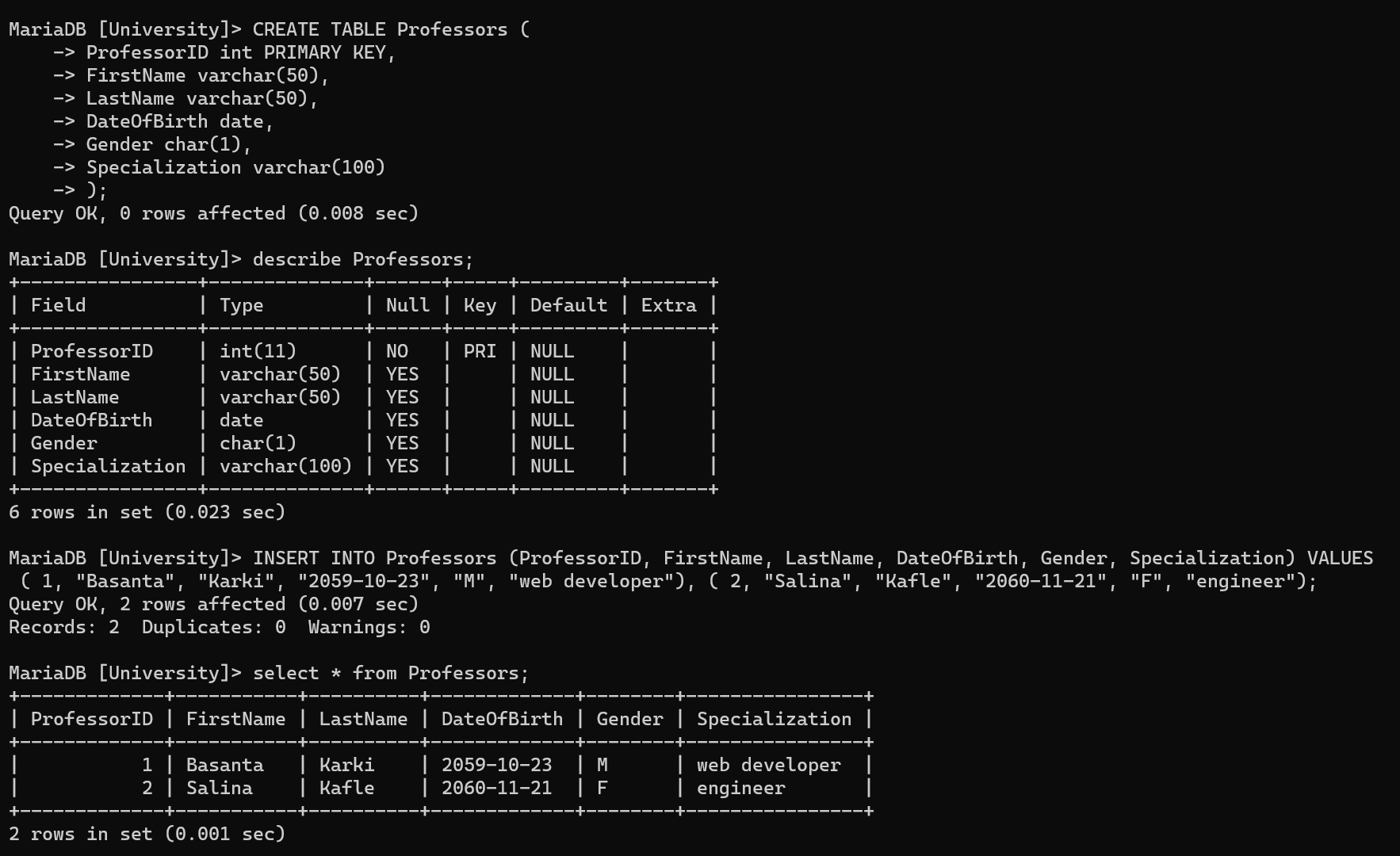
* Create an ENUM type named "GradeEnum" with values ('A', 'B', 'C', 'D', 'E', 'F') representing the possible grades in the "Enrollments" table.



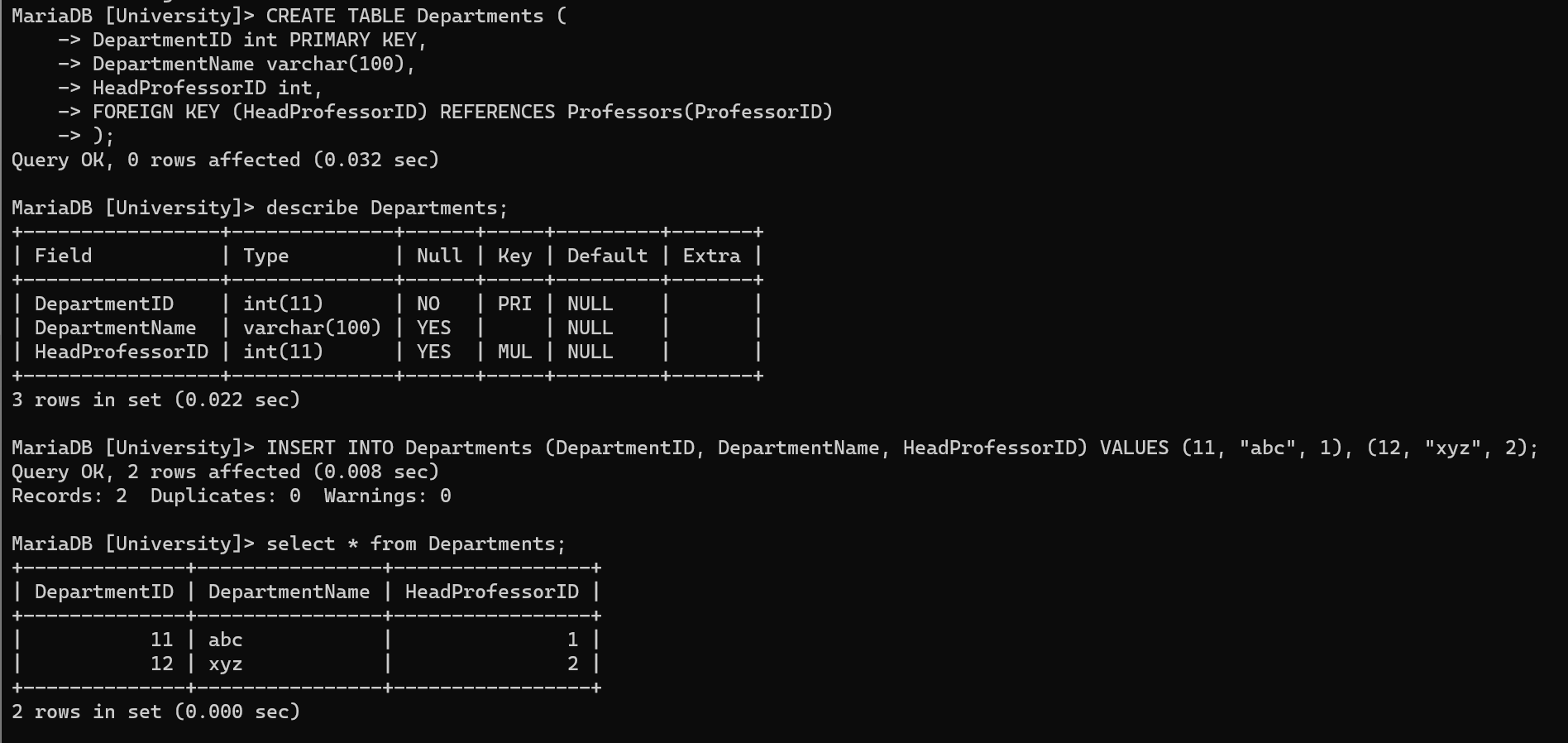
* Add a UNIQUE constraint to the combination of "StudentID" and "CourseID" in the "Enrollments" table to prevent duplicate enrollments.



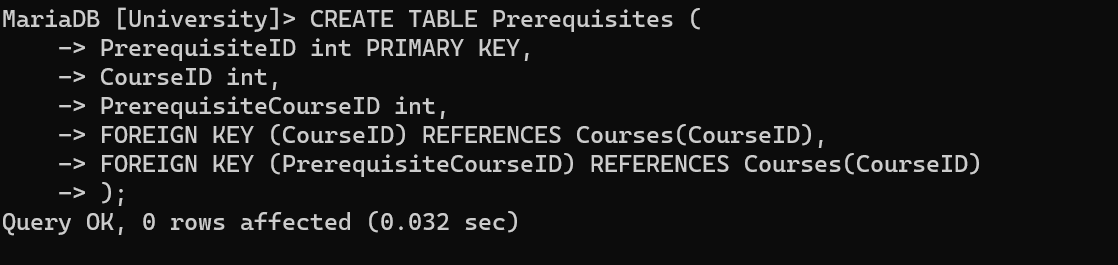
* Create a table named "Professors" with the following columns:
  + ProfessorID (integer, primary key)
  + FirstName (varchar(50))
  + LastName (varchar(50))
  + DateOfBirth (date)
  + Gender (char(1))
  + Specialization (varchar(100))

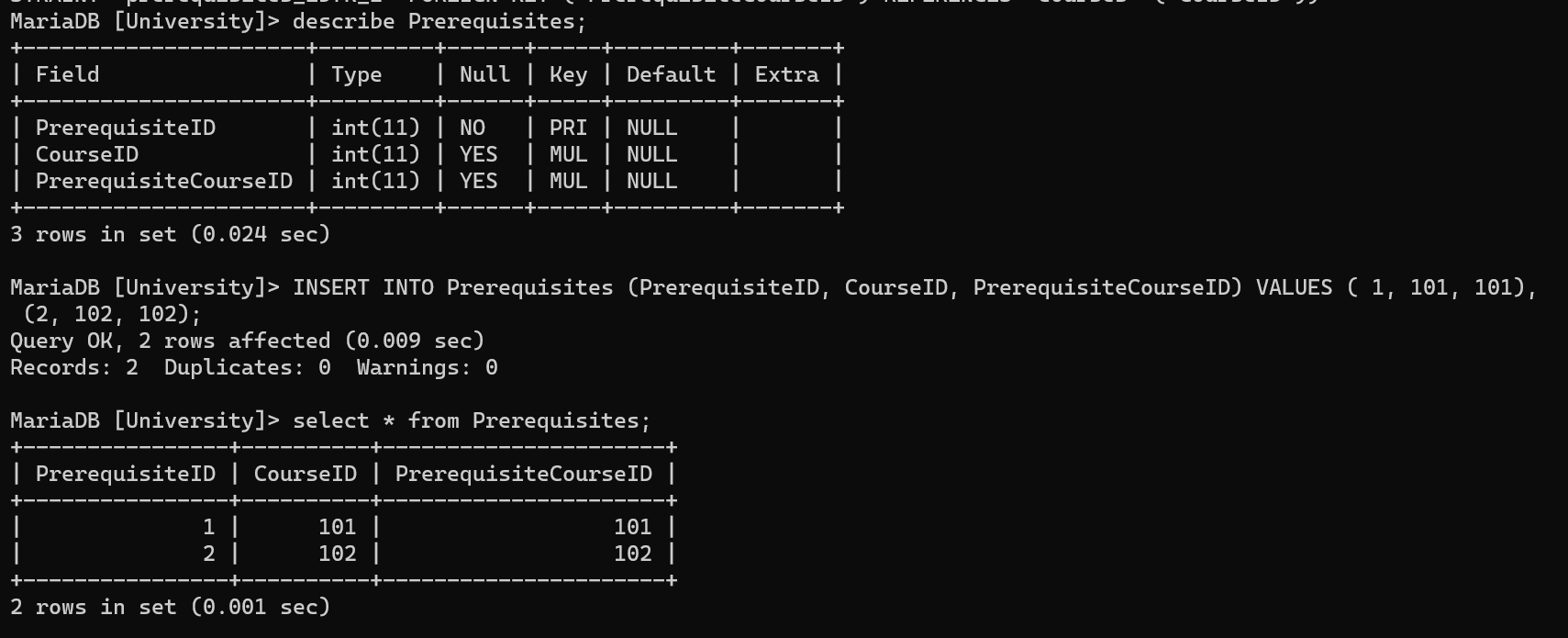


* Add a CHECK constraint to the "DateOfBirth" column in the "Students" and "Professors" tables to ensure that the date of birth is not in the future.
* Create a table named "Departments" with the following columns:
  + DepartmentID (integer, primary key)
  + DepartmentName (varchar(100))
  + HeadProfessorID (integer, foreign key referencing ProfessorID in the Professors table)



* Create a table named "Prerequisites" to establish relationships between courses. The table should have the following columns:
  + PrerequisiteID (integer, primary key)
  + CourseID (integer, foreign key referencing CourseID in the Courses table)
  + PrerequisiteCourseID (integer, foreign key referencing CourseID in the Courses table)





* Create a view named "StudentGrades" that displays the following information:
  + StudentID
  + FirstName
  + LastName
  + CourseID
  + CourseName
  + Grade

