**CPSC 50900 Database Systems Project**

**Name:** Jaya Soma Sai Haswanth Bandaru

**Initial Proposal:**

A new software development startup company who managed all of their previous projects in excel sheets has requested me to develop an internal project management tool for their startup to track all of their projects and keep the customer up-to-date about the progress. Other than that, the company also requires a centralised system to store all the documents (for example ER Diagrams, UML Diagrams, SPS). This system will manage different projects and each project will have multiple tasks which will be assigned to different employees. So, the company wants to keep track of all the tasks, their status, and who is working/worked on the task. In order to develop this system, a database solution is required to store the information of the projects, customers, employees, documents and tasks of a project.

Initially, the company aims to migrate all the old data through Excel/CSV files, so the initial data source of the system will be files. After data is completely migrated, an online web system will be used as a source of the data.

**Relational Database Design Process**

The system will be using the following 5 major entities:

1. Project
2. Customer
3. Employee
4. Tasks
5. Documents

**Project:**

The project entity will be used to store all the information relevant to the project. Following are the attributes for the project entity:

* ID - Unique ID that identifies a project
* Title - Name of the project
* Project Type - The type of the project (Development, Research, etc.)

**Customer:**

The customer entity will store all the information about the customer.

Following are the attributes for the customer entity:

* Name - The name of the client
* Email - The email address of the client
* Industry - The industry where the client works (Engineering, Medical, etc.)

**Employee:**

The employee entity will store all the information of the company’s employees.

* Name - The name of the employee
* Email - Email address of the employee
* Role - The role of the employee in the company (Manager, Developer, etc.)

**Tasks:**

The tasks associated with a project. It will have the following attributes:

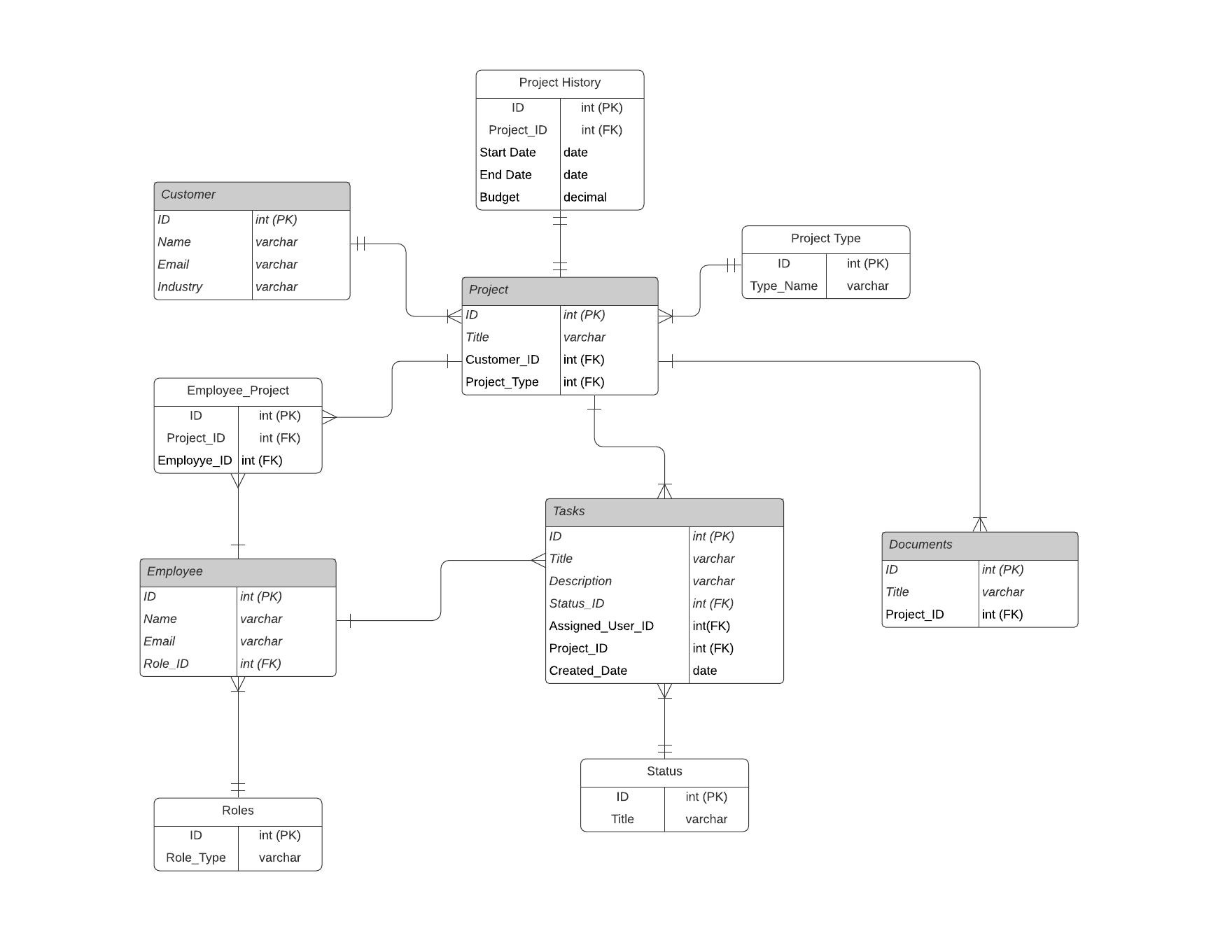
* ID - The unique ID of the task
* Title - The name of the task
* Assigned User - Name of the person who is working on it

**Documents:**

The documents associated with projects. It will have the following attributes:

* ID - The unique ID of the document
* Title - The name of the document
* Project Name - The name of the project by which the document is associated with

**ER DIAGRAM**

****

**Data Sources**

As described, the initial data source for the system will be delimited files. The system will get data from delimited files to store the information of the project which includes the title of project, information of the client like its name and email, the information of which employee worked on the task along with their names, emails and roles. The documents of the projects will get their title, project name from delimited files. After old data is completely migrated, the company aims to use a web portal as a data source for the system.

* The file for the project will contain the Title of the project, the Customer ID who is the actual sponsor of the project, and the type of the project which indicates the nature of the project.
* The file for the customer will contain the name of the customer, email address of the customer, and the industry from where the customer belongs to in a semi-colon separated format.
* The file for the employee will contain the name of the employee, the email of the employee and the role of the employee in the project in a semi-colon separated format.
* The task file will create all the information of the tasks to perform to complete the project, like title, description, date created, status, idof the assigned employee and the id of the project in a semi-colon separated format.
* The documents file will contain all the information regarding the documents a project has like title and project id in a semi-colon separated format.

**Data Definition Language Scripts**

* **Load Customers Data:**

LOAD DATA LOCAL INFILE

'C:/DATA/dev/work/CPSC 50900 Database Systems Project/Data Sources/customer.csv'

INTO TABLE customer FIELDS TERMINATED BY ';' (Name, Email, Industry);

This command will get the file from the local system, parse according to semicolons separately and insert the data in the customer table.

* **Load Projects Data:**

LOAD DATA LOCAL INFILE

'C:/DATA/dev/work/CPSC 50900 Database Systems Project/Data Sources/project.csv'

INTO TABLE project FIELDS TERMINATED BY ';' (Title, Customer\_ID, Project\_Type);

This command will get the file from the local system, parse according to semicolons separately and insert the data in the projects table.

* **Load Employee Data:**

LOAD DATA LOCAL INFILE

'C:/DATA/dev/work/CPSC 50900 Database Systems Project/Data Sources/employee.csv'

INTO TABLE employee FIELDS TERMINATED BY ';' (Name, Email, Role\_ID);

This command will get the file from the local system, parse according to semicolons separately and insert the data in the employee table.

* **Load Tasks Data:**

LOAD DATA LOCAL INFILE

'C:/DATA/dev/work/CPSC 50900 Database Systems Project/Data Sources/task.csv'

INTO TABLE task FIELDS TERMINATED BY ';' (Title, Description, Status\_ID, Assigned\_User\_ID, Project\_ID);

This command will get the file from the local system, parse according to semicolons separately and insert the data in the tasks table.

* **Load Documents Data:**

LOAD DATA LOCAL INFILE

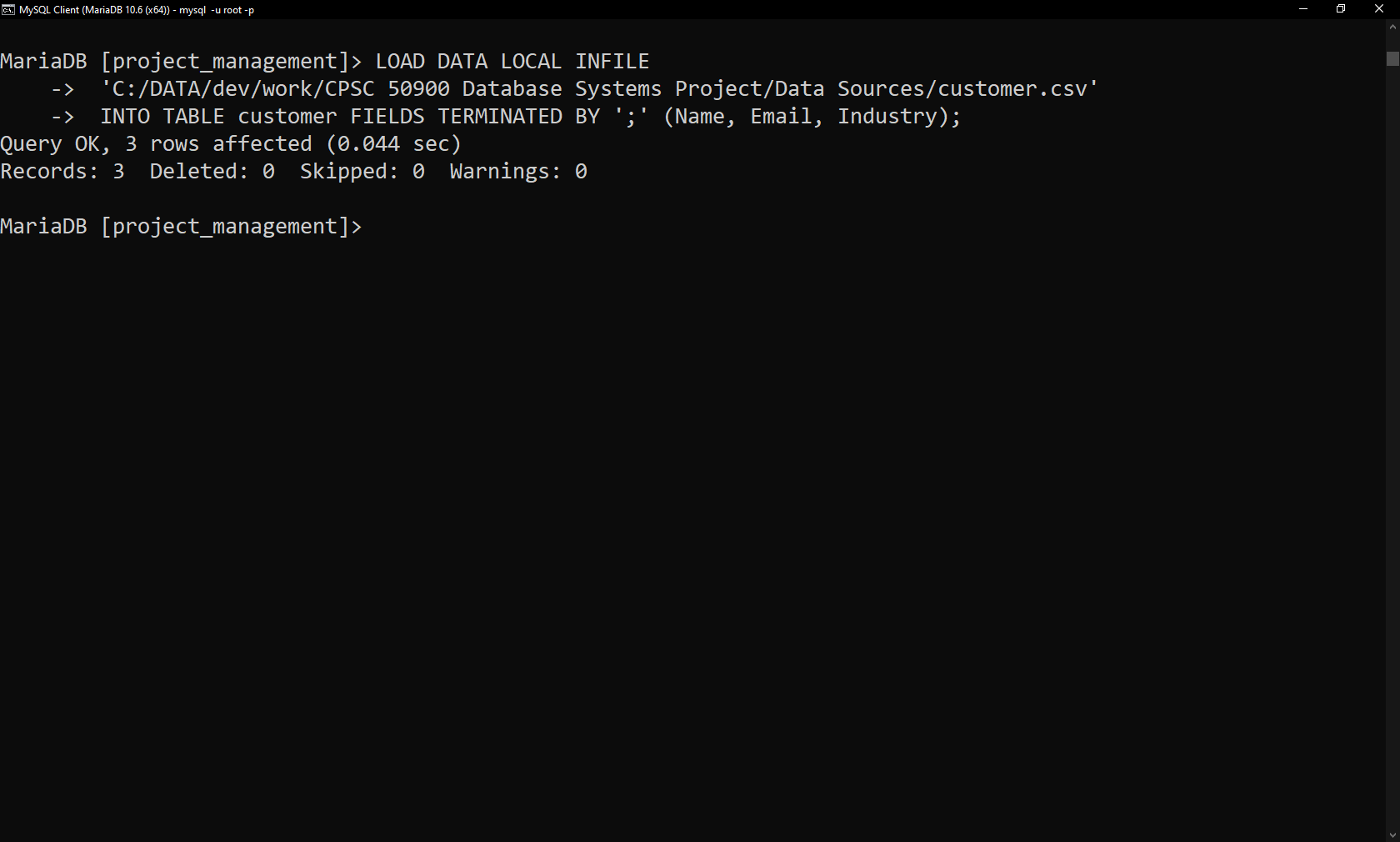
'C:/DATA/dev/work/CPSC 50900 Database Systems Project/Data Sources/documents.csv'

INTO TABLE documents FIELDS TERMINATED BY ';' (Title, Project\_ID);

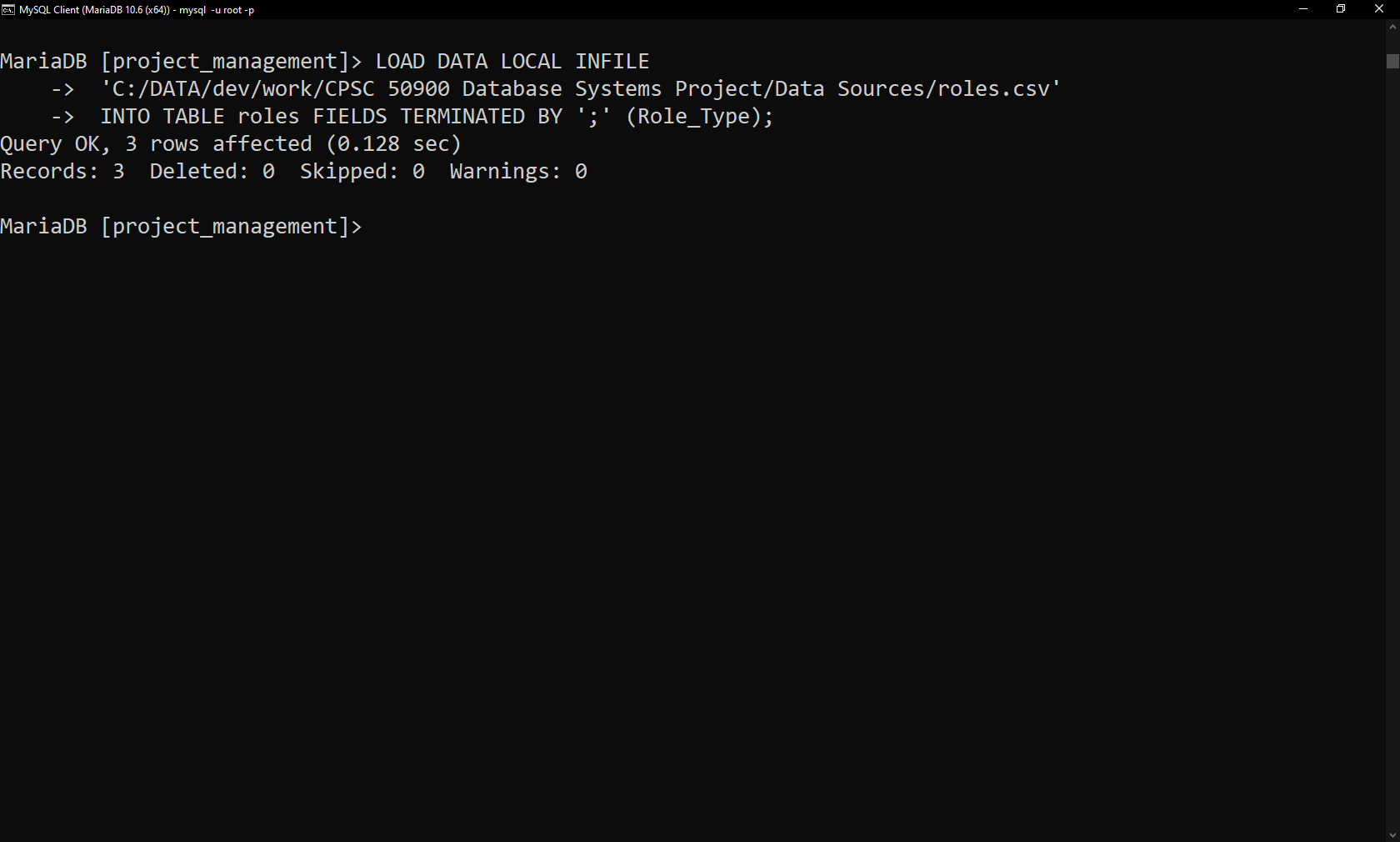
This command will get the file from the local system, parse according to semicolons separately and insert the data in the documents table.

**Successful data load screenshots:**

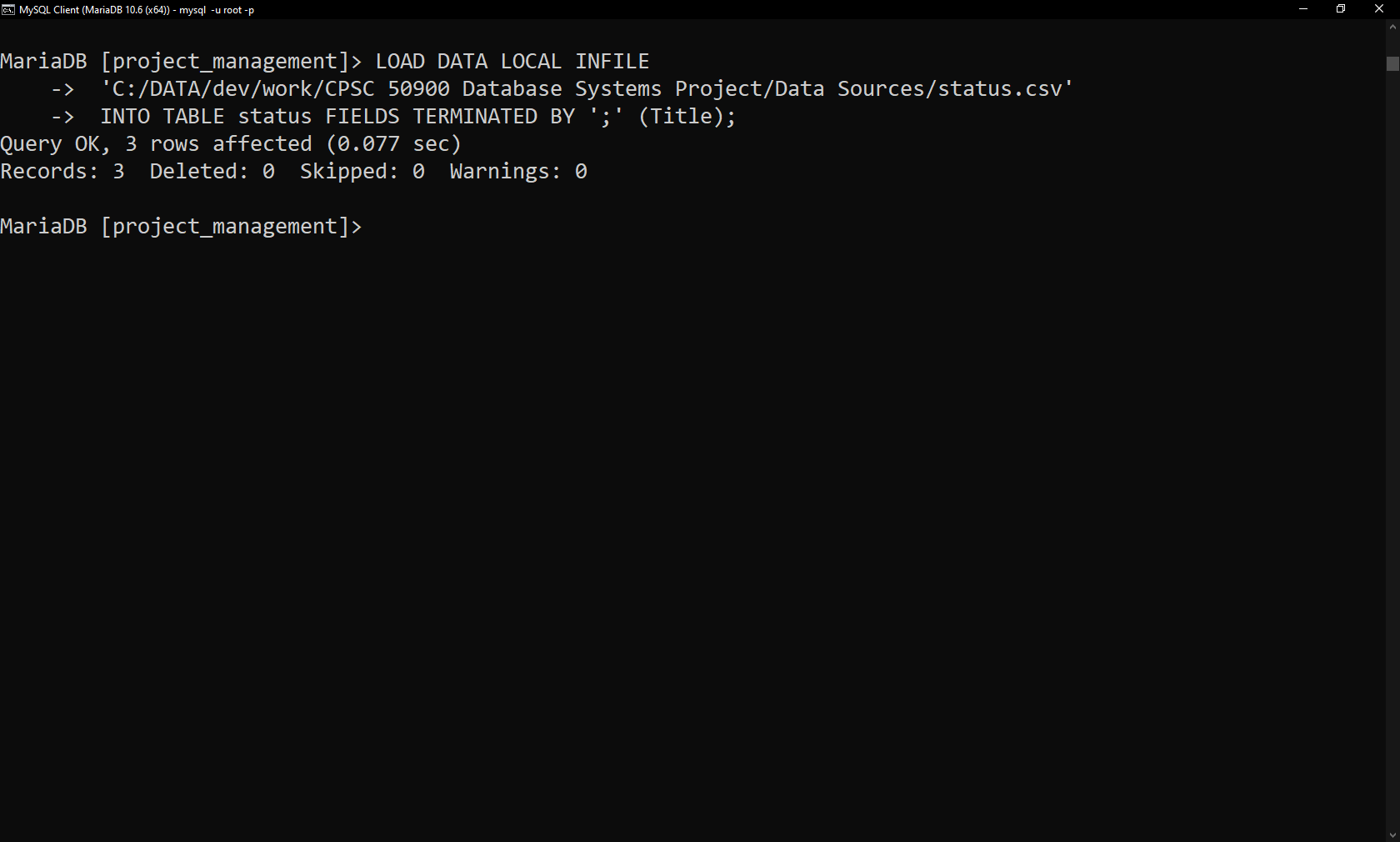
Load data in Customer Table



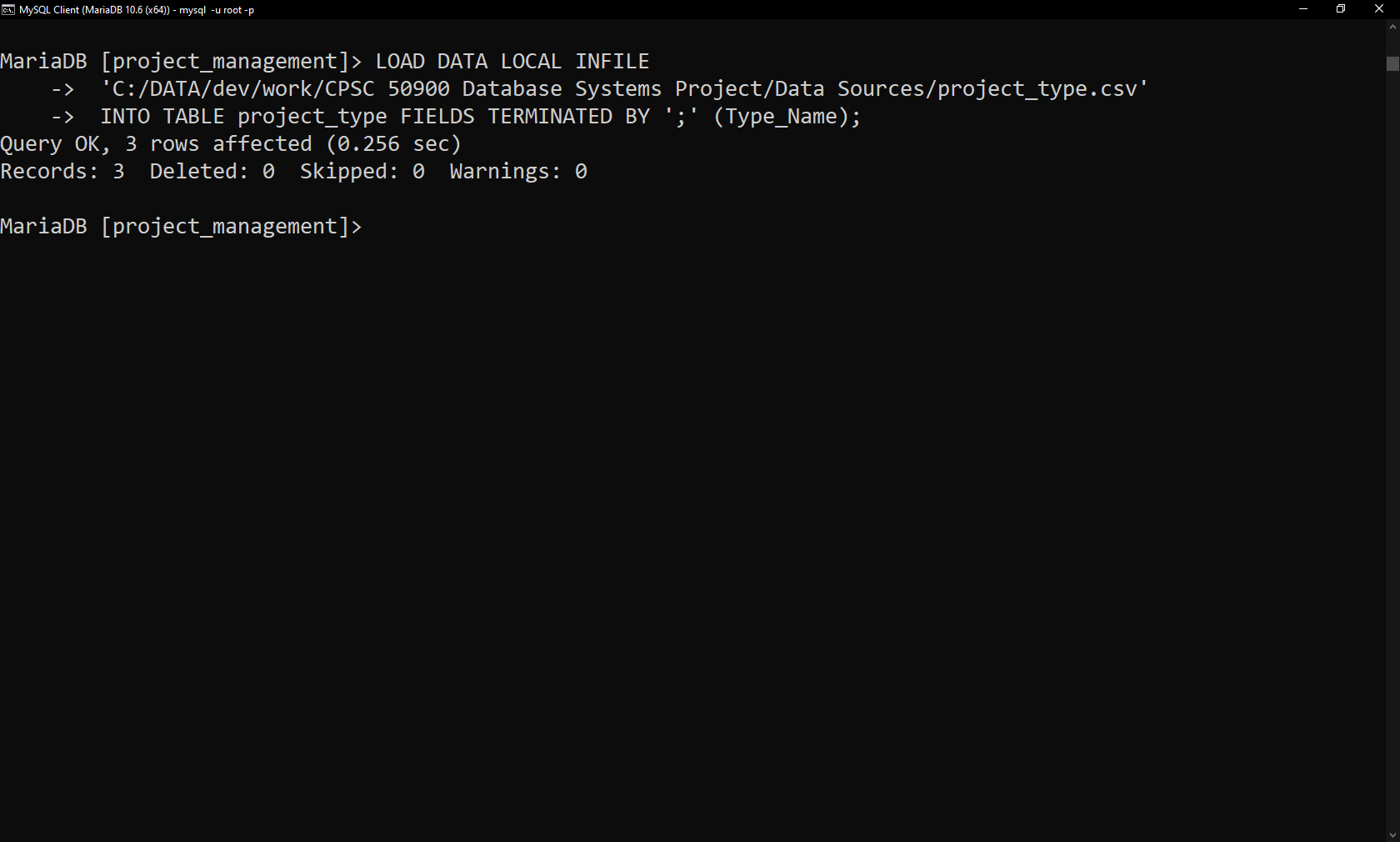
Load data in roles



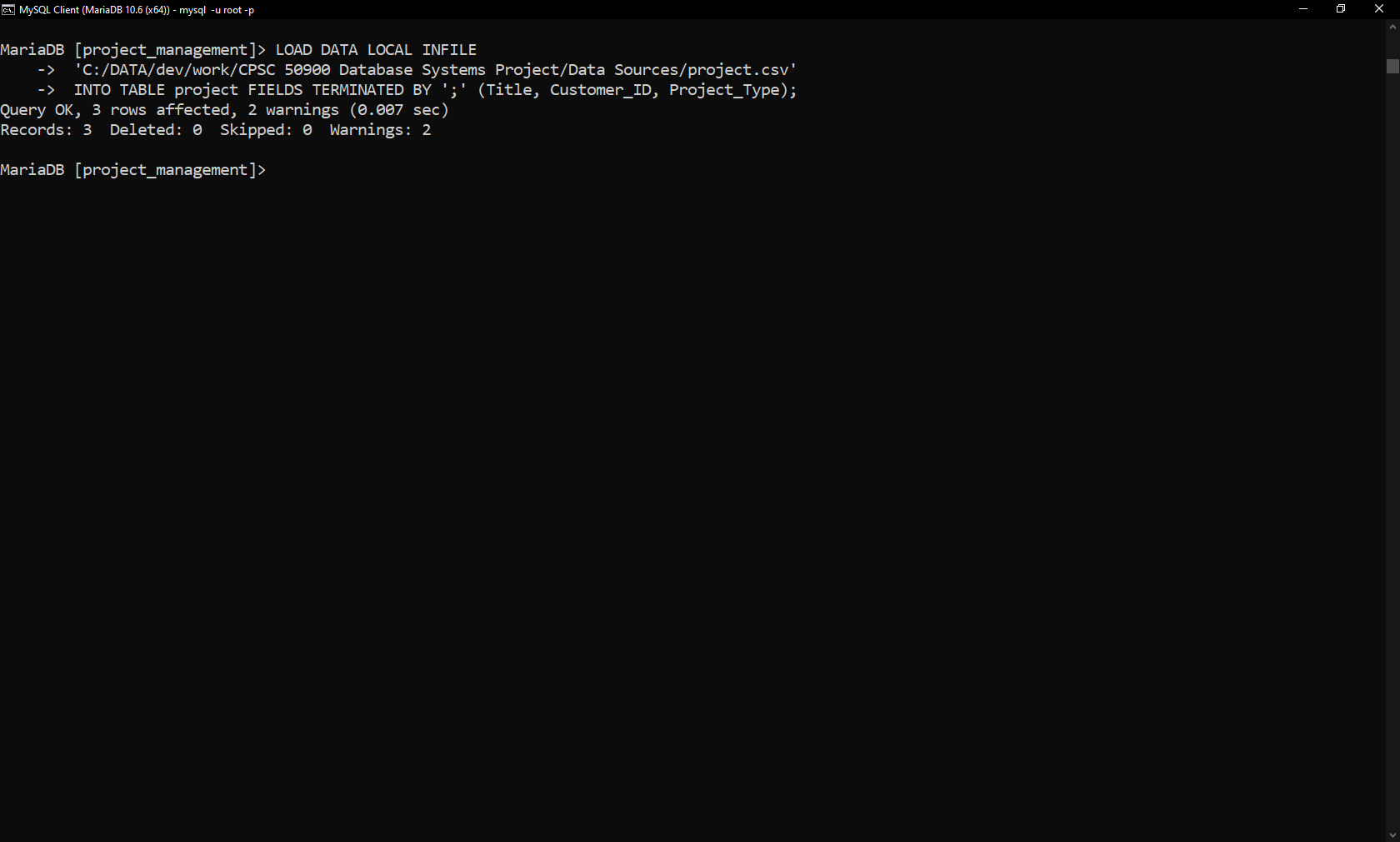
Load data in status



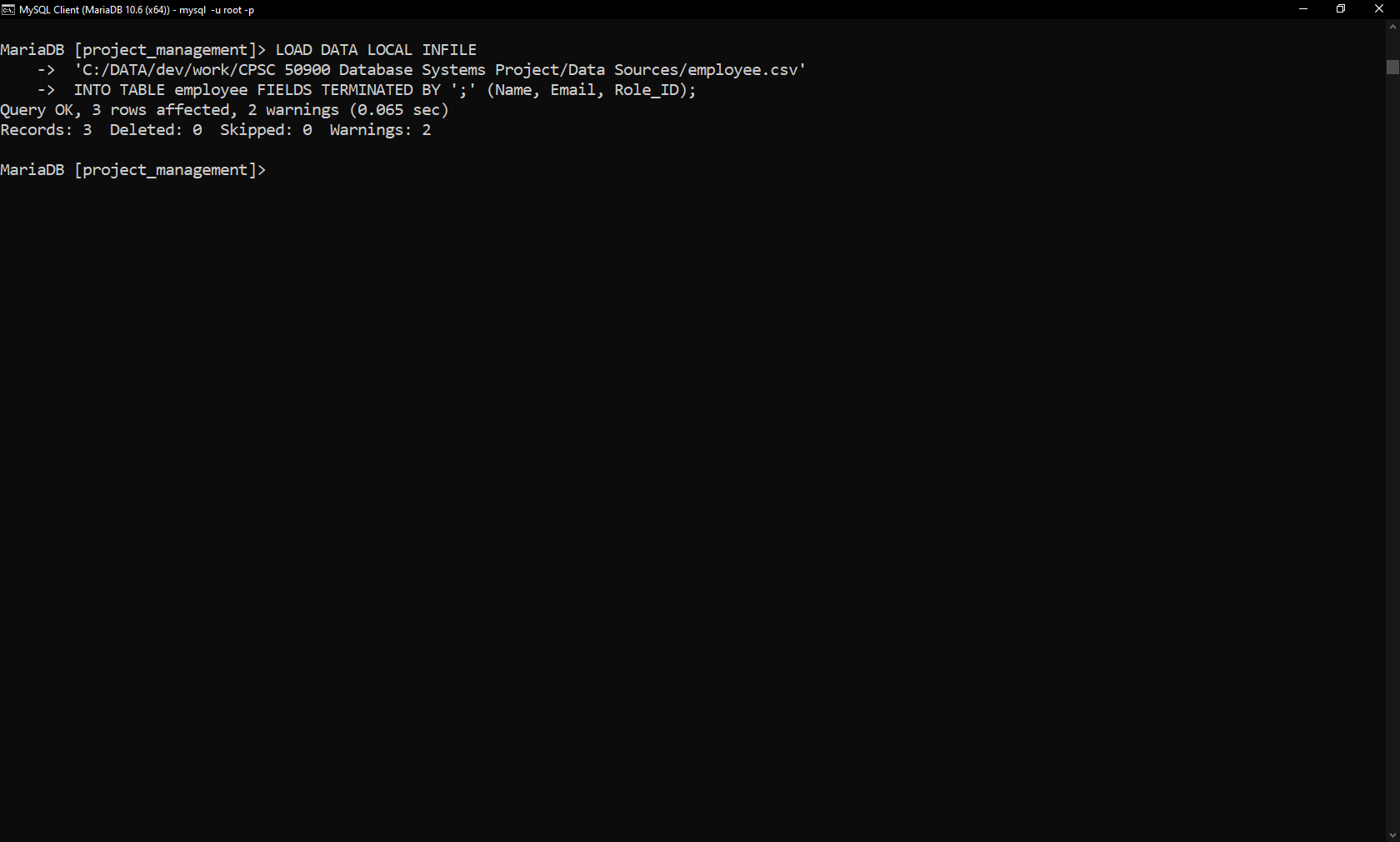
Load data in projet type



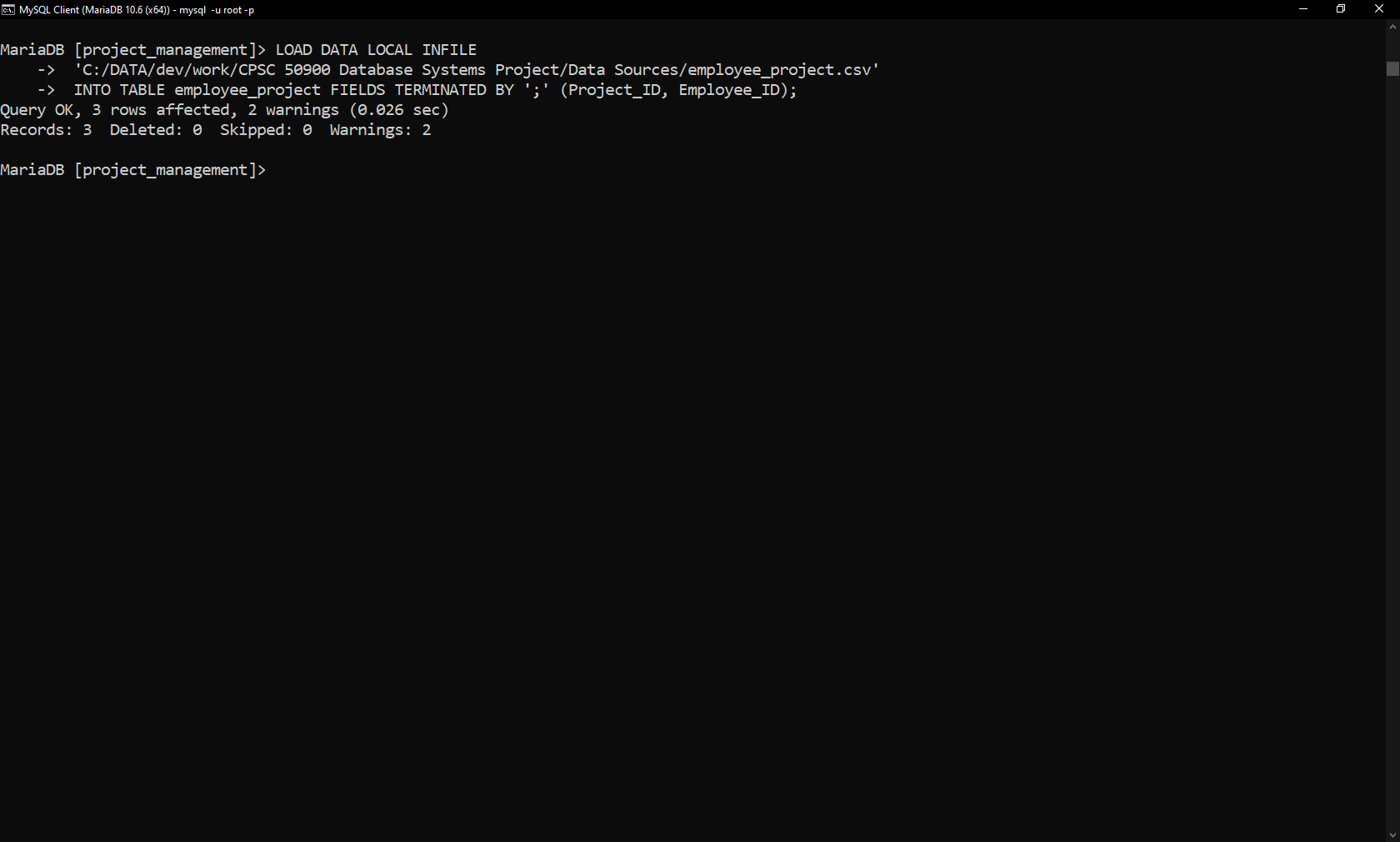
Load data in project



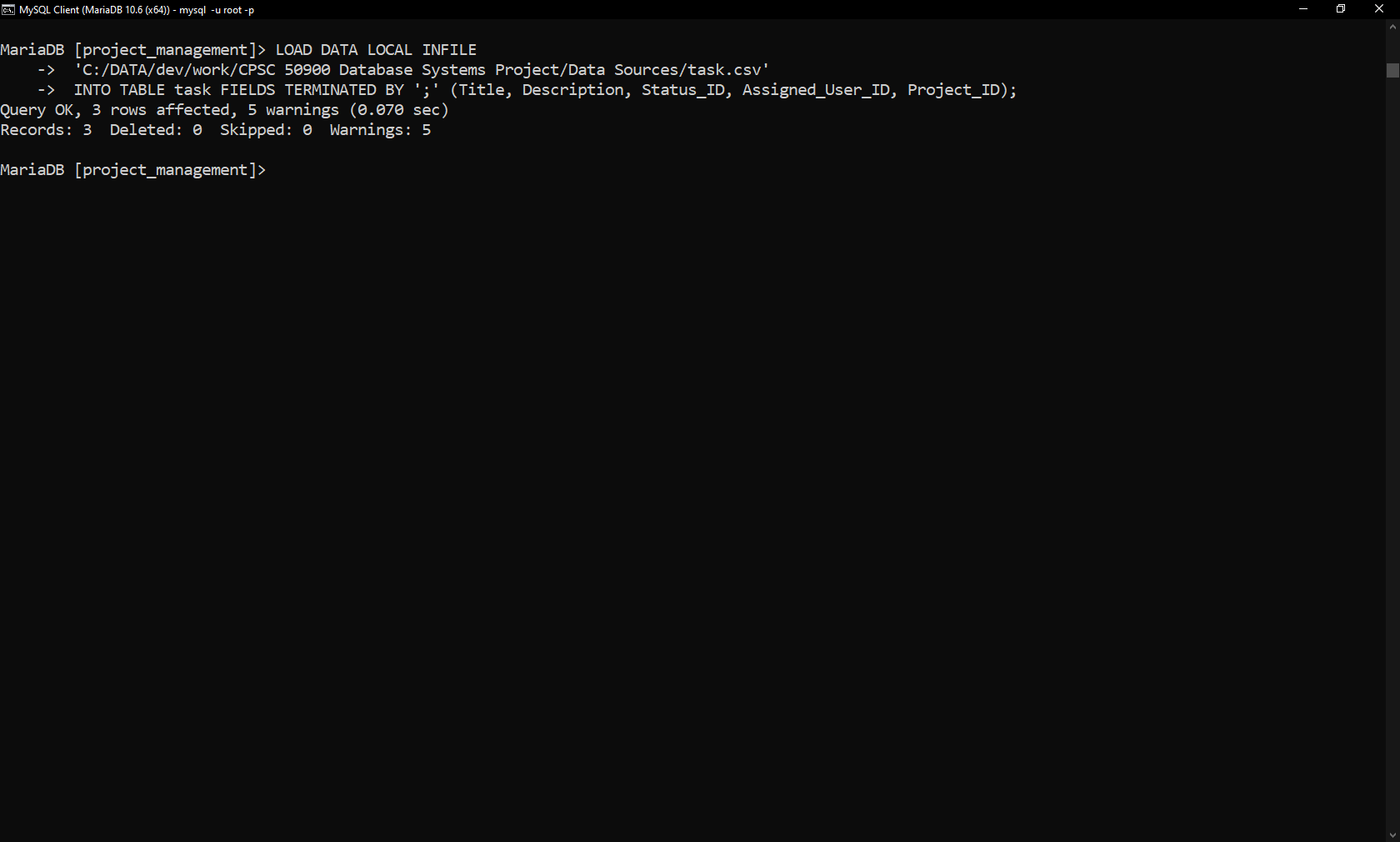
Load data in employee



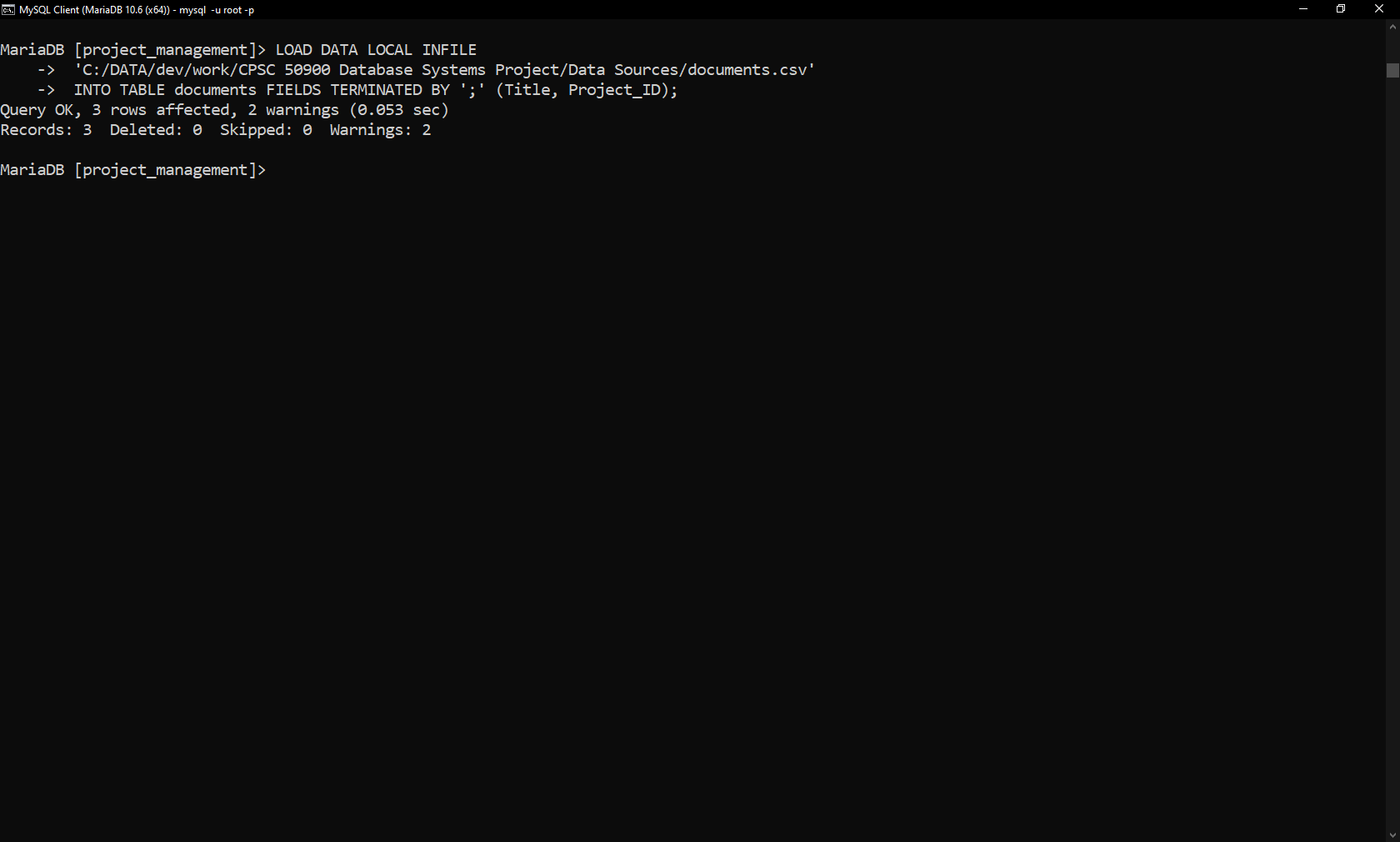
Load data in employee\_project



Load data in tasks



Load data in documents



**Data Manipulation Language Scripts**

**Insert into Query 1**

INSERT INTO task

(Title, Description, Status\_ID, Assigned\_User\_ID, Project\_ID)

VALUES

("Login Functionality", "Enable login to the portal",1, 2, 1);



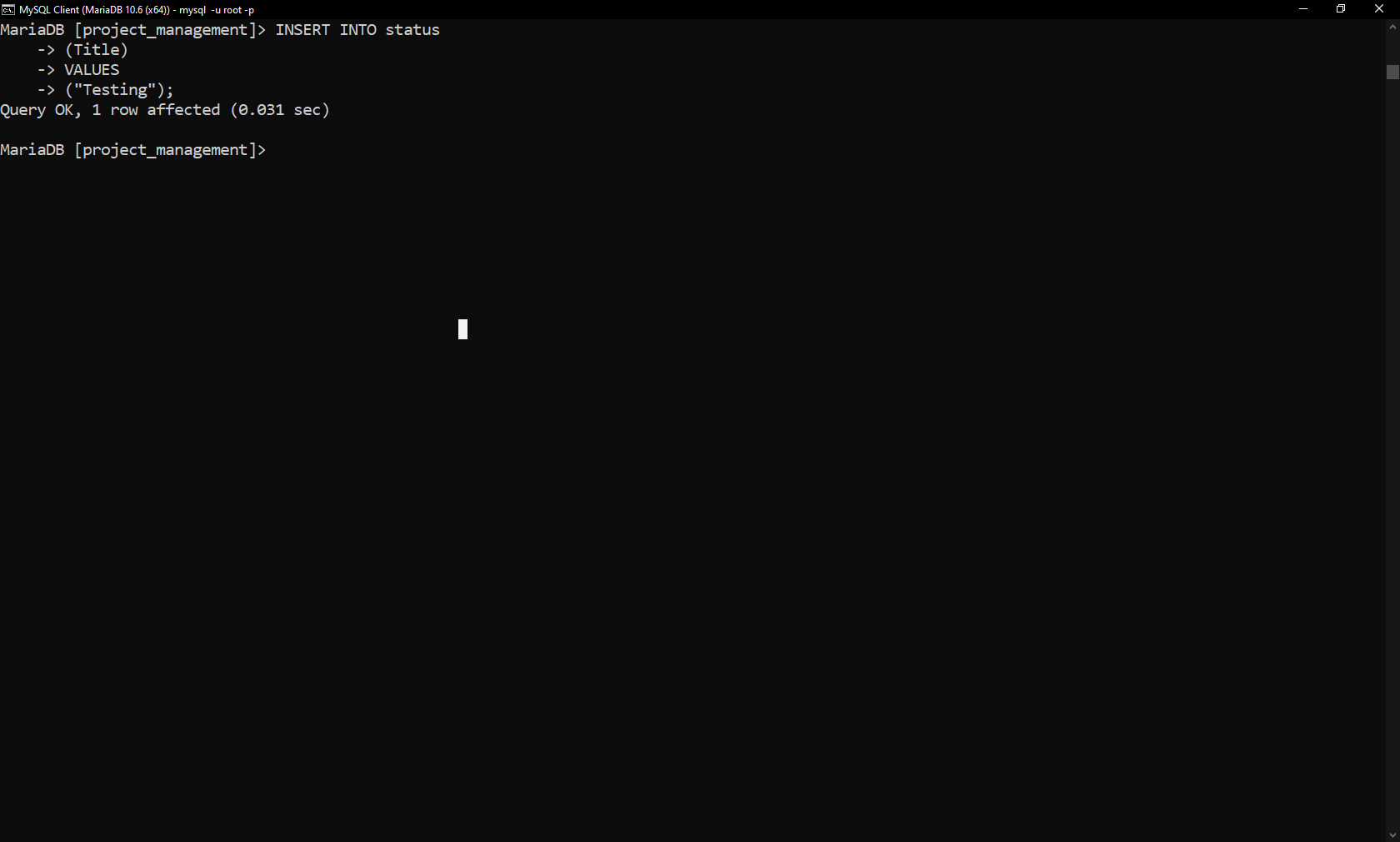
**Insert into Query 2**

INSERT INTO status

(Title)

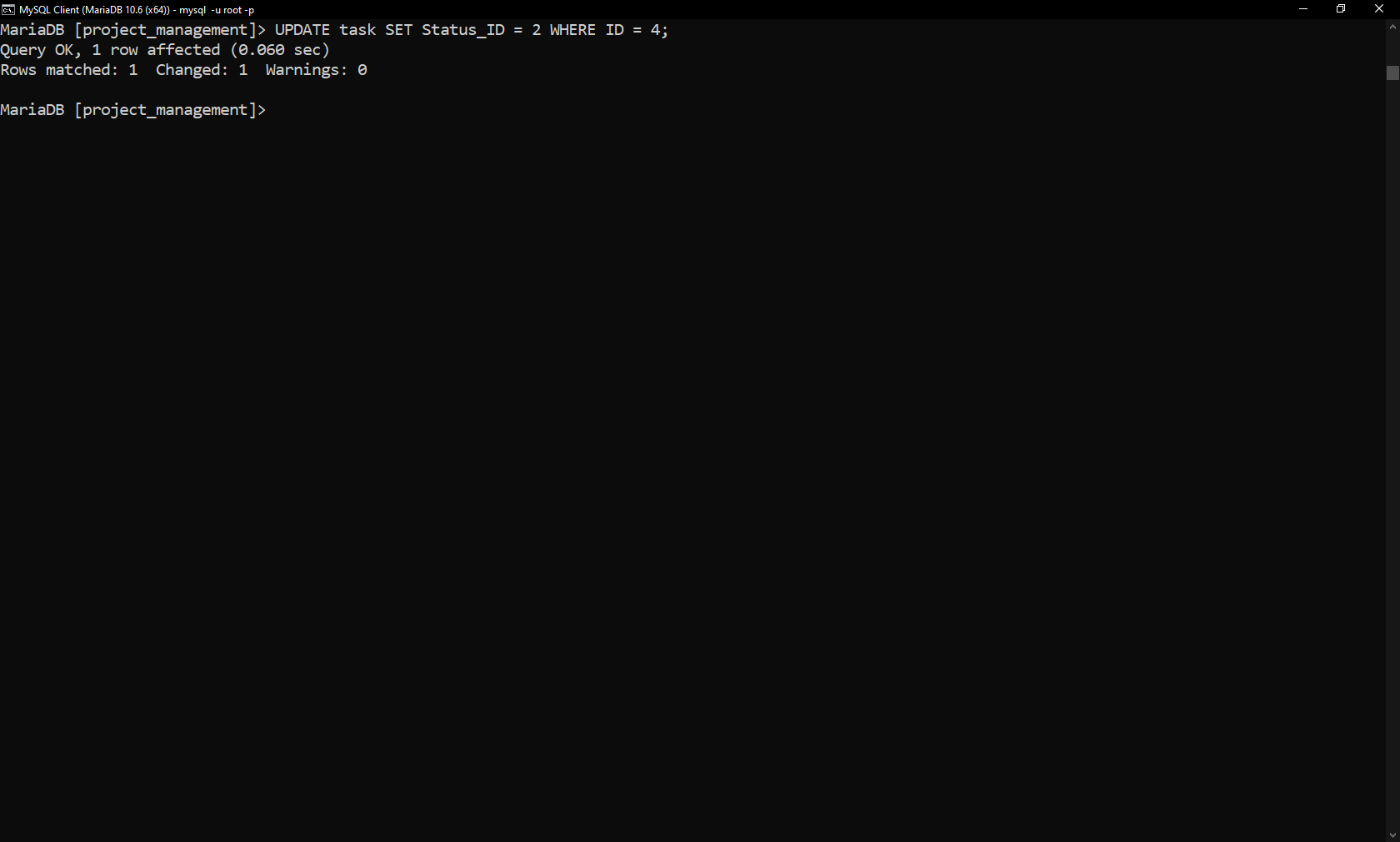
VALUES

("Testing");



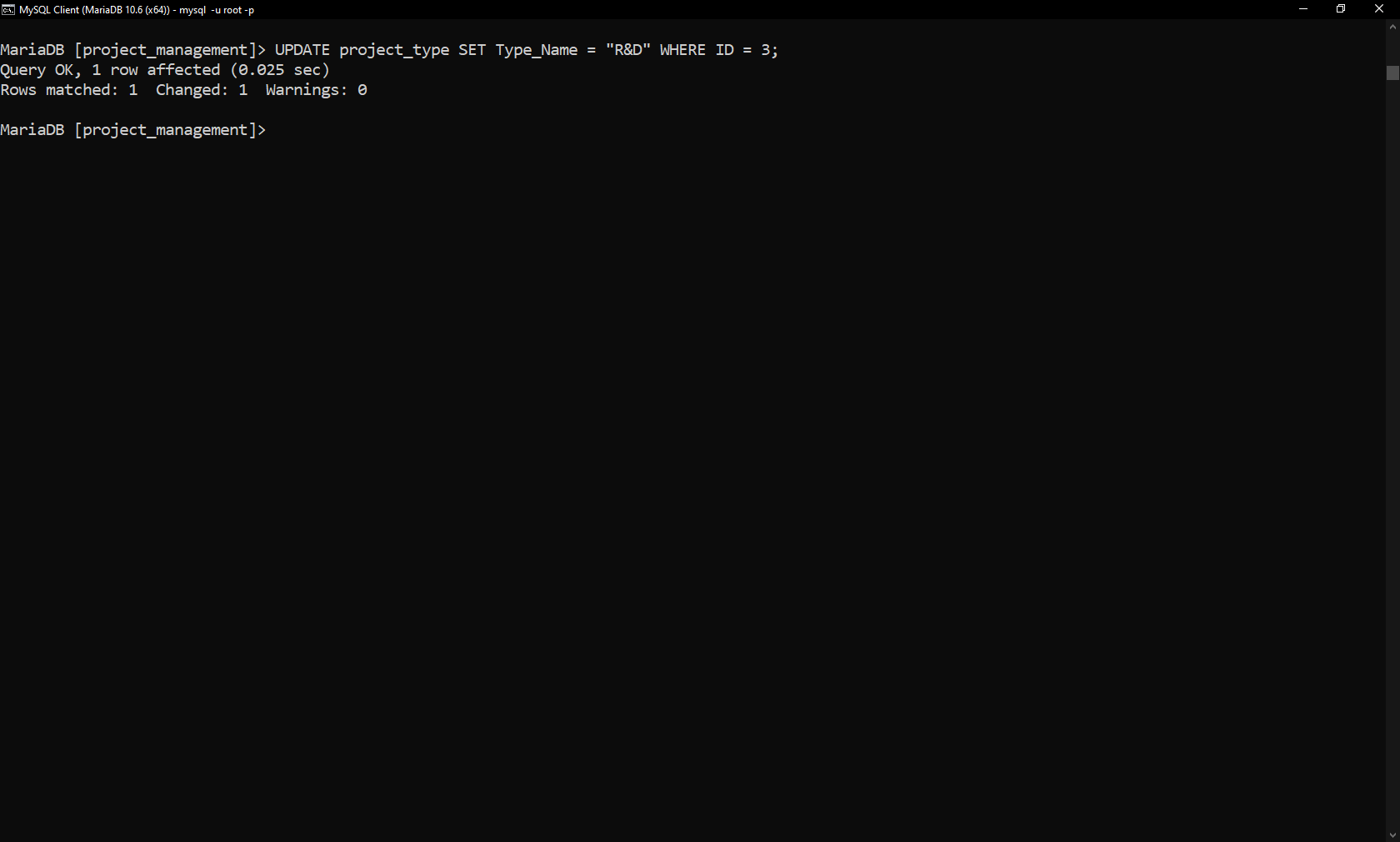
**Update Query 1**

UPDATE task SET Status\_ID = 2 WHERE ID = 4;



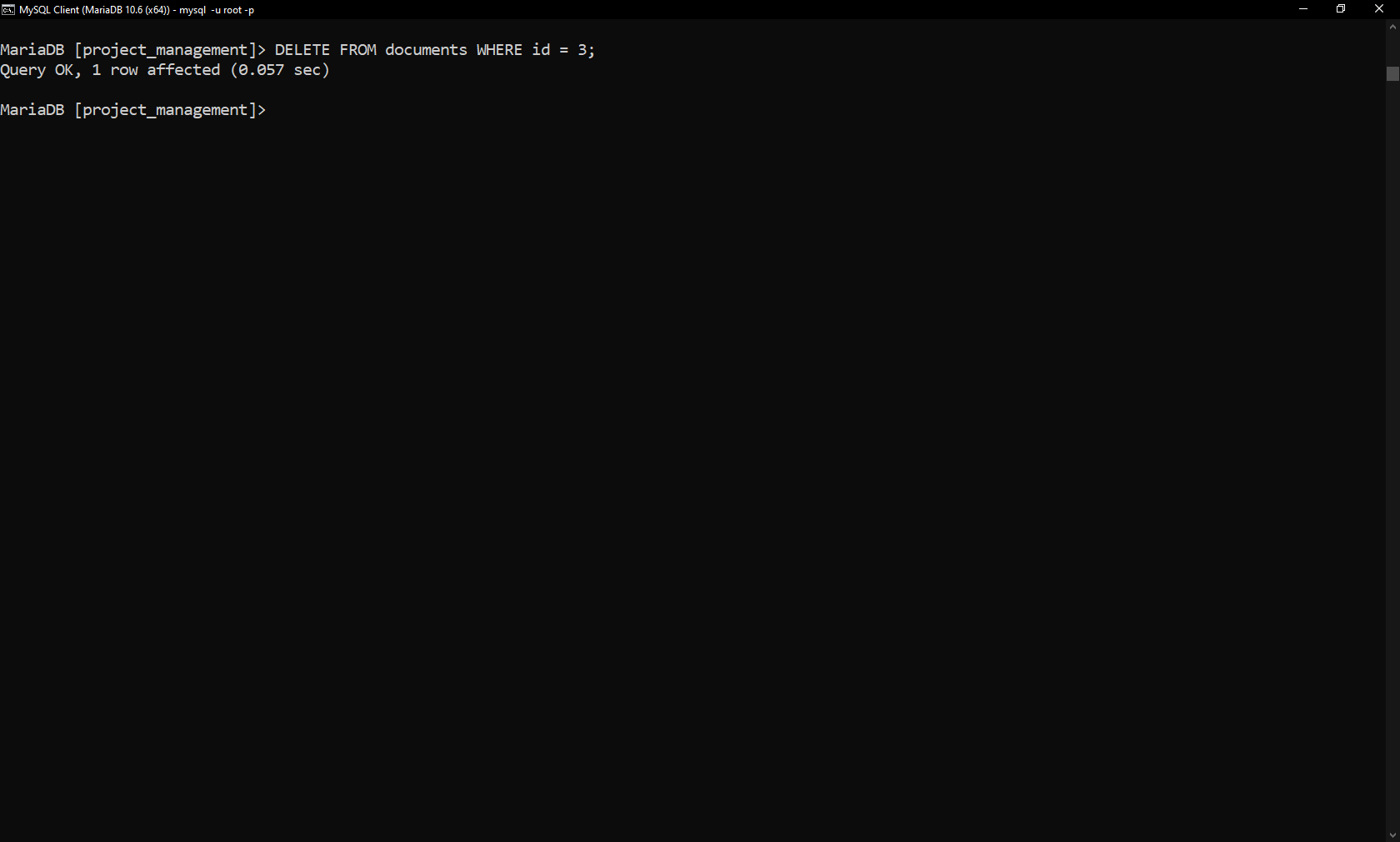
**Update Query 2**

UPDATE project\_type SET Type\_Name = "R&D" WHERE ID = 3;



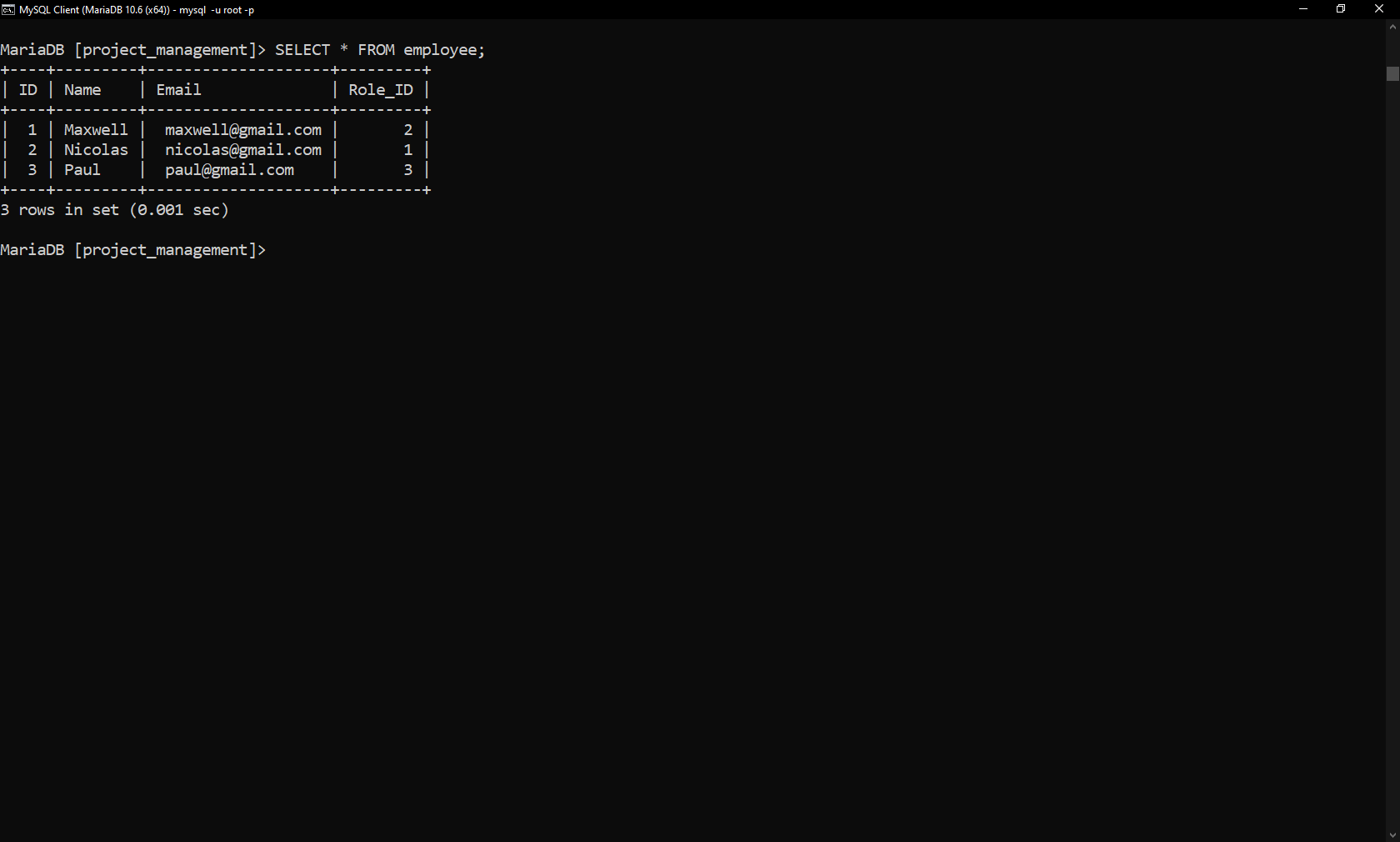
**Delete Query**

DELETE FROM documents WHERE id = 3;



**Select Query**

SELECT \* FROM employee;

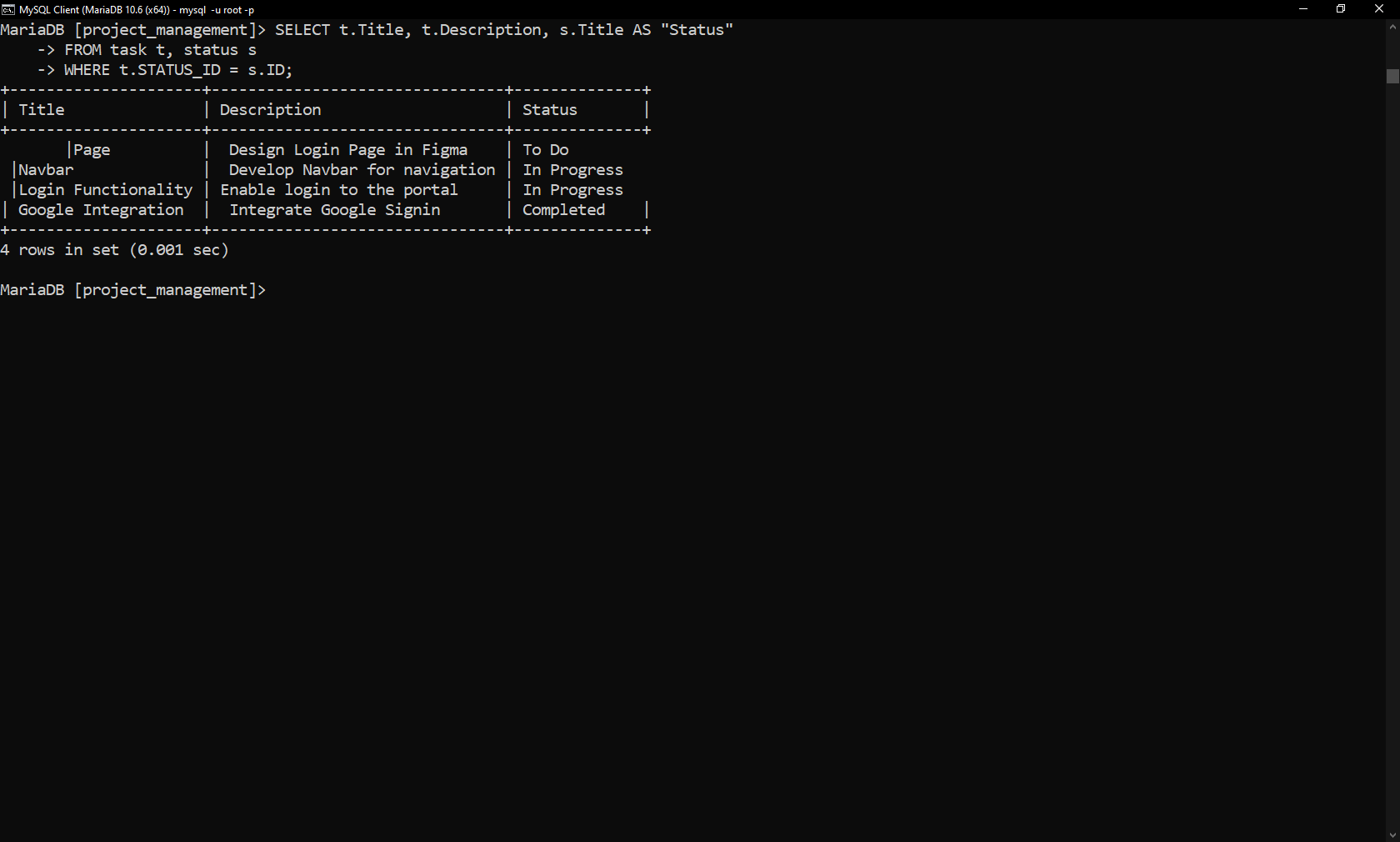


**Join Query 1**

SELECT t.Title, t.Description, s.Title AS "Status"

FROM task t, status s

WHERE t.STATUS\_ID = s.ID;

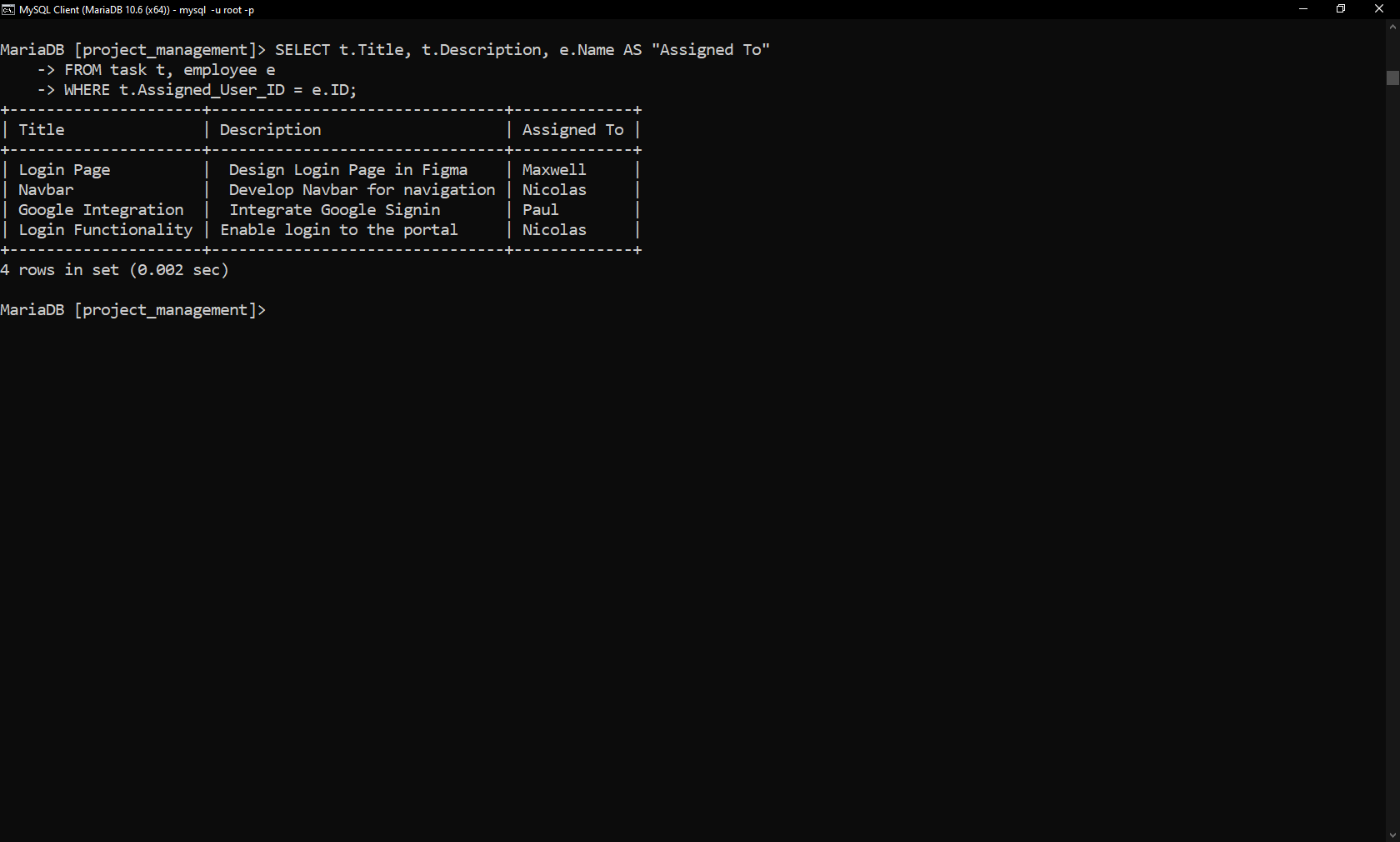


**Join Query 2**

SELECT t.Title, t.Description, e.Name AS "Assigned To"

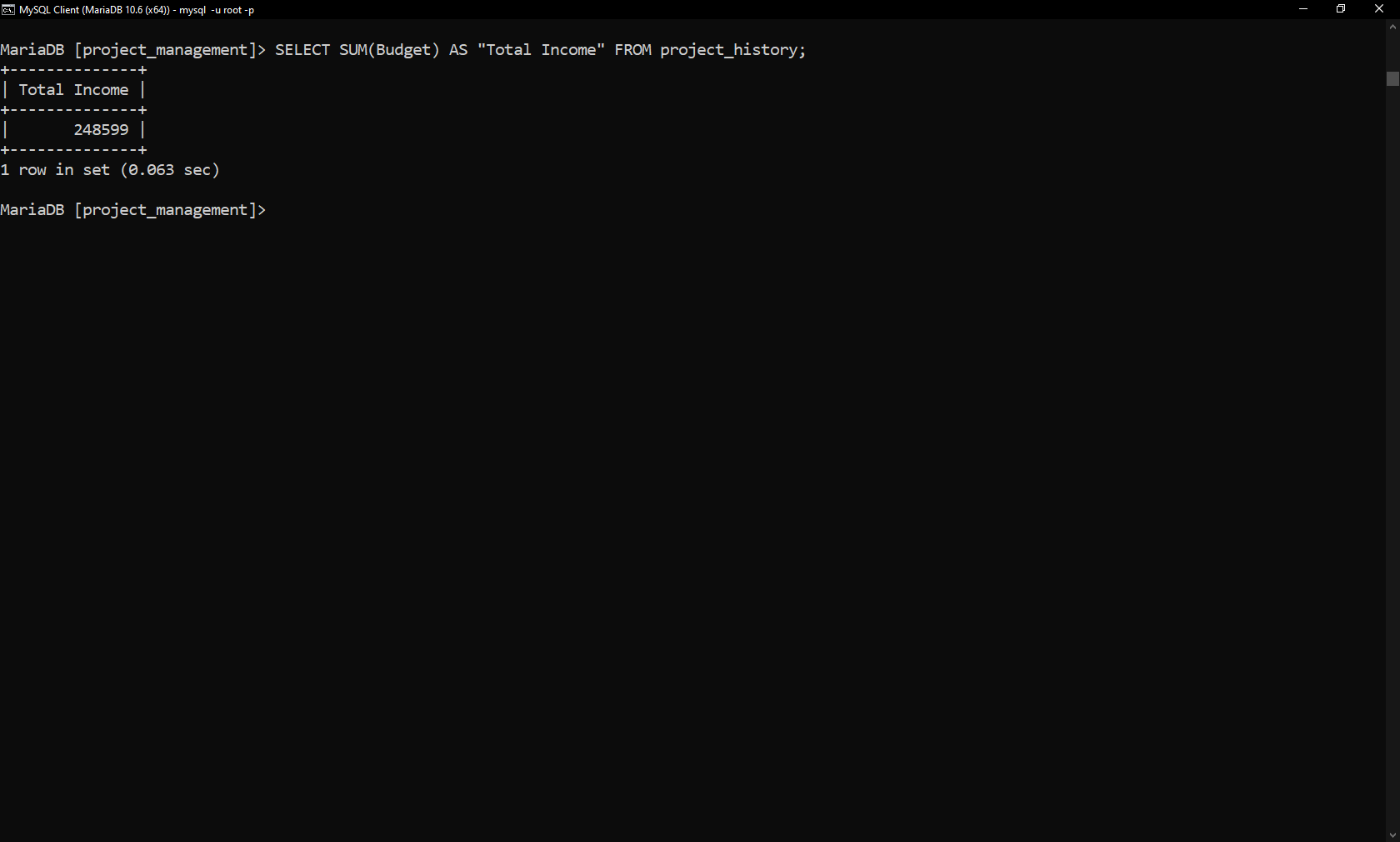
FROM task t, employee e

WHERE t.Assigned\_User\_ID = e.ID;



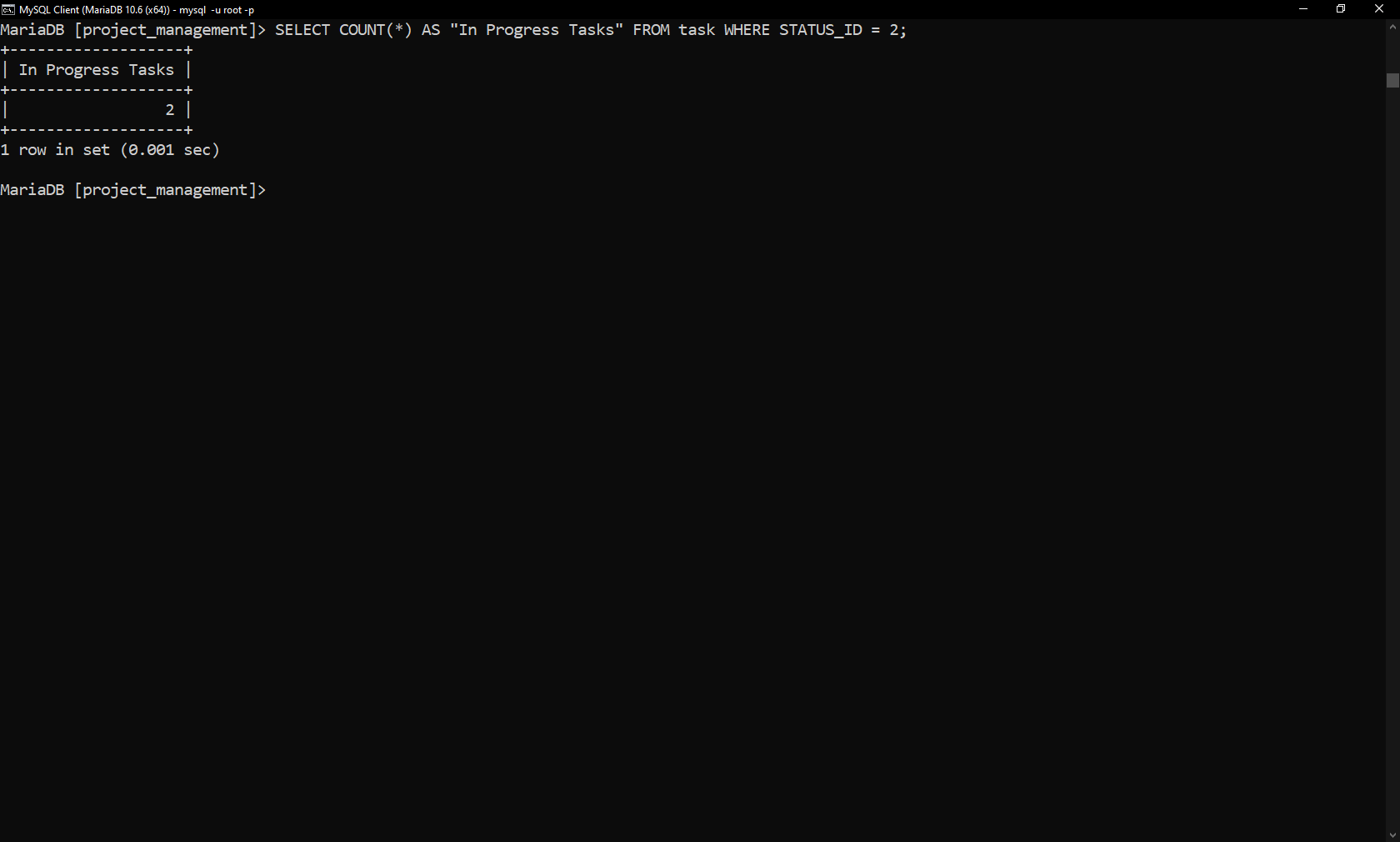
**Summary statement 1:**

SELECT SUM(Budget) AS "Total Income" FROM project\_history;



**Summary statement 2:**

SELECT COUNT(\*) AS "In Progress Tasks" FROM task WHERE STATUS\_ID = 2;

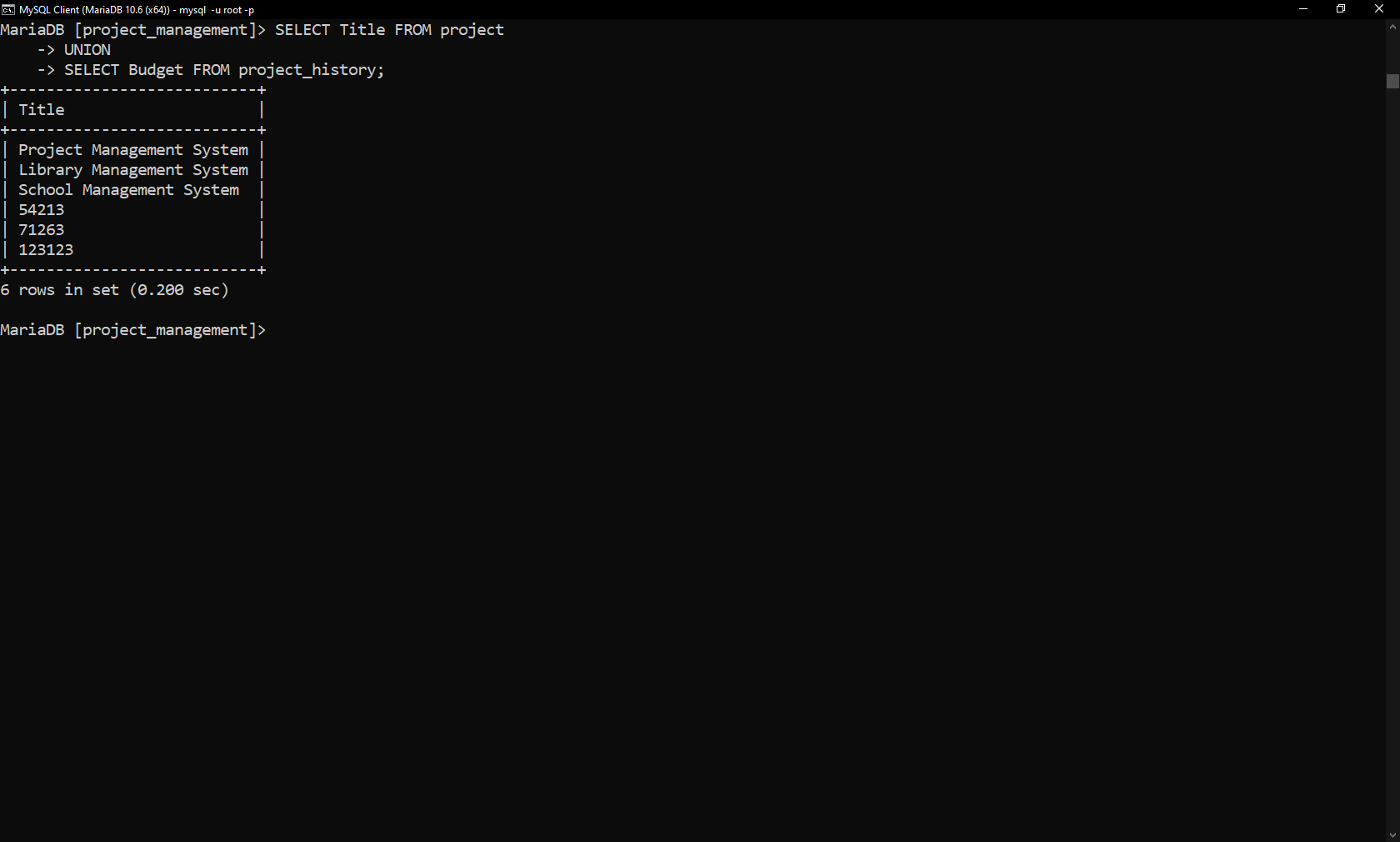


**Multi-table Query**

SELECT Title FROM project

UNION

SELECT Budget FROM project\_history;



**Query of choice**

SELECT p.Title as "Project Name", e.Name AS "Employee Name"

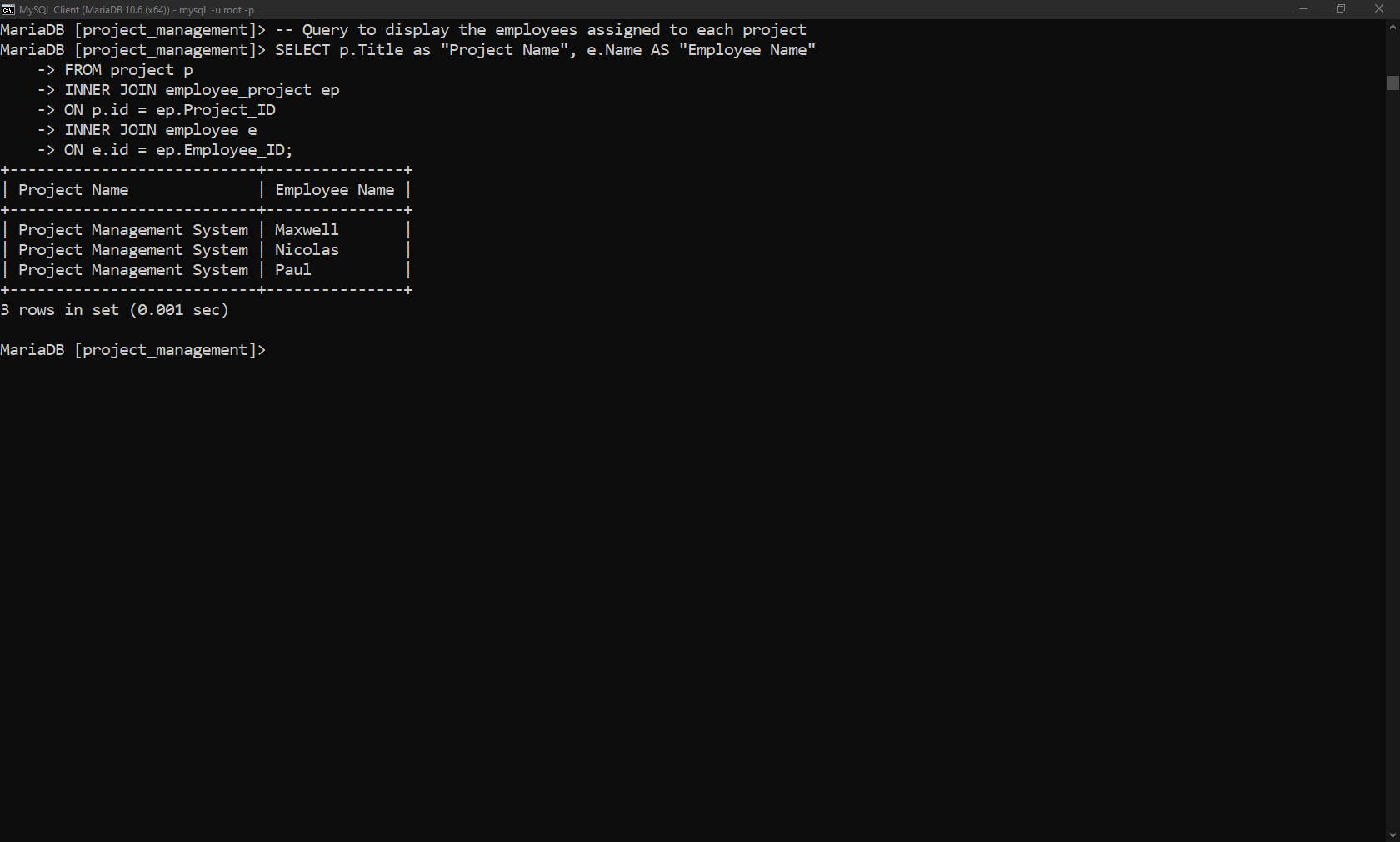
FROM project p

INNER JOIN employee\_project ep

ON p.id = ep.Project\_ID

INNER JOIN employee e

ON e.id = ep.Employee\_ID;



**INDEXES**

**Reason:**

For this system, I chose the title in 3 tables as an index because in future a web portal will be used with this database, therefore a user can search a project, task or document by title.

**SQL Statements:**

-- Query to create index for Title column of project Table

CREATE INDEX idx\_Project\_Title

ON project (Title);

-- Query to create index for Title column of documents Table

CREATE INDEX idx\_Doc\_Title

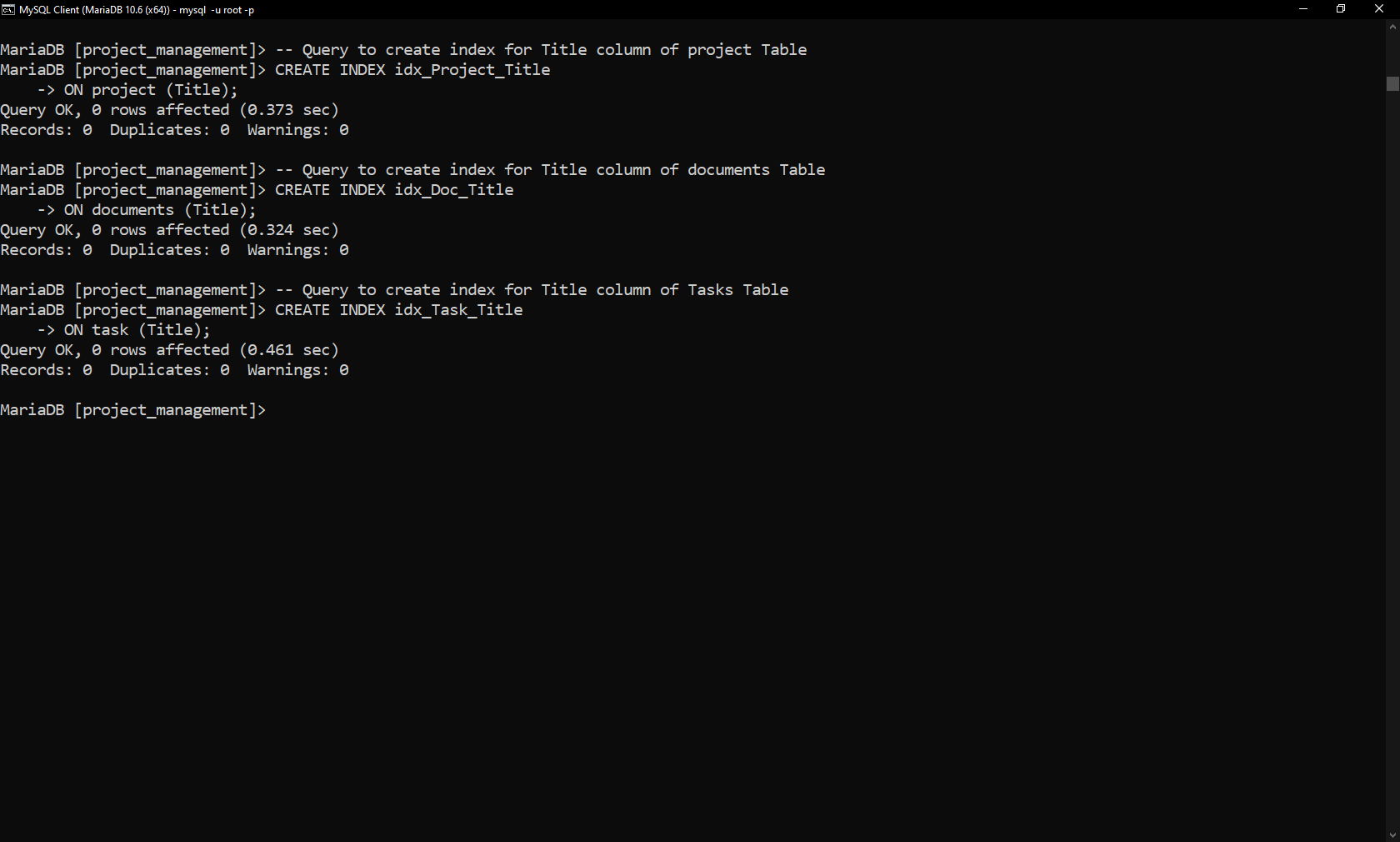
ON documents (Title);

-- Query to create index for Title column of Tasks Table

CREATE INDEX idx\_Task\_Title

ON task (Title);

**Screenshot:**



**VIEWS**

**SQL Statements:**

-- Query to create a view for Employee and Project Assignment

CREATE VIEW Project\_Employee\_Assignment AS

SELECT p.Title as "Project Name", e.Name AS "Employee Name"

FROM project p

INNER JOIN employee\_project ep

ON p.id = ep.Project\_ID

INNER JOIN employee e

ON e.id = ep.Employee\_ID;

-- Query to create a view for Employee and Task Assignment

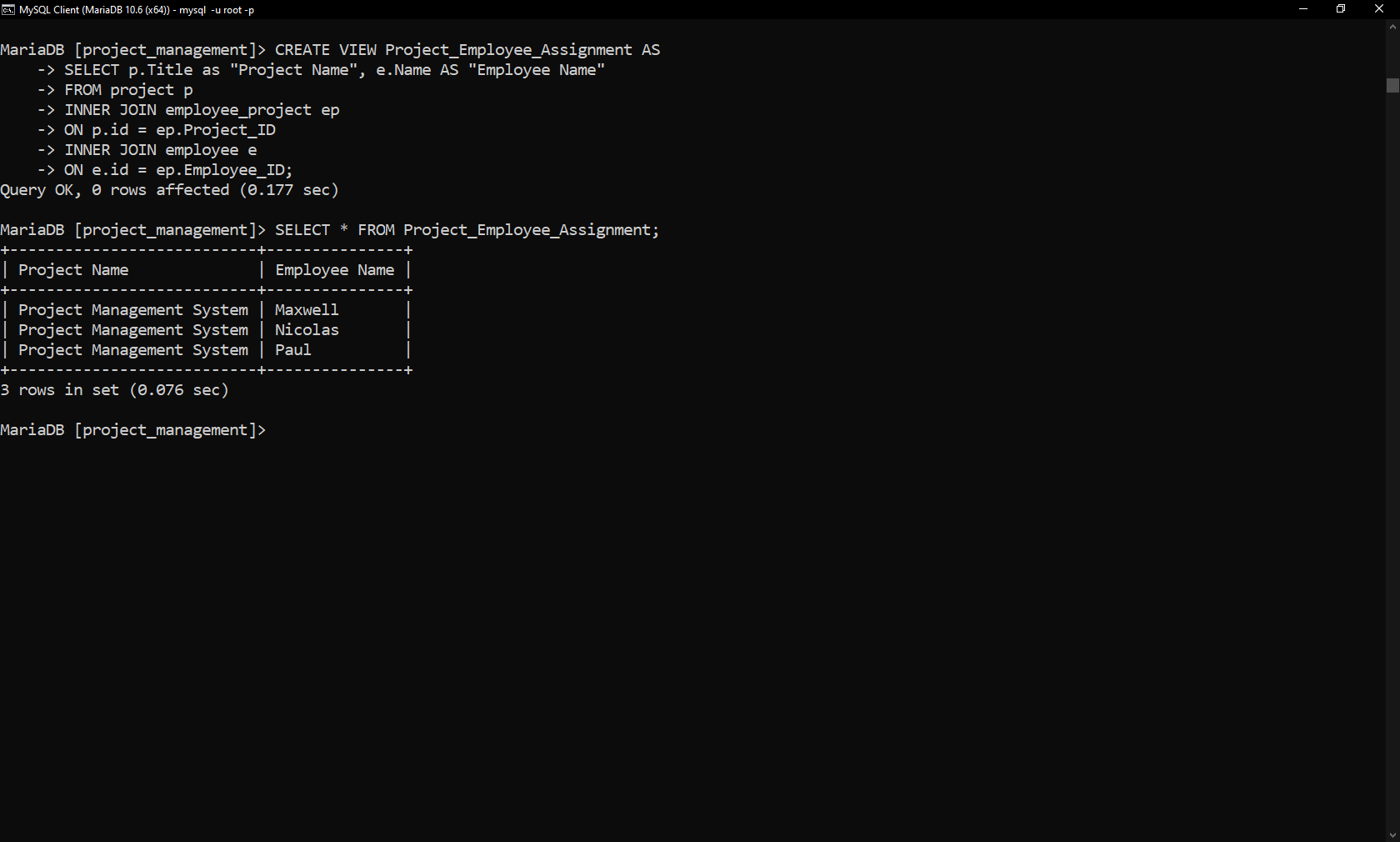
CREATE VIEW Employee\_Task\_Assignment AS

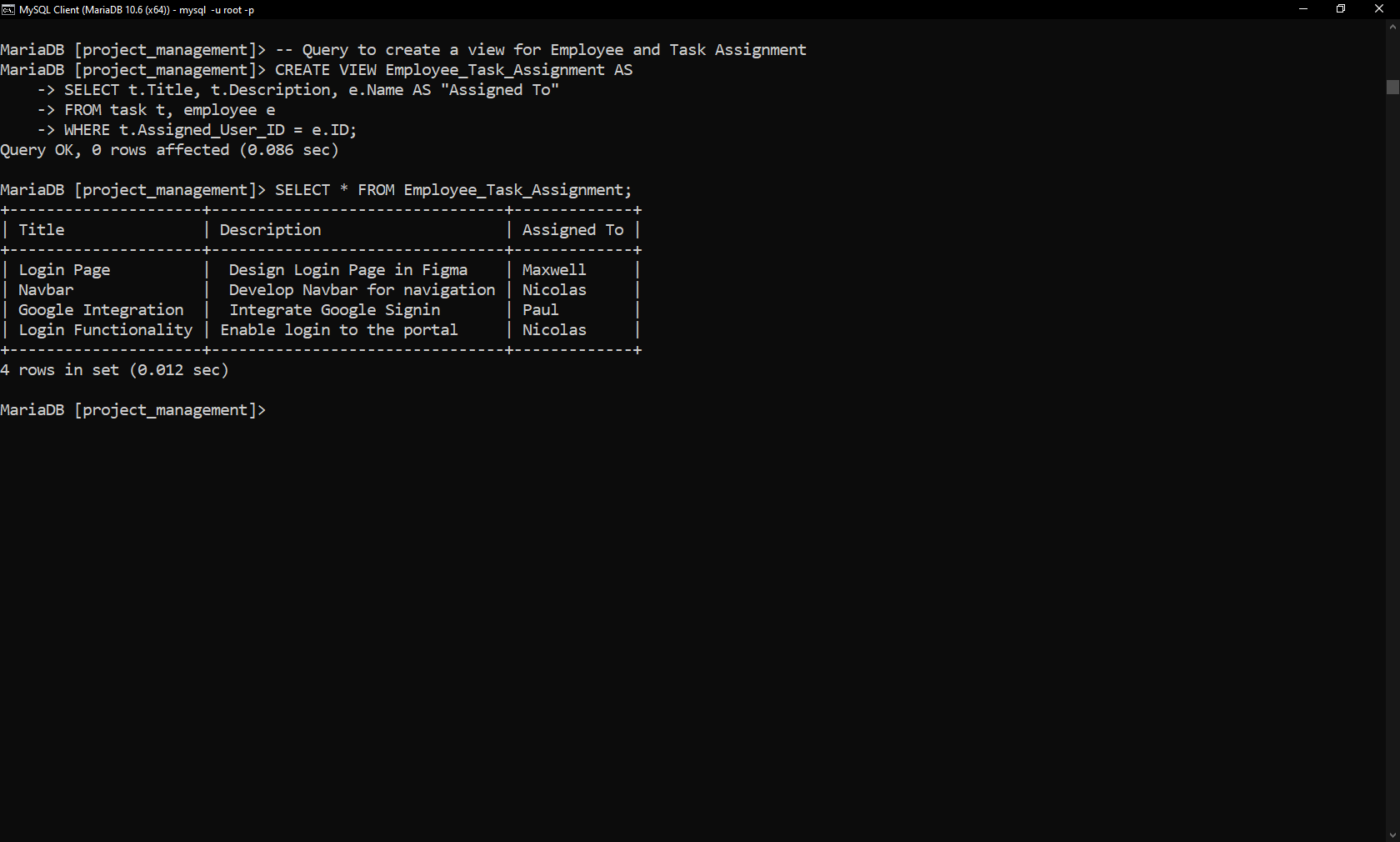
SELECT t.Title, t.Description, e.Name AS "Assigned To"

FROM task t, employee e

WHERE t.Assigned\_User\_ID = e.ID;

**Screenshots**





**Python script**

| import mysql.connector  # Create a database connection  connection = mysql.connector.connect(  host="localhost", user="root", password="123", database="project\_management"  )  db = connection.cursor()  # List representing the names of the tables to generate their reports  tables = [  "customer",  "documents",  "employee",  "employee\_project",  "project",  "project\_history",  "project\_type",  "roles",  "status",  "task",  ]  # Iterate through the list of the tables and execute  # Select query to get its data and save it to the file  for table in tables:  print("###############", table, "###############")  # Create a file  file = open(table, "w")  # SQL Query to get data  query = "SELECT \* FROM " + table  # Execute the query  db.execute(query)  result = db.fetchall()  # Save the result in the file  for row in result:  file.write(str(row) + "\n")  print(row) |
| --- |

**Output**

