**ASSIGNMENT-1**

* **CREATE A TABLES:-**

**Department**

**Employee**

**Salarylog**

**Emplog**

.open database1.db

CREATE TABLE Department (

Dept\_id text PRIMARY KEY,

Dept\_name TEXT NOT NULL

);

CREATE TABLE Employee (

Emp\_id text PRIMARY KEY,

Dept\_id INTEGER NOT NULL,

Emp\_name TEXT NOT NULL,

Mobile TEXT NOT NULL,

Email TEXT NOT NULL,

Age INTEGER NOT NULL CHECK (Age > 18 AND Age < 100),

City TEXT NOT NULL,

Salary REAL CHECK (Salary > 10000),

FOREIGN KEY (Dept\_id) REFERENCES Department(Dept\_id)

);

CREATE TABLE Salarylog (

Emp\_id INTEGER,

Old\_salary REAL,

New\_salary REAL,

Date TEXT,

FOREIGN KEY (Emp\_id) REFERENCES Employee(Emp\_id)

);

CREATE TABLE Employ (

Emp\_id INTEGER,

Emp\_name TEXT,

Dept\_id INTEGER,

Salary REAL,

Date TEXT,

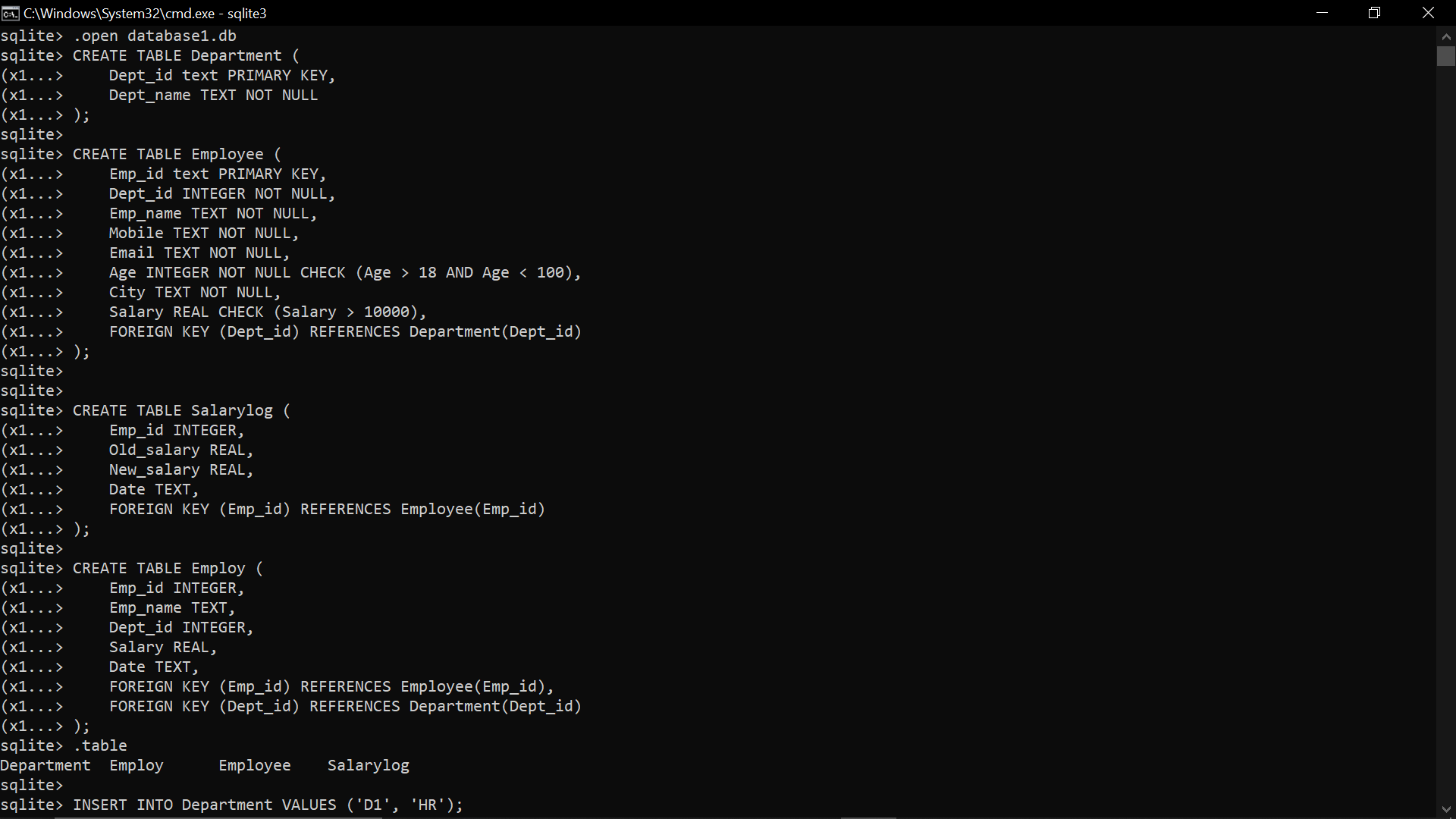
FOREIGN KEY (Emp\_id) REFERENCES Employee(Emp\_id),

FOREIGN KEY (Dept\_id) REFERENCES Department(Dept\_id)

);

.table

**OUTPUT:-**

****

1. TRIGGER

1. **Create trigger before insert on table department to check if the deot\_id starts with ‘D’ or not. If it not starts with ‘d’ then abort the insert.**

CREATE TRIGGER trg\_check\_deptid

BEFORE INSERT ON Department

FOR EACH ROW

BEGIN

SELECT

CASE

WHEN SUBSTR(NEW.Dept\_id, 1, 1) != 'D' THEN

RAISE(ABORT, 'Dept\_id must start with D')

END;

END;

1. **Create trigger before insert on table employee to check if the emp\_id starts with ‘E’ or not. If it not starts with ‘e’ then abort the insert.**

CREATE TRIGGER trg\_check\_empid

BEFORE INSERT ON Employee

FOR EACH ROW

BEGIN

SELECT

CASE

WHEN SUBSTR(NEW.Emp\_id, 1, 1) != 'E' THEN

RAISE(ABORT, 'Emp\_id must start with E')

END;

END;

1. **Create a trigger on employee table which track record of salary change of each employee. For salary logs use table salarylog.**

CREATE TRIGGER trg\_salary\_change

AFTER UPDATE OF Salary ON Employee

FOR EACH ROW

WHEN OLD.Salary != NEW.Salary

BEGIN

INSERT INTO Salarylog (Emp\_id, Old\_salary, New\_salary, Date)

VALUES (OLD.Emp\_id, OLD.Salary, NEW.Salary, DATE('now'));

END;

1. **Create a trigger on employee table which has back up of all the employees who are removed from the table. Use table emplog for the same.**

CREATE TRIGGER trg\_backup\_employee

BEFORE DELETE ON Employee

FOR EACH ROW

BEGIN

INSERT INTO Employ (Emp\_id, Emp\_name, Dept\_id, Salary, Date)

VALUES (OLD.Emp\_id, OLD.Emp\_name, OLD.Dept\_id, OLD.Salary, DATE('now'));

END;

**2. QUERIES**

1. **Insert appropriate records in department and employee tables.**

INSERT INTO Department VALUES ('D1', 'HR');

INSERT INTO Department VALUES ('D2', 'Account');

INSERT INTO Department VALUES ('D3', 'IT');

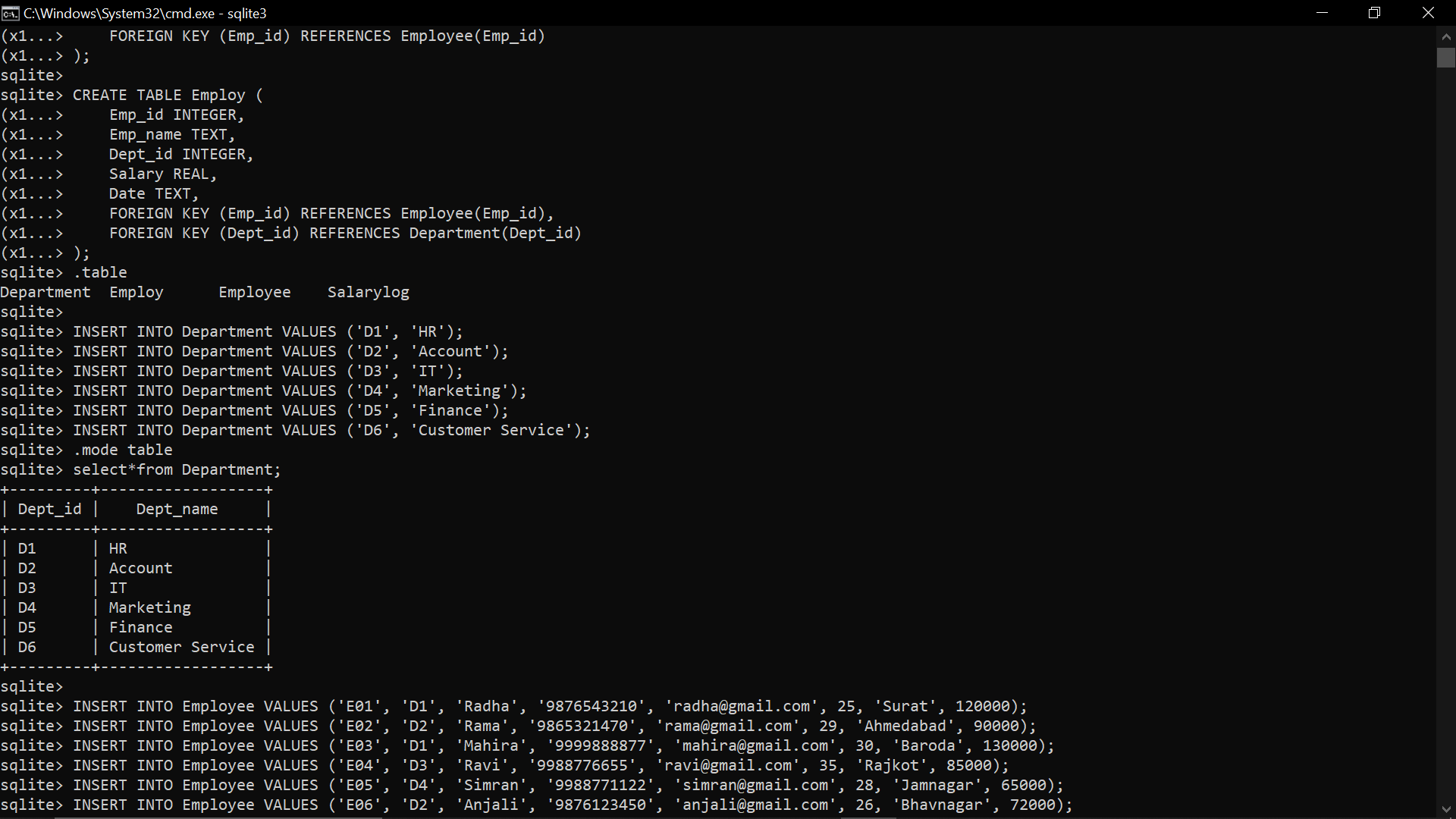
INSERT INTO Department VALUES ('D4', 'Marketing');

INSERT INTO Department VALUES ('D5', 'Finance');

INSERT INTO Department VALUES ('D6', 'Customer Service');

.mode table

select\*from Department;



INSERT INTO Employee VALUES ('E01', 'D1', 'Radha', '9876543210', 'radha@gmail.com', 25, 'Surat', 120000);

INSERT INTO Employee VALUES ('E02', 'D2', 'Rama', '9865321470', 'rama@gmail.com', 29, 'Ahmedabad', 90000);

INSERT INTO Employee VALUES ('E03', 'D1', 'Mahira', '9999888877', 'mahira@gmail.com', 30, 'Baroda', 130000);

INSERT INTO Employee VALUES ('E04', 'D3', 'Ravi', '9988776655', 'ravi@gmail.com', 35, 'Rajkot', 85000);

INSERT INTO Employee VALUES ('E05', 'D4', 'Simran', '9988771122', 'simran@gmail.com', 28, 'Jamnagar', 65000);

INSERT INTO Employee VALUES ('E06', 'D2', 'Anjali', '9876123450', 'anjali@gmail.com', 26, 'Bhavnagar', 72000);

INSERT INTO Employee VALUES ('E07', 'D5', 'Yash', '9845123460', 'yash@gmail.com', 40, 'Surat', 99000);

INSERT INTO Employee VALUES ('E08', 'D1', 'Kavita', '9781234567', 'kavita@gmail.com', 38, 'Vadodara', 78000);

INSERT INTO Employee VALUES ('E09', 'D4', 'Meera', '9876543100', 'meera@gmail.com', 32, 'Nadiad', 94000);

INSERT INTO Employee VALUES ('E10', 'D3', 'Vikram', '9966554433', 'vikram@gmail.com', 34, 'Anand', 102000);

INSERT INTO Employee VALUES ('E11', 'D2', 'Sita', '9990001112', 'sita@gmail.com', 22, 'Bharuch', 56000);

INSERT INTO Employee VALUES ('E12', 'D5', 'Arjun', '9811122233', 'arjun@gmail.com', 45, 'Surat', 88000);

INSERT INTO Employee VALUES ('E13', 'D1', 'Naina', '9933445566', 'naina@gmail.com', 31, 'Valsad', 87000);

INSERT INTO Employee VALUES ('E14', 'D3', 'Amit', '9776655443', 'amit@gmail.com', 50, 'Navsari', 110000);

INSERT INTO Employee VALUES ('E15', 'D2', 'Mona', '9988773322', 'mona@gmail.com', 24, 'Surat', 61000);

INSERT INTO Employee VALUES ('E16', 'D4', 'Komal', '9765432190', 'komal@gmail.com', 29, 'Ahmedabad', 99000);

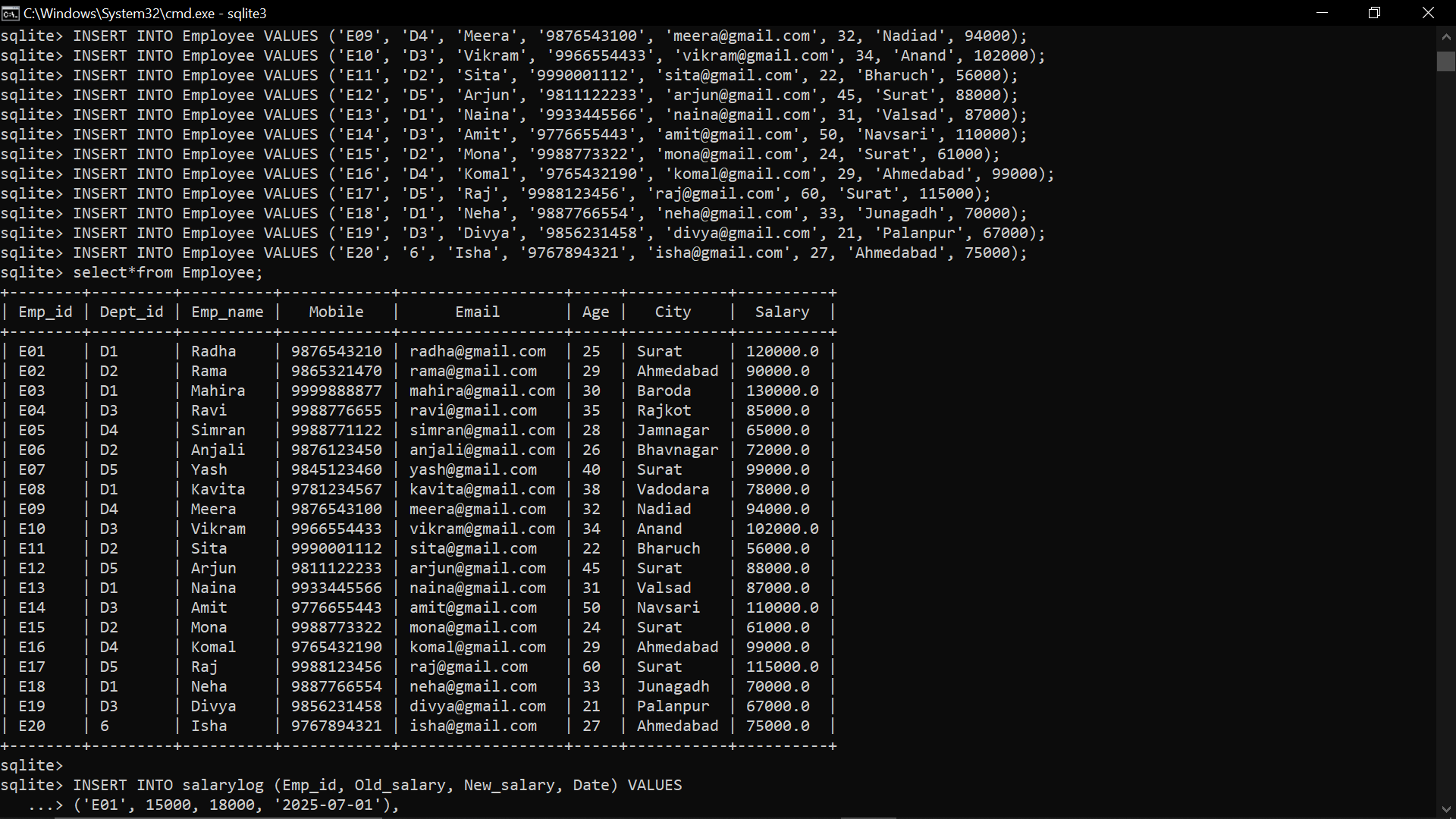
INSERT INTO Employee VALUES ('E17', 'D5', 'Raj', '9988123456', 'raj@gmail.com', 60, 'Surat', 115000);

INSERT INTO Employee VALUES ('E18', 'D1', 'Neha', '9887766554', 'neha@gmail.com', 33, 'Junagadh', 70000);

INSERT INTO Employee VALUES ('E19', 'D3', 'Divya', '9856231458', 'divya@gmail.com', 21, 'Palanpur', 67000);

INSERT INTO Employee VALUES ('E20', '6', 'Isha', '9767894321', 'isha@gmail.com', 27, 'Ahmedabad', 75000);

select\*from Employee;



INSERT INTO salarylog (Emp\_id, Old\_salary, New\_salary, Date) VALUES

('E01', 15000, 18000, '2025-07-01'),

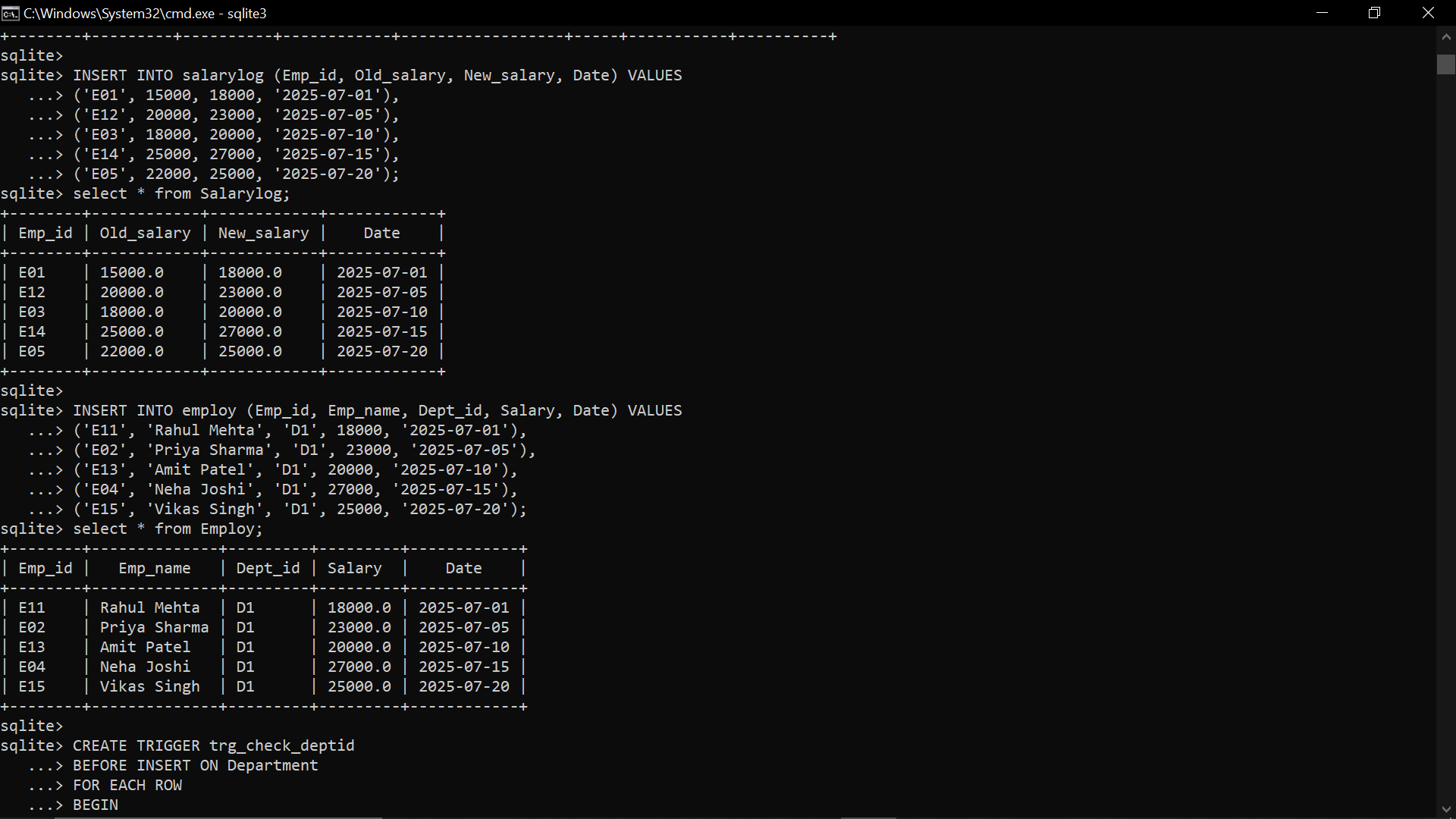
('E12', 20000, 23000, '2025-07-05'),

('E03', 18000, 20000, '2025-07-10'),

('E14', 25000, 27000, '2025-07-15'),

('E05', 22000, 25000, '2025-07-20');

select \* from Salarylog;



INSERT INTO employ (Emp\_id, Emp\_name, Dept\_id, Salary, Date) VALUES

('E11', 'Rahul Mehta', 'D1', 18000, '2025-07-01'),

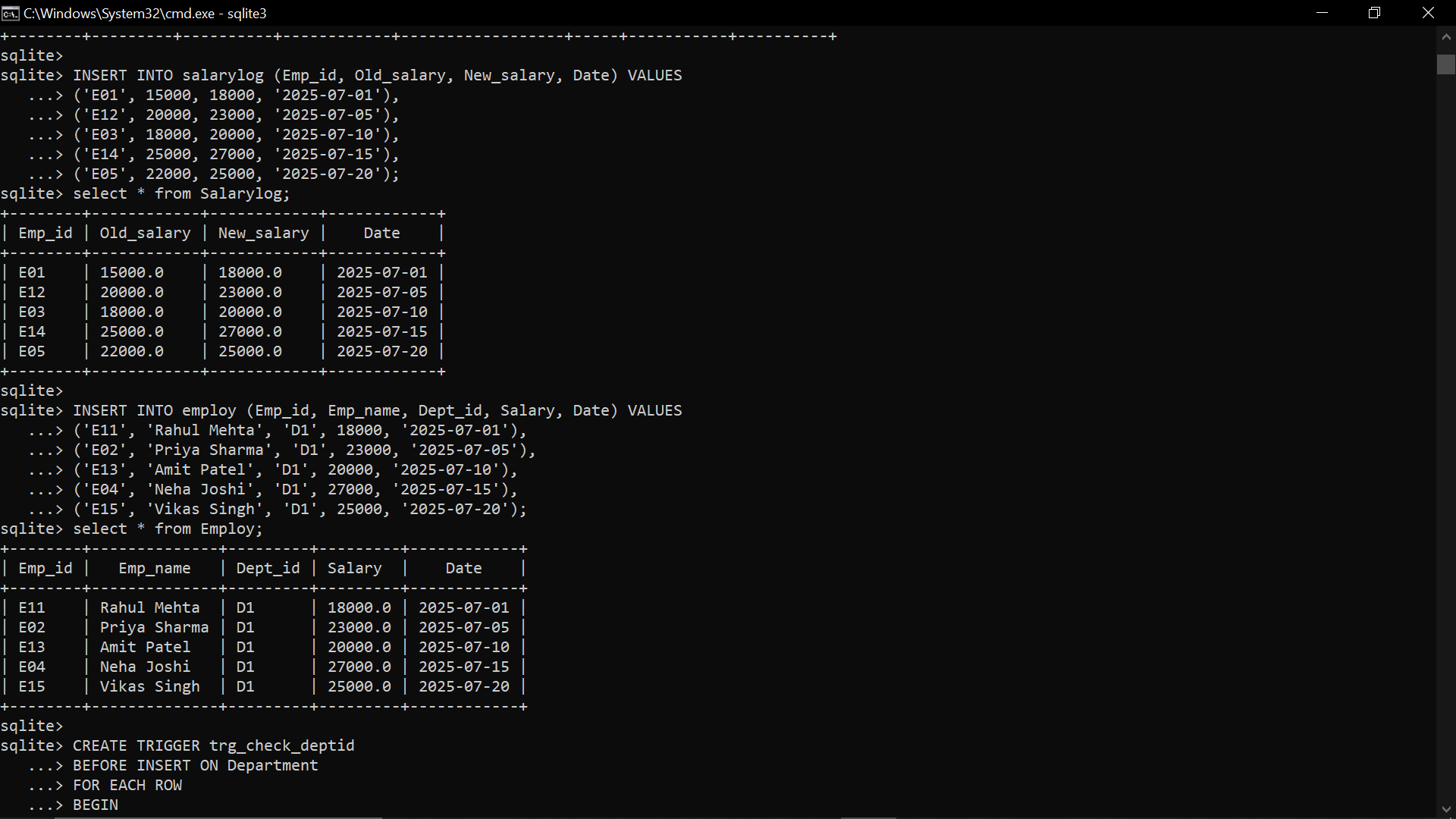
('E02', 'Priya Sharma', 'D1', 23000, '2025-07-05'),

('E13', 'Amit Patel', 'D1', 20000, '2025-07-10'),

('E04', 'Neha Joshi', 'D1', 27000, '2025-07-15'),

('E15', 'Vikas Singh', 'D1', 25000, '2025-07-20');

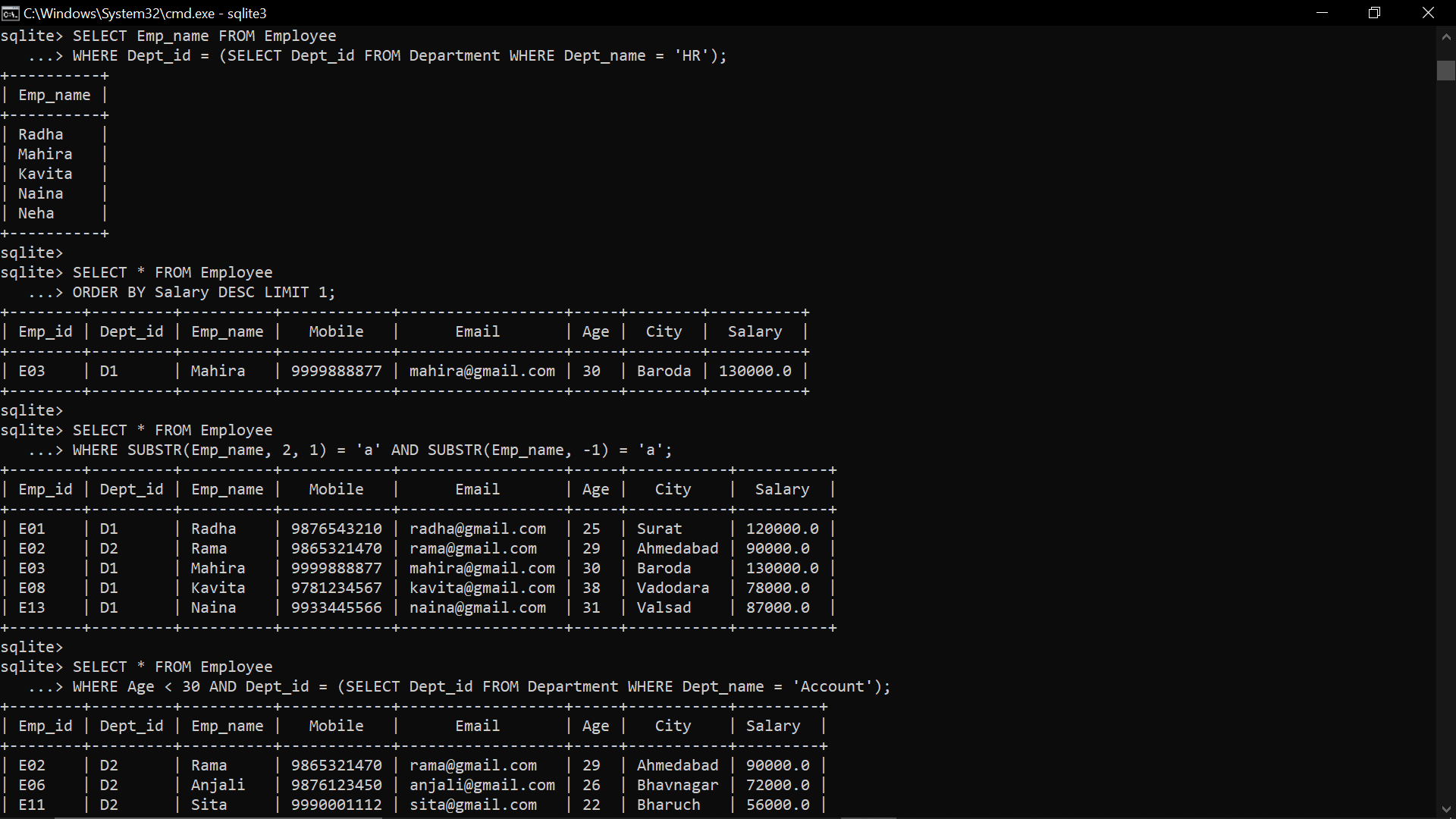
select \* from Employ;

****

1. **Find the employees name who works in “HR” department.**

SELECT Emp\_name FROM Employee

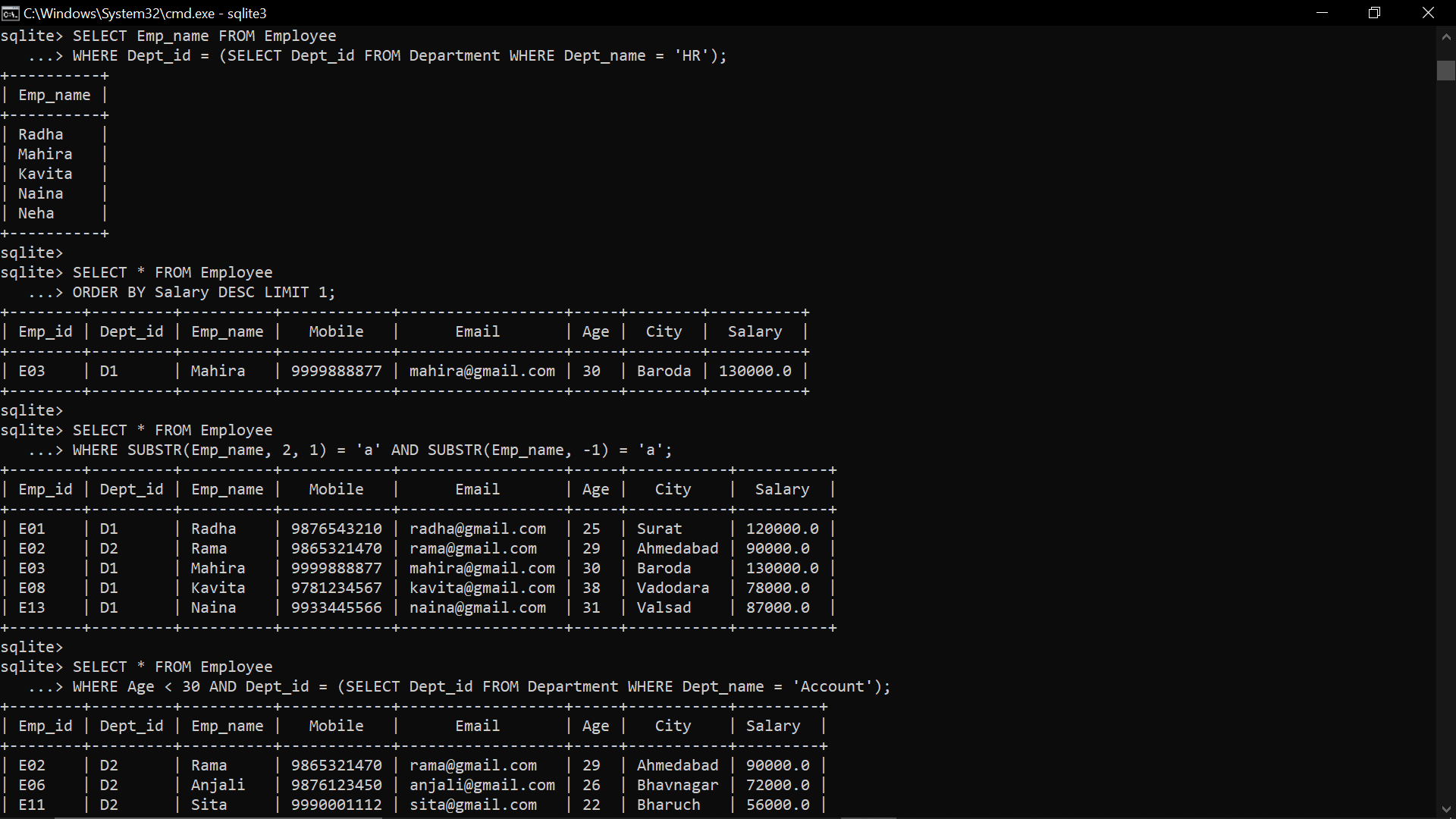
WHERE Dept\_id = (SELECT Dept\_id FROM Department WHERE Dept\_name = 'HR');



1. **Find the employee who has maximum salary.**

SELECT \* FROM Employee

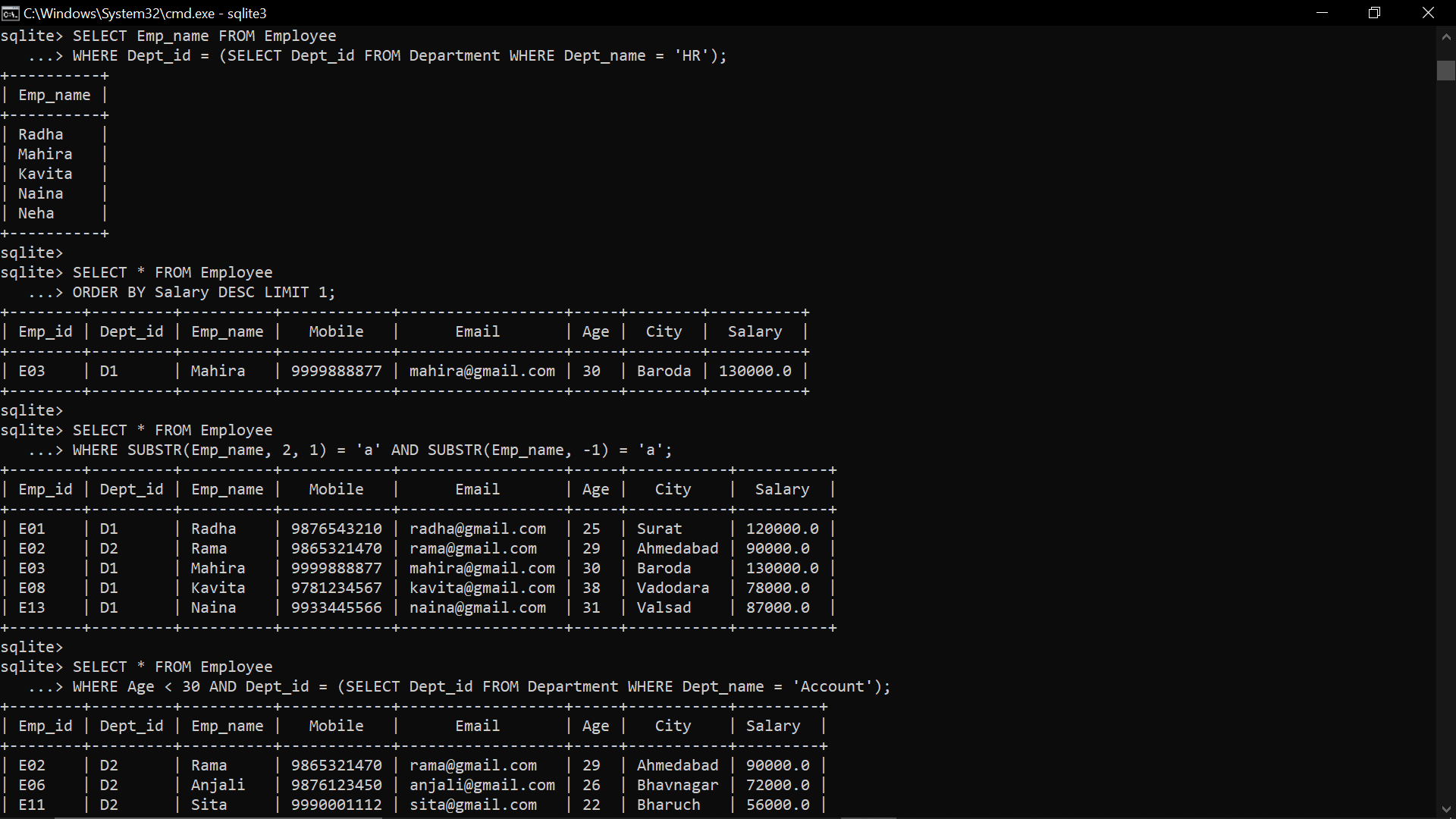
ORDER BY Salary DESC LIMIT 1;



1. **Find all the details of employees whose name’s second and last letter is ‘a’. Ex. Rama, Radha, Mahira...**

SELECT \* FROM Employee

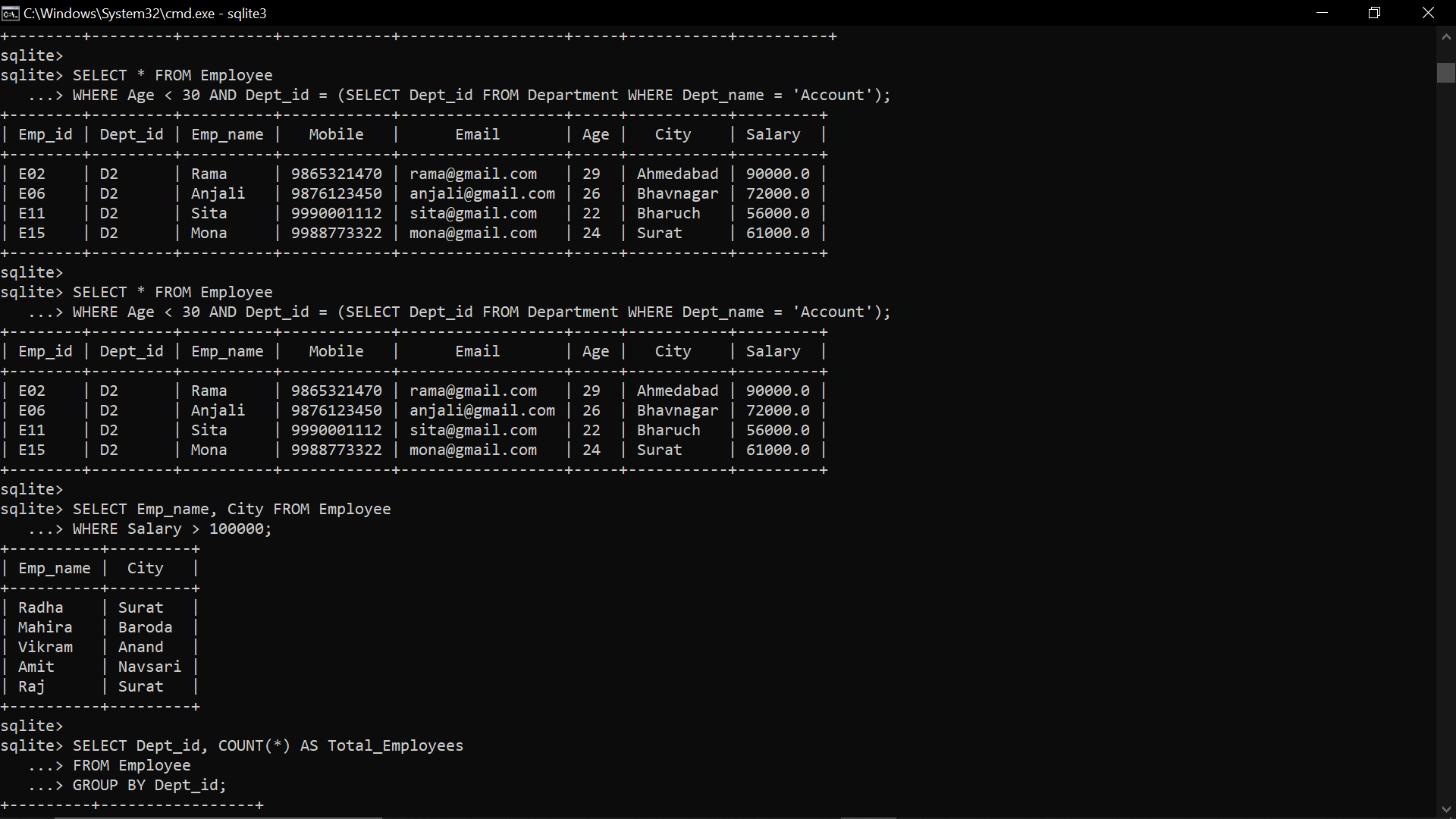
WHERE SUBSTR(Emp\_name, 2, 1) = 'a' AND SUBSTR(Emp\_name, -1) = 'a';



1. **Display all the employees whose age is less than 30 and working in Account department.**

SELECT \* FROM Employee

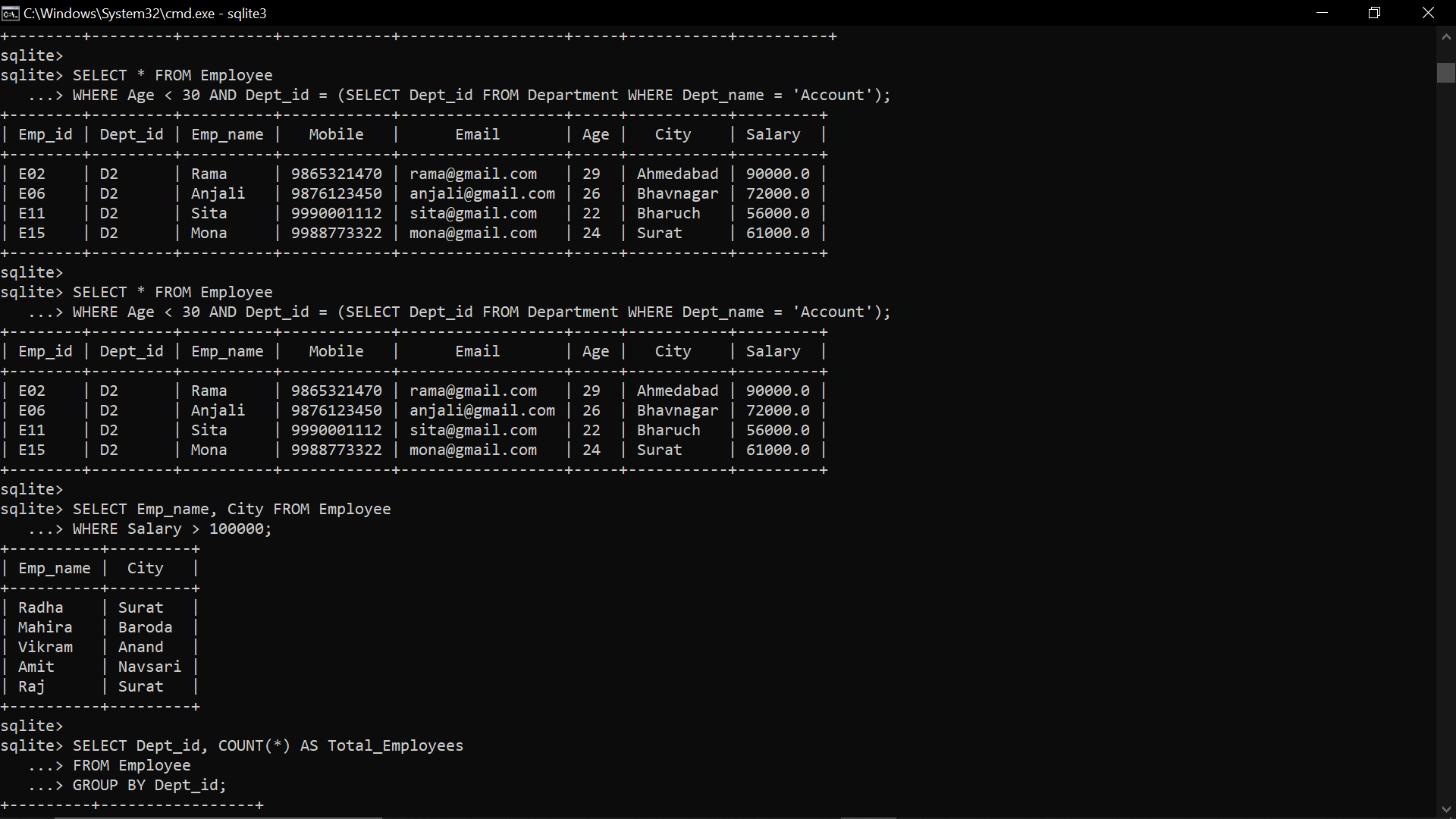
WHERE Age < 30 AND Dept\_id = (SELECT Dept\_id FROM Department WHERE Dept\_name = 'Account');



1. **Display customer name and city who have salary more than 100000.**

SELECT Emp\_name, City FROM Employee

WHERE Salary > 100000;

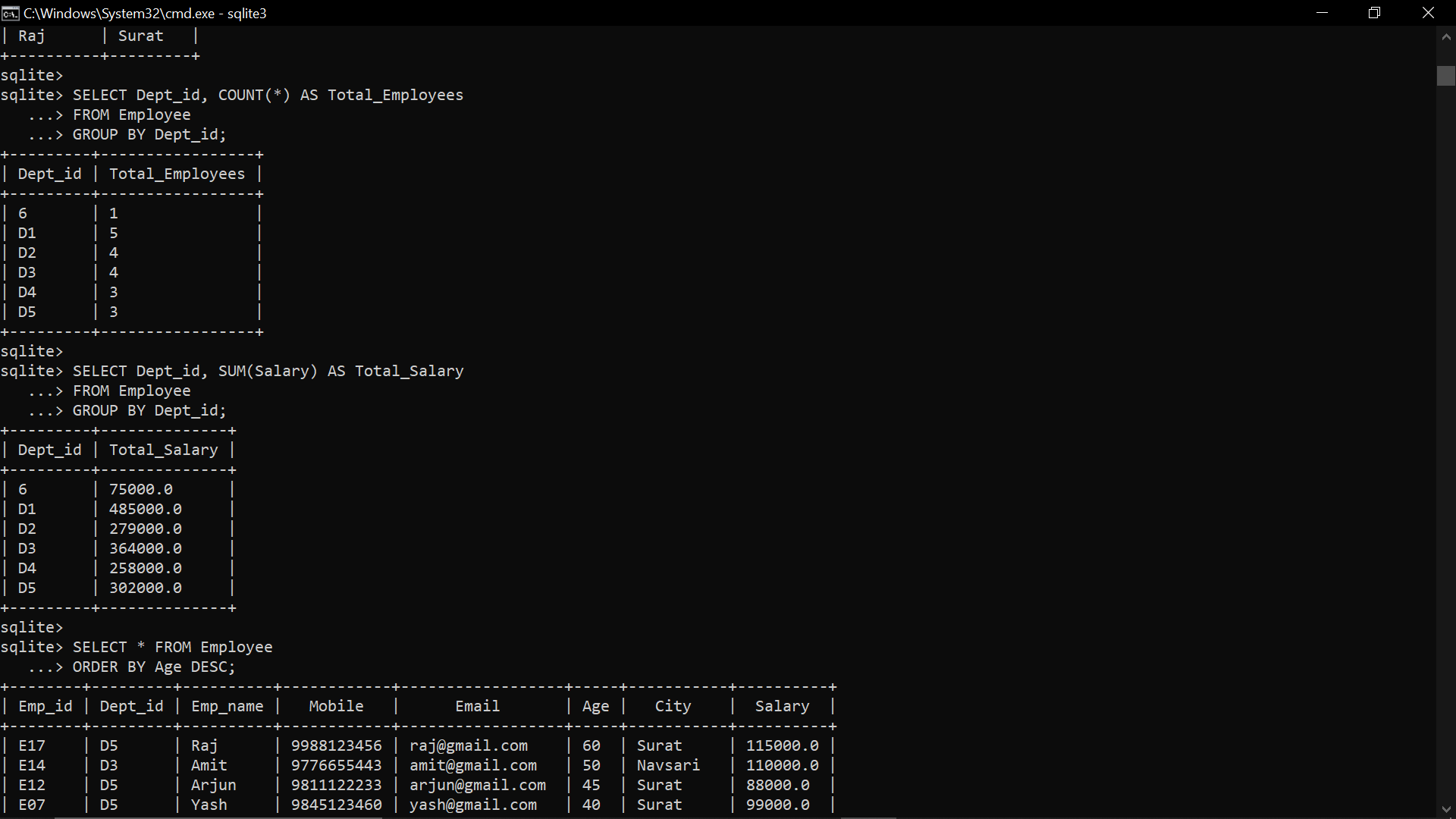


1. **Display total number of employees working in each department.**

SELECT Dept\_id, COUNT(\*) AS Total\_Employees

FROM Employee

GROUP BY Dept\_id;

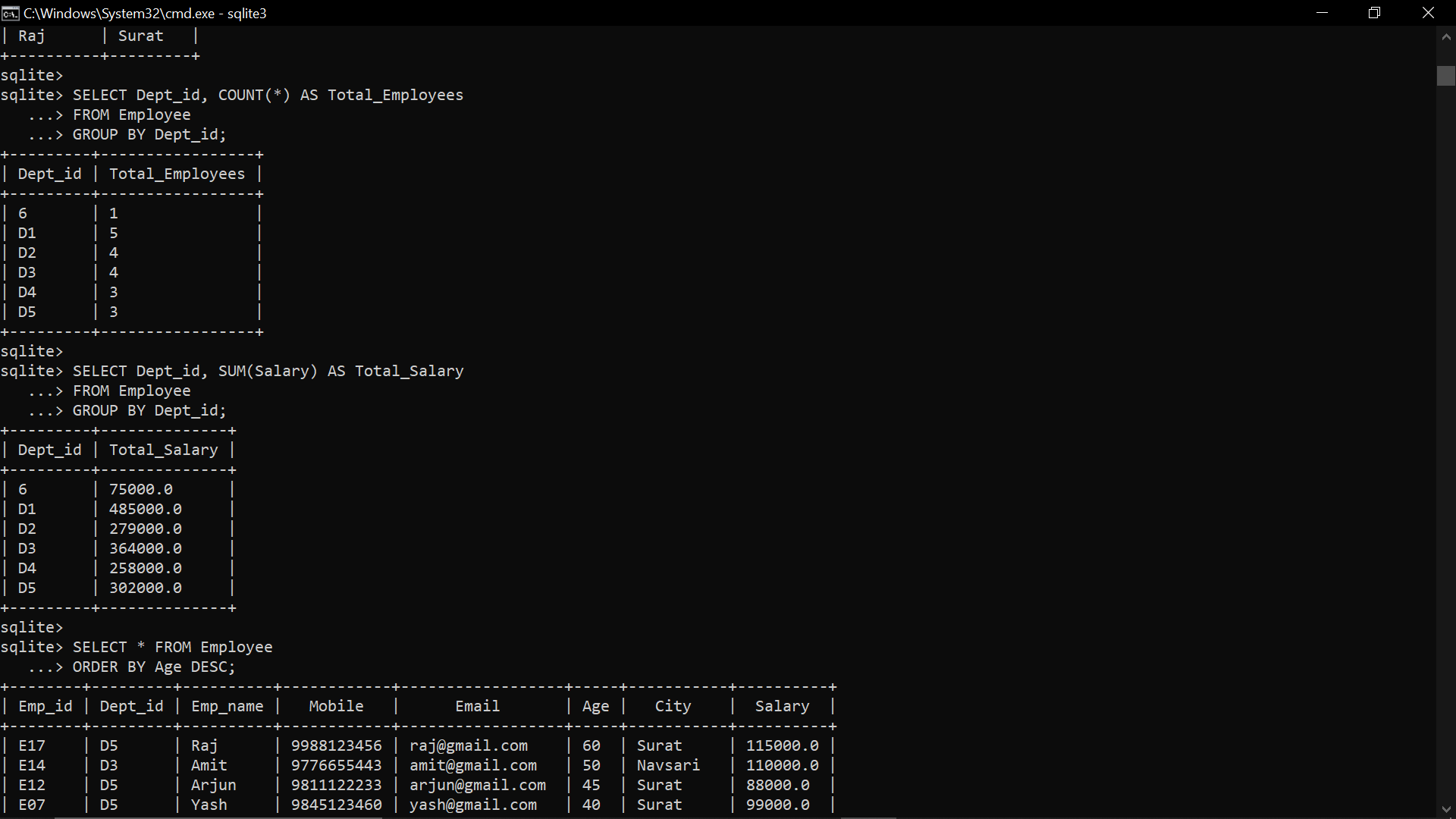


1. **Count total salary for each department.**

SELECT Dept\_id, SUM(Salary) AS Total\_Salary

FROM Employee

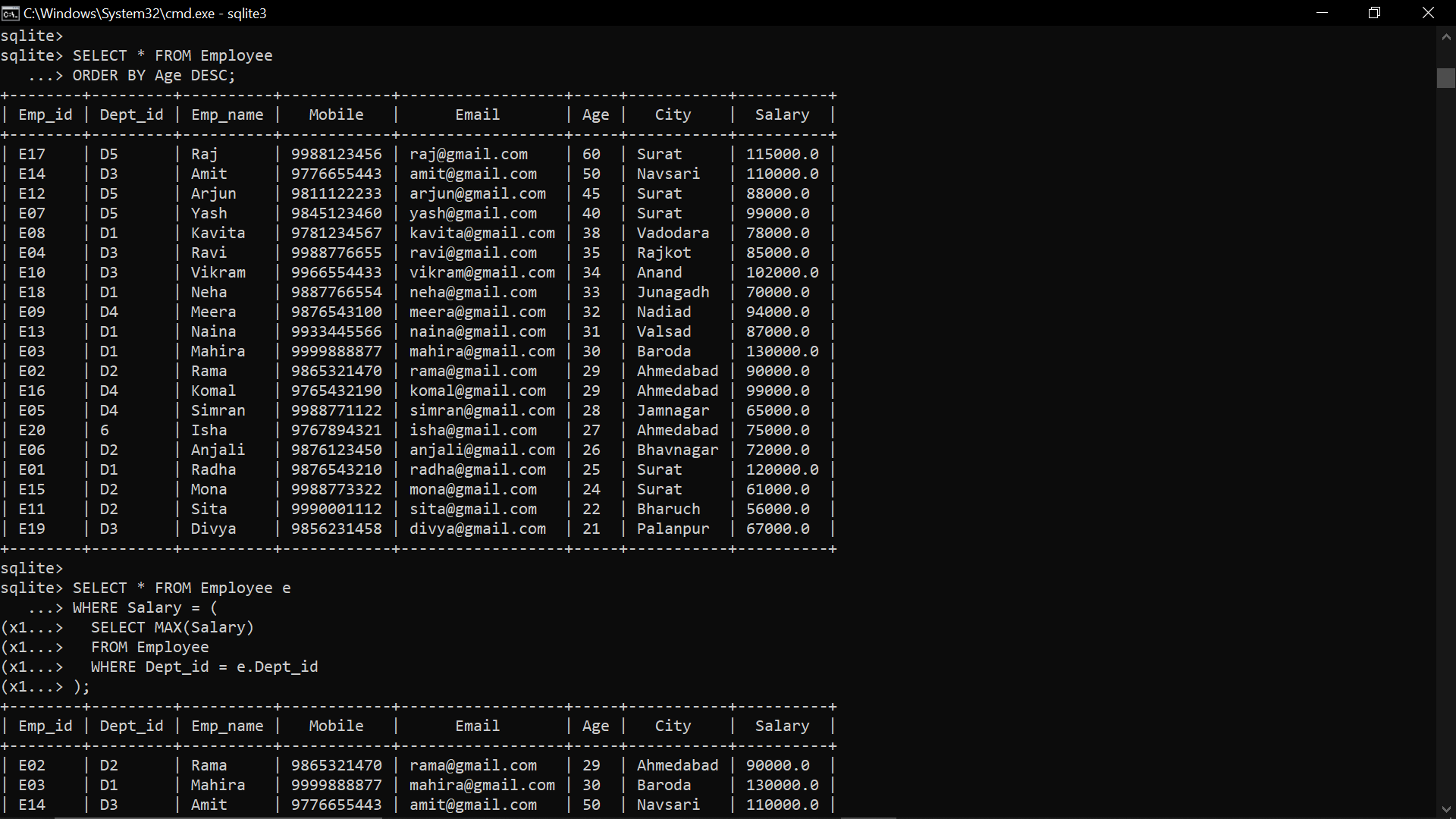
GROUP BY Dept\_id;



1. **Display all the employees in descending order of their age.**

SELECT \* FROM Employee

ORDER BY Age DESC;



1. **Display the employee from each department who is having maximum salary.**

SELECT \* FROM Employee e

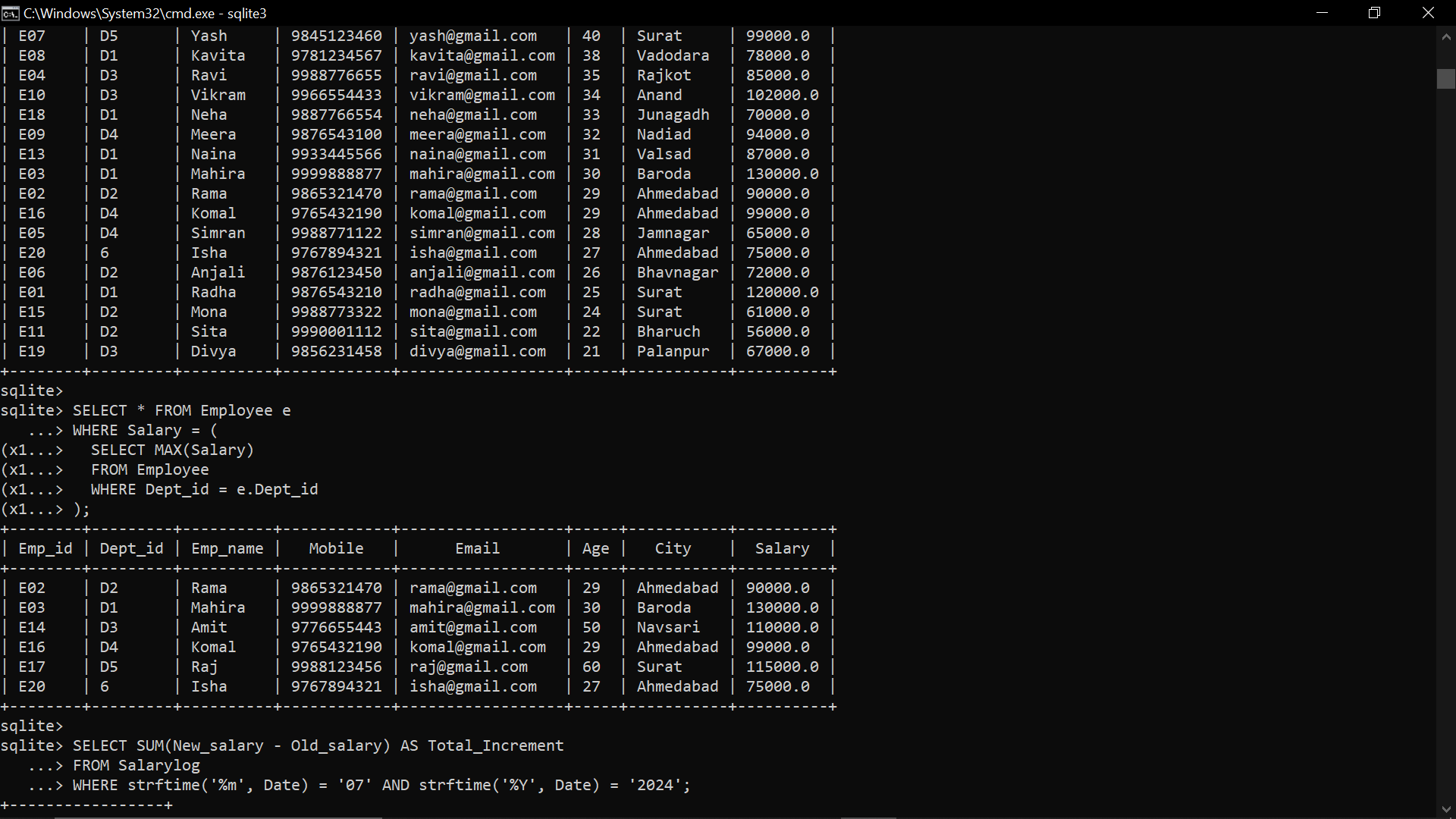
WHERE Salary = (

SELECT MAX(Salary)

FROM Employee

WHERE Dept\_id = e.Dept\_id

);

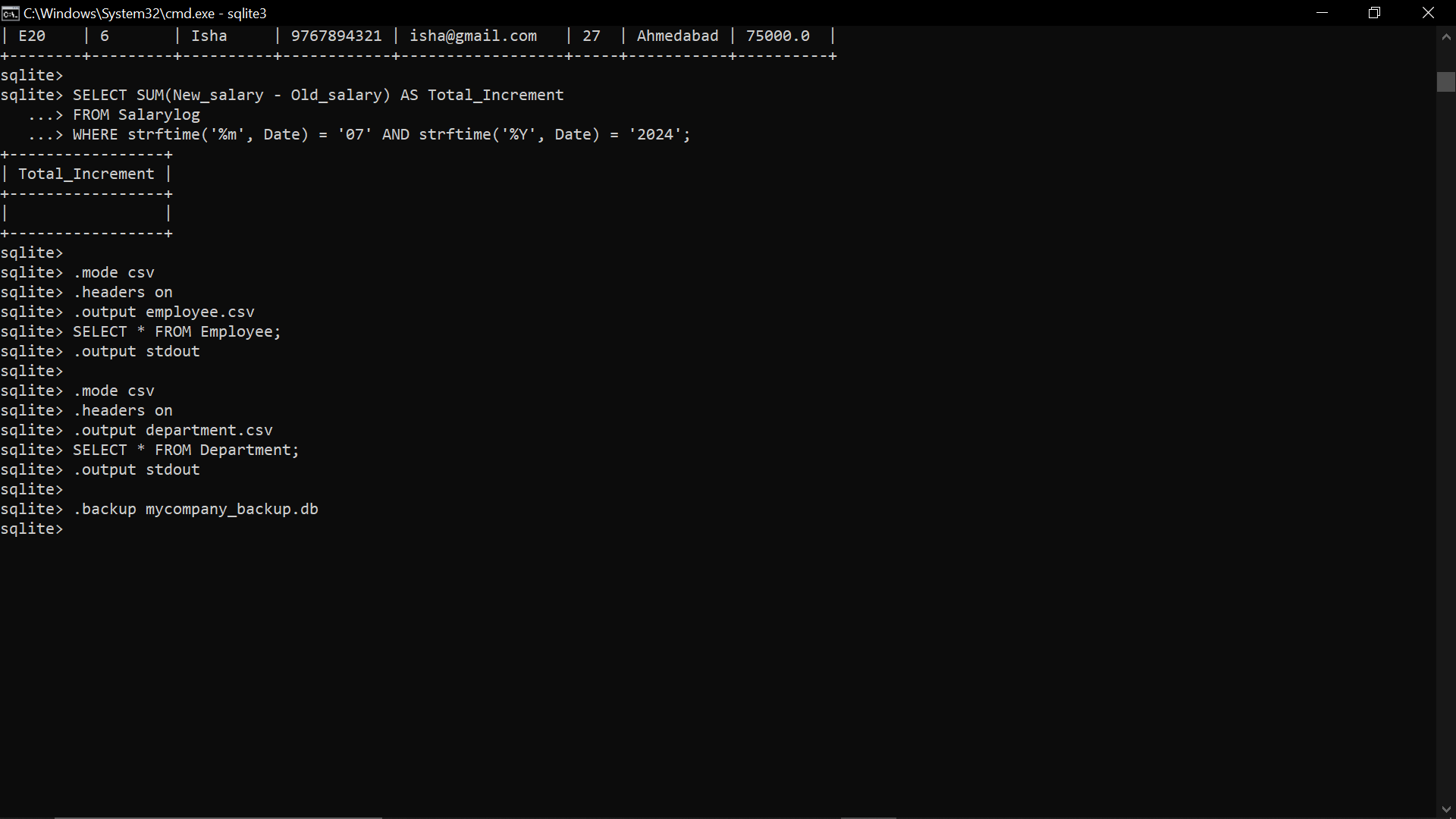


1. **Count total salary increment given to all employees in July month of 2024.**

SELECT SUM(New\_salary - Old\_salary) AS Total\_Increment

FROM Salarylog

WHERE strftime('%m', Date) = '07' AND strftime('%Y', Date) = '2024';



1. **Export employee table into employee.csv file.**

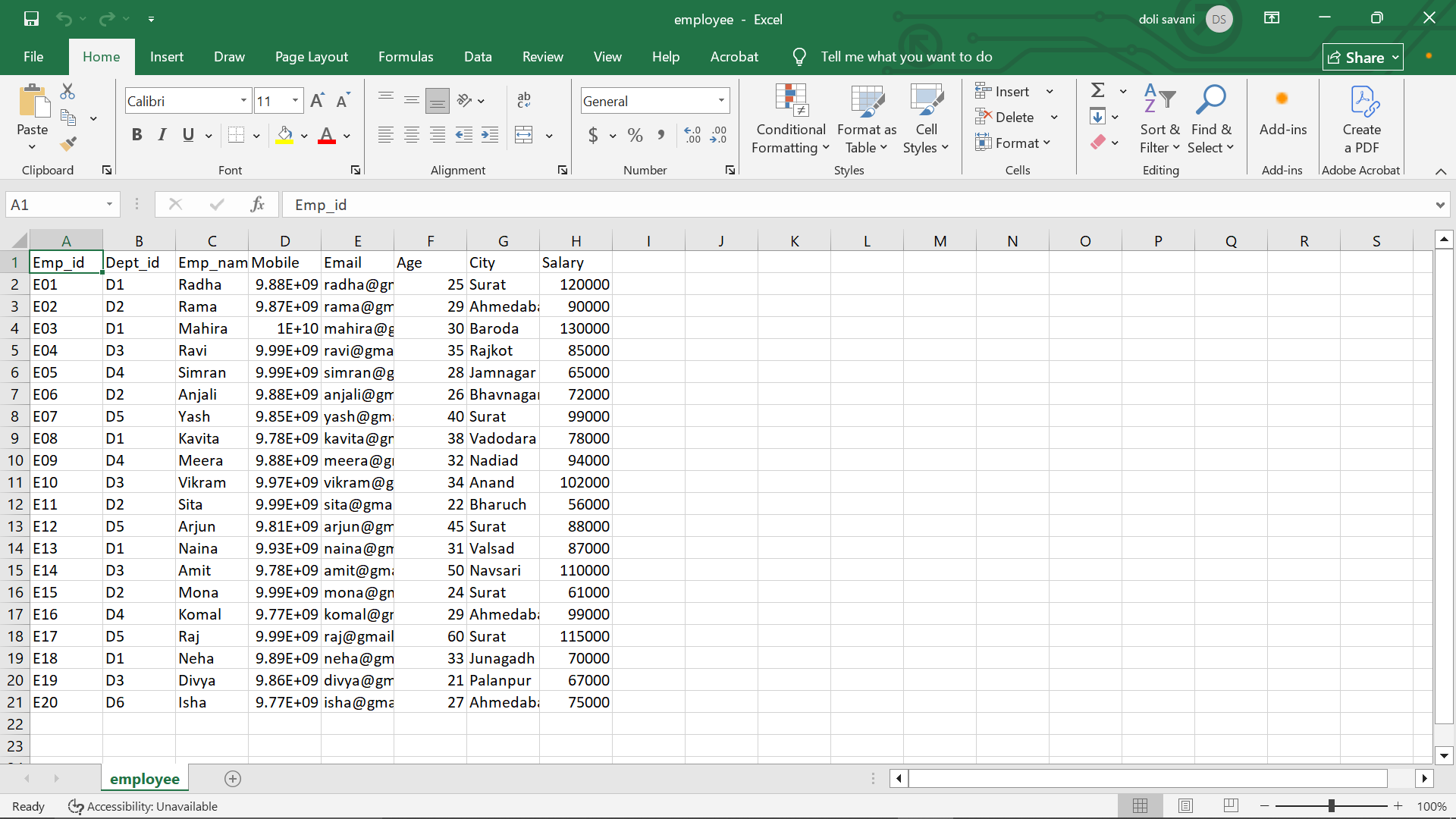
.mode csv

.headers on

.output employee.csv

SELECT \* FROM Employee;

.output stdout



1. **Export department table data into department.csv file.**

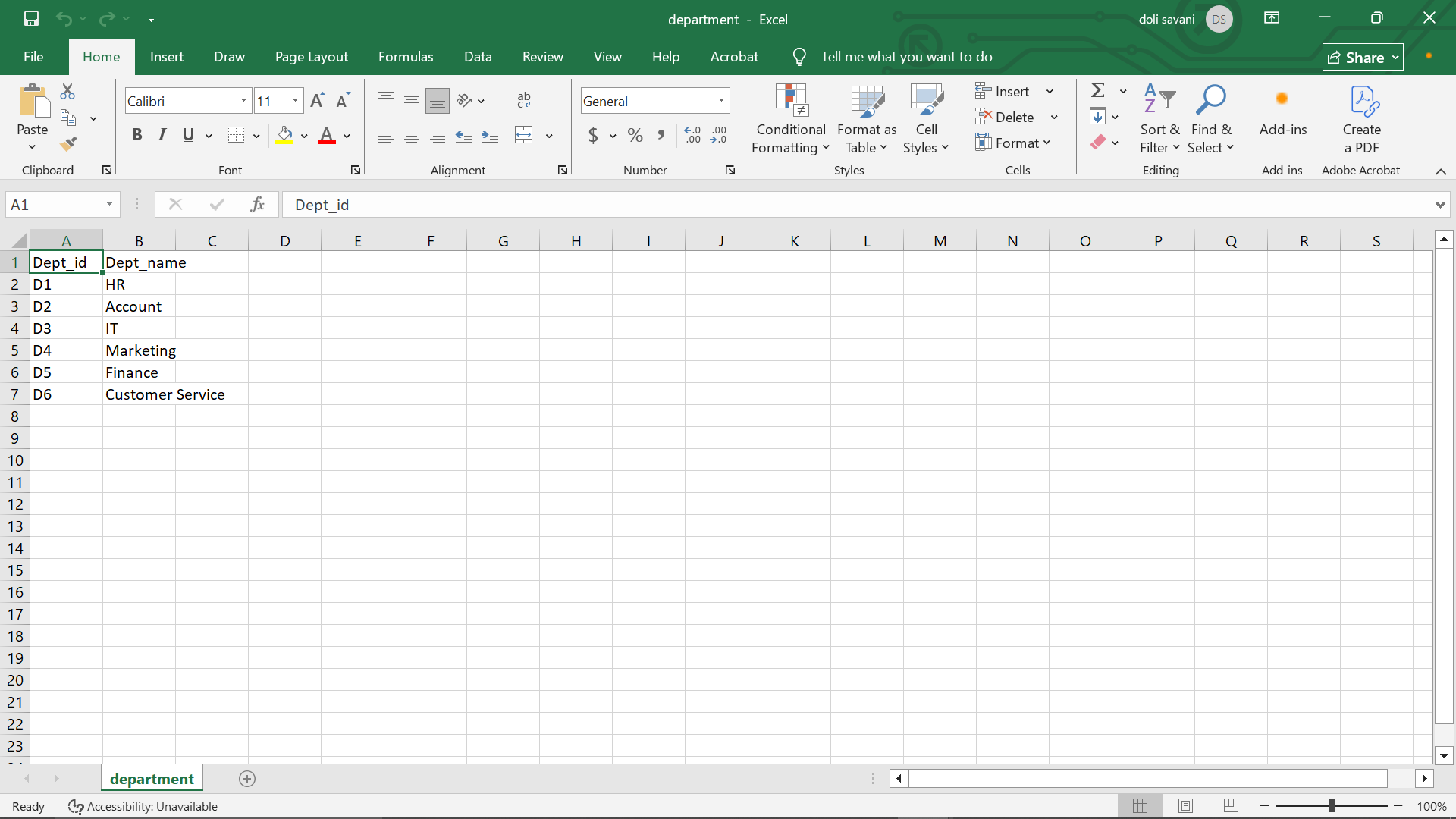
.mode csv

.headers on

.output department.csv

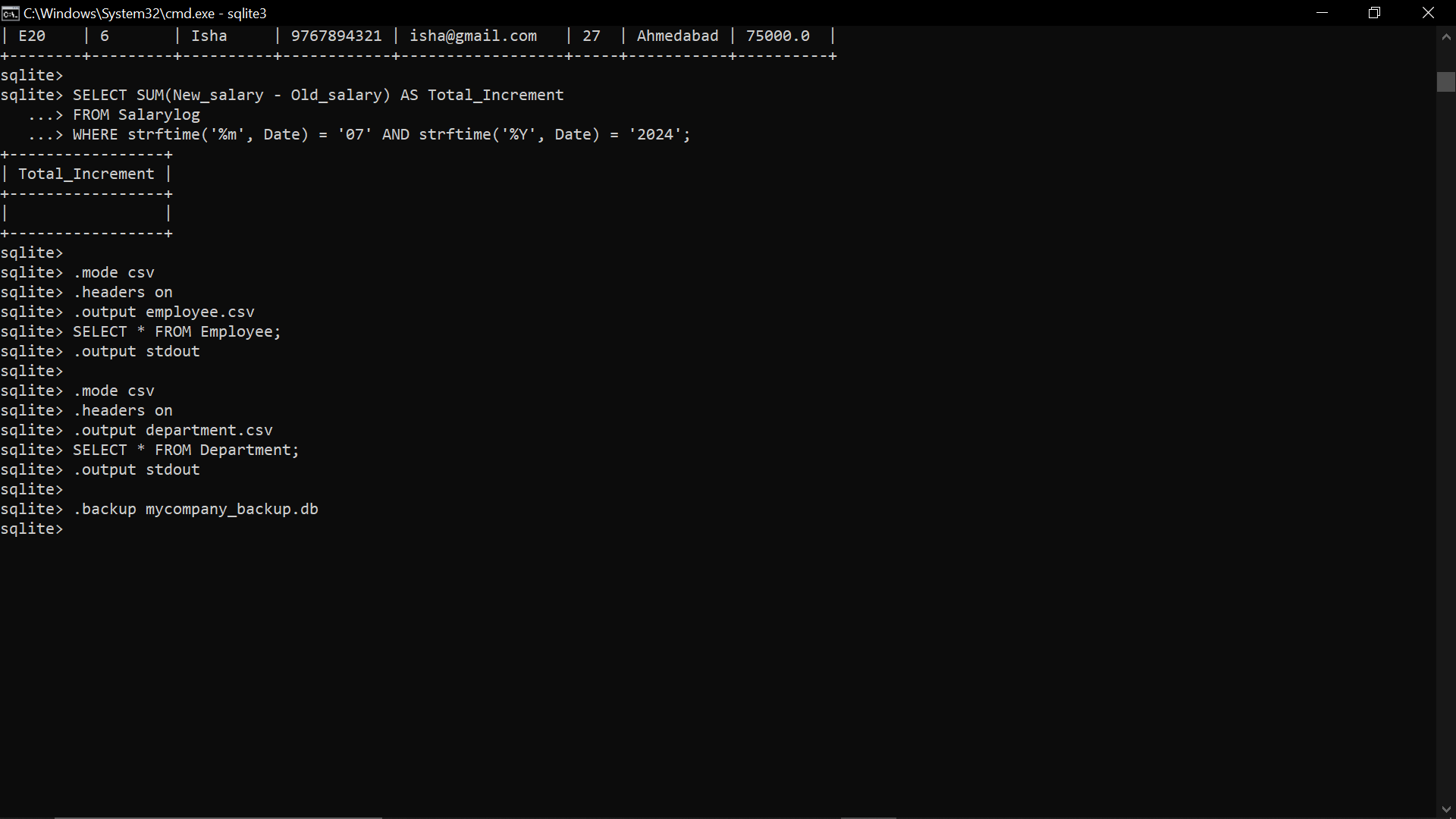
SELECT \* FROM Department;

.output stdout

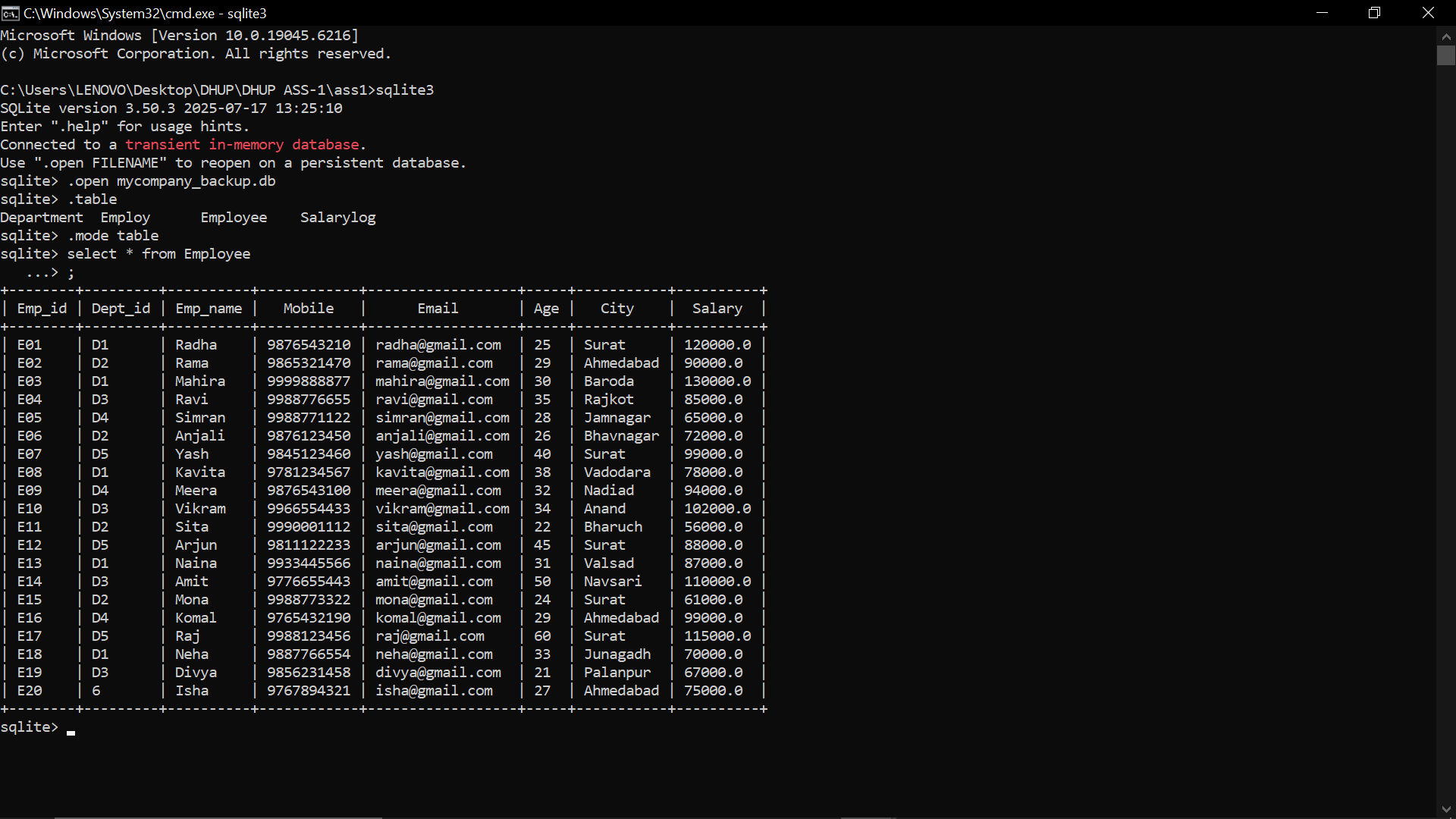


1. **Take backup of whole database in “mycompany” file.**

.backup mycompany\_backup.db

****

**Back\_up Database:-**

****