**SQL ASSIGNMENTS**

Create a database worker that should contain **first name, last name email, department, salary, Join Date** with 50 employees.

🡪 CREATE DATABASE WORKER;



A picture containing shape

Description automatically generated

CREATE TABLE Worker (

WORKER\_ID INT NOT NULL PRIMARY KEY AUTO\_INCREMENT,

FIRST\_NAME CHAR(25),

LAST\_NAME CHAR(25),

EMAIL VARCHAR(50),

SALARY INT(15),

JOINING\_DATE DATE,

DEPARTMENT VARCHAR(25)

);

Text

Description automatically generated

CREATE TABLE Worker (

WORKER\_ID INT NOT NULL PRIMARY KEY AUTO\_INCREMENT,

FIRST\_NAME CHAR(25),

LAST\_NAME CHAR(25),

SALARY INT(15),

JOINING\_DATE DATE,

DEPARTMENT VARCHAR(25)

);

Example:

INSERT INTO Worker (FIRST\_NAME, LAST\_NAME, EMAIL, SALARY, JOINING\_DATE, DEPARTMENT)

VALUES

('sathish', 'kumar', 'sathishkumar@guvi.com',60000,'2011-01-01', 'ENG'),

('ramesh', 'kumar', 'rameshkumar@guvi.com',25000,'2016-01-05', 'ADMIN'),

('Santhosh', 'kumar', 'Santhoshkumar@guvi.com',3000,'2017-06-25', 'SALES'),

('Sathish', 'Raj', 'SathishRaj@guvi.com',33000,'2018-01-08', 'ADMIN'),

('kiran', 'Arora', 'kiranArorag@guvi.com',22000,'2019-07-01', 'HR'),

('vivek', 'Arora', 'vivekArorag@guvi.com',32000,'2020-01-22', 'ENG'),

('smitha', 'ram', 'smitharam@guvi.com',35000,'2021-01-10', 'ADMIN'),

('niharika', 'verma', 'niharikaverma@guvi.com',45000,'2022-01-01', 'SALES'),

('suresh', 'kumar', 'sureshkumar@guvi.com',50000,'2014-01-18', 'FINANCE');

**Task-1**

1. Write an SQL query to fetch “FIRST\_NAME” from the Worker table using the alias name as <WORKER\_NAME>.

* SELECT FIRST\_NAME as WORKER\_NAME from worker;

Text

Description automatically generated

1. Write an SQL query to fetch unique values of DEPARTMENT from the Worker table.

* SELECT DISTINCT DEPARTMENT FROM WORKER;

🡪select count(distinct department) from worker;

* SELECT Count(\*) AS Distinct Departments

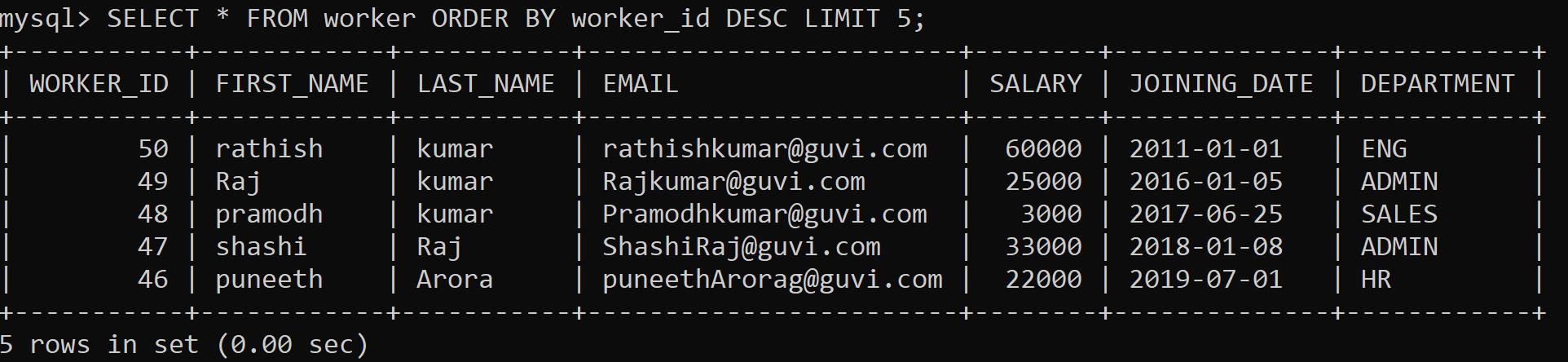
FROM (SELECT DISTINCT DEPARTMENT FROM worker);

Text

Description automatically generated

1. Write an SQL query to show the last 5 records from a table.

* SELECT \* from worker LIMIT 5;
* SELECT \* FROM worker ORDER BY worker\_id DESC LIMIT 5;



**Task-2**

1. Write an SQL query to print the first three characters of FIRST\_NAME from Worker

🡪 Select substring(FIRST\_NAME,1,3) from Worker;

A picture containing text

Description automatically generated

1. Write an SQL query to find the position of the alphabet (‘a’) in the first name

Select INSTR(FIRST\_NAME, BINARY'a') from Worker;

* FOR SQL
* SELECT CHARINDEX('a', 'worker') AS MatchPosition from worker;

Select INSTR(FIRST\_NAME, BINARY'a') from Worker where FIRST\_NAME = 'Amitabh';

* Shape

  Description automatically generated

1. Write an SQL query to print the name of employees who have the highest salary in each department.

🡪 Select department, Max(salary) from worker group by department;

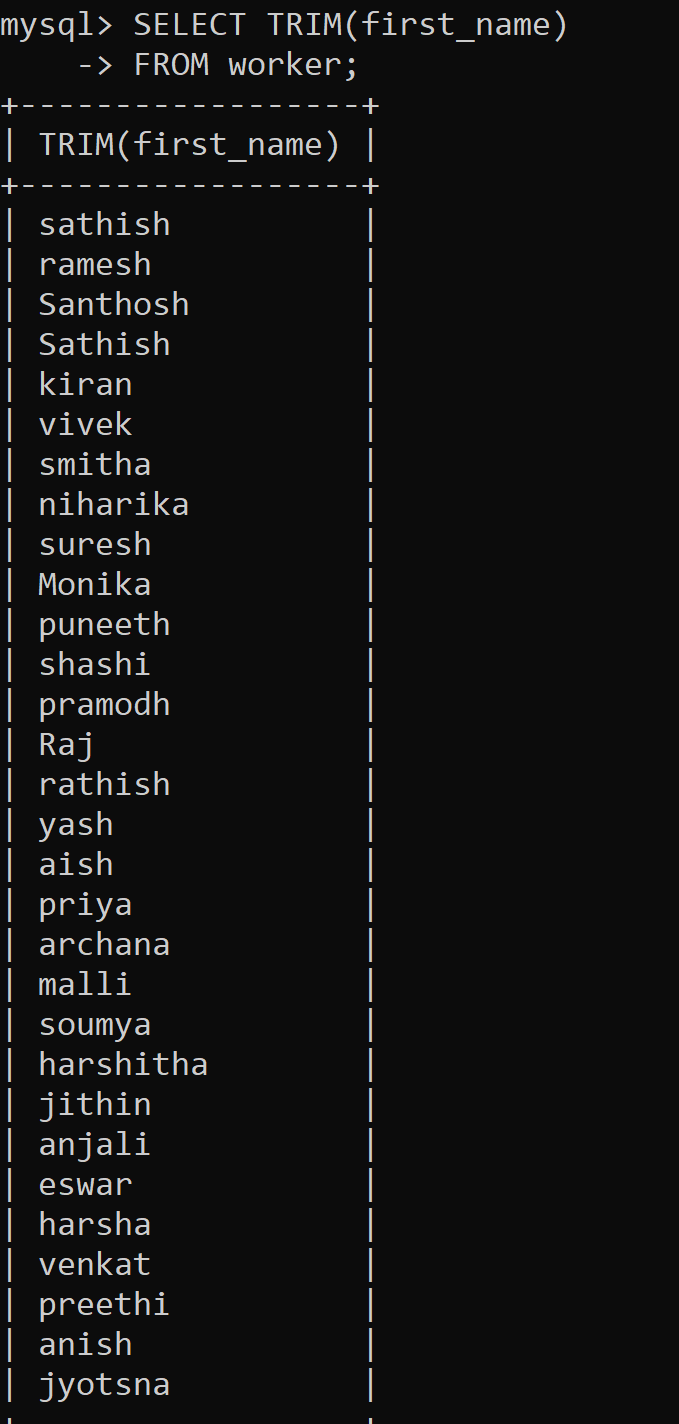
Text

Description automatically generated

**Task-3**

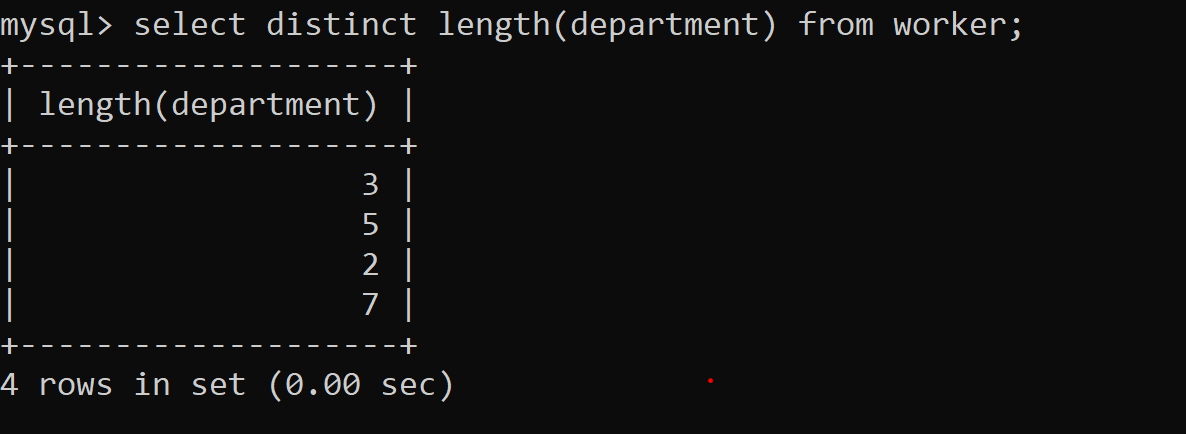
1. Write an SQL query to print the FIRST\_NAME from the Worker table after removing white spaces from the right side.

🡪 Select RTRIM(FIRST\_NAME) from worker;



1. Write an SQL query that fetches the unique values of DEPARTMENT from the Worker table and prints its length.

🡪 Select DISTINCT length(DEPARTMENT) from worker;



1. Write an SQL query to fetch nth max salaries from a table.

🡪 select max(salary) from worker;

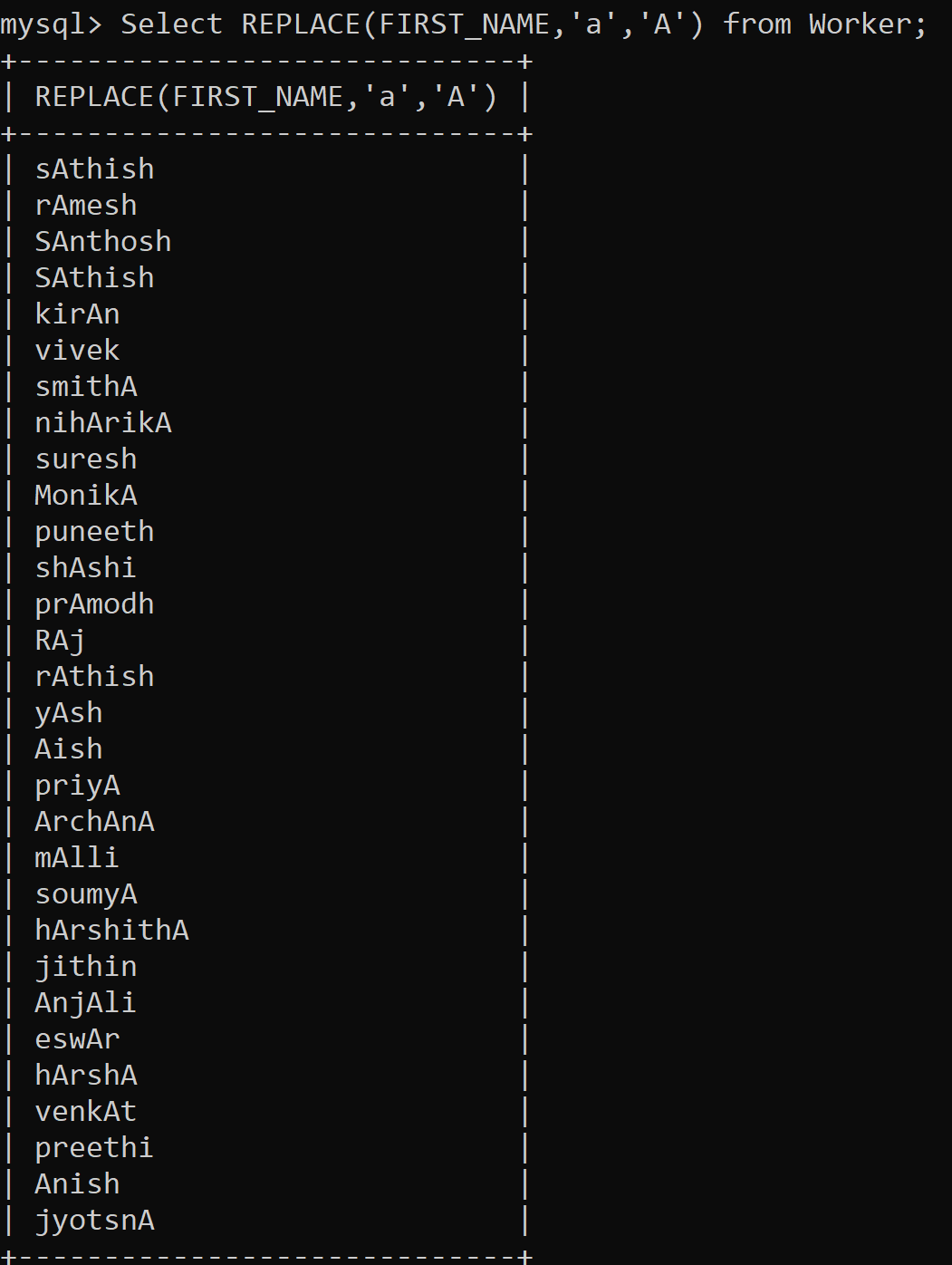
Graphical user interface, text

Description automatically generated

**Task-4**

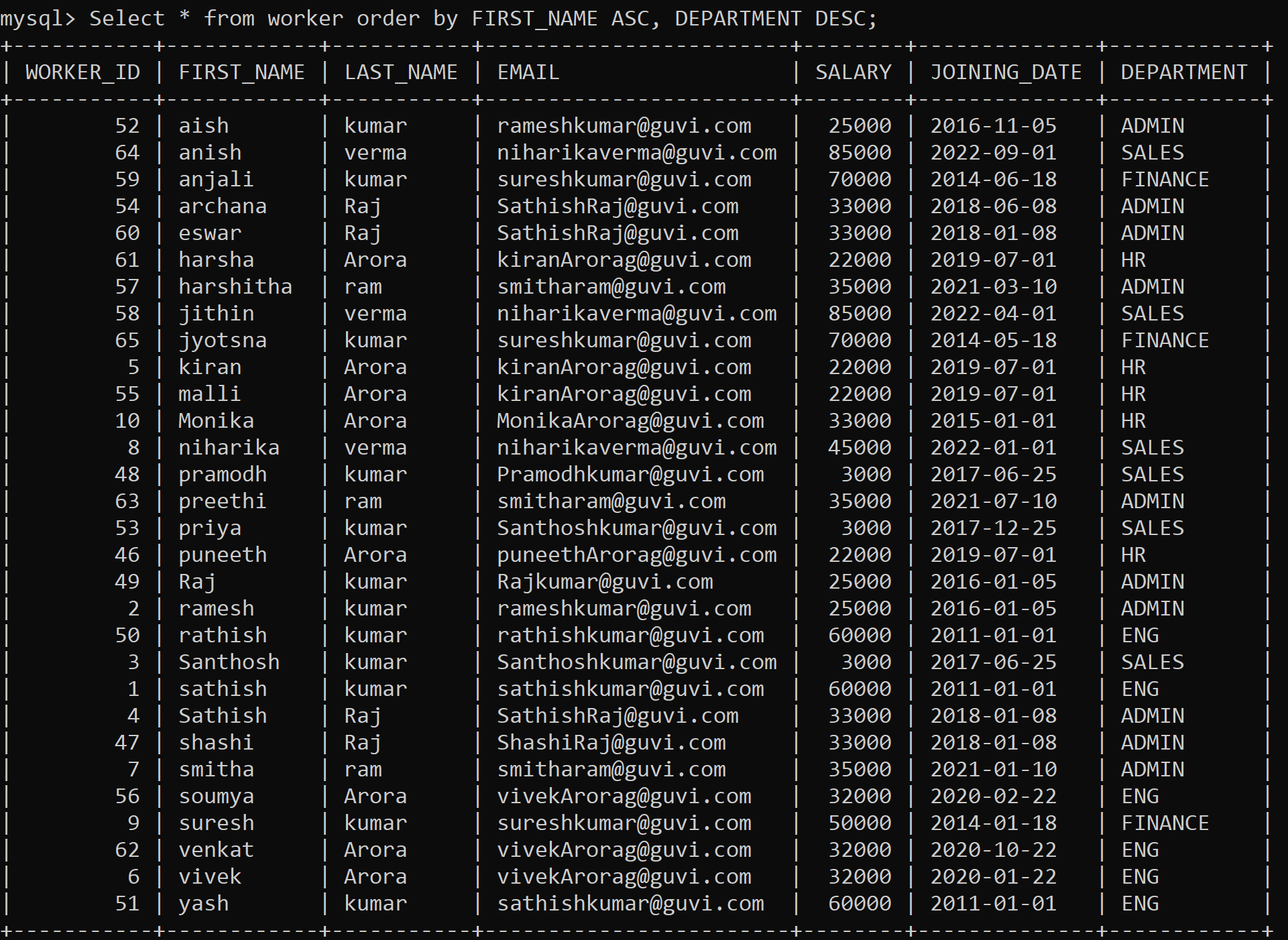
1. Write an SQL query to print the FIRST\_NAME from the Worker table after replacing ‘a’ with ‘A’.

* Select REPLACE(FIRST\_NAME , ‘a’, ‘A’) from worker;
* Select REPLACE(FIRST\_NAME, ‘a’, ’A’) from worker;



1. Write an SQL query to print all Worker details from the Worker table order FIRST\_NAME Ascending and DEPARTMENT Descending.

* Select \* from worker order by FIRST\_NAME ASC, DEPARTMENT DESC;
* Select \* from Worker order by FIRST\_NAME asc,DEPARTMENT desc;

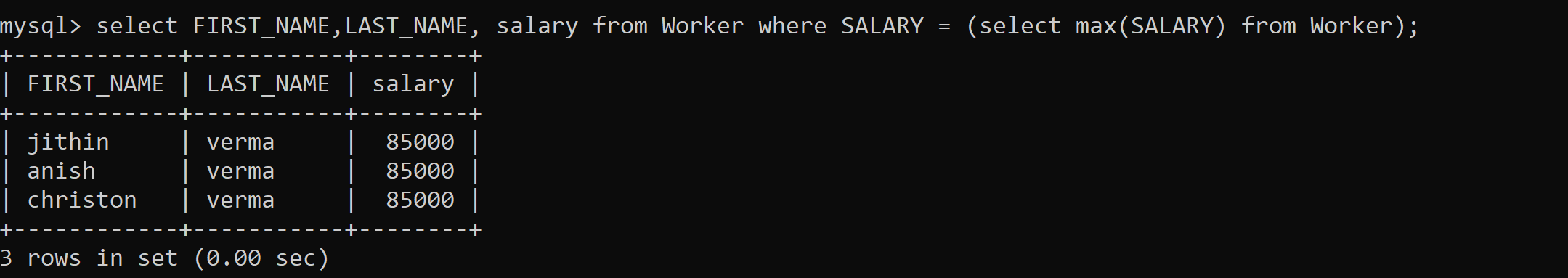


1. Write an SQL query to fetch the names of workers who earn the highest salary.

Select FIRST\_NAME , LAST\_NAME as worker\_name ,max(salary) from worker;

* Select FIRST\_NAME, max(salary) from worker;

select FIRST\_NAME,LAST\_NAME from Worker where SALARY = (select max(SALARY) from Worker);



)

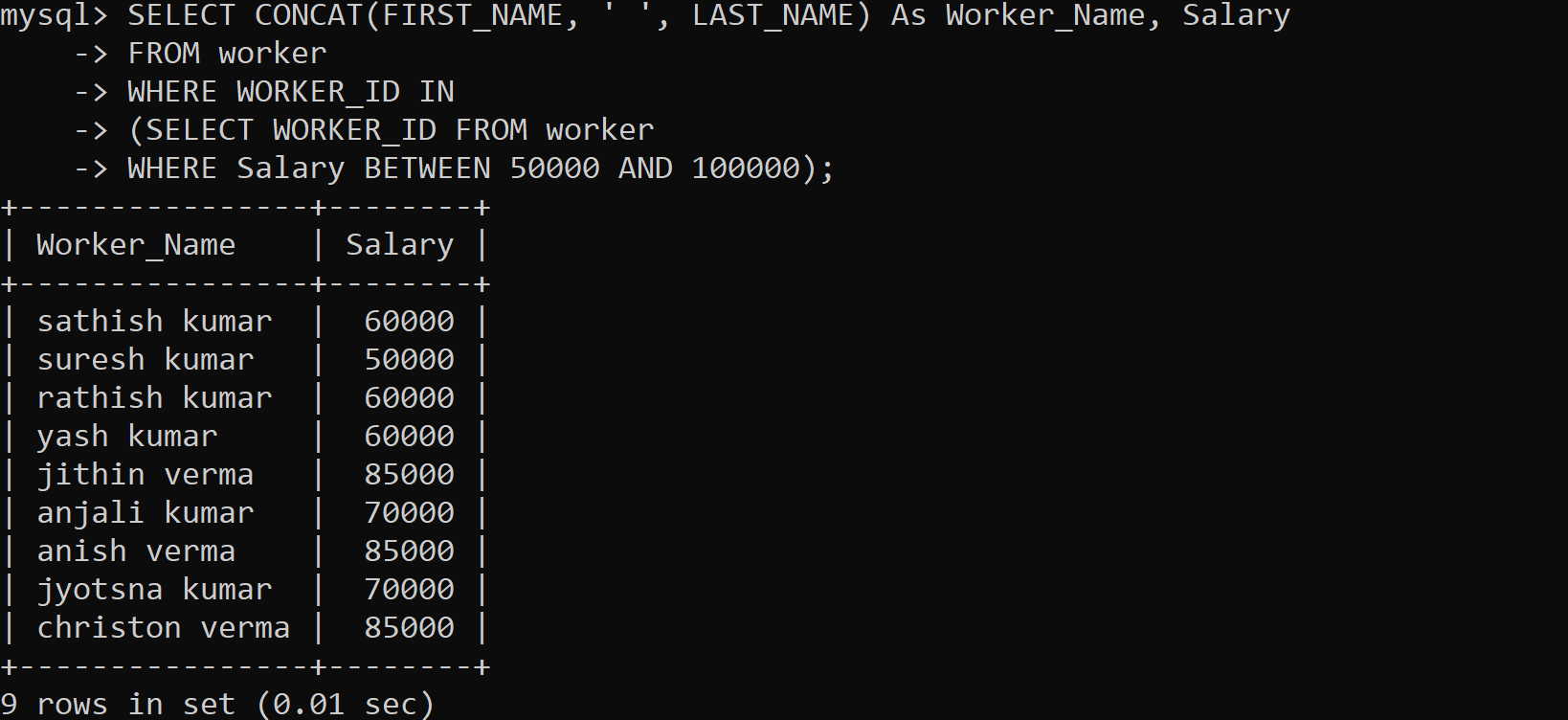
SELECT CONCAT(FIRST\_NAME, ' ', LAST\_NAME) As Worker\_Name, Salary

FROM worker

WHERE WORKER\_ID IN

(SELECT WORKER\_ID FROM worker

WHERE Salary BETWEEN 50000 AND 100000);

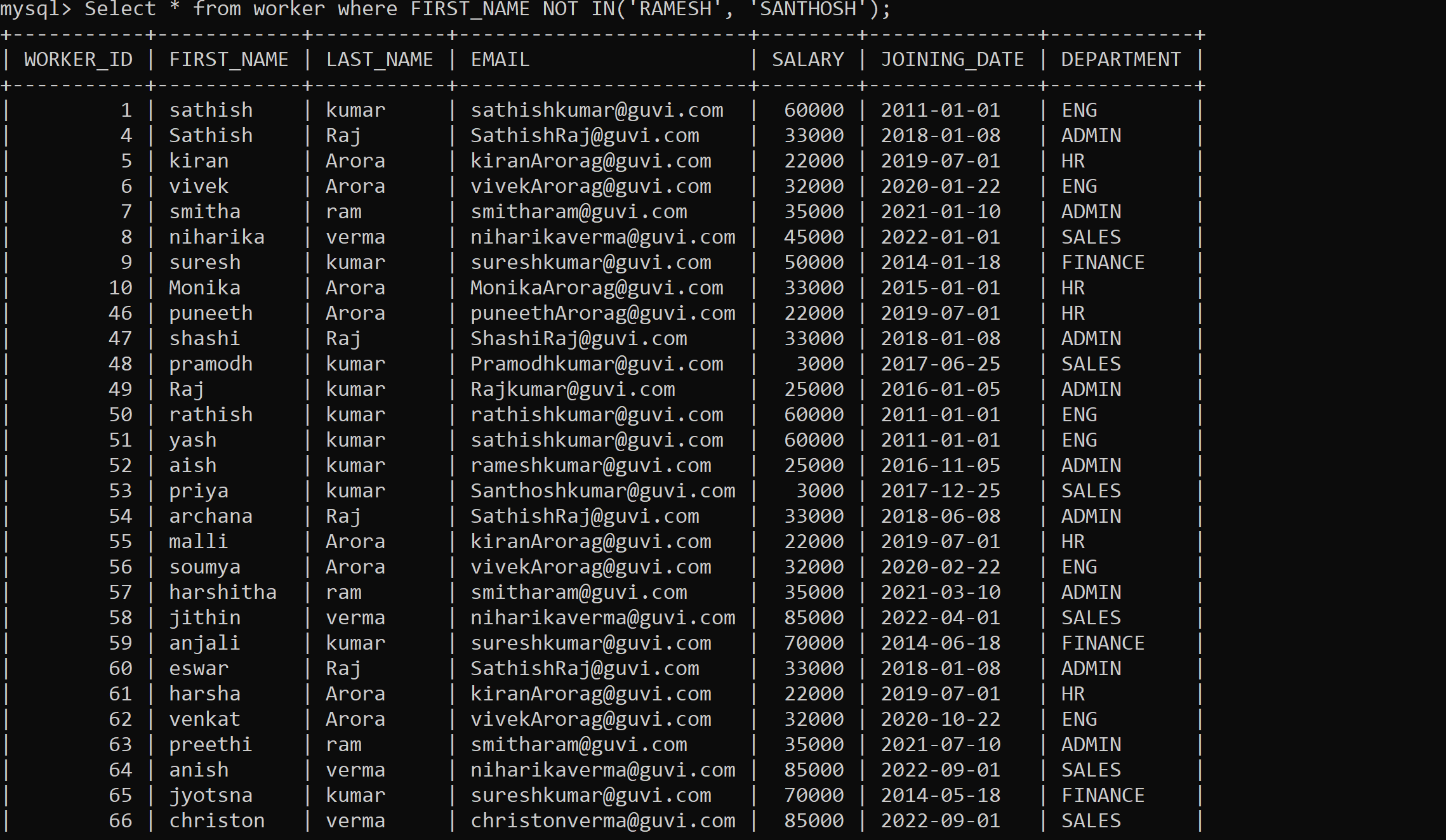


**Task-5**

1. Write an SQL query to print details of workers excluding first names, “Ramesh” and “Santhosh” from the Worker table.

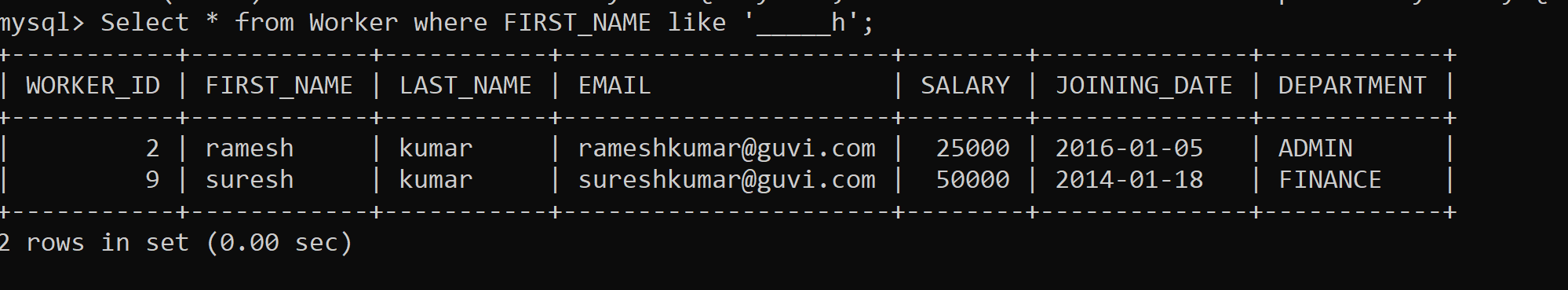
🡪Select \* from worker where FIRST\_NAME NOT IN('RAMESH', 'SANTHOSH');

* Select \* from worker where FIRST\_NAME NOT IN ('RAMESH', 'SANTHOSH');



1. Write an SQL query to print details of the Workers whose FIRST\_NAME ends with ‘h’ and contains six alphabets.

* Select \* from Worker where FIRST\_NAME like '\_\_\_\_\_h';



1. Write a query to validate Email of Employee (email should have first name last name and guvi.com example (first name=Kamal last name= raja and the mail id should be [kamalraja@guvi.com](mailto:kamalraja@guvi.com)).

* SELECT \* FROM worker

WHERE email LIKE '%@guvi.com';

* SELECT \* FROM people WHERE email LIKE '%\_@\_\_%.\_\_%'

A picture containing graphical user interface

Description automatically generated

**Task-6**

1. Write an SQL query to print details of the Workers who have joined in March ’2021.

* Select \* from Worker where year(JOINING\_DATE) = 2021 and month(JOINING\_DATE) = 3;

Graphical user interface, text

Description automatically generated

1. Write an SQL query to fetch duplicates that have matching data in some fields of a table.

* SELECT email,salary,JOINING\_DATE,DEPARTMENT, COUNT(\*)

FROM worker

GROUP BY email,salary,JOINING\_DATE,DEPARTMENT

HAVING

COUNT(\*) > 1;

Graphical user interface

Description automatically generated

SELECT email, COUNT(\*)

FROM worker

GROUP BY email

HAVING

COUNT(\*) > 1;

Text

Description automatically generated

1. How to remove duplicate rows from the Employees table.

SELECT

worker\_id

FROM (

SELECT

worker\_id,

ROW\_NUMBER() OVER (

PARTITION BY email

ORDER BY email) AS row\_num

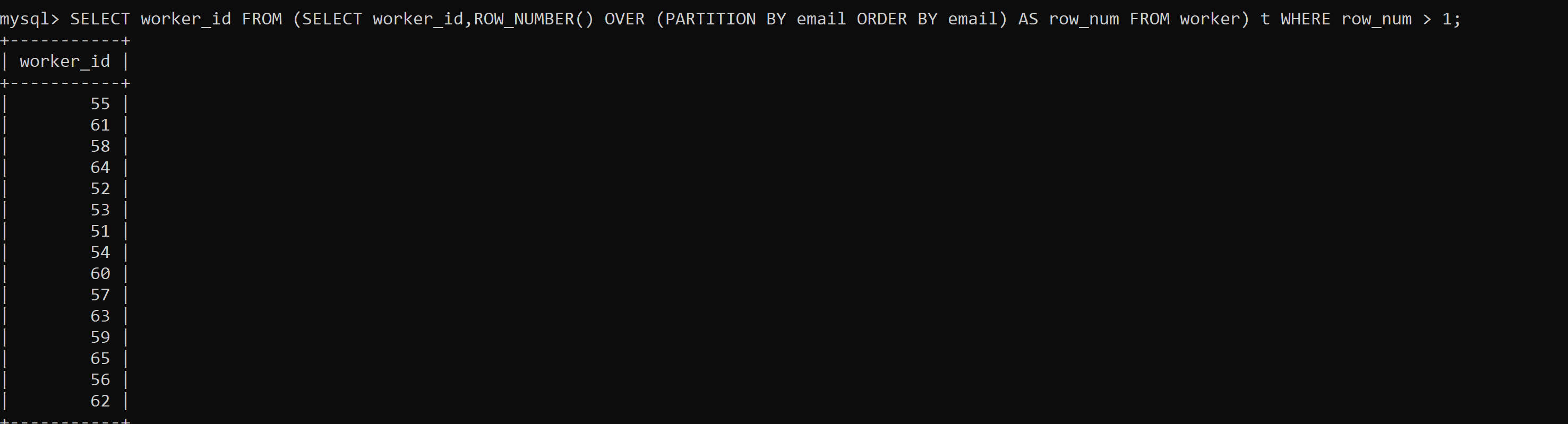
FROM

worker

) t

WHERE

row\_num > 1;



**Task-7**

1. Write an SQL query to show only odd rows from a table.

SELECT \* FROM Worker WHERE MOD (WORKER\_ID, 2) <> 0;

Text

Description automatically generated

1. Write an SQL query to clone a new table from another table.

The general query to clone a table with data is:

SELECT \* INTO WorkerClone FROM Worker;

The general way to clone a table without information is:

SELECT \* INTO WorkerClone FROM Worker WHERE 1 = 0;

An alternate way to clone a table (for MySQL) without is:

CREATE TABLE WorkerClone LIKE Worker;

INSERT workerclone SELECT \* FROM worker;

Text

Description automatically generated

**Task-8**

1. Write an SQL query to fetch intersecting records of two tables.

(SELECT \* FROM Worker)

INTERSECT

(SELECT \* FROM WorkerClone);

MYSQL:

SELECT worker.worker\_id

FROM worker

WHERE worker.worker\_id < 100

AND worker.worker\_id IN

(SELECT workerclone.worker\_id

FROM workerclone

WHERE workerclone.salary > 100);

A picture containing text

Description automatically generated

1. Write an SQL query to show records from one table that another table

does not have.

SELECT \* FROM Worker

MINUS

SELECT \* FROM Title;

SELECT worker\_id

FROM worker

WHERE NOT EXISTS

(SELECT \*

FROM workerclone

WHERE workerclone.worker\_id= worker.worker\_id);

Graphical user interface, text, application

Description automatically generated

**Task-9**

1. Write an SQL query to show the top n (say 15) records of a table.

Following MySQL query will return the top n records using the LIMIT method:

SELECT \* FROM Worker ORDER BY Salary DESC LIMIT 10;

Following SQL Server query will return the top n records using the TOP command:

SELECT TOP 10 \* FROM Worker ORDER BY Salary DESC;

Following Oracle query will return the top n records with the help of ROWNUM:

SELECT \* FROM (SELECT \* FROM Worker ORDER BY Salary DESC)

WHERE ROWNUM <= 10;

Graphical user interface, text

Description automatically generated

1. Write an SQL query to determine the nth (say n=10) highest salary from a table.

The following MySQL query returns the nth highest salary:

SELECT Salary FROM Worker ORDER BY Salary DESC LIMIT n-1,1;

The following SQL Server query returns the nth highest salary:

SELECT TOP 1 Salary

FROM (

SELECT DISTINCT TOP n Salary

FROM Worker

ORDER BY Salary DESC

)

ORDER BY Salary ASC;

SELECT Salary FROM Worker ORDER BY Salary DESC LIMIT 9,1;

Text

Description automatically generated

**Task-10**

1. Write an SQL query to determine the 8th highest salary without using TOP or LIMIT methods.

SELECT \* FROM (

SELECT ROW\_NUMBER() OVER (ORDER BY SALARY DESC) AS rownumber,Salary

FROM worker )

AS foo

WHERE rownumber = 8;

Text

Description automatically generated

1. Write an SQL query to fetch the list of employees with the same salary.

Select distinct W.WORKER\_ID, W.FIRST\_NAME, W.Salary

from Worker W, Worker W1

where W.Salary = W1.Salary

and W.WORKER\_ID != W1.WORKER\_ID;

A picture containing text

Description automatically generated

Attach the query with the output table screenshot with the assignment.