Name – Adarsh kumar pandey  
Exercise – 5

Reg. No . - 25MCMC29

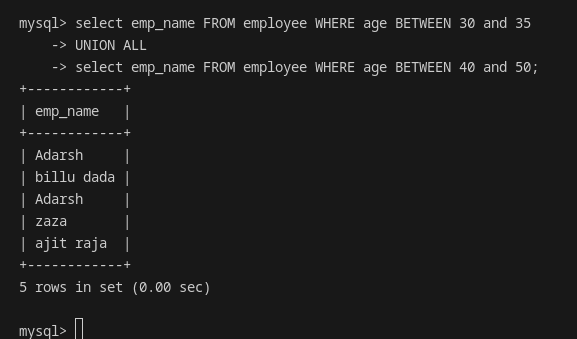
# employee (ID, emp\_name, age, street, city)

# works (ID, company\_name, salary)

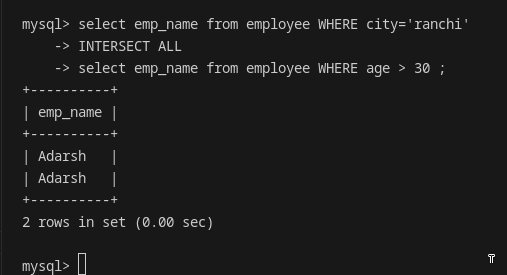
# company (company\_name, city)

# 2 . SQL Query Questions with Outputs

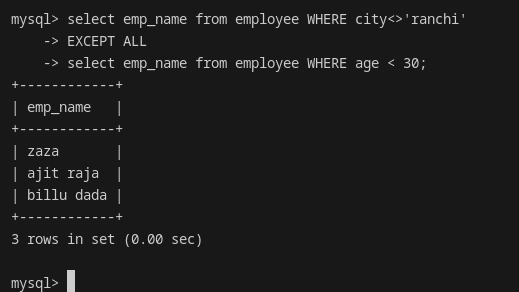
Q1. Use of union all operation findind employee who has age between 30 to 35 and 40 to 50



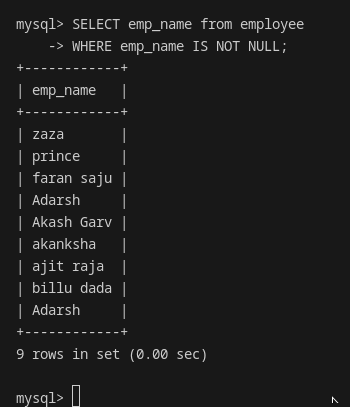
Q2. Use of intersection all to find all employee who are from ranchi and age > 30



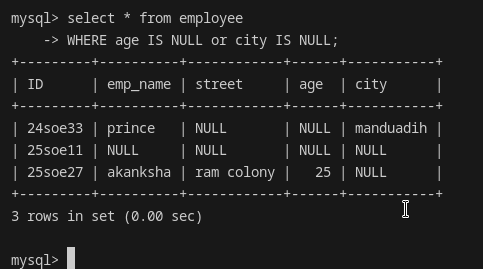
Q3. Use of except all



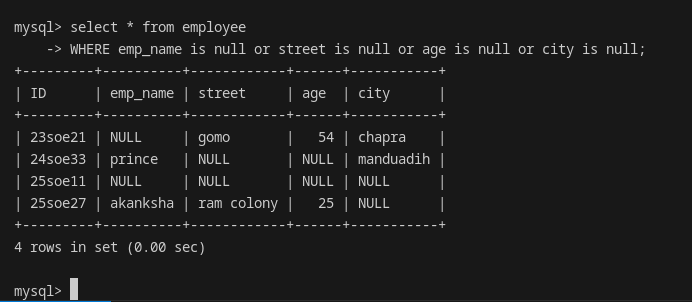
Q4. show employee name where name is not null



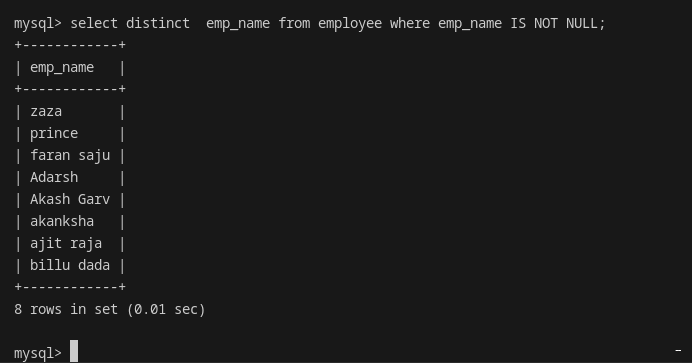
Q5. List details of eployee who's age is null or city is NULL



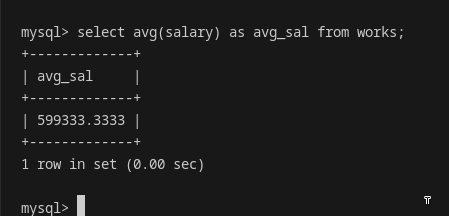
Q6. List all tuples who have null in any tupple



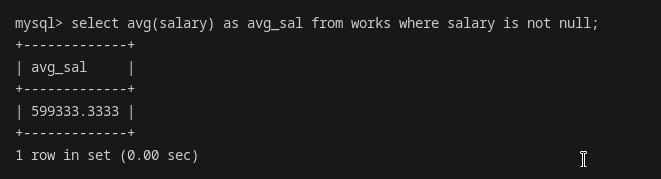
Q7. Perform not operation



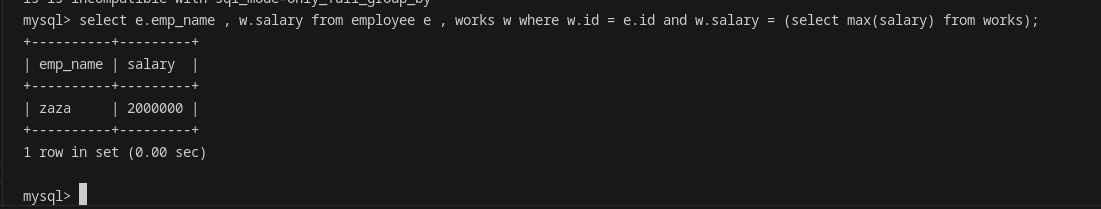
Q8. find average of salary



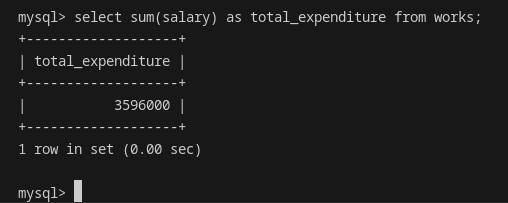
Q9. find avg of salary without null



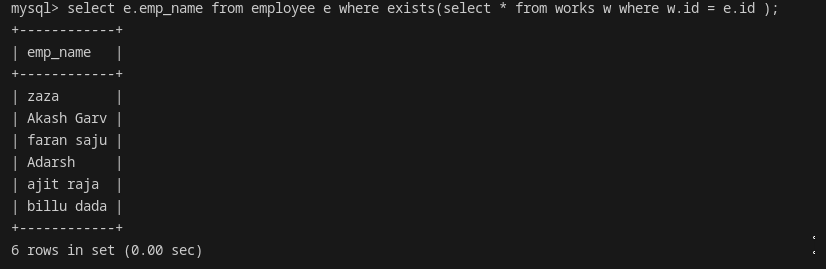
Q10. find name of employee who earns maximum salary



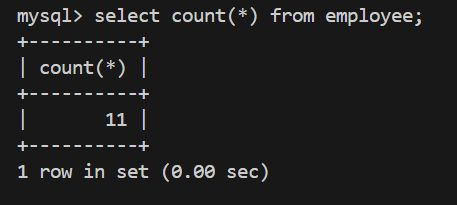
Q11. Calculate total of salary



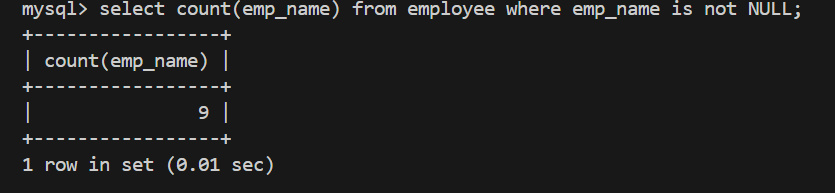
Q12. Get name of employee who work in some organisation using exists



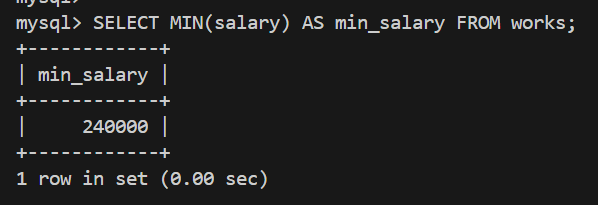
Q13. count no of tuples incluing null



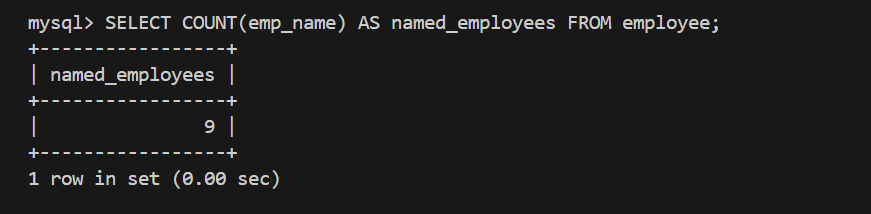
Q14. count no of tuples excluding null



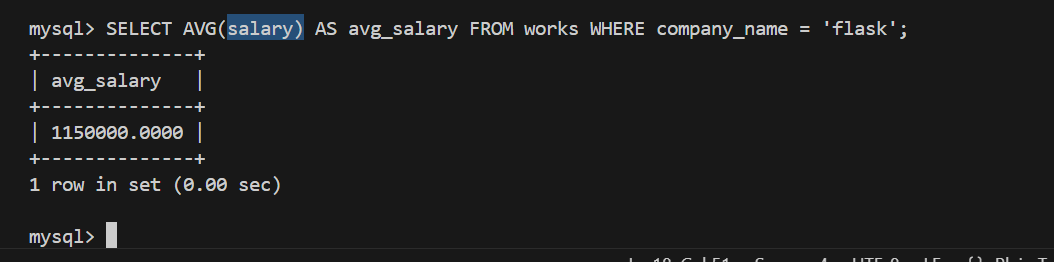
Q15. use of min : aggrigate



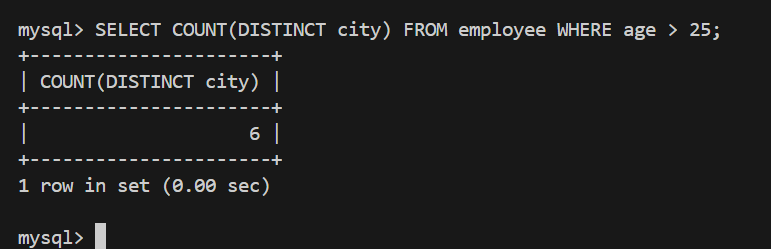
Q16. use of as clause with aggrigate function



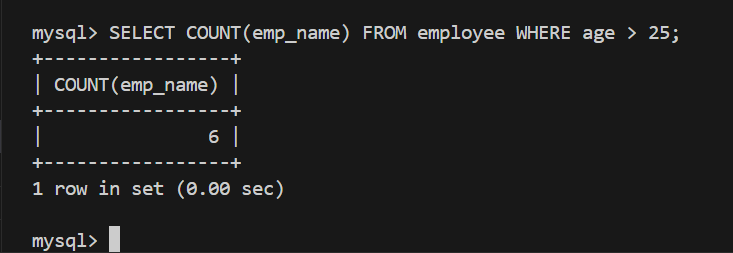
Q17. use of where caluse with aggrigate function



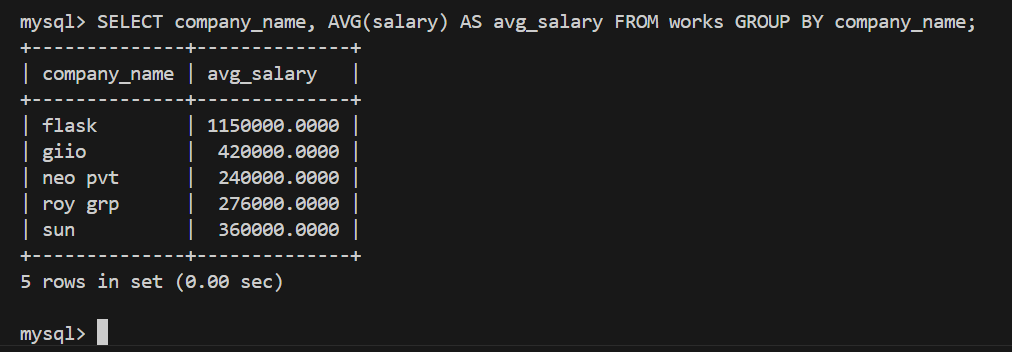
Q18. Use of COUNT(DISTINCT) with WHERE clause



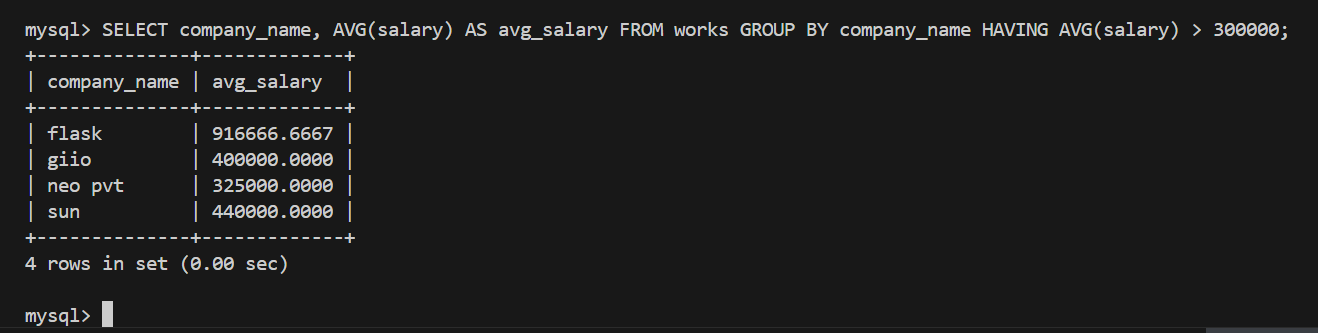
Q19. Use of COUNT with column and WHERE clause



Q20. Use of GROUP BY with AVG and AS alias



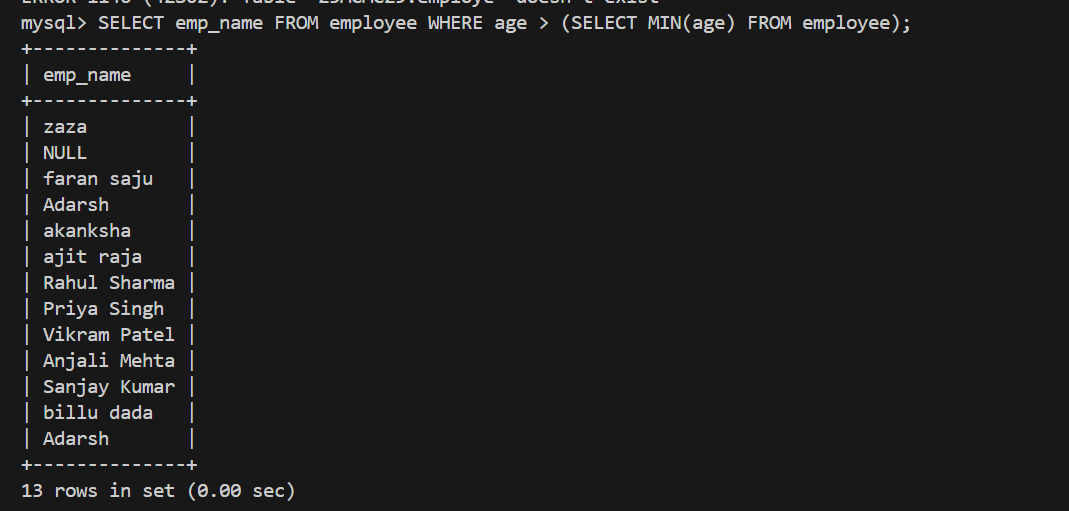
Q21.



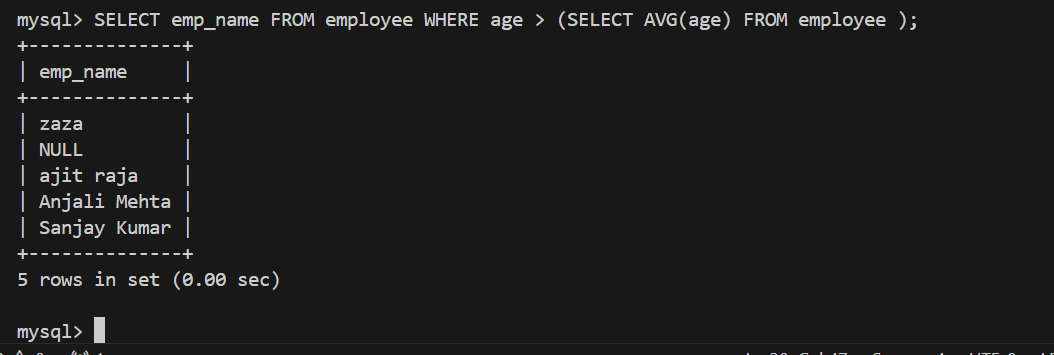
Q22. Use of subquery in FROM clause with AS alias



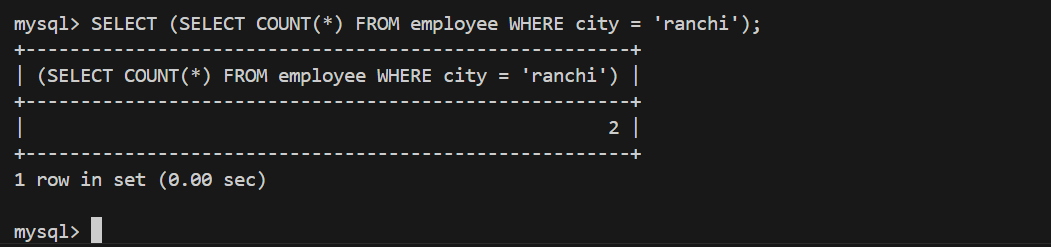
Q23. Use of subquery with MIN in WHERE clause



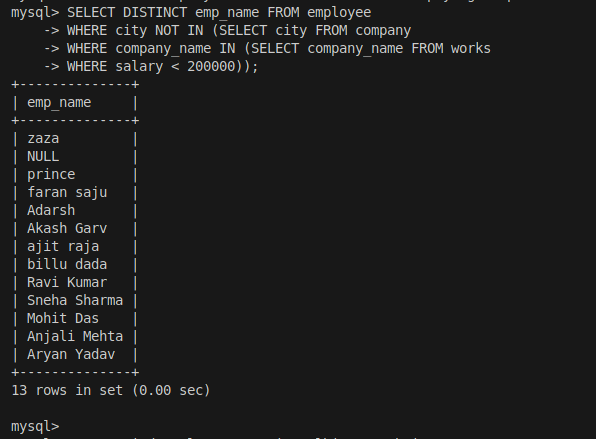
Q24. Use of subquery with AVG in WHERE clause



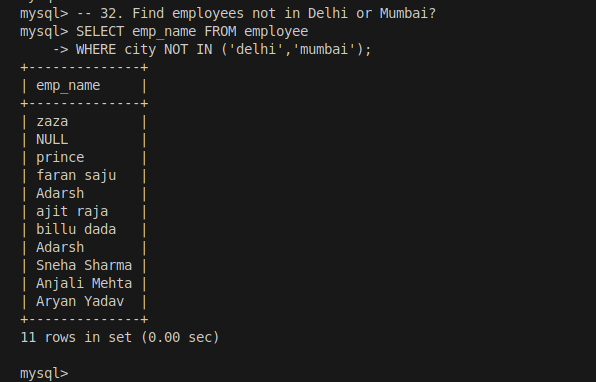
Q25. Use of scalar subquery in SELECT clause



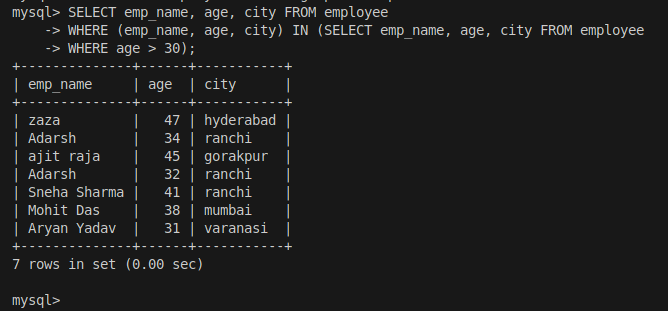
Q26. use of not in with nested subqueries



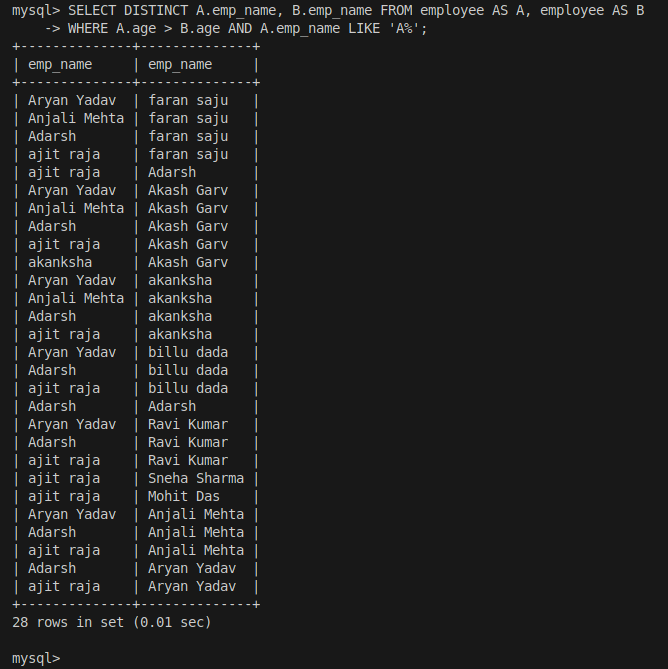
Q27. use of not in with literal values



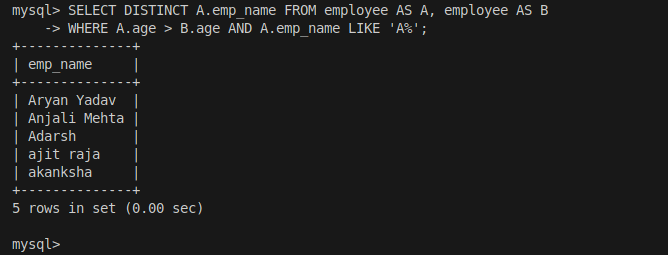
Q28. Use of row constructor with IN subquery



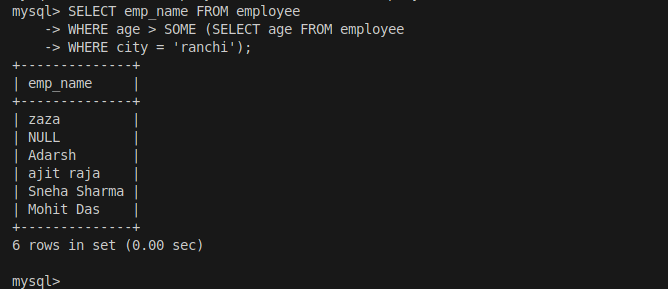
Q29. Use of self-join with table aliases



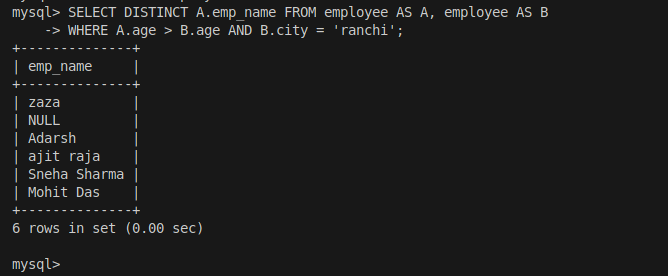
Q30. Use of self-join returning single column



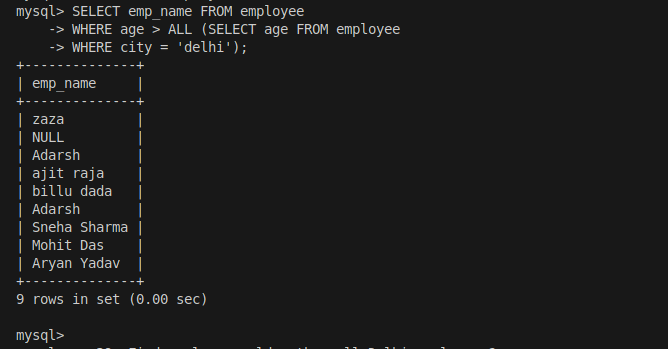
Q31. Use of SOME comparison operator



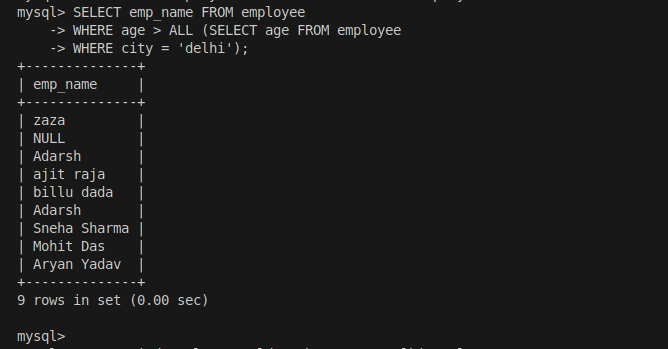
Q32. Use of self-join with specific condition



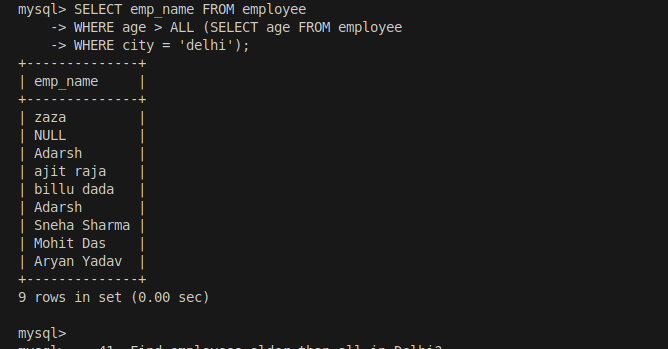
Q33. Use of ALL comparison operator



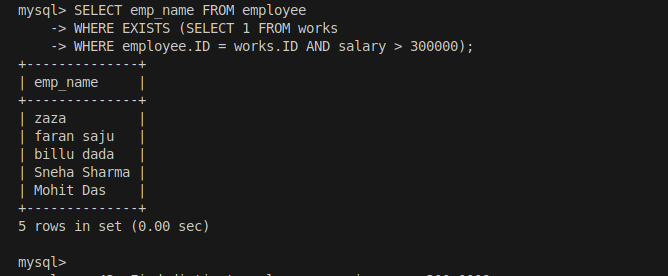
Q34. Use of ALL with subquery



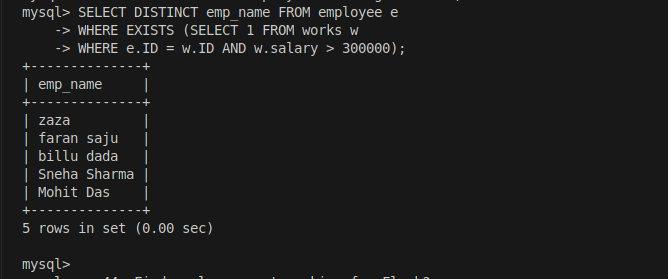
Q35. Use of ALL with subquery with where clause



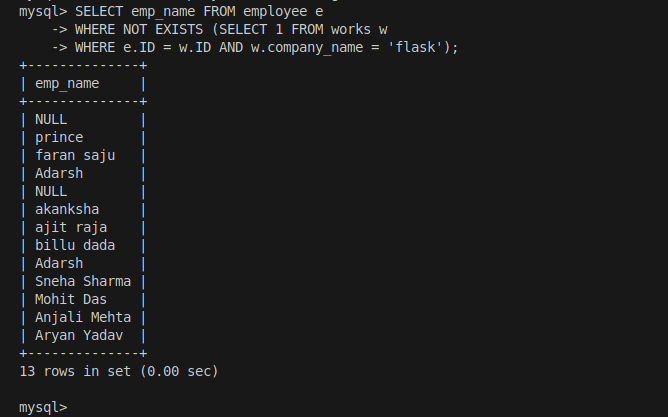
Q36. Use of EXISTS with correlated subquery



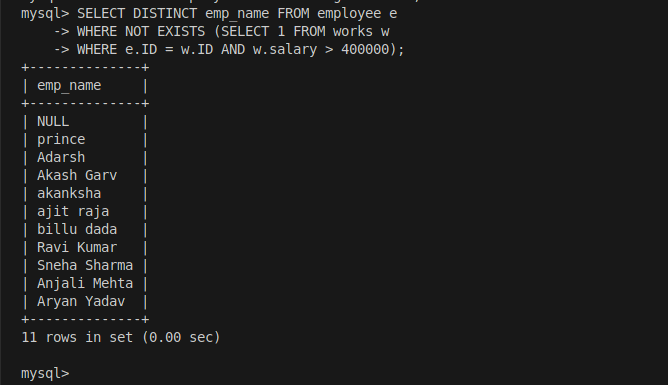
Q37. Use of EXISTS with temp table



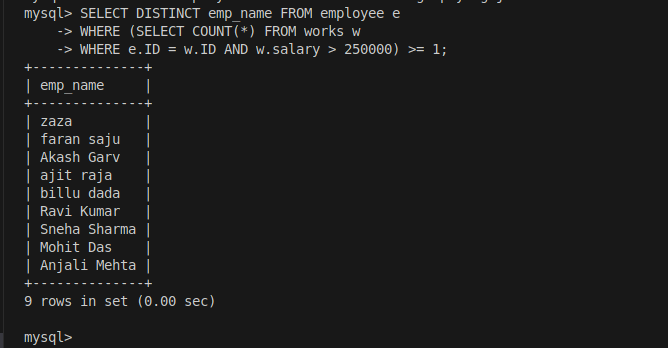
Q38. Use of NOT EXISTS with condition



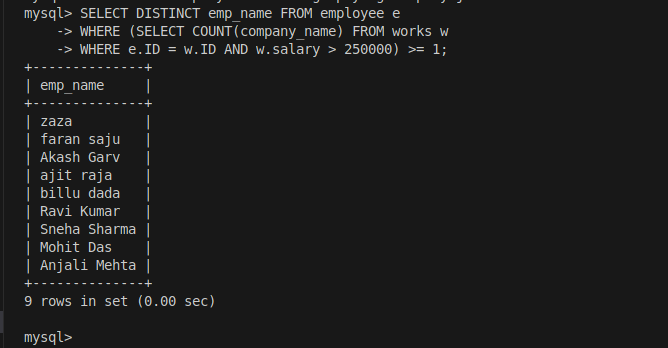
Q39. Use of NOT EXISTS with comparison



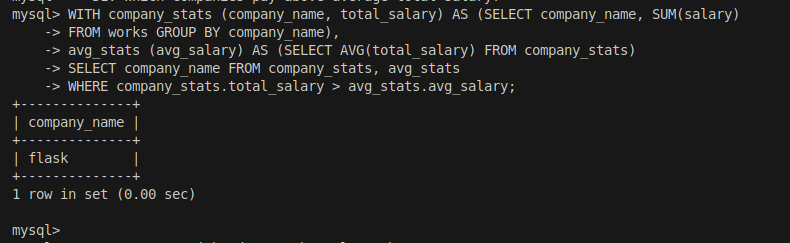
Q40. Use of subquery with COUNT(\*) in WHERE



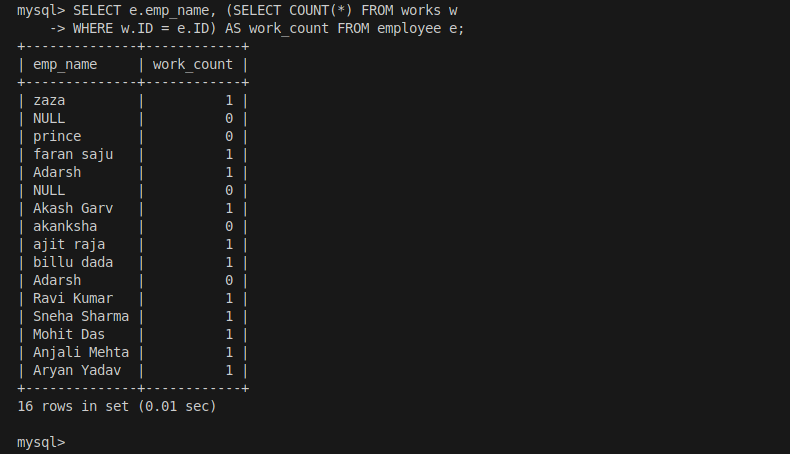
Q41. Use of subquery with COUNT(company name ) in WHERE



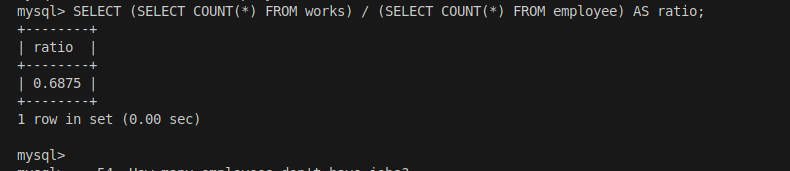
Q42. use of with to get who pay above average total salary



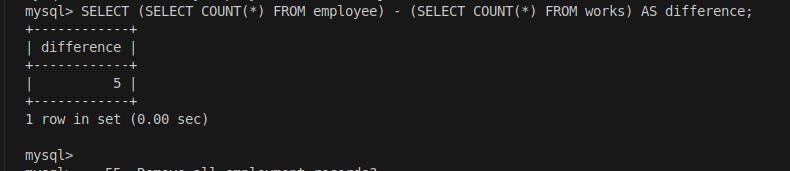
Q43. Use of correlated subquery in SELECT



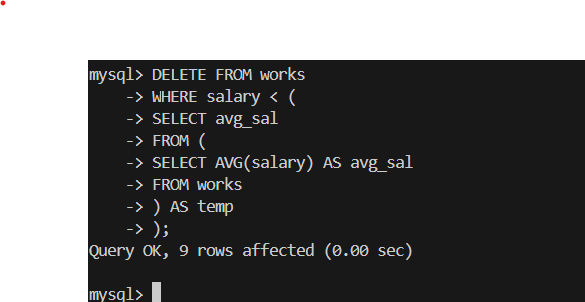
Q44. Use of multiple scalar subqueries in expression



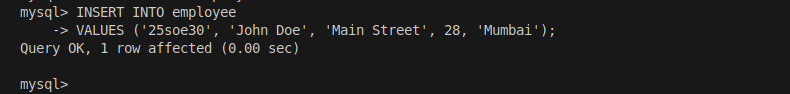
Q45. Use of scalar subqueries in arithmetic operation



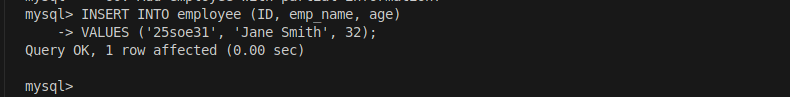
Q46. Use of DELETE with aggregate subquery



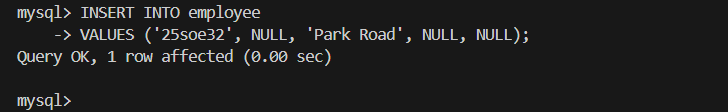
Q47. Use of INSERT with VALUES clause



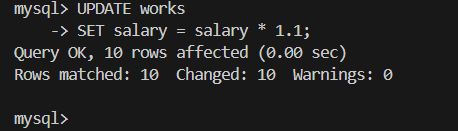
Q48. Use of INSERT with column list



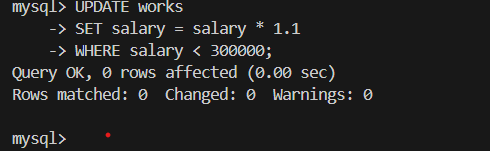
Q49. Use of INSERT with SELECT subquery



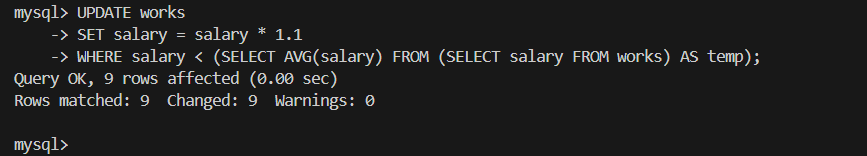
Q50. Use of UPDATE without WHERE clause



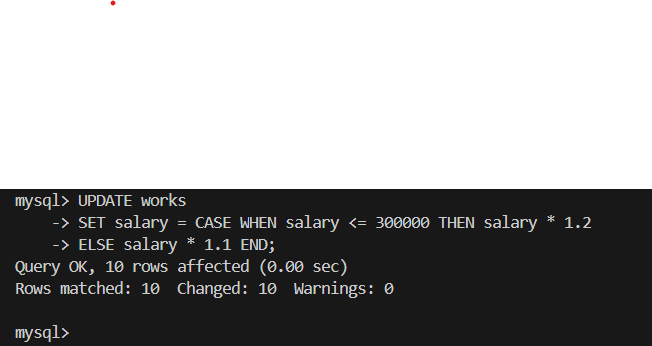
Q51. Use of UPDATE with WHERE condition



Q52. Use of UPDATE with aggregate subquery



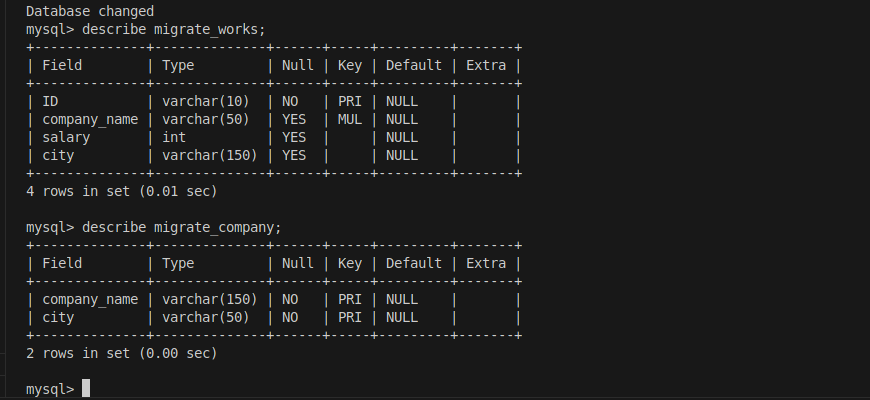
Q53. Use of UPDATE with CASE statement



## Now in the created employee database, provide the SQL queries for the following: a. Find the ID and name of each employee who does not work for “University of Hyderabad”. (1 point)

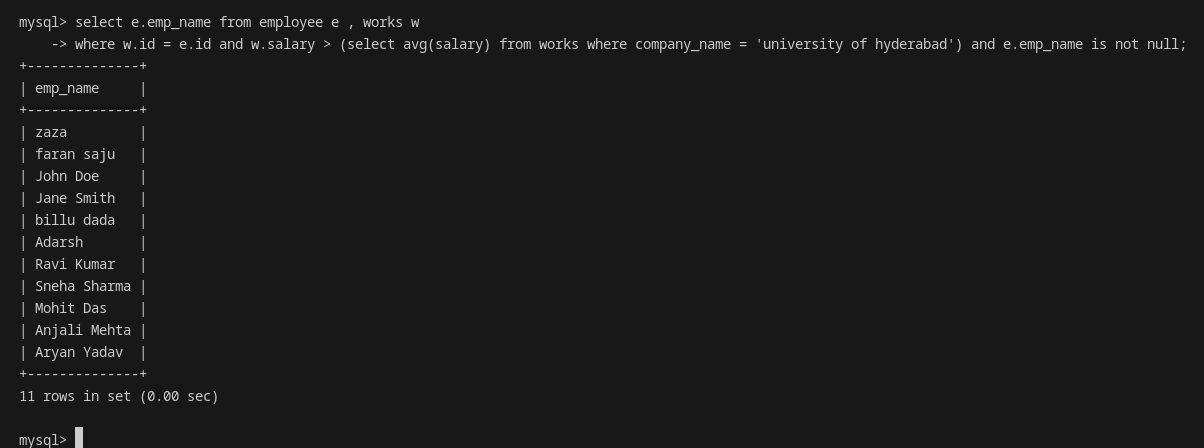
## b. Find the ID and name of each employee who earns at least as much as every employee in the database. (1 point)

## c. Assume that companies may be located in several cities. Find the name of each company that is located in every city in which “SBI Bank” is located. (1 point)



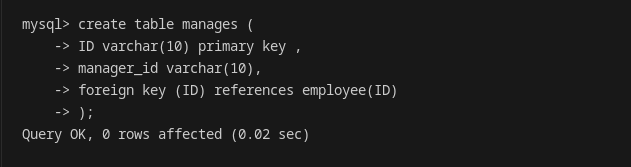
## d. Find the name of the company that has the most employees (or companies, in the case where there is a tie for the most). (1 point)

## e. Find the name of each company whose employees earn a higher salary, on average, than the average salary at “University of Hyderabad”. (1 point)



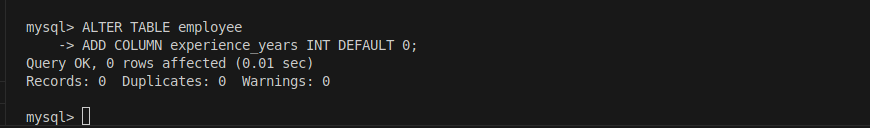
## 4. Now add a new relation to the database.

## manages(ID, manager\_id)



## **6 . Now in the created employee database, provide the SQL queries for the following.** **NB!** These queries need a bit more exploration of SQL queries on the internet.

## a. Think logically and add a new column to the employee table. Now update the rows in the table to also have some values for this new column. (1 point)



## 

## b. Change the attribute name ‘emp\_name’ to ‘person\_name’ in the employee table. (1 point)

