# Database Programming with SQL

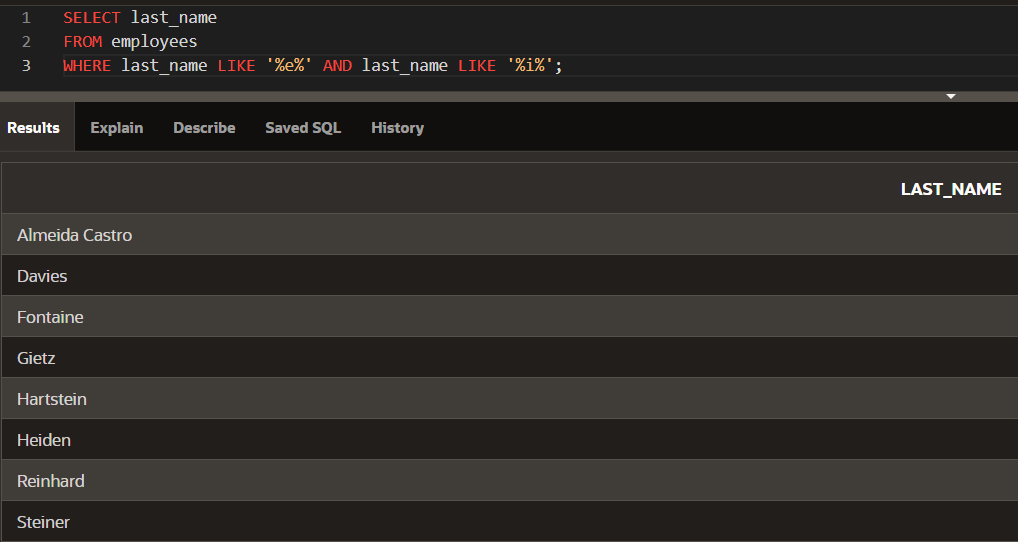
**3-1: Logical Comparisons and Precedence Rules**

1. Execute the two queries below. Why do these nearly identical statements produce two different results? Name the difference and explain why.

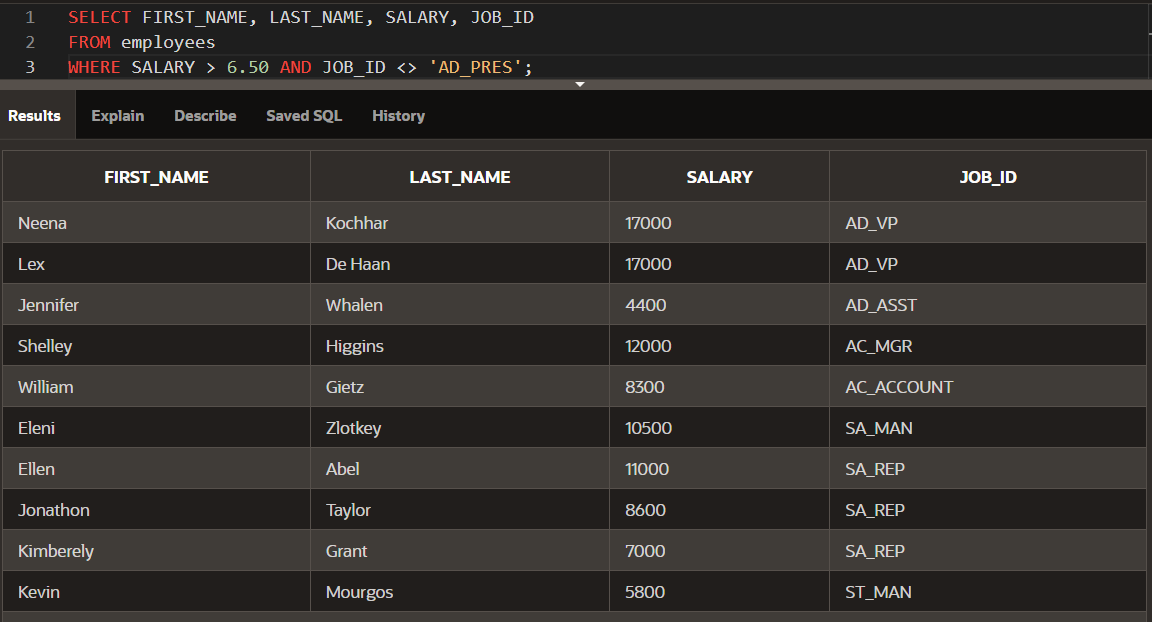
* SELECT code, description FROM d\_themes WHERE code >200 AND description IN('Tropical', 'Football', 'Carnival');
* SELECT code, description FROM d\_themes WHERE code >200 OR description IN('Tropical', 'Football', 'Carnival');

**AND** condition requires both conditions to be true for a row to be included in the result and on the other hand **OR** condition requires only one of the condition to be true for a row to be included in the result. The first query yields a smaller result because it filters rows more strictly as both conditions need to be true while the second query will yield a larger result as only one condition needs to be true for a row to be included in the result.

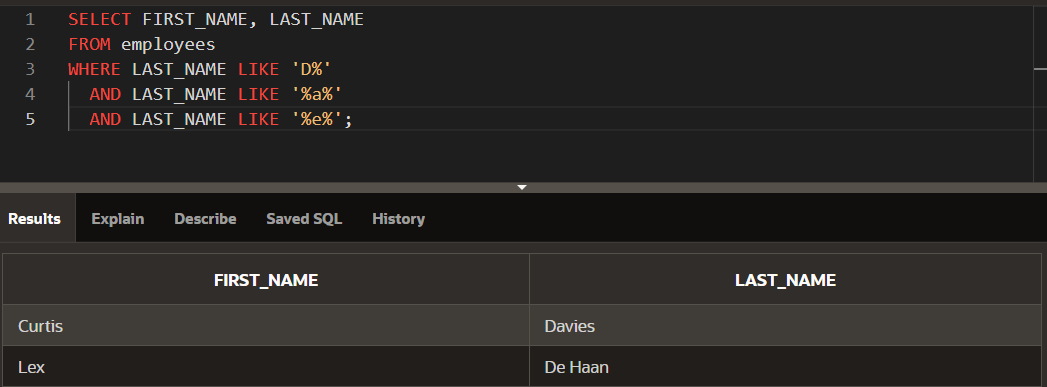
1. Display the last names of all Global Fast Foods employees who have “e” and “i” in their last names.



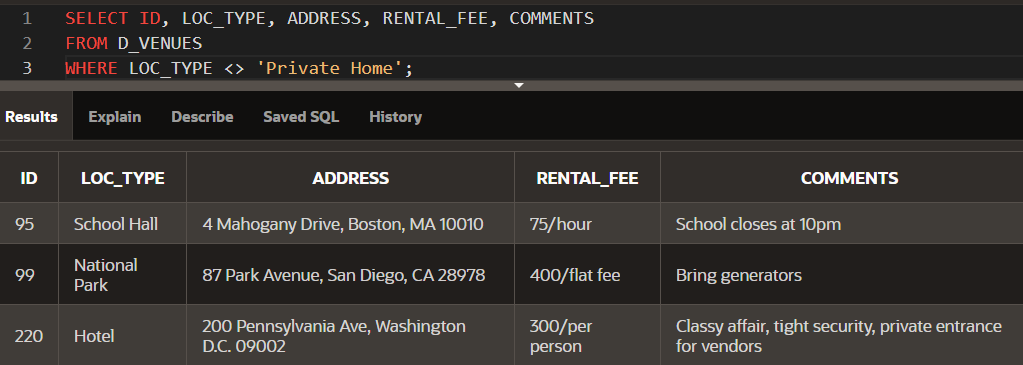
1. I need to know who the Global Fast Foods employees are that make more than $6.50/hour and their position is not order taker.



1. Using the employees table, write a query to display all employees whose last names start with “D” and have “a” and “e” anywhere in their last name



1. In which venues did DJs on Demand have events that were not in private homes?



1. Which list of operators is in the correct order from highest precedence to lowest precedence?
2. AND, NOT, OR
3. NOT, OR, AND
4. **NOT, AND, OR**

**For questions 7 and 8, write SQL statements that will produce the desired output.**

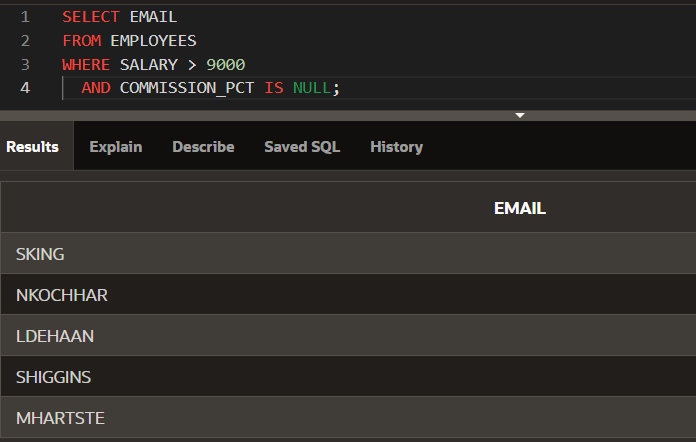
1. Who am I?

I was hired by Oracle after May 1998 but before June of 1999. My salary is less than $8000 per month, and I have an “en” in my last name.

***No one with this description exists.***

1. What's my email address?

Because I have been working for Oracle since the beginning of 1996, I make more than $9000 per month. Because I make so much money, I don't get a commission.

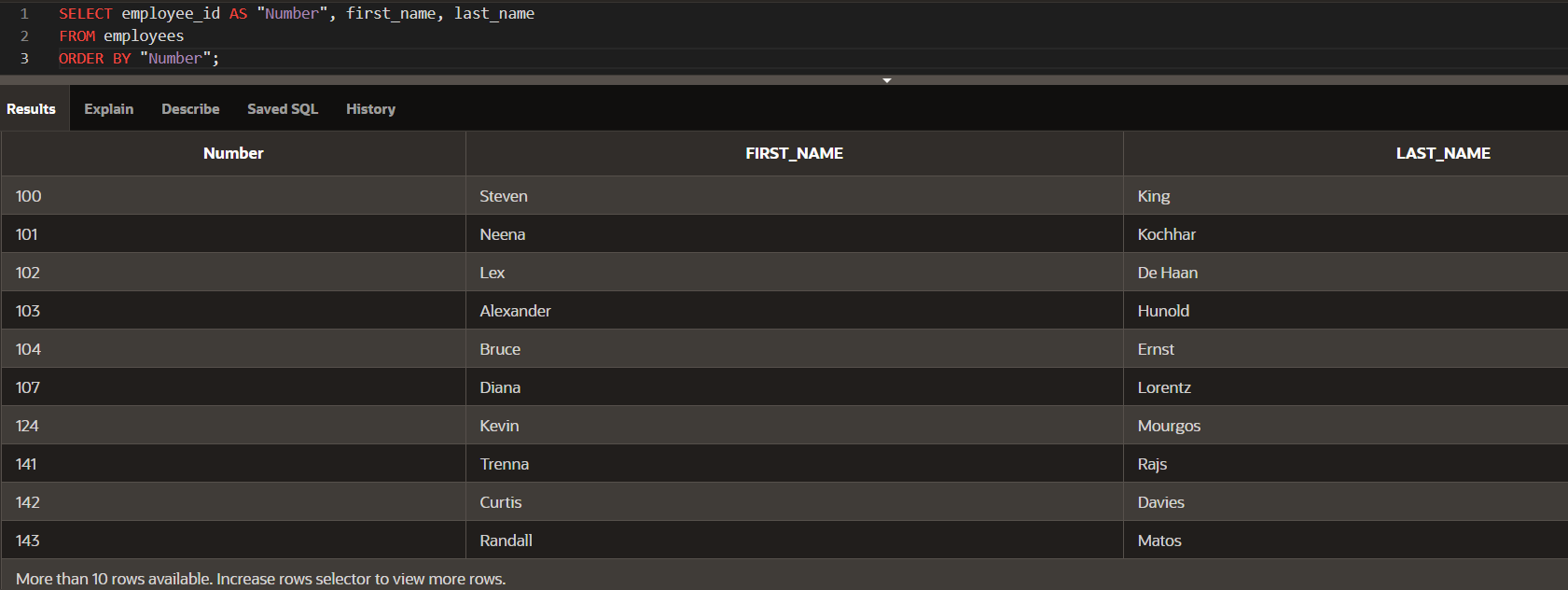


**3-2: Sorting Rows**

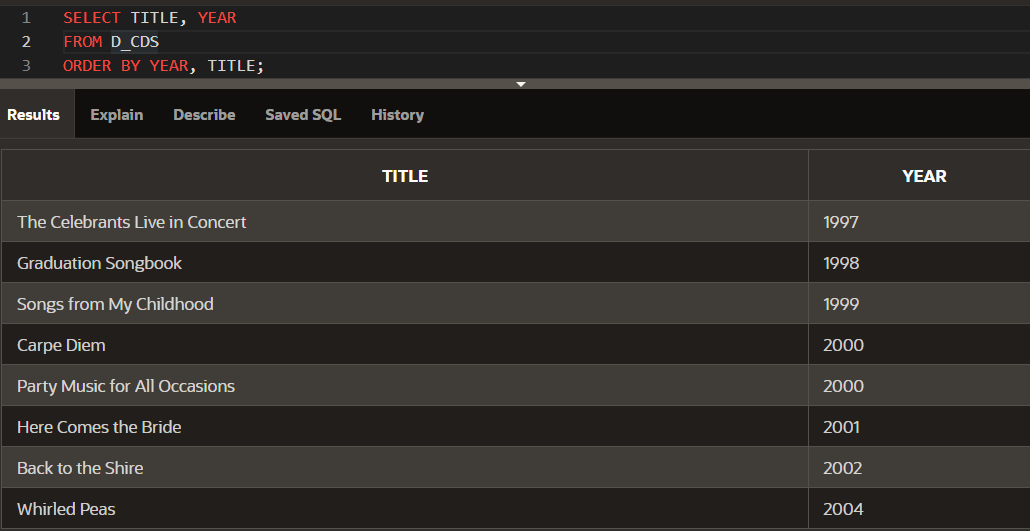
1. In the example below, assign the employee\_id column the alias of “Number.” Complete the SQL statement to order the result set by the column alias.

SELECT employee\_id, first\_name, last\_name

FROM employees



1. Create a query that will return all the DJs on Demand CD titles ordered by year with titles in alphabetical order by year.

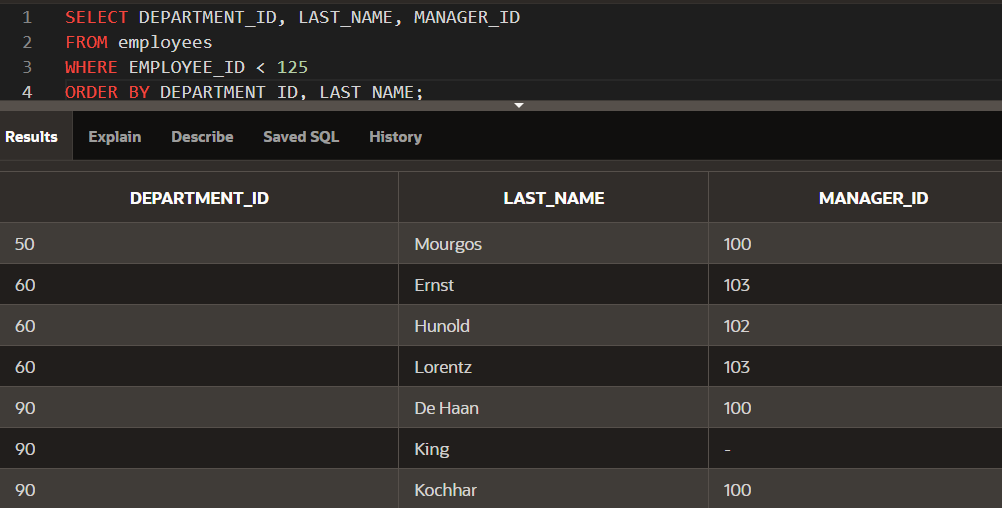


1. Order the DJs on Demand songs by descending title. Use the alias “Our Collection” for the song title.



1. Write a SQL statement using the employees table and the ORDER BY clause that could retrieve the information in the following table. Return only those employees with employee\_id<125.

|  |  |  |
| --- | --- | --- |
| DEPARTMENT\_ID | LAST\_NAME | MANAGER\_ID |
| 90 | Kochhar | 100 |
| 90 | King | (null) |
| 90 | De Haan | 100 |
| 60 | Lorentz | 103 |
| 60 | Hunold | 102 |
| 60 | Ernest | 103 |
| 50 | Mougos | 100 |



**Extension Activities:**

1. Limiting values with the WHERE clause is an example of:
2. Projection
3. Ordering
4. Joining
5. Grouping
6. **Selection**
7. You want to sort your CD collection by title, and then by artist. This can be accomplished using:
8. WHERE
9. SELECT
10. **ORDER BY**
11. DISTINCT
12. Which of the following are SQL keywords?
13. **SELECT**
14. ALIAS
15. COLUMN
16. **FROM**
17. Which of the following are true?
18. Multiplication and division take priority over addition.
19. Operators of the same priority are evaluated from left to right.
20. Parentheses can be used to override the rules of precedence.
21. **None of the above are false.**
22. The following query was written: SELECT DISTINCT last\_name FROM students
23. To select all the outstanding students
24. To choose last names that are duplicates
25. **To select last names without duplicates**
26. To select all last names
27. The following string was created using which SELECT clause? Abby Rogers is an order taker for Global Fast Foods
28. SELECT first\_name ||' ' ||last\_name ||' is an ' staff\_type ' for Global Fast Foods'
29. SELECT Abby Rogers is an ||staff\_type||' for Global Fast Foods'
30. SELECT first\_name, last\_name '||staff\_type||' for Global Fast Foods'
31. **SELECT first\_name ||' ' ||last\_name ||' is an '||staff\_type||' for Global Fast Foods'**
32. Which SELECT statement will always return the last names in alphabetical order?
33. SELECT last\_name AS ORDER BY FROM employees
34. **SELECT last\_name FROM employees ORDER BY last\_name**
35. SELECT last\_name FROM employees
36. SELECT ASC last\_name FROM employees
37. Which of the following SELECT clauses will return uppercase column headings?
38. SELECT id, last\_name, address, city, state, zip, phone\_number;
39. **SELECT ID, LAST\_NAME, ADDRESS, CITY, STATE, ZIP, PHONE\_NUMBER;**
40. SELECT Id, Last\_name, Address, City, State, Zip, Phone\_number;
41. SELECT id AS ID, last\_name AS NAME, address AS ADDRESS, city AS CITY, state AS STATE, zip AS ZIP, phone\_number AS PHONE\_NUMBER;
42. Which SELECT clause will return a column heading for employee\_id called “New Employees”?
43. SELECT last\_name AS "New Employees"
44. SELECT employee\_id AS New Employees
45. SELECT employee AS "New Employees"
46. **SELECT employee\_id AS "New Employees"**
47. Examine the following query: SELECT last\_name, job\_id, salary FROM employees WHERE job\_id = 'SA\_REP' OR job\_id = 'AD\_PRES' AND salary >15000; Which results could not have been returned from this query?
48. **Joe Everyone, sales representative, salary 15000**
49. Jane Hendricks, sales manager, salary 15500
50. Arnie Smithers, administration president, 20000
51. Jordan Lim, sales representative, salary 14000
52. Finish this query so it returns all employees whose last names start with “St”.

SELECT last\_name

FROM employees

**WHERE last\_name LIKE 'St%';**

1. What salary values will not be returned from this query?

SELECT last\_name, first\_name, salary

FROM employees

WHERE salary BETWEEN 1900 AND 2100;

**Any salary values below 1900 and above 2100 will not be returned.**

1. Correct each WHERE clause:
2. WHERE department\_id NOT IN 101,102,103;

**WHERE department\_id NOT IN (101, 102, 103);**

1. WHERE last\_name = King

**WHERE last\_name = 'King'**

1. WHERE start date LIKE "05-May-1998”

**WHERE start\_date LIKE '05-May-1998'**

1. WHERE salary IS BETWEEN 5000 AND 7000

**WHERE salary BETWEEN 5000 AND 7000**

1. WHERE id =! 10

**WHERE id != 10**

**3-3: Introduction to Functions**

1. For each task, choose whether a single-row or multiple row function would be most appropriate:
   1. Showing all of the email addresses in upper case letters

**Single-row function**: This task involves transforming each individual email into uppercase, so a single-row function like UPPER() would be appropriate.

* 1. Determining the average salary for the employees in the sales department

**Multiple-row function**: Since you're calculating the average salary across multiple employees, a multiple-row function like AVG() would be used.

* 1. Showing hire dates with the month spelled out (September 1, 2004)

**Single-row function**: This requires formatting individual hire dates, so a single-row function like TO\_CHAR() with a date format would work.

* 1. Finding out the employees in each department that had the most seniority (the earliest hire date)

**Multiple-row function**: To find the employee with the earliest hire date, a multiple-row function like MIN() should be used.

* 1. Displaying the employees’ salaries rounded to the hundreds place

**Single-row function**: Since this task involves rounding individual salaries, a single-row function like ROUND() would be appropriate.

* 1. Substituting zeros for null values when displaying employee commissions.

**Single-row function**: You would use a function like NVL() to handle null values on a row-by-row basis, making it a single-row function.

1. The most common multiple-row functions are: AVG, COUNT, MAX, MIN, and SUM. Give your own definition for each of these functions.

* **AVG()**: This function calculates the average of all numeric values in a specified column.
* **COUNT()**: It returns the total number of rows that match the specified condition.
* **MAX()**: This function retrieves the highest value from the specified column.
* **MIN()**: It returns the smallest value from the specified column.
* **SUM()**: This function sums up all numeric values in the column.

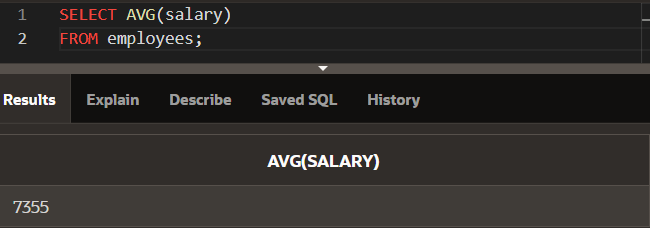
1. Test your definitions by substituting each of the multiple-row functions into this query.

SELECT FUNCTION(salary)

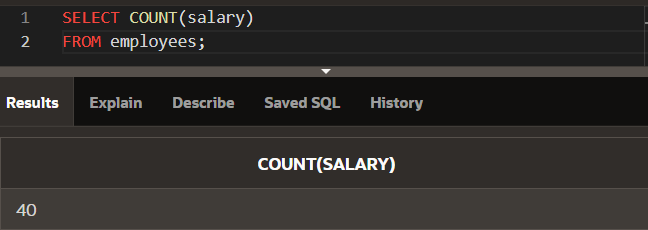
FROM employees

Write out each query and its results.

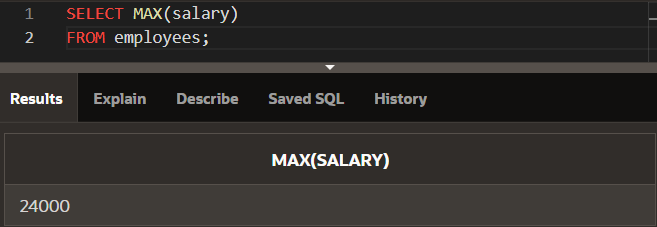
* AVG()



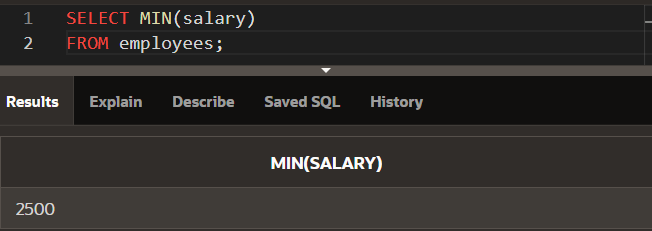
* COUNT()



* MAX()



* MIN()



* SUM()

