**SQL Lab Work.**

You can use Lex playground for SQL to complete this lab work. Pl. remember, SQL syntax may vary based on the database that you use. You may have to tweak the sql accordingly.

[TOC - SQL Server Playground | Lex (infosysapps.com)](https://lex.infosysapps.com/web/en/app/toc/lex_auth_012926670761689088103/overview)

* If you have already worked on this task, then, pl. do not work on it again.
* Pl. try to complete all tasks. If you are not completing 100%, it would impact your rating.
* Pl. submit your leave request, timesheet , WFH request etc. to your unit manager. pl. do not submit it to me.
* Once work is done, pl. send SQL queries and query results in a word document to me via teams or email. Also, update accelerate portal with % completion and actual work hours.

**Lab work:**

1. Create the **departments** table

|  |
| --- |
| CREATE TABLE departments (  department\_id INT PRIMARY KEY,  department\_name VARCHAR(50) NOT NULL  );  INSERT INTO departments (department\_id, department\_name)  VALUES  (1, 'Sales'),  (2, 'Marketing'),  (3, 'Finance'),  (4, 'Human Resources'); |

1. Create the **jobs** table.

|  |
| --- |
| CREATE TABLE jobs (  job\_id INT PRIMARY KEY,  job\_title VARCHAR(50) NOT NULL,  min\_salary DECIMAL(10, 2),  max\_salary DECIMAL(10, 2)  );  INSERT INTO jobs (job\_id, job\_title, min\_salary, max\_salary)  VALUES  (1, 'Sales Representative', 30000.00, 60000.00),  (2, 'Marketing Manager', 50000.00, 90000.00),  (3, 'Financial Analyst', 40000.00, 80000.00),  (4, 'Human Resources Specialist', 35000.00, 70000.00); |

1. Create the **employees** table.

|  |
| --- |
| CREATE TABLE employees (  employee\_id INT PRIMARY KEY,  employee\_name VARCHAR(50) NOT NULL,  department\_id INT,  job\_id INT,  manager\_id INT,  hire\_date DATE,  salary DECIMAL(10, 2),  last\_update TIMESTAMP DEFAULT CURRENT\_TIMESTAMP,  FOREIGN KEY (department\_id) REFERENCES departments(department\_id),  FOREIGN KEY (job\_id) REFERENCES jobs(job\_id),  FOREIGN KEY (manager\_id) REFERENCES employees(employee\_id)  );  INSERT INTO employees (employee\_id, employee\_name, department\_id, job\_id, manager\_id, hire\_date, salary)  VALUES  (1001, 'John Smith', 1, 1, NULL, '2020-01-01', 50000.00),  (1002, 'Jane Doe', 2, 2, 1001, '2019-05-15', 60000.00),  (1003, 'David Johnson', 3, 3, 1001, '2021-02-10', 55000.00),  (1004, 'Sarah Williams', 1, 1, 1001, '2022-03-20', 52000.00),  (1005, 'Michael Brown', 4, 4, 1002, '2020-08-01', 45000.00); |

1. Create the **sales** table.

|  |
| --- |
| CREATE TABLE sales (  sale\_id INT PRIMARY KEY,  employee\_id INT,  amount DECIMAL(10, 2),  sale\_date DATE,  FOREIGN KEY (employee\_id) REFERENCES employees(employee\_id)  );  INSERT INTO sales (sale\_id, employee\_id, amount, sale\_date)  VALUES  (1, 1001, 1000.00, '2022-01-10'),  (2, 1002, 1500.00, '2022-02-15'),  (3, 1003, 800.00, '2021-12-05'),  (4, 1001, 1200.00, '2023-03-20'),  (5, 1004, 900.00, '2022-11-25'), |

1. Create the ‘countries’ table.

|  |
| --- |
| CREATE TABLE countries (  country\_id INT PRIMARY KEY,  country\_name VARCHAR(50) NOT NULL  );  INSERT INTO countries (country\_id, country\_name)  VALUES  (1, 'United States'),  (2, 'United Kingdom'),  (3, 'Canada'),  (4, 'Australia'); |

1. Create a table ‘regions’.

|  |
| --- |
| CREATE TABLE regions (  region\_id INT PRIMARY KEY,  region\_name VARCHAR(50) NOT NULL  );  INSERT INTO regions (region\_id, region\_name)  VALUES  (1, 'North America'),  (2, 'Europe'),  (3, 'Asia'); |

1. Create a Table ‘locations’.

|  |
| --- |
| CREATE TABLE locations (  location\_id INT PRIMARY KEY,  street\_address VARCHAR(100),  postal\_code VARCHAR(20),  city VARCHAR(50) NOT NULL,  state\_province VARCHAR(50),  country\_id INT,  FOREIGN KEY (country\_id) REFERENCES countries(country\_id)  );  INSERT INTO locations (location\_id, street\_address, postal\_code, city, state\_province, country\_id)  VALUES  (1, '123 Main St', '12345', 'New York', 'New York', 1),  (2, '456 Elm St', '56789', 'London', NULL, 2),  (3, '789 Oak St', '98765', 'Toronto', 'Ontario', 3),  (4, '321 Pine St', '54321', 'Sydney', 'New South Wales', 4); |

1. Try to add more records to the table using insert scripts.
2. List all employees in the database.
3. Retrieve the names of all departments.
4. List all employees who have a salary greater than $50,000.
5. Retrieve the total number of employees in the company.
6. List the employees' names, their department names, and their job titles.
7. Retrieve the average salary for each department
8. List the employees who joined the company in the year 2020.
9. Retrieve the highest salary in the company
10. List the department names along with the number of employees in each department.
11. List the department name, job title, and average salary for each department.
12. Retrieve the employee names, their managers' names, and the department names for all employees.
13. List the employees who have a salary greater than the average salary in their department.
14. Retrieve the top 5 departments with the highest number of employees.
15. List the employee names and their hire dates in the format "Month Year" for employees who were hired in the year 2022.
16. List the employees who have a higher salary than their respective manager.
17. Retrieve the employee names and the number of years they have been with the company (based on their hire date).
18. List the employees who have the same job title and are in the same department.
19. Retrieve the employees who have not been assigned to any department
20. List the top 5 employees with the highest total sales amount.
21. Retrieve the employee names and their respective managers' names, along with the total number of employees managed by each manager.
22. List the employees who have at least twice the average salary of their department.
23. Retrieve the top 3 departments with the highest average sales amount per employee.
24. List the employees who have had the highest sales amount in their department in a given year (e.g., 2022).
25. Retrieve the employee names and their corresponding performance rank based on the total sales amount, from highest to lowest.
26. List the employees who have been with the company for more than 5 years and have achieved sales in each year since they joined.
27. Retrieve the employee names along with the percentage of their sales amount compared to the total sales amount of their department.
28. Create a stored procedure called **GetEmployeeCountByDepartment** that takes a department ID as input and returns the total count of employees in that department.
29. Execute the stored procedure created in the above step and verify the results.
30. Create a stored procedure called **GetSalesByYear** that takes a year as input and returns the total sales amount for each month of that year.
31. Execute the stored procedure created in the above step and verify the results.
32. Create a stored procedure called **UpdateEmployeeSalary** that updates the salary of an employee based on their employee ID.
33. Execute the stored procedure created in the above step and verify the results
34. Create a stored procedure called **GetEmployeeDetails** that retrieves the employee details (name, department, job title) for a given employee ID.
35. Execute the stored procedure created in the above step and verify the results
36. Create an index on the **employees** table for the **department\_id** column.
37. Partition the **sales** table by range using the **sale\_date** column, with three partitions for different date ranges.
38. Create a trigger on the **employees** table that automatically updates the **last\_update** column to the current timestamp whenever an employee's record is updated.
39. Update an employee's record to see the trigger in action.
40. Execute a query on the **employees** table to observe the performance improvement due to indexing. You can verify by finding the execution time difference with and without index.
41. Insert records into the **sales\_partitioned** table and observe the distribution among partitions.
42. Retrieve the names of employees who have the highest salary in each department
43. List the department names along with the average salary of employees in each department, ordered by the average salary in descending order.
44. Retrieve the names of employees who have sales records with an amount greater than $1,000.
45. List the names of employees who have the same manager.
46. Retrieve the employee names and the count of their sales records, only including employees with at least 5 sales records
47. List the department names and the number of employees whose salary is above the average salary in their respective department.
48. Retrieve the employee names and their corresponding years of experience (based on the difference between the current year and their hire year).
49. List the employee names who have sales records in both the year 2021 and 2022.