**EMPLOYEES.sql**

Certainly! Let's categorize the SQL tasks based on the concepts they involve:

### Basic SQL Queries (1-10):

1. Retrieve all columns from the "employees" table.

2. Select distinct job titles from the "employees" table.

3. Count the total number of employees.

4. Find the minimum and maximum salaries in the company.

5. Calculate the average salary of all employees.

6. Retrieve the employee with the highest salary.

7. List the unique department numbers.

8. Find the number of employees in each department.

9. Select the employees whose last name starts with 'S'.

10. Count the number of distinct cities in the "employees" table.

### Joins (11-16):

11. Retrieve the names of employees along with their department names.

12. List employees who have not been assigned to any department.

13. Find the managers and their direct reports.

14. Display employees and their job titles along with department names.

15. List employees who have the same manager.

16. Find employees who have not been assigned a manager.

### Aggregation and Grouping (17-23):

17. Calculate the total salary expense for each department.

18. Find the average salary for each job title.

19. Count the number of employees in each department and job title.

20. Identify the department with the highest average salary.

21. List the department with the most employees.

22. Retrieve the total number of employees hired each year.

23. Calculate the total salary for each department and year.

### Subqueries (24-27):

24. Find employees whose salary is above the average salary.

25. Retrieve employees who have the same job title as the manager.

26. List employees who have the highest salary in their department.

27. Select employees who have been promoted more than once.

### Conditional Statements (28-32):

28. Retrieve employees who joined the company before 2000.

29. List employees who have a salary greater than 50000 and less than 80000.

30. Find employees whose last names start with 'A' or 'M'.

31. Select employees who are not managers.

32. Identify employees who have received a bonus.

### String Functions (33-37):

33. Retrieve the first names and last names of employees as a single column.

34. List employees whose last names are at least 5 characters long.

35. Display the employee names in uppercase.

36. Find employees whose first names end with 'a'.

37. Retrieve employees with a space in their last name.

### Date Functions (38-42):

38. List employees hired in the year 2000.

39. Calculate the age of each employee.

40. Find employees who have been with the company for more than 10 years.

41. Identify employees whose birthdays are in the current month.

42. Retrieve the tenure of each employee in years and months.

### Set Operations (43-47):

43. Find common employees between two departments.

44. List employees who are either in the 'Sales' or 'Marketing' department.

45. Retrieve employees who are in the 'HR' department but not in 'IT'.

46. Identify employees who are in both the 'Finance' and 'Marketing' departments.

47. List all employees except those in the 'Management' department.

### Window Functions (48-52):

48. Rank employees by salary within each department.

49. Calculate the running total salary for each department.

50. Find the top 3 highest salaries in each department.

51. Retrieve the employee with the highest salary in each year.

52. List employees with the difference in salary from the previous row.

### Indexing and Performance (53-55):

53. Identify columns that are indexed in the "employees" table.

54. Retrieve the execution plan for a complex query.

55. Analyze the performance of a query and suggest improvements.

### Constraints (56-59):

56. Add a unique constraint to the "employee\_id" column.

57. Create a check constraint to ensure that salaries are positive.

58. Add a foreign key constraint to the "manager\_id" column.

59. Remove a constraint from a column.

### Advanced Queries (60-64):

60. Retrieve the second highest salary in the company.

61. Find the median salary of all employees.

62. List employees who have never taken a vacation.

63. Calculate the average salary excluding the highest and lowest salaries.

64. Identify employees who have the same hire date.

### Updating and Deleting Data (65-67):

65. Update the salary of all employees by a certain percentage.

66. Delete employees who have been with the company for less than a year.

67. Update the job title of employees in a specific department.

### Views (68-71):

68. Delete all records from the "employees" table.

69. Create a view that shows the employee names and their department names.

70. Update data through a view and check if the underlying table is modified.

71. List all views in the database and their definitions.

### Transactions (72-74):

72. Begin a transaction, update salaries, and then commit the changes.

73. Rollback a transaction that results in an error.

74. Set isolation levels for a specific transaction.

### Stored Procedures and Functions (75-77):

75. Create a stored procedure that retrieves employee information based on a given department.

76. Rollback a transaction that results in an error.

77. Develop a function to calculate the age of an employee.

### Triggers (78-79):

78. Create a trigger that updates a log table when an employee is inserted.

79. Develop a trigger that prevents salary updates above a certain threshold.

### Security (80-81):

80. Grant SELECT permission on the "employees" table to a new user.

81. Revoke UPDATE permission on the "salary" column from a user.

### JSON and XML (82-83):

82. Retrieve employee data as JSON.

83. Parse and extract information from a JSON column.

### Analytic Functions (84-85):

84. Use the LEAD or LAG function to compare an employee's salary with the next or previous row.

### Hierarchical Queries (86-87):

86. Display the organizational hierarchy of employees.

87. Find all subordinates of a specific manager.

### User-Defined Types (88-89):

88. Create a user-defined data type for employee roles.

89. Use the user-defined type in a table.

### Pivot and Unpivot (90-91):

90. Pivot the data to display total salaries for each department and job title.

91. Unpivot the data to display individual salaries for each department and job title.

### Dynamic SQL and Indexing Strategies (92-93):

92. Create a dynamic SQL statement to retrieve data based on user input.

93. Analyze the database schema and suggest appropriate indexes.

### Materialized Views and Data Migration (94-95):

94. Create a materialized view for frequently used complex queries.

95. Migrate data from one table to another while maintaining referential integrity.

### Temporal Tables and Full-Text Search (96-97):

96. Implement temporal tables to track changes in employee data over time.

97. Perform a full-text search for employees with specific skills.

### Database Design, Data Validation, and Query Optimization (98-100):

98. Propose improvements to the existing database schema for better performance.

99. Implement constraints to validate data integrity.

100. Optimize a complex query for better performance.