**https://livesql.oracle.com/**

**Plain select**  
• Select unique job IDs from the Employees table  
• Select 10 records from the Employees table sorted in descending order by hire\_date  
• Build a query which returns employee\_id, first\_name and salary from employees table  
• Build a query which shows salary for the whole year (sal \* 12) for each employee. Add columns employee\_id, first\_name, last\_name  
• Select all data from table employees  
• Select 1+1 and add a description to the column  
**Select with where clause**  
• Select all data for employees who work in 30th department  
• Display employees whose last name is equal to King or Gates or Bell  
• Select employees with salary less than 10000  
• Select employees without managers  
• Display data for 10th department  
• Display all jobs with minimal salary more than 5500  
• Display information about employees who have “John” as first name and “Chen” as last name  
• Display information about employees who have salary in the range from 4000 to 13000, don’t have commission\_pct and work in department 60.  
• Select all “CLERK” employees (job\_id should contain clerk)  
Using functions and text formatting  
• Show employees who joined the company in 1998 (3 variants using extract, to\_char, between)  
• Display employees who have been working in the company for more than 25 years  
• Display days from the beginning of the year till today  
• Format the salary column displaying only integer part (to\_char(salary, '99999'))  
• Select all employees in the following format: “first\_name last\_name : salary”, where both names and salaries are left padded  
• Select an employee and their substring of job\_id from underscore symbol (“\_”)  
• Select an employee and their job\_id for job\_id beginning with AD\_. Replace the abbreviation (AD) with “ADMINISTRATION"  
• Select an employee’s name in upper case and their experience  
• Select the first\_name, last\_name, salary and bonuses columns for employees. Bonuses should be  
calculated for experience (25+ years - $1000, 20-25 years - $500)  
• Create a query that displays the employees’ last names and commission amounts. If an employee does not earn a commission, show “No Commission”. Label the column: COMM. (Write two variants with coalescence and case)  
• Write a query which will display info about names, surnames and job\_id based on the Employees table. If the job\_id column is equal to PU\_MAN - display Purchasing Manager, if PU\_CLERK - Purchasing Clerk, else - other  
**Grouping and aggregation**  
• Select the average/max/min job experience  
• Select the maximum length of employee name  
• Display the average minimum salary (use the Jobs table)  
• Display the number of employees who joined later than 01.01.1998  
• Display the max/min/avg salary by department  
• Display department\_ids where average salary is more than 10000  
• Display department\_id, year and the number of employees who joined in that year  
• Select years when the number of employees joined is more than 15  
• Window functions. Use window functions to rank employees by salary from max to min  
• Window functions. Use window functions to add a row number of employees ordered by surname  
• Window functions. Use window functions to enumerate employees inside their departments. Order them by surname.  
**Joins**  
• Display an employee’s name, salary and their full job title  
• Display addresses of all departments (location\_id, region\_name, country\_name, city, state\_province, street\_address) Join locations, countries, and regions  
• Display employees and their job titles if their salary is more than 15000  
• Display a department name and its manager’s first and last name  
• Display employees and their department name for employees whose experience is less than 20 years  
• Display an employee whose hire date is less than their manager’s hire date  
• Select employees who report to “Steven King”  
• Select employees who work in departments located in “Southlake”  
• Select jobs for which employees submitted in 1997  
• Display the amount of money that is spent for salary by departments. (departments’ names and the sum of salaries)  
• Display the amount of employees who work in a department. Show the department’s name  
• Display all departments’ names and, if possibly, employees who work in them (employee name, job\_id, department\_name, department\_id)  
• Display all employees and, if possibly, their managers  
• Display an employee’s first and last name, ending and starting date of all their jobs and their titles. If the employee doesn’t have a job history then it should still be in the output with their job title  
• Display country\_name, city and department\_name  
• Display the employee’s name and country where they work  
• Use a natural join to display countries and its regions  
**Subqueries**  
• Display the name and surname of a manager who manages more than 5 employees  
• Display the city of the employee whose employee id is 105  
• Display the highest average salary among departments  
• Display the department with the maximum number of employees  
• Show details of employees who have the highest salary  
• Display the current job of employees who worked as IT programmers in the past  
• Show all employees who work in a certain department in a single row  
• Show a report where Steven King has two phone numbers (one from the table, and the second should be – “845.444.3581”)  
Set operations  
• Create a report which will display the amount of salaries by departments and the total amount of salaries (in the last row)  
• Select a list of countries that have no departments in it. Use set operations  
• Select a list of departments that do not contain the “IT\_PROG” job\_id  
• Create a report of employees working in departments 10, 50, 20 (right in that order)  
Common table expressions and hierarchical queries  
• Display the hierarchy of employees working in 80th department  
• Create the hierarchy of employees reporting to the manager with employee\_id=101. Use text formatting to show the level of hierarchy. Sort hierarchy in the correct order by manager\_id.  
• Show all managers for employee “John Chen”  
**DML queries**  
• Insert a new row in the Employees table  
• Change the salary of employee 101 to 20000  
• Increase the salary of “IT\_PROG” employees by 15%  
• Change the location of the department that spends the biggest amount of money on salaries  
• Transfer the employee with id 180 to department 80 and increase their salary on 2000  
• Add a new location to the “US” country;  
• Remove locations without departments in them  
• Remove an employee by employee\_id  
• Remove an employee from department 20 who has no direct subordinates  
• Remove 5 employees with the lowest salaries