Oracle SQL exercises

1. Display the last name concatenated with the job ID, separated by a comma and space, and name the column Employee and Title.

**Query:**

select CONCAT(CONCAT(CONCAT(last\_name,','),' ' ),job\_id) AS Employee\_and\_Title from EMPLOYEES;

1. Create a query to display all the data from the EMPLOYEES table. Separate each column by a comma. Name the column THE\_OUTPUT.

**Query:**

SELECT employee\_id || ',' || first\_name || ',' || last\_name || ',' || email || ',' || phone\_number || ','|| job\_id || ',' || manager\_id || ',' || hire\_date || ',' || salary || ',' || commission\_pct || ',' || department\_id THE\_OUTPUT FROM employees;

1. Create a query to display the last name and salary for all employees whose salary is not in the range of 5,000 and 12,000.

**Query:**

SELECT last\_name, salary from EMPLOYEES where salary NOT BETWEEN 5000 and 12000;

1. Display the employee last name, job ID, and start date of employees hired between February 20, 1998, and May 1, 1998. Order the query in ascending order by start date.

**Query:**

Select last\_name, job\_id, hire\_date from EMPLOYEES WHERE hire\_date between '20-FEB-1998' and '01-MAY-1998' ORDER BY hire\_date;

1. Display the last name and department number of all employees in departments 20 and 50 in alphabetical order by name.

**Query:**

Select last\_name, department\_id from EMPLOYEES where department\_id ='20' or department\_id = '50' order by Last\_name;

1. Display the last name and job title of all employees who do not have a manager.

**Query:**

Select last\_name,job\_id from EMPLOYEES where manager\_id is null;

1. Display the last name, salary, and commission for all employees who earn commissions. Sort data in descending order of salary and commissions.

**Query:**

Select last\_name,salary,commission\_pct from employees where commission\_pct is not null order by salary desc,commission\_pct desc;

1. For each employee, display the employee number, last\_name, salary, and salary increased by 15% and expressed as a whole number. Label the column New Salary.

**Query:**

Select employee\_id, last\_name,salary,Round(salary \*1.15,0) "New Salary" from employees;

1. Modify your above query to add a column that subtracts the old salary from the new salary. Label the column Increase.

**Query:**

Select employee\_id,last\_name,salary,round(((salary\*15)/100)+salary,0) "New salary", round(((salary\*15)/100)+salary,0)-salary "increase" from EMPLOYEES;

1. Write a query that displays the employee’s last names with the first letter capitalized and all other letters lowercase and the length of the name for all employees whose name starts with *J*, *A*, or *M*. Give each column an appropriate label. Sort the results by the employees’ last names.

**Query:**

Select Initcap(last\_name) "Name", length(last\_name) "Length" from EMPLOYEES where last\_name like 'J%' OR last\_name like 'A%' or last\_name like 'M%' order by last\_name;

1. Create a unique listing of all jobs that are in department 80. Include the location of the department in the output.

**Query:**

Select distinct(job\_id),location\_id from EMPLOYEES join DEPARTMENTS using (department\_id) where department\_id='80';

1. Display the employee last name and department name for all employees who have an “a” (lowercase) in their last names.

**Query:**

select last\_name, department\_name from employees where last\_name like '%a%';

1. Write a query to display the last name, job, department number, and department name for all employees who work in Toronto.

**Query:**

Select e.last\_name,e.job\_id,e.department\_id,d.department\_name ,l.city from hr.EMPLOYEES e join  hr.DEPARTMENTS d on (e.department\_id = d.department\_id) join hr.LOCATIONS l on (d.location\_id = l.location\_id)  where lower(l.city) = 'toronto';

1. Create a query that displays employee last names, department numbers, and all the employees who work in the same department as a given employee. Give each column an appropriate label.

**Query:**

SELECT e.department\_id department, e.last\_name employee, c.last\_name colleague FROM employees e JOIN employees c ON (e.department\_id = c.department\_id) WHERE

e.employee\_id <> c.employee\_id ORDER BY e.department\_id, e.last\_name, c.last\_name;

1. Display the names and hire dates for all employees who were hired before their managers, along with their manager’s names and hire dates. Label the columns Employee, Emp Hired, Manager, and Mgr Hired, respectively.

**Query:**

SELECT w.last\_name "Employee", w.hire\_date "Employee\_Hired", m.last\_name "Manager", m.hire\_date "Manager\_Hired" FROM employees w JOIN employees m ON

(w.manager\_id = m.employee\_id) WHERE w.hire\_date < m.hire\_date;

1. Display the highest, lowest, sum, and average salary of all employees. Label the columns Maximum, Minimum, Sum, and Average, respectively. Round your results to the nearest whole number

**Query:**

Select Round(max(Salary),0) "Maximum", Round(min(salary),0) "Minimum", Round(sum(salary),0) "Sum", Round(avg(salary),0) "Average" from employees;

1. Determine the number of managers without listing them. Label the column Number of Managers.

**Query:**

Select count(distinct manager\_id) "Number\_of\_managers" from employees;

1. Create a query that will display the total number of employees and, of that total, the number of employees hired in 1995, 1996, 1997, and 1998. Create appropriate column headings.

**Query:**

SELECT COUNT(\*) "Total", SUM(DECODE(TO\_CHAR(hire\_date, 'YYYY'),1995,1,0))"1995", SUM(DECODE(TO\_CHAR(hire\_date, 'YYYY'),1996,1,0))"1996",

SUM(DECODE(TO\_CHAR(hire\_date, 'YYYY'),1997,1,0))"1997",

SUM(DECODE(TO\_CHAR(hire\_date, 'YYYY'),1998,1,0))"1998"

FROM employees;

1. Create a matrix query to display the job, the salary for that job based on department number, and the total salary for that job, for departments 20, 50, 80, and 90, giving each column an appropriate heading.

**Query:**

SELECT job\_id "Job", SUM(DECODE(department\_id , 20, salary)) "Dept 20", SUM(DECODE(department\_id , 50, salary)) "Dept 50", SUM(DECODE(department\_id , 80, salary)) "Dept 80", SUM(DECODE(department\_id , 90, salary)) "Dept 90", SUM(salary) "Total" FROM employees GROUP BY job\_id;

1. Write a query that displays the employee numbers and last names of all employees who work in a department with any employee whose last name contains a “*u”*.

**Query:**

Select employee\_id, last\_name from EMPLOYEES where department\_id in (select department\_id from EMPLOYEES where last\_name like '%u%');

1. Write a query to display the last name, department number, and salary of any employee whose department number and salary both match the department number and salary of any employee who earns a commission.

**Query:**

Select last\_name,department\_id,salary from hr.EMPLOYEES where department\_id NOT in (select department\_id from hr.EMPLOYEES where commission\_pct is null ) and salary NOT in (select salary from hr.EMPLOYEES where commission\_pct is null);

1. Create a query to display the last name, hire date, and salary for all employees who have the same salary and commission as Kochhar.

**Query:**

Select last\_name,hire\_date,salary from employees where salary in (select salary from employees where employee\_name ='Kochhar') and commission in (select commisiion from employees where employee\_name ='Kochhar');

1. Write a query to find all employees who earn more than the average salary in their departments. Display last name, salary, department ID, and the average salary for the department. Sort by average salary. Use aliases for the columns retrieved by the query as shown in the sample output.

**Query:**

SELECT last\_name, department\_id, salary

FROM hr.employees e

WHERE salary > (select avg(salary) from hr.employees e2 where e2.department\_id = e.department\_id) order by salary

**Connecting with Oracle Server**

1. Request for PL/SQL VDI

2. To login to **SQL developer** through command prompt

d:\users\temp>C:

c:>cd sqldeveloper

c:\sqldeveloper >sqldeveloper.exe or c:\sqldeveloper >sqldeveloper

Goto new connection and enter credentials (username : hr Password: hr) as below

