Clear screenshots of successful run of SQL query and output is required in a single file. Zero will be assigned otherwise. You may use SQL developer or SQL plus. If you want to be independent of mySeneca apps or Seneca Oracle instance, install Oracle XE in your laptop and use SQL plus. Include your answer to Question 6,7,8 in the same file.

1. Display the department name, city, street address and postal code for all

departments. Use the JOIN and USING form of syntax.

Sort the output by department name descending.

SELECT department\_name, street\_address, city, postal\_code

FROM departments

JOIN locations USING (location\_id)

ORDER BY department\_name DESC;

A screenshot of a computer

Description automatically generated

2. Display full name of the employees, their hire date and salary together with

their department name, but only for departments which names start with **A** or **S**. Full name should be in format of :

**First / Last.** Use the JOIN and ON form of syntax.

Sort the output by department name and then by last name.

SELECT e.first\_name || ' / ' || e.last\_name AS full\_name, e.hire\_date, e.salary, d.department\_name

FROM employees e

JOIN departments d ON (e.department\_id = d.department\_id)

WHERE d.department\_name LIKE 'A%' OR d.department\_name LIKE 'S%'

ORDER BY d.department\_name, e.last\_name;

A screenshot of a computer

Description automatically generated

3. Display full name of the manager of each department in provinces Ontario,

California and Washington plus department name, city, postal code and province name. Full name should be in format of :

**Last, First.** Use the JOIN and ON form of syntax.

Sort the output by city and then by department name.

SELECT e.last\_name || ', ' || e.first\_name AS full\_name, d.department\_name, l.city, l.postal\_code, l.state\_province

FROM employees e

JOIN departments d ON d.manager\_id = e.employee\_id

JOIN locations l ON l.location\_id = d.location\_id

WHERE l.state\_province IN ('Ontario', 'California', 'Washington')

ORDER BY l.city, d.department\_name;

A screenshot of a computer

Description automatically generated

4. Display the department name and Highest, Lowest and Average pay per each

department. Name these results *High, Low* and *Avg.*

Use JOIN and ON form of the syntax.

Sort the output so that department with highest average salary are shown first.

SELECT d.department\_name, MAX(e.salary) AS High, MIN(e.salary) AS Low, AVG(e.salary) AS Avg

FROM employees e

JOIN departments d ON e.department\_id = d.department\_id

GROUP BY d.department\_name

ORDER BY Avg DESC;

A screenshot of a computer

Description automatically generated

5. Display the employee last name and employee number along with their

manager’s last name and manager number. Label the columns Employee,

Emp#, Manager, and Mgr#, respectively.

SELECT emp.last\_name AS Employee, emp.employee\_id AS Emp#, mgr.last\_name AS Manager, mgr.employee\_id AS Mgr#

FROM employees emp

JOIN employees mgr ON emp.manager\_id = mgr.employee\_id;

A screenshot of a computer

Description automatically generated

6. What is the difference between join and subquery?

Join combines rows from two or more tables based on a common field. It usually shows faster, especially with large datasets. Joins are written in the from clause of a query. They are usually harder to read and understand because they can be complex. However, they are very efficient and make good use of the database's abilities.

Subquery is a query inside another query, often found in the where clause. Subqueries break complex queries into smaller parts, making them easier to write and understand. They are usually used in SELECT, WHERE clauses. While subqueries offer more flexibility, they can be slower, especially if they return many rows.

7. Explain what is wrong with the following statement?

**Select employee\_id, last\_name**

**from employees**

**where salary = (select max(salary)**

**from employees**

**group by department\_id)**

The subquery returns multiple values, but the = operator expects only one value. The subquery select max(salary) from employees group by department\_id gives the highest salary for each department, resulting in many rows.

To fix this,

Select E.employee\_id, E.last\_name

from employees E

JOIN (

select department\_id, max(salary) as max\_salary

from employees

group by department\_id

) D

on E.department\_id = D.department\_id and E.salary = D.max\_salary;

8. What is the difference between IN and ANY operator? Where would you use them? How about NOT IN and ALL operators?

ANY operator returns true if any of the values specified meets the condition, using =, <>, <, >, <=, >=. IN Operator allows specifying multiple values in the WHERE clause. NOT IN operator negates the IN condition. ALL operator returns true if all the subquery values meet the condition.